Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing

MICHAEL C. MOZER

In algorithmic music composition, a simple technique involves selecting notes sequentially according to a transition table that specifies the probability of the next note as a function of the previous context. An extension of this transition-table approach is described, using a recurrent autopredictive connectionist network called CONCERT. CONCERT is trained on a set of pieces with the aim of extracting stylistic regularities. CONCERT can then be used to compose new pieces. A central ingredient of CONCERT is the incorporation of psychologically grounded representations of pitch, duration and harmonic structure. CONCERT was tested on sets of examples artificially generated according to simple rules and was shown to learn the underlying structure, even where other approaches failed. In larger experiments, CONCERT was trained on sets of J. S. Bach pieces and traditional European folk melodies and was then allowed to compose novel melodies. Although the compositions are occasionally pleasant, and are preferred over compositions generated by a third-order transition table, the compositions suffer from a lack of global coherence. To overcome this limitation, several methods are explored to permit CONCERT to induce structure at both fine and coarse scales. In experiments with a training set of waltzes, these methods yielded limited success, but the overall results cast doubt on the promise of note-by-note prediction for composition.

KEYWORDS: Music composition, neural networks, recurrent networks, psychoacoustic representation, multi-scale processing.

1. Introduction

In creating music, composers bring to bear a wealth of knowledge of musical conventions. Some of this knowledge is based on the experience of the individual, some is culture specific, and perhaps some is universal. No matter what the source, this knowledge acts to constrain the composition process, specifying, for example, the musical pitches that form a scale, the pitch or chord progressions that are agreeable, and stylistic conventions like the division of a symphony into movements and the AABB form of a gavotte. If we hope to build automatic composition systems that create agreeable tunes, it will be necessary to incorporate knowledge of musical conventions into the systems. The difficulty is in deriving this knowledge in an

M. C. Mozer, Department of Computer Science, and Institute of Cognitive Science, University of Colorado, Boulder, CO 80309-0430, USA. E-mail: mozer@cs.colorado.edu.

explicit form: even human composers are unaware of many of the constraints under which they operate (Loy, 1991).

In this article, a connectionist network that composes melodies with harmonic accompaniment is described. The network is called CONCERT, an acronym for *con*nectionist *c*omposer of *er*udite *t*unes. (The 'er' may also be read as *er*ratic or *er*satz, depending on what the listener thinks of its creations.) Musical knowledge is incorporated into CONCERT via two routes. First, CONCERT is trained on a set of sample melodies from which it extracts regularities of note and phrase progressions; these are melodic and stylistic constraints. Second, representations of pitch, duration and harmonic structure that are based on psychological studies of human perception have been built into CONCERT. These representations, and an associated theory of generalization proposed by Shepard (1987), provide CONCERT with a basis for judging the similarity among notes, for selecting a response, and for restricting the set of alternatives that can be considered at any time. The representations thus provide CONCERT with psychoacoustic constraints.

The experiments reported here are with single-voice melodies, some with harmonic accompaniment in the form of chord progressions. The melodies range from 10-note sequences to complete pieces containing roughly 150 notes. A complete composition system should describe each note by a variety of properties—pitch, duration, phrasing, accent—along with more global properties such as tempo and dynamics. In the experiments reported here, the problem has been stripped down somewhat, with each melody described simply as a sequence of pitch–duration–chord triples. The burden of the present work has been to determine the extent to which CONCERT can discover the structure in a set of training examples.

Before the details of CONCERT are discussed a description is given of a traditional approach to algorithmic music composition using Markov transition tables, the limitations of this approach, and how these limitations may be overcome in principle using connectionist learning techniques.

2. Transition Table Approaches to Algorithmic Music Composition

A simple but interesting technique in algorithmic music composition is to select notes sequentially according to a transition table that specifies the probability of the next note as a function of the current note (Dodge & Jerse, 1985; Jones, 1981; Lorrain, 1980). For example, the transition probabilities depicted in Table I constrain the next pitch to be one step up or down the C-major scale from the current pitch. Generating a sequence according to this probability distribution therefore results in a musical random walk. Transition tables may be hand-constructed according to certain criteria, as in Table I, or they may be set up to embody a particular musical style. In the latter case, statistics are collected over a set of examples (hereafter, the training set) and the transition-table entries are defined to be the transition probabilities in these examples.

Table I. Transition probability from current pitch to the next

Next	Current pitch						
pitch	C	D	E	F	G	A	B
C	0	0.5	0	0	0	0	0.5
D	0.5	0	0.5	0	0	0	0
E	0	0.5	0	0.5	0	0	0
F	0	0	0.5	0	0.5	0	0
G	0	0	0	0.5	0	0.5	0
A	0	0	0	0	0.5	0	0.5
B	0.5	0	0	0	0	0.5	0

The transition table is a statistical description of the training set. In most cases, the transition table will lose information about the training set. To illustrate, consider the two sequences ABC and EFG. The transition table constructed from these examples will indicate that A goes to B with probability 1, B to C with probability 1, and so forth. Consequently, given the first note of each sequence, the table can be used to recover the complete sequence. However, with two sequences like BAC and DAE, the transition table can only say that following an A either an E or a C occurs, each with a 50% likelihood. Thus, the table cannot be used to reconstruct the examples unambiguously.

Clearly, in melodies of any complexity, musical structure cannot be fully described by the pairwise statistics. To capture additional structure, the transition table can be generalized from a two-dimensional array to n dimensions. In the n-dimensional table, often referred to as a table of order $n - 1$, the probability of the next note is indicated as a function of the previous $n - 1$ notes. By increasing the number of previous notes taken into consideration, the table becomes more context sensitive, and therefore serves as a more faithful representation of the training set.[1] Unfortunately, extending the transition table in this manner gives rise to two problems. First, the size of the table explodes exponentially with the amount of context and rapidly becomes unmanageable. With, say, 50 alternative pitches, 10 alternative durations and a third-order transition table—modest sizes on all counts—7.5 billion entries are required. Second, a table representing the high-order structure masks the tremendous amount of low-order structure present. To elaborate, consider the sequence

A F G B F G C F G D F G♯ E F G

One would need to construct a third-order table to represent this sequence faithfully. Such a table would indicate that, for example, the sequence GBF is always followed by G. However, there are first-order regularities in the sequence that a third-order table does not make explicit, namely the fact that an F is almost always followed by a G. The third-order table is thus unable to predict what will follow, say, AAF, although a first-order table would sensibly predict G. There is a trade-off between the ability to represent the training set faithfully, which usually requires a high-order table, and the ability to generalize in novel contexts, which profits from a low-order table. What one would really like is a scheme by which only the relevant high-order structure is represented (Lewis, 1991).

Kohonen (1989) and Kohonen *et al.*, (1991) have proposed exactly such a scheme. The scheme is a symbolic algorithm that, given a training set of examples, produces a collection of rules—a context-sensitive grammar—sufficient for reproducing most or all of the structure inherent in the set. These rules are of the form *context* → *next_note*, where *context* is a string of one or more notes, and *next_note* is the next note implied by the context. Because the context length can vary from one rule to the next, the algorithm allows for varying amounts of generality and specificity in the rules. The algorithm attempts to produce deterministic rules—rules that always apply in the given context. Thus, the algorithm will not discover the regularity F → G in the above sequence because it is not absolute. One could conceivably extend the algorithm to generate simple rules like F → G along with exceptions (e.g. DF → G♯), but the symbolic nature of the algorithm still leaves it poorly equipped to deal with statistical properties of the data. Such an ability is not critical if the algorithm's goal is to construct a set of rules from which the training set can be exactly reconstructed. However, if one views music composition as an intrinsically random process, it is inappropriate to model every detail of the training set. Instead, the goal ought to be to capture the most important—i.e. statistically regular—structural properties of the training set.

Both the transition-table approach and Kohonen's musical grammar suffer from two further drawbacks. First, both algorithms are designed so that a particular note, n, cannot be used to predict note $n + i$ unless all intervening notes, $n + 1, \ldots, n + i - 1$, are also considered. In general, one would expect that the most useful predictor of a note is the immediately preceding note, but cases exist where notes $n, \ldots, n + k$ are more useful predictors of note $n + i$ than notes $n + k + 1, \ldots, n + i - 1$ (e.g. a melody in which high-pitch and low-pitch phrases alternate, as in the solo violin partitas of J. S. Bach). The second, and perhaps more serious, drawback is that a symbolic representation of notes does not facilitate generalization from one musical context to perceptually similar contexts. For instance, the congruity of octaves is not encoded, nor is the abstract notion of intervals such as a 'minor third'.

Connectionist learning algorithms offer the potential of overcoming the various limitations of transition-table approaches and Kohonen musical grammars. Connectionist algorithms are able to discover relevant structure and statistical regularities in sequences (e.g. Elman, 1990; Mozer, 1989). Indeed, connectionist algorithms can be viewed as an extension of the transition-table approach, a point also noted by Dolson (1989). Just as the transition-table approach uses a training set to calculate the probability of the next note in a sequence as a function of the previous notes, so does the network to be described, CONCERT. The connectionist approach, however, is far more flexible in principle: the form of the transition function can permit the consideration of varying amounts of context, the consideration of non-contiguous context, and the combination of low-order and high-order regularities.

The connectionist approach also promises better generalization through the use of distributed representations (Hinton *et al.*, 1986). In a local representation, where each note is represented by a discrete symbol, the sort of statistical contingencies that can be discovered are among notes. However, in a distributed representation, where each note is represented by a set of continuous feature values, the sort of contingencies that can be discovered are among features. To the extent that two notes share features, featural regularities discovered for one note may transfer to the other note.

3. Note-by-note Composition

The Markov transition table and the Kohonen algorithm both use a note-by-note technique in which notes are produced sequentially and linearly, from the start of a piece to the end, each note depending on the preceding context. Todd (1989) and Bharucha and Todd (1989) first explored this technique in a connectionist framework. Since then, it has been adopted by many other connectionist researchers (e.g. Stevens & Wiles, 1993). CONCERT also uses the note-by-note technique; it differs from earlier work primarily in that it uses an assortment of state-of-the-art connectionist tricks to achieve a sophisticated implementation of the technique (e.g. back-propagation through time, probabilistic interpretation of the network outputs, a maximum likelihood training criterion, representational considerations) and tests the technique on a variety of relatively large-scale problems. By using a powerful architecture and learning algorithm, the goal of the research is to see how far the note-by-note technique can be pushed. Previous research has been fairly uncritical in accepting and examining a network's performance; the simple fact that a network creates novel output tends to be interpreted as success. In this work, CONCERT's performance on simple, well-structured sequences is evaluated according to an objective criterion, and on complex musical examples according to the ears of experienced human listeners.

Despite the research effort expended on note-by-note composition, it might seem an unlikely technique to succeed. Music has a rich, hierarchical structure, from the level of notes within a theme, to themes within a phrase, to phrases within a movement, to movements within a symphony. One might well be skeptical that a sequential, linear composer could keep track of multiple levels of structure. In principle, however, the connectionist approach can; the present work is a test of whether it can do so in practice.

This type of linear technique has shown surprising and interesting results for natural-language processing. Elman (1990, 1993) has trained sequential networks on strings of letters or words and has found that the networks could infer grammatical and semantic structure, even recursive structure. However, this work has focused primarily on the sentence level; one can hardly imagine that it would scale up to handle, say, semantic structure of paragraphs or short stories. Fortunately, music is unlike natural language in several simplifying respects: the set of atomic elements is finite, relatively small and unambiguous, and psychoacoustic and stylistic regularities abound. Consequently, constraints among elements are stronger. Thus, it seems *a priori* plausible that at least several levels of structure might be inferrable by a linear technique for music.

4. The CONCERT Architecture

CONCERT is a recurrent network architecture that learns to behave as an autopredictor (Elman, 1990). A melody is presented to it, one note at a time, and its task at each point in time is to predict the next note in the melody. Using a training procedure described below, CONCERT's connection strengths are adjusted so that it can perform this task correctly for a set of training examples. Each example consists of a sequence of notes. The current note in the sequence is represented in the input layer of CONCERT, and the prediction of the next note is represented in the output layer. The input and output layers both represent three aspects of a note: its pitch, its duration and its harmonic chord accompaniment. As Figure 1 indicates, the next note is encoded in two different ways: the next-note-distributed (or NND) layer contains CONCERT's internal representation of the note—divided into three pools of units, forming distributed representations of pitch, duration and harmony—while the next-note-local (or NNL) layer contains one unit for each alternative pitch, duration and chord. The representation of a note in the NND layer, as well as in the input layer, is based on psychophysical studies, described further below. For now, it should suffice to say that this representation is distributed, i.e. a note is indicated by a pattern of activity across the units. Because such patterns of activity can be quite difficult to interpret, the NNL layer provides an alternative, explicit representation of the possibilities.

The context layer can represent relevant aspects of the input history, that is, the temporal context in which a prediction is made. When a new note is presented in

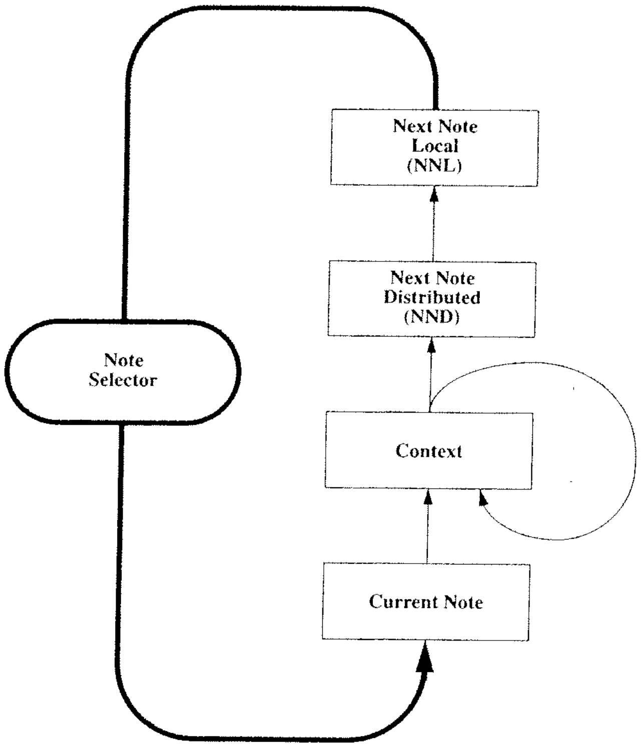


Figure 1. The CONCERT architecture. Rectangles indicate a layer of units, directed lines indicate full connectivity from one layer to another. The selection process is external to CONCERT and is used to choose among the alternatives proposed by the network during composition.

the input layer, the activity pattern currently in the context layer is integrated with the new note to form a new context representation. In general terms

$$\mathbf{c}(n) = f(\mathbf{c}(n-1), \mathbf{x}(n))$$

Where $\mathbf{x}(n)$ is a vector representing the nth note in the input sequence, $\mathbf{c}(n)$ is the context activity pattern following processing of input note n—which I refer to as step n—and f is a member of the class of functions that can be implemented by the connectionist hardware. At the start of each sequence the context layer is cleared, i.e. $\mathbf{c}(0) = 0$.

CONCERT could readily be wired up to behave as a kth order transition table. In this case, the function f is defined to implement a k-element stack in the context layer. This stack would hold on to notes $n - k + 1$ through n. The connections from the context layer to the output layer would then have to be set up to realize a look-up table in which each combination of previous notes maps to the appropriate probability distribution over the next note. However, the architecture is more general than a transition table because f is not limited to implementing a stack and the mapping from the context layer to the output need not be an arbitrary look-up table. From myriad possibilities, the training procedure attempts to find a set of connections that are adequate for performing the next-note prediction task. This involves determining which aspects of the input sequence are relevant for making future predictions and constructing the function f appropriately. Subsequently, the context layer will retain only task-relevant information. This contrasts with Todd's (1989) work on connectionist composition in which the recurrent context connections are pre-wired and fixed, which makes the nature of the information Todd's model retains independent of the examples on which it is trained.

Once CONCERT has been trained, it can be run in composition mode to create new pieces. This involves first seeding CONCERT with a short sequence of notes, perhaps the initial notes of one of the training examples. From this point on, the output of CONCERT can be fed back to the input, allowing CONCERT to continue generating notes without further external input. Generally, the output of CONCERT does not specify a single note with absolute certainty; instead, the output is a probability distribution over the set of candidates. It is thus necessary to select a particular note in accordance with this distribution. This is the role of the selection process depicted in Figure 1.

4.1. Unit Activation Rules

The activation rule for the context units is

$$c_i(n) = s\left[\sum_j w_{ij}\, x_j(n) + \sum_j v_{ij}\, c_j(n-1)\right] \tag{1}$$

where $c_i(n)$ is the activity of context unit i at step n, $x_j(n)$ is the activity of input unit j at step n, w_{ij} is the connection strength from unit j of the input to unit i of the context layer, v_{ij} is the connection strength from unit j to unit i within the context layer, and s is the standard logistic activation function rescaled to the range $[-1,1]$. Units in the NND layer follow a similar rule:

$$nnd_i(n) = s\left[\sum_j u_{ij}\, c_j(n)\right]$$

where $nnd_i(n)$ is the activity of NND unit i at step n and u_{ij} is the strength of connection from context unit j to NND unit i.

The NND and NNL representations can be broken into three component vectors, corresponding to pitch, duration and chord representations. The transformation from the NND pitch representation to the NNL pitch representation is described here; the transformation for the duration and chord representations is similar. The pitch transformation is achieved by first computing the distance between the NND pitch representation $\mathbf{nndp}(n)$, and the target (distributed) representation of each pitch i, $\mathbf{p}_i$:

$$d_i = \| \mathbf{nndp}(n) - \mathbf{p}_i \|$$

where $\|\cdot\|$ denotes the L_2 vector norm. This distance is an indication of how well the NND representation matches a particular pitch. The activation of the NNL unit corresponding to pitch i, $nnlp_i$, increases as the distance decreases:

$$nnlp_i(n) = \frac{e^{-d_i}}{\sum_j e^{-d_j}}$$

This normalized exponential transform was first proposed by Bridle (1990) and Rumelhart (in press). It produces an activity pattern over the NNL units in which each unit has activity in the range [0,1] and the activity of all units sums to 1. Consequently, the NNL activity pattern can be interpreted as a probability distribution—in this case, the probability that the next note has a particular pitch. An analogous transformation is performed to determine the activity of NNL units that represent note duration and accompanying chord. The distance measure and the exponential function have their basis in psychological theory (Shepard, 1987), a point that is elaborated on shortly.

4.2. *Training Procedure*

CONCERT is trained using a variation of the back-propagation algorithm (Rumelhart *et al.*, 1986) which adjusts the connection strengths within CONCERT so that the network can perform the next-note prediction task for a set of training examples. The algorithm requires first defining a measure of the network's performance—of how good a job the network does at predicting each note in each of the training examples. Commonly, a squared difference measure of error is used:

$$E_{lms} = \sum_{q,n,j} (nnlp_j(n, q) - \delta(j, P(n, q)))^2 + \sum_{q,n,j} (nnld_j(n, q) - \delta(j, D(n, q)))^2$$
$$+ \sum_{q,n,j} (nnlc_j(n, q) - \delta(j, C(n, q)))^2$$

where q is an index over pieces in the training set, n is an index over notes within a piece, and j is an index over pitch, duration or chord units in the NNL layer; $P(n, q)$, $D(n, q)$, $C(n, q)$ are the indices of the target pitch, duration and chord for note n of piece q; $\delta(a,b) = 1$ if $a = b$ or 0 otherwise. This measure is minimized when the outputs of the units corresponding to the correct predictions are 1 and the outputs of all other units are 0.

Another performance measure is sensible in the context of output units that have a probabilistic interpretation (Bridle, 1990; Rumelhart, in press). Because each NNL unit's output represents the probabilistic expectation of a pitch, performance depends on predicting the appropriate notes with high probability. This suggests the likelihood performance measure

$$L = \prod_{q,n} nnlp_{P(n,q)}(n, q)\; nnld_{D(n,q)}(n, q)\; nnlc_{C(n,q)}(n, q)$$

which is the joint probability of making the correct prediction for all notes of all pieces.[2] A log-likelihood criterion

$$E = -\log L = -\sum_{q,n} \log nnlp_{P(n,q)}(n, q) + \log nnld_{D(n,q)}(n, q) + \log nnlc_{C(n,q)}(n, q)$$

is used instead because it is easier to work with, and has the same extrema as L.

Back-propagation specifies how the weights in the network should be changed to reduce E. This involves computing the gradient of E with respect to the weights in the network: $\partial E/\partial \mathbf{W}$, $\partial E/\partial \mathbf{V}$, and $\partial E/\partial \mathbf{U}$. The first step in this process is computing the gradient with respect to the activity of units in the NND layer, and then propagating this gradient back to the weights in layers below. For the NND pitch units

$$\frac{\partial E}{\partial \mathbf{nndp}(n, q)} = \left[\frac{\mathbf{nndp}(n, q) - \mathbf{p}_{P(n,q)}}{d_{P(n,q)}} - \sum_i nnlp_i(n, q) \frac{\mathbf{nndp}(n, q) - \mathbf{p}_i}{d_i} \right]$$

Back-propagation still cannot be used to train CONCERT directly, because CONCERT contains recurrent connections and the algorithm applies only to feedforward networks. Several variations of the algorithm have been proposed for dealing with recurrent networks (Williams & Zipser, in press). Here, we use the 'back-propagation through time' (BPTT) procedure of Rumelhart *et al.* (1986), which transforms a recurrent network into an equivalent feedforward network.[3] This training procedure computes the true gradient of the objective function with respect to the various network weights. This means that if an input note at step l has any predictive utility at some later time n, then in principle the algorithm should adjust the connections so that note l is maintained in the network's memory. Contingencies over time should be discovered when they exist. There is a weaker version of BPTT that passes error back only a fixed number of steps (e.g. the training procedure used by Elman (1990)), which in principle makes contingencies across longer time spans more difficult to maintain and discover. CONCERT has been furnished with the most powerful connectionist recurrent-network learning procedure in order to endow it with the best possible chance of success.

5. **Representing Musical Elements**

Following this description of CONCERT's architecture, dynamics and training procedure, the issue of representing a musical piece arises. A piece is defined as a sequence of elements, each of which is characterized by a melody pitch, a melody duration and a harmonic chord accompaniment. The pitch and duration specify the notes of the melody, each of which is accompanied by a chord (or silence). This encoding synchronizes the melody and the harmony. Although chord changes generally occur at a slower rate than changes in the melody line, this is encoded simply by repeating chords for each note of the melody until the chord changes. The three elements—pitch, duration and chord representation—are discussed in turn.

5.1. *Pitch Representation*

To accommodate a variety of music, CONCERT needs the ability to represent a range of about four octaves. Using standard musical notation, these pitches are labeled as follows: C1, D1, . . ., B1, C2, D2, . . . B2, C3, . . . C5, where C1 is the lowest pitch and C5 the highest. Sharps and flats are denoted with ♯ and ♭, respectively, e.g. C♯3 and G♭2. Within an octave, there are twelve chromatic steps; the range C1–C5 thus includes 49 pitches.

Perhaps the simplest representation of pitch is to have one unit for each possibility. The pitch C1 would be represented by the activity vector $[1\ 0\ 0\ .\ .\ .]^T$, C♯1 by the vector $[0\ 1\ 0\ .\ .\ .]^T$, and so forth. An alternative would be to represent pitch by a single unit whose activity was proportional to the frequency of the pitch. One might argue that the choice of a pitch representation is not critical because back-propagation can, in principle, discover an alternative representation well suited to the task (Hinton, 1987). In practice, however, researchers have found that the choice of external representation is a critical determinant of the network's ultimate performance (e.g. Denker *et al.*, 1987; Mozer, 1987). Quite simply, the more task-appropriate information that is built into the network, the easier the job the learning algorithm has.

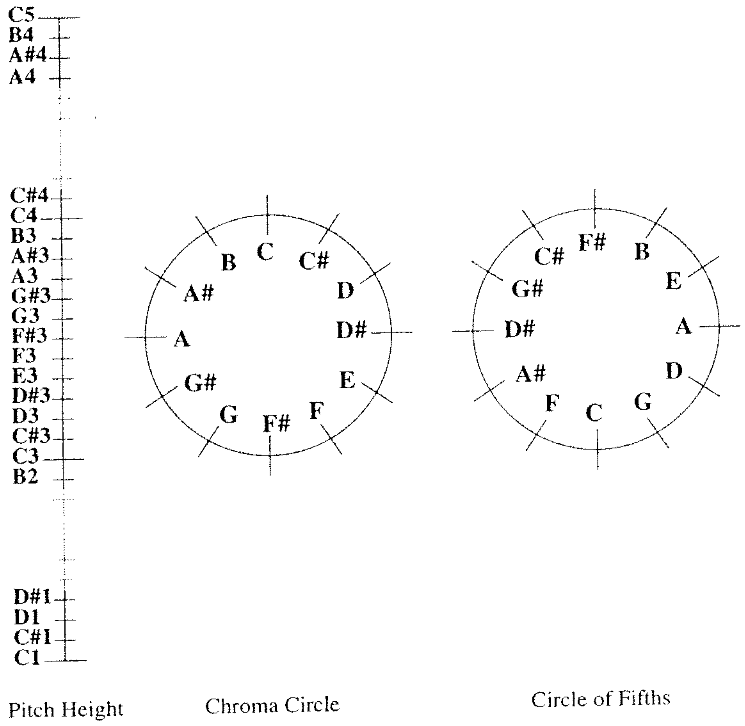
Laden and Keefe (1989) advocate the approach of including as much information as possible from psychoacoustics into the design of networks for music perception and cognition. They have developed a model of chord classification that categorizes triads as major, minor or diminished chords. Classification performance is superior with the use of a representation that explicitly encodes harmonics of the fundamental pitches.

In accord with this approach, and because CONCERT is being asked to make predictions about melodies that people have composed or to generate melodies that people perceive as pleasant, CONCERT has been furnished with a psychologically motivated representation of pitch. This means that notes that people judge to be similar should have similar representations in the network, indicating that the representation in the head matches the representation in the network. The local representation scheme proposed earlier clearly does not meet this criterion. In the local representation, every pair of pitches is equally similar (using either the distance or angle between vectors as a measure of similarity), yet people perceive pairs of notes like C1 and C♯1 to be more similar than, say, C1 and A4. Other obvious representations of pitch do not meet the criterion either. For example, a direct encoding of frequency does not capture the similarity that people hear between octaves.

Shepard (1982) has studied the similarity of pitches by asking people to judge the perceived similarity of pairs of pitches. He has proposed a theory of generalization (Shepard, 1987) in which the perceived similarity of two items decreases exponentially with the distance between them in an internal or 'psychological' representational space.[4] For the internal representation of pitch, Shepard has proposed a five-dimensional space, depicted in Figure 2. In this space, each pitch specifies a point along the pitch height (or PH) dimension, an (x,y) coordinate on the chroma circle (or CC) and an (x,y) coordinate on the circle of fifths (or CF). We will refer to this representation as PHCCCF, after its three components. The PH component specifies the logarithm of the frequency of a pitch; this logarithmic transform places tonal half-steps at equal spacing from one another along the PH axis. In the CC, neighboring pitches are a tonal half-step apart. In the CF, the perfect fifth of a pitch is the next pitch immediately counterclockwise.[5] The proximity of two pitches in the five-dimensional PHCCCF space can be determined simply by computing the Euclidean distance between their representations.

Shepard presents detailed arguments for the psychological validity of the PHCCCF representation. Let us briefly look at some of its benefits. Consider first the PH and CC components. In this three-dimensional subspace, pitches form a helix in which the winding of the helix is due to the CC and the height is due to the PH. As pitches proceed up the chromatic scale, they wind up the helix. Pitches exactly one octave apart are directly above one another on the helix; that is, they have the same locus on the CC but different values of PH. For this reason, octaves have similar representations. Depending on how the PH component is scaled relative to the CC (i.e. how elongated the helix is), pitches like C1 and C2 may even be closer in the representational space than pitches like C1 and B1, even though C1 is closer to B1 in frequency.

The CF endows the representation with other desirable properties. First, the circle localizes the tones in a musical key. Any seven adjacent tones correspond to a particular key. For instance, the tones of the C-major and A-minor diatonic scales—C, D, E, F, G, A and B—are grouped together on the CF. The most common pentatonic keys are similarly localized. Second, and perhaps more critical, the CF can explain the subjective equality of the intervals of the diatonic scale. To elaborate, Shepard points out that people tend to hear the successive steps of the major scale as equivalent, although with respect to log frequency, some of the intervals are only half as large as others. For example, in C major, the E–F and B–C steps are half tones apart (minor seconds) while all others are a whole tone apart (major seconds). The combination of the PH and the CF permits a representation in which the distance between all major and minor seconds is the same. This is achieved by using a scale ratio of approximately 3:1 for the CC relative to the CF.



Pitch Height Chroma Circle Circle of Fifths

Figure 2. Shepard's (1982) pitch representation.

One desirable property of the overall PHCCCF representation is that distances between pitches are invariant under transposition. Consider any two pitches, say, D2 and G♯4. Transposing the pitches preserves the distance between them in the PHCCCF representation. Thus, the distance from D2 to G♯4 is the same as from E2 to A♯4, from D1 to G♯3, and so forth. See Bharucha (1991) for a further discussion of the psychological issues involved in the representation of musical pitch.

The relative importance of the PH, CC and CF components can be varied by adjusting the diameters of the CC and the CF. For example, if the two circles have the same diameter, then, in terms of the CC and CF components, the distance between C and G is the same as the distance between C and B. This is because B is one notch from the C on the CC and five notches on the CF, while the G is five notches away on the CC and one on the CF. However, if the diameter of the CC is increased, then C is closer to B than to G (based on the distance in the four-dimensional CC and CF subspace); if the diameter is decreased, C is closer to G than to B. If the diameters of both circles are decreased relative to the PH scale, then pitch frequency becomes the most important determinant of similarity. Shepard argues that the weighting of the various components depends on the particular musical task and the listener's expertise. Based on Shepard's evidence, a reasonable representation for expert musicians is to weigh the CF and CC components equally, and to set the diameter of the CC and CF components equal to the distance of one octave in PH. This is the scale shown in Figure 2.

The final issue to discuss is how the PHCCCF representation translates into an activity vector over a set of connectionist units. A straightforward scheme is to use five units, one for pitch height and two pairs to encode the (x, y) coordinates of the pitch on the two circles.[6]

One problem with this scheme is that, if the units have the usual sigmoidal activation function, equal spacing of tones in pitch height or on the circles in unit activity space is not preserved in unit net input space. This means that context units attempting to activate NND units do not reap the full benefit of the representation (e.g. transposition invariance). A second problem with the simple five-unit scheme is that each unit encodes a coordinate value directly; there are seven discrete values for the *x*- and *y*-coordinates of the circles, 49 for the pitch height. Consequently, minor perturbations of the activity vector could lead to misinterpretations.

Due to these problems, an alternative representation of the CC and CF components has been adopted. The representation involves six binary-valued units to represent a tone on each circle; the representation for CC tones is shown in Table II. This representation preserves the essential distance relationships among tones on the CC: the distance between two tones is monotonically related to the angle between the tones. Because each unit has to encode only two distinct values, the representation is less sensitive to noise than is one in which each unit encodes a coordinate of the circle.

Unfortunately, there is no similar scheme that can be used to encode PH in a Boolean space of reasonably low dimensionality that preserves intrinsic distance relationships. Consequently, a single linear unit has been used for PH. Although this means that the PH unit can take on 49 distinct values, it is not critical that the unit represent a value with great accuracy. The PH unit essentially conveys information about the octave; information about the pitch within an octave can be gleaned from the values on the other dimensions. Consequently, a precise response of the PH unit is not crucial. Its activity is scaled to range from −9.798 for C1 to +9.798 for C5. This scaling achieves the desired property previously described that the distance in the CC or CF component between pitches on opposite sides of the circle equals the distance between pitches one octave apart in the PH component.[7]

Table II. Representation of tones on the CC

Tone	Representation					
C	−1	−1	−1	−1	−1	−1
C♯	−1	−1	−1	−1	−1	+1
D	−1	−1	−1	−1	+1	+1
D♯	−1	−1	−1	+1	+1	+1
E	−1	−1	+1	+1	+1	+1
F	−1	+1	+1	+1	+1	+1
F♯	+1	+1	+1	+1	+1	+1
G	+1	+1	+1	+1	+1	−1
G♯	+1	+1	+1	+1	−1	−1
A	+1	+1	+1	−1	−1	−1
A♯	+1	+1	−1	−1	−1	−1
B	+1	−1	−1	−1	−1	−1

Table III. PHCCCF representation for selected pitches

Pitch	PH	CC						CF					
C1	−9.978	+1	+1	+1	−1	−1	−1	−1	−1	−1	+1	+1	+1
F♯1	−7.349	−1	−1	−1	+1	+1	+1	+1	+1	+1	−1	−1	−1
G2	−2.041	−1	−1	−1	−1	+1	+1	−1	−1	−1	−1	+1	+1
C3	0	+1	+1	+1	−1	−1	−1	−1	−1	−1	+1	+1	+1
D♯3	1.225	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
E3	1.633	−1	+1	+1	+1	+1	+1	+1	−1	−1	−1	−1	−1
A4	8.573	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
C5	9.798	+1	+1	+1	−1	−1	−1	−1	−1	−1	+1	+1	+1
Rest	0	+1	−1	+1	−1	+1	−1	+1	−1	+1	−1	+1	−1

The PHCCCF representation consists of 13 units altogether. Sample activity patterns for some pitches are shown in Table III. Rests (silence) are assigned a unique code, listed in the last row of the table, that is maximally different from all pitches. The end of a piece is coded by a series of rests.

As with any distributed representation, there are limitations as to how many and which pitches can be represented simultaneously. The issue arises because the NND layer needs to be able to encode a set of alternatives, not just a single pitch. If, say, A1, D2 and E2 are equally likely as the next note, the NND layer must indicate all three possibilities. To do so, it must produce an activity vector that is nearer to $\mathbf{p}_{A1}$, $\mathbf{p}_{D2}$ and $\mathbf{p}_{E2}$ than to other possibilities. The point in PHCCCF space that is simultaneously closest to the three pitches is simply the average vector, $(\mathbf{p}_{A1} + \mathbf{p}_{D2} + \mathbf{p}_{E2})/3$. Table IV shows the pitches nearest to the average vector. As hoped for, A1, D2 and E2 are the nearest three. This is not always the case, though. Table V shows the pitches nearest to the average vector which represents the set {A1, D2, D♯2}. This illustrates the fact that certain clusters of pitches are more compact in the PHCCCF space than others. The PHCCCF representation not only introduces a similarity structure over the pitches, but also a limit on the combinations of pitches

Table IV. Distance from representation of {A1,D2,E2} to nearest 10 pitches

Rank	Pitch	Distance
1	D2	2.528
2	E2	2.779
3	A1	3.399
4	B1	3.859
5	C2	4.130
6	C♯2	4.422
7	A2	4.422
8	E1	4.441
9	G1	4.497
10	G2	4.497

Table V. Distance from representation of {A1, D2, D♯2} to nearest 10 pitches

Rank	Pitch	Distance
1	D2	2.373
2	C2	3.277
3	E2	3.538
4	C♯2	3.654
5	B1	3.714
6	D♯2	3.774
7	A1	3.946
8	F2	4.057
9	A♯1	4.146
10	G1	4.323

that can be considered simultaneously. Arbitrary limitations are a bad thing in general, but here the limitations are theoretically motivated.

One serious shortcoming of the PHCCCF representation is that it is based on the similarity between pairs of pitches presented in isolation. Listeners of music do not process individual notes in isolation; notes appear in a musical context which suggests a musical key which in turn contributes to an interpretation of the note. Some psychologically motivated work has considered the effects of context or musical key on pitch representation (Krumhansl, 1990; Krumhansl & Kessler, 1982; Longuet-Higgins, 1976, 1979). CONCERT could perhaps be improved by incorporating the ideas in this work. Fortunately, it does not require discarding the PHCCCF representation altogether, because the PHCCCF representation shares many properties in common with the representations suggested by Krumhansl and Kessler and by Longuet-Higgins.

5.2. Duration Representation

Although considerable psychological research has been directed toward the problem of how people perceive duration in the context of a rhythm (e.g. Fraisse, 1982; Jones & Boltz, 1989; Pressing, 1983), there appears to be no psychological theory of representation of individual note durations comparable to Shepard's work on pitch. Shepard suggests that there ought to be a representation of duration analogous to the PHCCCF representation, although no details are discussed in his work. The representation of duration built into CONCERT attempts to follow this suggestion.

The representation is based on the division of each beat (quarter-note) into twelfths. A quarter-note thus has a duration of 12/12, an eighth-note 6/12, an eighth-note triplet 4/12, and so forth (Figure 3). Using this characterization of note durations, a five-dimensional space can be constructed, consisting of three components (Figure 4). In this representation, each duration specifies a point along the duration height dimension, an (x, y) coordinate on the 1/3 beat circle and an (x, y) coordinate on the 1/4 beat circle. The duration height is proportional to the logarithm of the duration. This logarithmic transformation follows the general psychophysical law (Fechner's law) relating stimulus intensity to perceived sensation. The point on the $1/n$ beat circle is the duration after subtracting out the greatest integer multiple of $1/n$. For example, the duration 18/12 is represented by the point 2/12 on the 1/3 beat circle and the point 0/12 on the 1/4 beat circle. The two circles

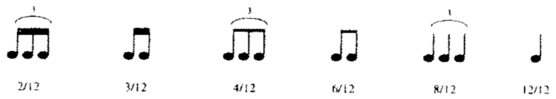


Figure 3. The characterization of note durations in terms of twelfths of a beat. The fractions shown correspond to the duration of a single note of a given type.

result in similar representations for related durations. For example, eighth-notes and quarter-notes (the former half the duration of the latter) share the same value on the 1/4 beat circle; eighth-note triplets and quarter-note triplets share the same value on the 1/3 beat circle; and quarter-notes and half-notes share the same values on both the 1/4 and 1/3 beat circles.

This five-dimensional space is encoded directly by five units in CONCERT. It was not necessary to map the 1/3 or 1/4 beat circle into a higher-dimensional binary space, as was done for the CC and the CF (Table II), because the beat circles are sparsely populated. Only two or three values need to be distinguished along the x and y-dimensions of each circle, which is well within the capacity of a single unit.

Several alternative approaches to rhythm representation are worthy of mention. A straightforward approach is to represent time implicitly by presenting each pitch on the input layer for a number of time steps proportional to the duration. Thus, a half-note might appear for 24 time steps, a quarter-note for 12, an eighth-note for 6. Todd (1989) followed an approach of this sort, although he did not quantize time so finely. He included an additional unit to indicate whether a pitch was articulated or tied to the previous pitch. This allowed for the distinction between, say, two successive quarter-notes of the same pitch and a single half-note. The drawback of this implicit representation of duration is that time must be sliced into

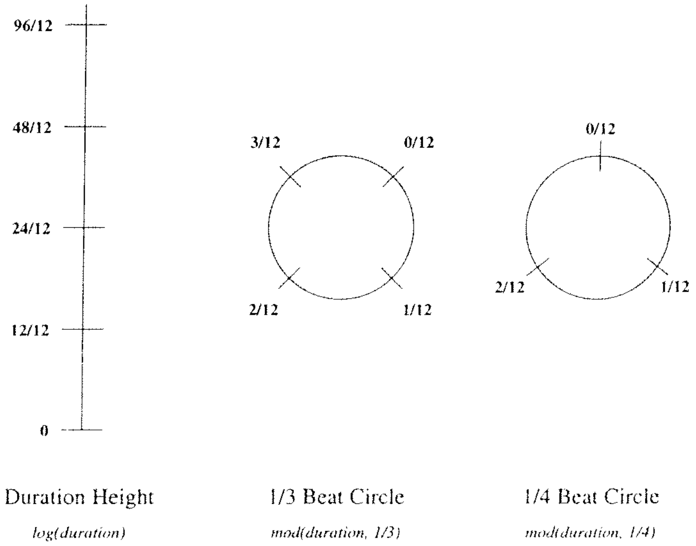


Figure 4. The duration representation used in CONCERT.

fairly small intervals to allow for a complete range of alternative durations, and as a result, the number of steps in a piece of music becomes quite large. This makes it difficult to learn contingencies among notes. For instance, if four successive quarter-notes are presented, to make an association between the first and the fourth, a minimum of 24 input steps must be bridged.

One might also consider a representation in which a note's duration is encoded with respect to the role it plays within a rhythmic pattern. This requires the explicit representation of larger rhythmic patterns. This approach seems quite promising, although it moves away from the note-by-note prediction paradigm that this work examines.

5.3. Chord Representation

The chord representation is based on Laden and Keefe's (1989) proposal for a psychoacoustically grounded distributed representation. Firstly, the Laden and Keefe proposal will be summarized and then it will be explained how it was modified for CONCERT.

The chords used here are in root position and are composed of three or four component pitches; some examples are shown in Table VI. Consider each pitch separately. In wind instruments, bowed string instruments and singing voices, a particular pitch will produce a harmonic spectrum consisting of the fundamental pitch (e.g. for C3, 440 Hz), and harmonics that are integer multiples of the fundamental (880 Hz, 1320 Hz, 1760 Hz and 2200 Hz). Laden and Keefe projected the continuous pitch frequency to the nearest pitch class of the chromatic scale, e.g. 440 to C3, 880 to C4, 1320 to G3, 1760 to C5 and 2200 to E5, where the projections to G3 and E5 are approximate. Using an encoding in which there is an element for each pure pitch class, a pitch was represented by activating the fundamental and the first four harmonics. The representation of a chord consisted of the superimposition of the representations of the component pitches. To allow for the range C3–C7, 49 elements are required. In Laden and Keefe's work, a neural network with this input representation better learns to classify chords as major, minor or diminished than a network with an acoustically neutral input representation.

Several modifications of this representation were made for CONCERT. The octave information was dropped, reducing the dimensionality of the representation from 49 to 12. The strength of representation of harmonics was exponentially weighted by their harmonic number; the fundamental was encoded with an activity of 1.0, the first harmonic with an activity of 0.5, the second harmonic with an activity of 0.25, and so forth. Activities were rescaled from a 0-to-1 range to a −1-to-1 range. Finally, an additional element was added to the representation based on a psychoacoustic study of perceived chord similarity. Krumhansl *et al.* (1982) found that people tended to judge the tonic, subdominant and dominant chords (i.e. C, F and G in the key of C) as being quite similar. This similarity was not present in the Laden and Keefe representation. Hence, an additional element was added to force these chords closer together. The element had value +1.5 for these three chords (as well as C7, F7 and G7), −1.5 for all other chords. Hierarchical clustering shows some of the similarity structure of the 13-dimensional representation (Figure 5).

Table VI. Elements of chords

Chord	Component pitches			
C major	C3	E3	G3	
C minor	C3	E♭3	G3	
C augmented	C3	E3	G♯3	
C diminished	C3	E♭3	G♭3	
C7	C3	E3	G3	B♭3

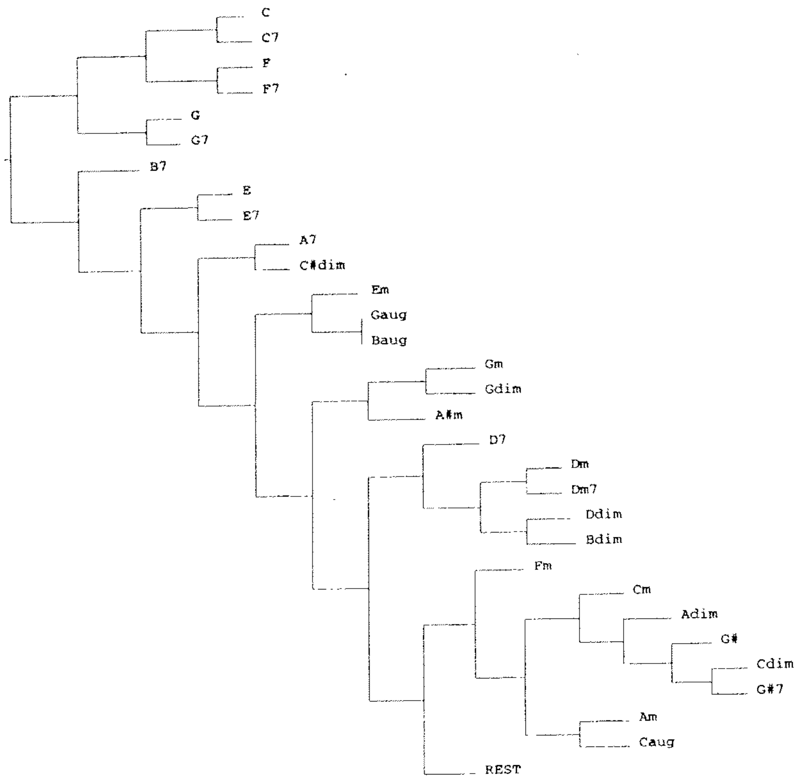


Figure 5. Hierarchical clustering of the representations of various chords used in simulation studies.

6. Basic Simulation Results

Many decisions had to be made in constructing CONCERT. In pilot simulation experiments, various aspects of CONCERT were explored, including variants in the representations, such as using two units to represent the circles in the PHCCCF representation instead of six; alternative error measures, such as the mean-squared error; and the necessity of the NNL layer. Empirical comparisons supported the architecture and representations described earlier.

One potential pitfall in the research area of connectionist music composition is the uncritical acceptance of a network's performance. It is absolutely essential that a network be evaluated according to some objective criterion. One cannot judge the enterprise to be a success simply because the network is creating novel output. Even random note sequences played through a synthesizer sound interesting to many listeners. Thus an examination of CONCERT's performance was begun by testing CONCERT on simple artificial pitch sequences, with the aim of verifying that it can discover the structure known to be present. In these sequences, there was no variation in duration and harmony; consequently, the duration and chord components of the input and output were ignored.

6.1. *Extending a C-major Diatonic Scale*

To start with a simple experiment, CONCERT was trained on a single sequence consisting of three octaves of a C-major diatonic scale: C1 D1 E1 F1 . . . B3. The target at each step was the next pitch in the scale: D1 E1 F1 G1 . . . C4. CONCERT is said to have learned the sequence when, at each step, the activity of the NNL unit representing the target at that step is more active than any other NNL unit. In 10 replications of the simulation with different random initial weights, 15 context units, a learning rate of 0.005 and no momentum, CONCERT learned the sequence in about 30 passes. Following training, CONCERT was tested on four octaves of the scale. CONCERT correctly extended its predictions to the fourth octave, except that in 4 of the 10 replications, the final note, C5, was transposed down an octave. Table VII shows the CONCERT's output for two octaves of the scale. Octave 3 was part of the training sequence, but octave 4 was not. Activities of the three most active NNL pitch units are shown. Because the activities can be interpreted as probabilities, one can see that the target is selected with high confidence.

CONCERT was able to learn the training set with as few as two context units, although surprisingly, generalization performance tended to improve as the number of context units was increased. CONCERT was also able to generalize from a two-octave training sequence, but it often transposed pitches down an octave.

6.2. *Learning the Structure of Diatonic Scales*

In this simulation, CONCERT was trained on a set of diatonic scales in various keys over a one-octave range, e.g. D1 E1 F♯1 G1 A1 B1 C♯2 D2. Thirty-seven such

scales can be made using pitches in the C1–C5 range. The training set consisted of 28 scales—roughly 75% of the corpus—selected at random, and the test set consisted of the remaining 9. In 10 replications of the simulation using 20 context units, CONCERT mastered the training set in approximately 55 passes. Generalization performance was tested by presenting the scales in the test set one pitch at a time and examining CONCERT's prediction. This is not the same as running CONCERT in composition mode because CONCERT's output was not fed back to the input; instead, the input was a predetermined sequence. Of the 63 pitches to be predicted in the test set, CONCERT achieved remarkable performance: 98.4% correct. The few errors were caused by transposing pitches one full octave or one tonal half-step.

Table VII. Performance on octaves 3 and 4 of C-major diatonic scale

Input pitch	Output unit activities					
C3	D3	0.961	C3	0.017	E3	0.014
D3	E3	0.972	D3	0.012	F3	0.007
E3	F3	0.982	D♯3	0.008	G3	0.006
F3	G3	0.963	F3	0.015	A3	0.010
G3	A3	0.961	G3	0.024	B3	0.012
A3	B3	0.972	A3	0.025	C4	0.002
B3	C4	0.979	A♯3	0.010	C♯4	0.005
C4	D4	0.939	C4	0.040	E4	0.009
D4	E4	0.968	D4	0.018	F4	0.006
E4	F4	0.971	D♯4	0.016	E4	0.005
F4	G4	0.931	F4	0.037	F♯4	0.015
G4	A4	0.938	G4	0.044	B4	0.007
A4	B4	0.915	A4	0.080	A♯4	0.003
B4	C5	0.946	A♯4	0.040	B4	0.011

To compare CONCERT with a transition-table approach, a second-order transition table was built from the training set data and its performance measured on the test set. The transition-table prediction (i.e. the pitch with highest probability) was correct only 26.6% of the time. The transition table is somewhat of a straw man for this task: a transition table that is based on absolute pitches is simply unable to generalize correctly. Even if the transition table encoded relative pitches, a third-order table would be required to master the environment. Kohonen's musical grammar faces the same difficulties as a transition table.

A version of CONCERT was tested using a local pitch representation in the input and NND layers instead of the PHCCCF representation. The local representation had 49 pitch units, one per tone. Although the NND and NNL layers may seem somewhat redundant with a local pitch representation, the architecture was not changed to avoid confounding the comparison between representations with other possible factors. Testing the network in the manner described above, generalization performance with the local representation and 20 context units was only 54.4%. Experiments with smaller and larger numbers of context units resulted in no better performance.

6.3. *Learning Random Walk Sequences*

In this simulation, 10-element sequences were generated according to a simple rule: the first pitch was selected at random, and then successive pitches were either one step up or down the C-major scale from the previous pitch, the direction chosen at random. The pitch transitions can easily be described by a transition table, as illustrated in Table I. CONCERT, with 15 context units, was trained for 50 passes through a set of 100 such sequences. If CONCERT has correctly inferred the underlying rule, its predictions should reflect the plausible alternatives at each point in a sequence. To test this, a set of 100 novel random walk sequences was presented. After each note *n* of a sequence, CONCERT's performance was evaluated by matching the top two predictions—the two pitches with highest activity—against the actual note *n* + 1 of the sequence. If note *n* + 1 was not one of the top two predictions, the prediction was considered to be erroneous. In 10 replications of the simulation, the mean performance was 99.95% correct. Thus, CONCERT was clearly able to infer the structure present in the patterns. CONCERT performed equally well, if not better, on random walks in which chromatic steps (up or down a tonal half step) were taken. CONCERT with a local representation of pitch achieved 100% generalization performance.

6.4. *Learning Interspersed Random Walk Sequences*

The sequences in this simulation were generated by interspersing the elements of two simple random walk sequences. Each interspersed sequence had the following form: a_1, b_1, a_2, b_2, . . ., a_5, b_5, where a_1 and b_1 are randomly selected pitches, a_{i+1} is one step up or down from a_i on the C-major scale, and likewise for b_{i+1} and b_i. Each sequence consisted of 10 pitches. CONCERT, with 25 context units, was trained on 50 passes through a set of 200 examples and was then tested on an additional 100. In contrast to the simple random walk sequences, it is impossible to predict the second pitch in the interspersed sequences (b_1) from the first (a_1). Thus, this prediction was ignored for the purpose of evaluating CONCERT's performance. CONCERT achieved a performance of 94.8% correct. Excluding errors that resulted from octave transpositions, performance improves to 95.5% correct. CONCERT with a local pitch representation achieves a slightly better performance of 96.4%.

To capture the structure in this environment, a transition-table approach would need to consider at least the previous two pitches. However, such a transition table is not likely to generalize well because, if it is to be assured of predicting a note at step n correctly, it must observe the note at step $n-2$ in the context of every possible note at step $n-1$. Hence, a second-order transition table was constructed from CONCERT's training set. Using a testing criterion analogous to that used to evaluate CONCERT, the transition table achieved a performance level on the test set of only 67.1% correct. Kohonen's musical grammar would face the same difficulty as the transition table in this environment.

6.5. *Learning AABA Phrase Patterns*

The melodies in this simulation were formed by generating two phrases, call them A and B, and concatenating the phrases in an AABA pattern. The A and B phrases consisted of five-note ascending chromatic scales, the first pitch selected at random. The complete melody then consisted of 21 elements—four phrases of five notes followed by a rest marker—an example of which is

F♯2 G2 G♯2 A2 A♯2 F♯2 G2 G♯2 A2 A♯2 C4 C♯4 D4 D♯4 E4 F♯2 G2 G♯2 A2 A♯2 REST.

These melodies are simple examples of sequences that have both fine and coarse structure. The fine structure is derived from the relations among pitches within a phrase, the coarse structure is derived from the relations among phrases. This pattern set was designed to examine how well CONCERT could cope with multiple levels of structure and long-term dependencies, of the sort that is found (albeit to a much greater extent) in real music.

CONCERT was tested with 35 context units. The training set consisted of 200 examples and the test set another 100 examples. Ten replications of the simulation were run for 300 passes through the training set.

Because of the way that the sequences were organized, certain pitches could be predicted based on local context, whereas other pitches required a more global memory of the sequence. In particular, the second to fifth pitches within a phrase could be predicted based on knowledge of the immediately preceding pitch. To predict the first pitch in the repeated A phrases and to predict the rest at the end of a sequence, more global information is necessary. Thus, the analysis was split to distinguish between pitches that required only local structure and pitches that required more global structure. Generalization performance was 97.3% correct for the local components, but only 58.4% for the global components.

6.6. *Discussion*

Through the use of simple, structured training sequences, it is possible to evaluate the performance of CONCERT. The initial results from CONCERT are encouraging. CONCERT is able to learn structure in short sequences with strong regularities, such as a C-major scale and a random walk in pitch. Two examples of structure were presented that CONCERT can learn but that cannot be captured by a simple transition table or by Kohonen's musical grammar. One example involved diatonic scales in various keys, the other involved interspersed random walks.

CONCERT clearly benefits from its psychologically grounded representation of pitch. In the task of extending the C-major scale, CONCERT with a local pitch representation would simply fail. In the task of learning the structure of diatonic scales, CONCERT's generalization performance drops by nearly 50% when a local representation is substituted for the PHCCCF representation. The PHCCCF representation is not always a clear win: generalization performance on random walk sequences improves slightly with a local representation. However, this result is not inconsistent with the claim that the PHCCCF representation assişts CONCERT in learning structure present in human-composed melodies. The reason is that random walk sequences are hardly based on the sort of musical conventions that gave rise to the PHCCCF representation, and hence the representation is unlikely to be beneficial. In contrast, musical scales are at the heart of many musical conventions; it makes sense that scale-learning profits from the PHCCCF representation. Human-composed melodies should fare similarly. Beyond its ability to capture aspects of human pitch perception, the PHCCCF representation has the advantage over a local representation of reducing the number of free parameters in the network. This can be an important factor in determining network generalization performance.

The result from the AABA-phrase experiment is disturbing, but not entirely surprising. Consider the difficulty of correctly predicting the first note of the third repetition of the A phrase. The listener must remember not only the first note of the A phrase, but also that the previous phrase has just ended (that five consecutive notes ascending the chromatic scale were immediately preceding) and that the current phrase is not the second or fourth one of the piece (i.e. that the next note starts the A phrase). Minimally, this requires a memory that extends back 11 notes. Moreover, most of the intervening information is irrelevant.

7. Capturing Higher-order Musical Organization

In principle, CONCERT trained with back-propagation should be capable of discovering arbitrary contingencies in temporal sequences, such as the global structure in the AABA phrases. In practice, however, many researchers have found that back-propagation is not sufficiently powerful, especially for contingencies that span long temporal intervals and that involve high-order statistics. For example, if a network is trained on sequences in which one event predicts another, the relationship is not hard to learn if the two events are separated by only a few unrelated intervening events, but as the number of intervening events grows, a point is quickly reached where the relationship cannot be learned (Mozer, 1989, 1992, 1993; Schmidhuber, 1992). Bengio *et al.* (1994) present theoretical arguments for inherent limitations of learning in recurrent networks.

This poses a serious limitation on the use of back-propagation to induce musical structure in a note-by-note prediction paradigm because important structure can be found at long time-scales as well as short. A musical piece is more than a linear string of notes. Minimally, a piece should be characterized as a set of musical phrases, each of which is composed of a sequence of notes. Within a phrase, local structure can probably be captured by a transition table, e.g. the fact that the next note is likely to be close in pitch to the current, or that if the past few notes have been in ascending order, the next note is likely to follow this pattern. Across phrases, however, a more global view of the organization is necessary.

The difficult problem of learning coarse as well as fine structure has been addressed recently by connectionist researchers (Mozer, 1992; Mozer & Das, 1993; Myers, 1990; Ring, 1992; Schmidhuber, 1992; Schmidhuber *et al.*, 1993). The basic idea in many of these approaches involves building a reduced description (Hinton, 1988) of the sequence that makes global aspects more explicit or more readily detectable. In the case of the AABA structure, this might involve taking the sequence of notes composing A and redescribing them simply as 'A'. Based on this reduced description, recognizing the phrase structure AABA would involve little more than recognizing the sequence AABA. By constructing the reduced description, the problem of detecting global structure has been turned into the simpler problem of detecting local structure.

The challenge of this approach is to devise an appropriate reduced description. I describe here the simple scheme in Mozer (1992) as a modest step toward a solution. This scheme constructs a reduced description that is a bird's eye view of the musical piece, sacrificing a representation of individual notes for the overall contour of the piece. Imagine playing back a song on a tape recorder at double the regular speed. The notes are to some extent blended together and indistinguishable. However, events at a coarser time-scale become more explicit, such as a general ascending trend in pitch or a repeated progression of notes. Figure 6 illustrates the idea. The curve in Figure 6(a), depicting a sequence of individual pitches, has been smoothed and compressed to produce the graph in Figure 6(b). Mathematically, 'smoothed and compressed' means that the waveform has been low-pass filtered and sampled at a lower rate. The result is a waveform in which the alternating upwards and downwards flow is unmistakable.

Multiple views of the sequence can be realized in CONCERT using context units that operate with different time constants. With a simple modification to the context unit activation rule (equation (1))

$$c_i(n) = \tau_i c_i(n-1) + (1-\tau_i)\, s\left[\sum_j w_{ij}\, x_j(n) + \sum_j v_{ij}\, c_j(n-1)\right] \qquad (2)$$

where each context unit i has an associated time constant, τ_i, that ranges from 0 to 1 and determines the responsiveness of the unit—the rate at which its activity changes. With $\tau_i = 0$, the activation rule reduces to equation (1) and the unit can sharply change its response based on a new input. With large τ_i, the unit is sluggish, holding on to much of its previous value and thereby averaging the response to the net input over time. At the extreme of $\tau_i = 1$, the second term drops out and the

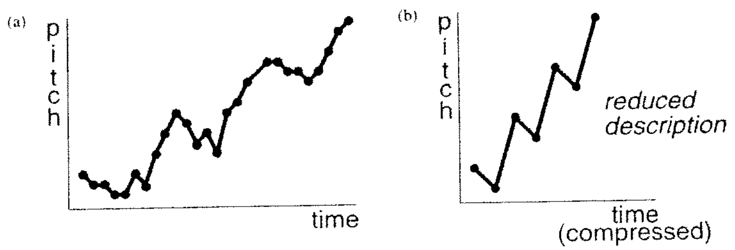


Figure 6. (a) A sequence of individual notes. The vertical axis indicates the pitch, the horizontal axis the time. Each point corresponds to a particular note. (b) A smoothed, compact view of the sequence.

unit's activity becomes fixed. Thus, large τ_i smooth out the response of a context unit over time. This is one property of the waveform in Figure 6(b) relative to the waveform in Figure 6(a).

The other property, the compactness of the waveform, is also achieved by a large τ_i, although somewhat indirectly. The key benefit of the compact waveform in Figure 6(b) is that it allows a longer period of time to be viewed in a single glance, thereby explicating contingencies occurring during this interval. Equation (2) also facilitates the learning of contingencies over longer periods of time. To see why this is the case, consider the relation between the error derivative with respect to the context units at step n, $\partial E/\partial \mathbf{c}(n)$, and the error back-propagated to the previous step, $n-1$. One contribution to $\partial E/\partial c_i(n-1)$, from the first term in equation (2), is

$$\frac{\partial E}{\partial c_i(n)}\,\frac{\partial}{\partial c_i(n-1)}\,[\tau_i c_i(n-1)] = \tau_i\,\frac{\partial E}{\partial c_i(n)}$$

This means that when τ_i is large, most of the error signal in context unit i at note n is carried back to note $n-1$. Thus, the back-propagated error signal can make contact with points further back in time, facilitating the learning of more global structure in the input sequence.

Several comments regarding this approach are now given:

- Figure 6 is inaccurate in that it represents a filtering of the raw input sequence. The idea proposed here actually involves a filtering of the transformed input (i.e. the representation in the context layer). Regardless, the basic intuition applies in either case.
- Time constants have been incorporated into the activation rules of other connectionist architectures. Most pertinent is Bharucha and Todd's (1989) use of fixed time constants to implement an exponentially decaying memory of input context. McClelland's (1979) cascade model makes use of time constants in a feedforward network. The continuous-time networks of Pearlmutter (1989) and Pineda (1987) are based on a differential equation update rule, of which Equation (2) is a discrete-time version. Mozer (1989) proposed an architecture for sequence recognition that included linear-integrator units. However, none of this work has exploited time constants to control the temporal responsivity of individual units.

- Although Figure 6 depicts only two time-scales, context units can operate at many different time-scales, with smaller values of τ_i specializing the units to be sensitive to local properties of the sequence and larger values specializing the units to be sensitive to more global properties. One obvious possibility is to use back-propagation to determine the appropriate values of τ_i; however, my suspicion is that there are many local optima blocking the path to the best τ_i.
- This approach specializes each context unit for a particular time-scale. Nonetheless, it allows for interactions between representations at different time-scales, as each context unit receives input from all others. Thus, CONCERT can in principle learn to encode relationships between structure at different scales.
- Equation (2) suggests one particular type of reduced description, consisting of a smoothed and compressed representation of the context unit response over time. This is a simple-minded reduced description; ideally, one would like the reduced description to characterize meaningful 'chunks' or events in the input sequence. This idea is expanded later in the article.

7.1. AABA Phrase Patterns Revisited

In the experiment with AABA phrase patterns described earlier, the CONCERT architecture contained 35 context units, all with $\tau = 0$. In this experiment, called the reduced description (or RD) version, 30 context units had $\tau = 0$ and 5 had $\tau = 0.8$. The experiment was otherwise identical. Table VIII compares the generalization performance of the original and RD networks, for predictions involving local and global structure. Performance involving global structure was significantly better for the RD version ($F(1, 9) = 179.8$, $p < 0.001$), but there was only a marginally reliable difference for performance involving local structure ($F(1, 9) = 3.82$, $p = 0.08$). The global structure can be further broken down to prediction of the end of the sequence and prediction of the first pitch of the repeated A phrases. In both cases, the performance improvement for the RD version was significant: 88.0% versus 52.9% for the end of sequence ($F(1, 9) = 220$, $p < 0.001$); 69.4% versus 61.2% for the first pitch ($F(1, 9) = 77.6$, $p < 0.001$).

Experiments with different values of τ in the range 0.7–0.95 yielded qualitatively similar results, as did experiments in which the A and B phrases were formed by random walks in the key of C. Overall, modest improvements in performance are observed, yet the global structure is never learned as well as the local, and it is clear that CONCERT's capabilities are no match for those of people in this simple domain.

8. Larger Simulation Experiments

In the next three sections, simulations are described that use real music as training data, and the reduced description technique described in the previous section is used.

Table VIII. Performance on AABA phrases

Structure	Original net (%)	RD net (%)
Local	97.3	96.7
Global	58.4	75.6

Table IX. Bach training examples

Piece	Number of notes
Minuet in G major (no. 1)	126
Minuet in G major (no. 2)	166
Minuet in D minor	70
Minuet in A minor	84
Minuet in C minor	80
March in G major	153
March in D major	122
March in Eb major	190
Musette in D major	128
Little prelude in C major	121

8.1. Composing Melodies in the Style of Bach

The melody lines of 10 simple pieces by J. S. Bach were used to train CONCERT (Table IX). The set of pieces is not particularly coherent; it includes a variety of musical styles. The primary thing that the pieces have in common is their composer. The original pieces had several voices, but the melody generally appeared in the treble voice. Importantly, to naïve listeners, the extracted melodies sounded pleasant and coherent without the accompaniment.

In the training data, each piece was terminated with a sequence of three rests. This allowed CONCERT to learn not only the notes within a piece but also when the end of the piece was reached. Further, each major piece was transposed to the key of C major and each minor piece to the key of A minor. This was done to facilitate learning because the pitch representation does not take into account the notion of musical key; hopefully, a more sophisticated pitch representation would avoid the necessity of this step.

Two fixed input units were included in this simulation. One indicated whether the piece was in a major versus minor key, another whether the piece was in 3/4 meter versus 2/4 or 4/4. These inputs did not change value for a given piece. To keep CONCERT on beat, an additional input unit was active for notes that were on the downbeat of each measure. If a note was tied from one measure to the next, it was treated as two events; this ensured that each downbeat would correspond to a distinct input event.[8]

Learning the examples involves predicting a total of 1260 notes altogether. CONCERT was trained with 40 hidden units, 35 with $\tau = 0$ and 5 with $\tau = 0.8$, for 3000 passes through the training set. The learning rate was gradually lowered from 0.0004 to 0.0002. By the completion of training, CONCERT could correctly predict about 95% of the pitches and 95% of the durations. Attempts were made to train CONCERT with a greater proportion of RD context units and with greater τ values, but training performance suffered.

New pieces were created by presenting the first four notes of one of the training examples to seed CONCERT, and then allowing CONCERT to run in composition mode. Selection was made independently for the pitch and duration of each note, according to their respective probability distributions, with the additional constraint that durations were ruled invalid if the resulting note crossed measure boundaries (this was allowed only when the component of the duration representation indicating a tied note was active). Two examples of compositions produced by CONCERT are shown in Figure 7. CONCERT specifies the end of a composition by producing

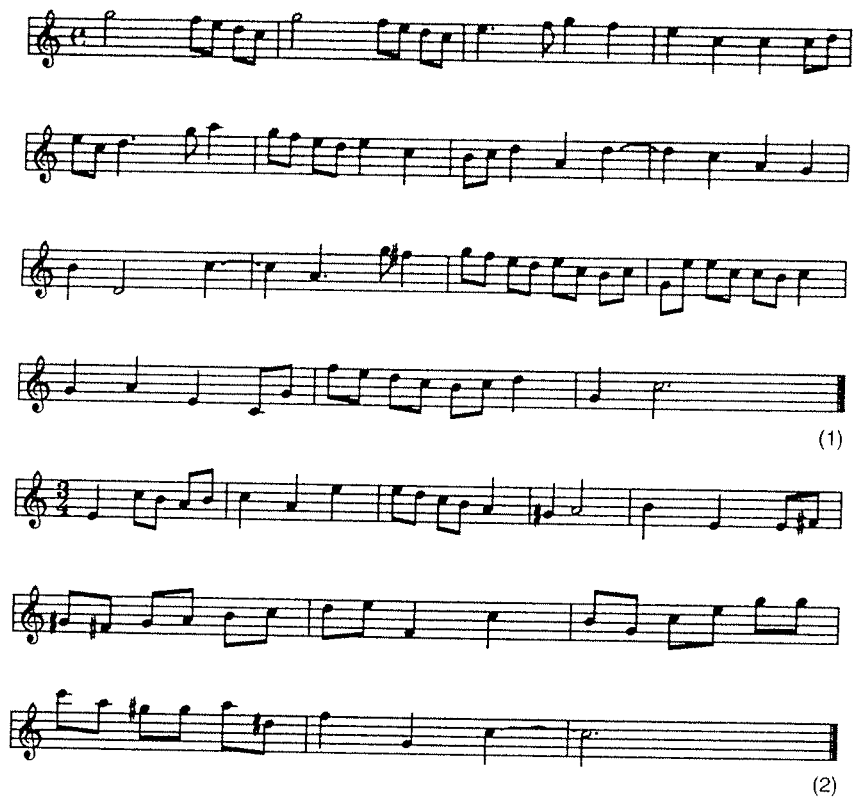


Figure 7. Two sample compositions produced by CONCERT based on the Bach training set.

a sequence of rests. The compositions rapidly diverge from the training example used as seed, due to the probabilistic note selection process, although some compositions occasionally contain brief excerpts from the training examples.

Compositions were also generated based on the Bach examples using a transition-table approach. Two third-order transition tables were built from the training data, one for durations and one for pitches. (Combining the two in a single table is not feasible because the resulting table is too large—over 110 million cells—and is too sparse to be useful.) Twelve musically untrained listeners were asked to state their preference for the CONCERT compositions or the transition-table compositions. Two representative examples of each technique were played. Order of presentation was counterbalanced across listeners. All twelve chose CONCERT, some with ambivalence, others with a strong preference. Listeners commented that the CONCERT compositions were more coherent and had a more consistent beat. The final cadences of the CONCERT composition were also noted as superior, no doubt because at this point in the piece CONCERT did actually consider more than third-order structure.

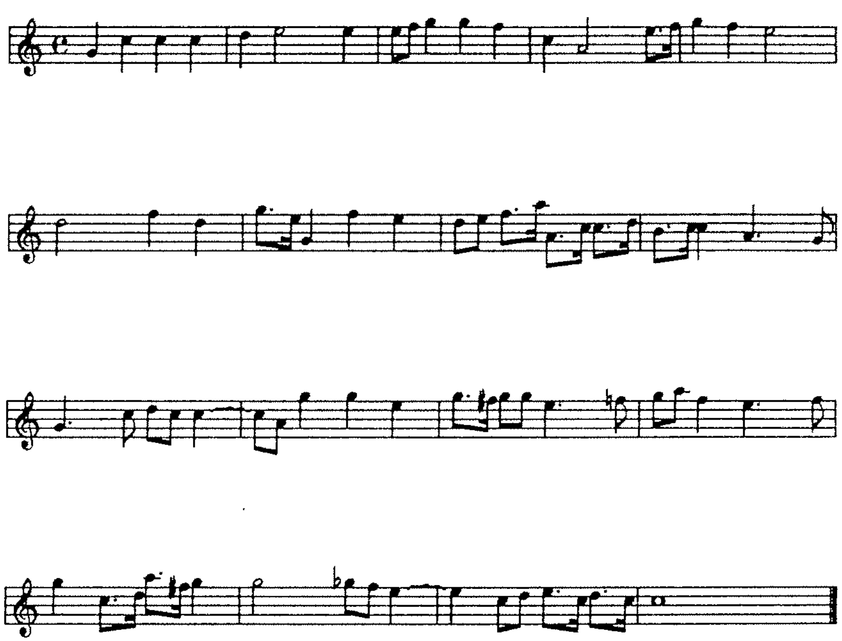


Figure 8. A sample composition produced by CONCERT based on a training set of traditional European folk melodies.

8.2. Composing Melodies in the Style of European Folk Tunes

In a second experiment, CONCERT was trained on a set of 25 traditional European folk melodies (included in *Der Fluiten Luschof* of J. van Eyck). The pieces were somewhat shorter than the Bach examples, having an average of 74 notes per piece. All pieces were in the key of C major and had 4/4 meter. Because of the uniform mode and meter of the pieces, there was no need to represent these features explicitly on the input, as was done for the Bach examples.

CONCERT was trained with 50 hidden units, 45 with $\tau = 0$ and 5 with $\tau = 0.8$, for about 2000 passes through the training set. By the completion of training, CONCERT could correctly predict 93% of the pitches and 90% of the durations in the training set. Compositions using the trained network sounded reasonable, occasionally having more appeal than the training examples. Figure 8 shows a sample composition.

8.3. Explicit Training on Higher-order Structure: The Waltz Experiment

The two experiments with real musical data were a mixed success. Although clearly superior to a third-order transition table, CONCERT failed to produce music with high-order structure. One cannot claim a qualitative difference between the CONCERT and transition-table compositions.

The above experiments utilized only five RD context units. One might conjecture that higher-level structure might be learned better with more RD units.

However, even with no RD units, CONCERT achieved a training performance of 95%; replacing ordinary context units with RD units did not improve, and generally harmed, training performance. The problem appears to be that the prediction task can be performed very well using only local structure.

To encourage the consideration of structure on a longer time-scale, a final experiment introduced harmonic accompaniment to the melody line. The training examples were a set of 25 waltzes by various composers, collected in a 'Fakebook' with a rhythmically simplified melody line and accompanying chord progressions. The change in chords occurred at a slower rate than the change in notes of the melody: in the training set, the average duration of a melody note was 1.4 beats, while the average duration between chord changes was 5.9 beats. Consequently, one might hope that in order to learn the structure of the chord progression, it would be necessary to span longer periods of time, and, hence, it would be necessary for CONCERT to extract higher-level structure from the pieces.

Each note in the input was accompanied by the current chord. Chord duration was not explicitly represented; the duration of a chord was simply the sum of the durations of the consecutive notes that were associated with the chord.

Figure 9 shows two compositions produced by CONCERT based on the waltz data set. There was little evidence in the compositions that significant global structure was learned.

8.4. Discussion

While CONCERT performs well on simple, structured, artificial sequences, the prognosis looks bleaker for natural music. One critic described CONCERT's creations as "compositions only their mother could love". To summarize more delicately, few listeners would be fooled into believing that the pieces had been composed by a human. While the local contours made sense, the pieces were not musically coherent, lacking thematic structure and having minimal phrase structure and rhythmic organization.

It appears that the CONCERT architecture and training procedure do not scale well as the length of the pieces grows and as the amount of higher-order structure increases. This comes as no surprise to some: learning the structure of music through a note-by-note analysis is formally the same task as learning to read and understand English from a letter-by-letter analysis of text. Nonetheless, many researchers are pursuing a linear note-by-note approach to music analysis and composition. This is not entirely naïve, as connectionist algorithms are in principle capable of discovering the multiple levels of structure in music. However, the experiments reported here show no cause for optimism in practice, despite the use of state-of-the-art connectionist architectures and training algorithms, and attempts to encourage CONCERT to learn global structure.

The present results clearly signal difficulty in using sequential networks to match transition probability distributions of arbitrary order. Even more sophisticated approaches are clearly needed. Several directions are mentioned that one might consider in the domain of music analysis and composition.

- *Multiple recurrent hidden layers.* The current CONCERT architecture is limited in that the update rule for the context layer is a squashed linear mapping. That is, the new context is a linear function of the old context and current input, passed through a squashing function. To give CONCERT more flexibility in

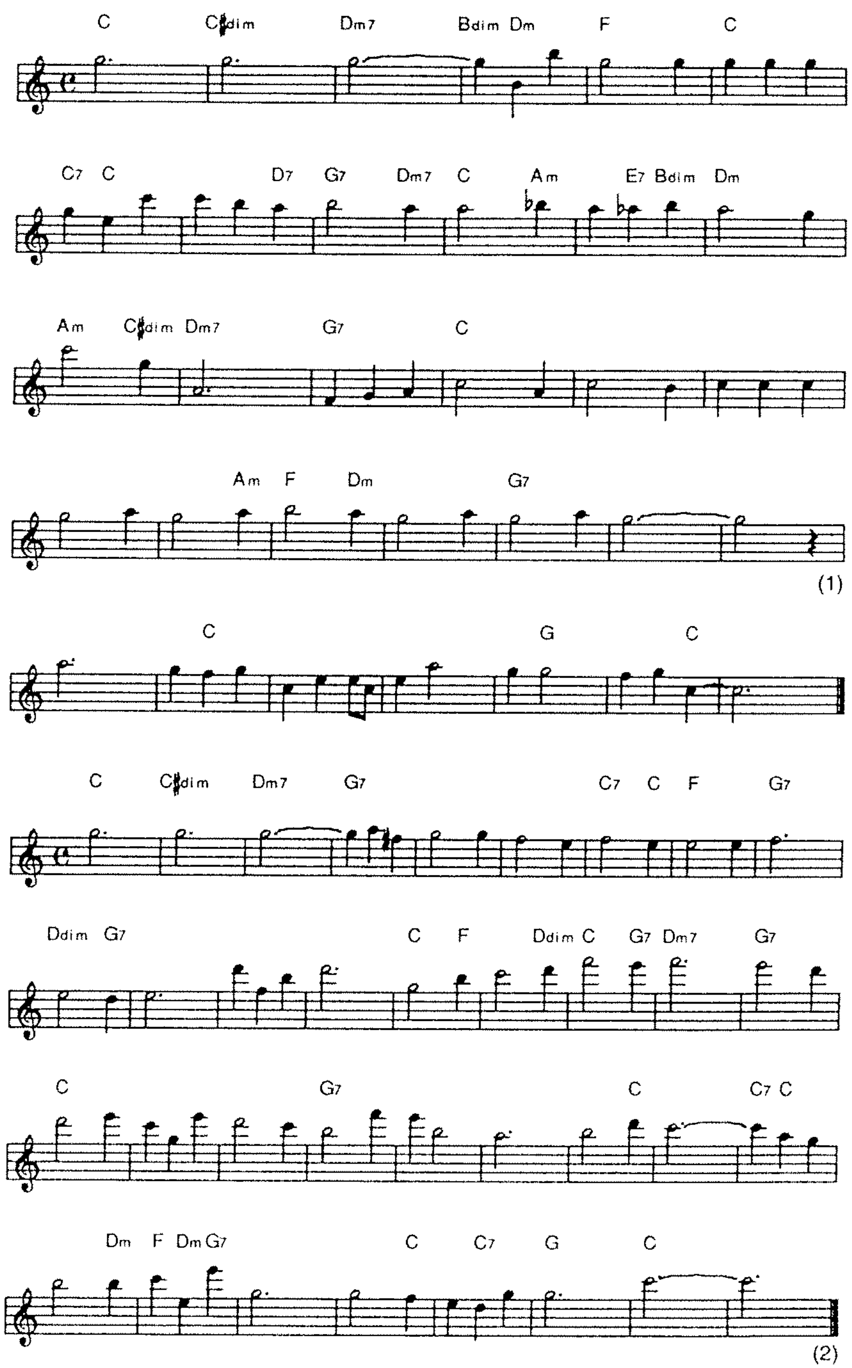


Figure 9. Two compositions produced by CONCERT based on a training set of waltzes. For both compositions, CONCERT was given the same initial three measures, but the compositions rapidly diverged.

the update rule, one might consider multiple hidden layers in the recurrent loop (for related work, see Cottrell & Tsung, 1993; Tsung & Cottrell, 1993).

- *Alternative approaches to remembering the past.* There are many connectionist approaches to constructing a memory of past events for the purpose of predicting the future (Mozer, 1993; Principe *et al.*, 1994). CONCERT with the RD units would be considered a TIS-exponential architecture according to Mozer's taxonomy. Another promising approach might be to include a small buffer in the input and/or context layers to hold a history of recent activities, allowing local temporal structure to be learned without wasting the resources of the recurrent connections.
- *Cascaded sequential networks.* Burr and Miyata (1993) and Todd (1991) have proposed cascaded sequential networks for the purpose of representing hierarchically organized structure. Gjerdingen (1992) has suggested a related approach using a hierarchy of masking fields in an ART 3 architecture. The basic idea is for lower levels of the hierarchy to encode fine structure and higher levels to encode coarser structure. While only small-scale tests of this idea have been performed, it seems promising and worth pursuing.
- *Chunking architectures.* In the cascaded sequential networks described above, the network designers specify in advance the hierarchical decomposition of a linear sequence. Todd (1991) assumes that an explicit decomposition is included as part of the training process. Burr and Miyata (1993) assume that successively higher levels operate on fixed, slower time-scales: the lowest level net is updated once a beat, the next net once every three beats, the next net every twelve beats, and so forth. Rather than providing the system with a hierarchical decomposition, one would really like the system to discover the decomposition itself. Ideally, each level of the hierarchy should encode meaningful 'chunks' or events in the sequence—the nature of a chunk being determined by the statistics of the environment—and the next higher level should operate on these chunks. Schmidhuber *et al.* (1993) describe an architecture of this sort in which the higher level analyzes only components of the input that cannot be interpreted by the lower level, yielding an automatic decomposition of sequences. This architecture has not been tested on musical sequences. Mozer and Das (1993) present an explicit chunking mechanism that has the capability of creating new symbols to represent an abstraction of a sequence of input elements. The mechanism operates on the generated symbols just as it does on the input elements, allowing it to chunk the chunks recursively. The creation of new symbols achieves a reduced description that is more flexible than the time-constant method proposed in this article. The time-constant method attempts to derive global statistics, whereas the chunking mechanism derives only local statistics, but does so over abstractions of the input. We are presently exploring how the chunking mechanism performs on musical sequences.
- *Explicit representation of structure.* In the final simulation study, CONCERT was trained to predict harmonic accompaniment, which naturally operates at a coarser time-scale than the melody itself. The hope was that by including this coarser structure as an explicit component of the task, CONCERT would be forced to learn it. One can push this idea further and include structure at even coarser time-scales, e.g. the sort of description that musicologists might use to characterize or analyze a piece, such as phrase boundaries, themes, inversions, key modulations. Given training data annotated with these descriptions, CONCERT would have the opportunity to learn global structure explicitly.

- *Staged training.* Music appears to have a tremendous amount of local structure that can mask the presence of global structure. The evidence for this comes from the fact that CONCERT performed very well on training sets even without paying attention to global structure. One might get around this problem by staging training in CONCERT, first training RD units with large time constants, and then gradually introducing units with smaller (and zero) time constants in the course of training. This would force CONCERT to examine the more global structure from the start.
- *Representations of musical elements in context.* In the current work, the encoding of pitch, duration and harmony is independent of the temporal context in which the elements are embedded. This is clearly wrong from a psychological perspective: in music, as in every domain of cognition, context and expectations affect the interpretation of perceptual elements. A truer cognitive architecture would allow interactions between the processes that determine the encoding and the processes—modeled in CONCERT—that generate expectancies (see Bharucha (1987, 1991) for a relaxation model that has this general flavor). Another way of embodying this interaction is to consider the representation of musical elements in a musical context. The representations of pitch and chords in CONCERT are based on psychoacoustic studies that consider only pairwise similarities. Psychological studies of pitch and harmony in a musical context (e.g. Krumhansl, 1990; Krumhansl & Kessler, 1982; Longuet-Higgins, 1976, 1979) could potentially be of value in incorporating the preceding input history into the network's representations. Similarly, structured representations of rhythm (e.g. Lerdahl & Jackendoff, 1983; McAuley, 1993) might help to impose higher-level organization on the input sequences.

Acknowledgements

This research was supported by NSF Presidential Young Investigator award IRI-9058450 and grant 90-21 from the James S. McDonnell Foundation. My thanks to Paul Smolensky, Peter Todd, Debbie Breen, Yoshiro Miyata and two anonymous reviewers for helpful comments regarding this work, to Darren Hardy and Hal Eden for technical assistance, and to staff, faculty and students of the computer science department for serving as subjects in informal preference experiments.

Notes

1. Following Smolensky (1988), the phrase 'faithful representation' is used to mean that the represented items can be accurately reconstructed from the representation. A faithful transition-table representation of a set of examples would be one that, given the first few notes of any example, could unambiguously determine the remainder of the example.
2. Of course, this interpretation assumes independence of the predictions, which is certainly not true in CONCERT. However, Bridle (1990) provides another justification, somewhat less intuitive, for this performance measure in terms of an information-theoretic criterion.
3. An unforgivable pun: Rob Goldstone suggested calling CONCERT's training procedure 'Bach propagation'.
4. This is one justification for the exponential function in the NNL layer.
5. The perfect fifth is a musically significant interval. The frequency ratio of a note to its perfect fifth is 2:3, just as the frequency ratio of a note to its octave is 1:2.
6. The reader may wonder why points on a circle need to be represented in a two-dimensional space. After all, the points lie on a 1-D continuum, albeit embedded in a 2-D space. Without such an embedding, however, distance relationships between points cannot be preserved. If the circle is cut

and flattened into a 1-D continuum, formerly adjacent points on opposite sides of the cut will end up far apart.

7. Although a PH scale factor of 9.798 was used for the target NND representation, $\mathbf{p}_i$, a PH scale factor of 1.0 was used for the input representation. This was based on empirical studies of what scale factors yielded the best performance. The primary reason that a PH scale factor other than 1.0 on the inputs causes difficulties is that the resulting error surface is poorly conditioned when different units have different activity ranges (Widrow & Stearns, 1985).
8. To maintain information about note ties, an additional component of the duration representation signaled whether a note was tied from the previous.

References

Bengio, Y., Simard, P. & Frasconi, P. (1993) Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*. 5, 157–161.

Bharucha, J.J. (1987) MUSACT: a connectionist model of musical harmony. In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, pp. 508–517. Hillsdale, NJ: Erlbaum.

Bharucha, J.J. (1991) Pitch, harmony, and neural nets: a psychological perspective. In P.M. Todd & D.G. Loy (Eds), *Music and Connectionism*, pp. 84–99. Cambridge, MA: MIT Press/Bradford Books.

Bharucha, J.J. & Todd, P.M. (1989) Modeling the perception of tonal structure with neural nets. *Computer Music Journal*, 44–53.

Bridle, J. (1990) Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In D.S. Touretzky (Ed.), *Advances in Neural Information Processing Systems 2*, pp. 211–217. San Mateo, CA: Morgan Kaufmann.

Burr, D. & Miyata, Y. (1993) Hierarchical recurrent networks for learning musical structure. In C. Kamm, G. Kuhn, B. Yoon, S.Y. Kung & R. Chellappa (Eds), *Neural Networks for Signal Processing III*. Piscataway, NJ: IEEE.

Cottrell, G.W. & Tsung, F.-S. (1993) Learning simple arithmetic procedures. *Connection Science*, **5**, 37–58.

Denker, J., Schwartz, D., Wittner, B., Solla, S., Howard, R., Jackel, L. & Hopfield, J. (1987) Automatic learning, rule extraction, and generalization. *Complex Systems*, **1**, 877–922.

Dodge, C. & Jerse, T.A. (1985) *Computer Music: Synthesis, Composition, and Performance*. New York: Shirmer Books.

Dolson, M. (1989) Machine Tongues XII: neural networks. *Computer Music Journal*, **13**, 28–40.

Elman, J.L. (1990) Finding structure in time. *Cognitive Science*, **14**, 179–212.

Elman, J.L. (1993) Learning and development in neural networks: the importance of starting small. *Cognition*, **48**, 71.

Fraisse, P. (1982) Rhythm and tempo. In D. Deutsch (Ed.), *The Psychology of Music*, pp. 149–180. New York: Academic Press.

Gjerdingen, R.O. (1992) Learning syntactically significant temporal patterns of chords: a masking field embedded in an ART3 architecture. *Neural Networks*, **5**, 551–564.

Hinton, G. (1987) Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pp. 1–12. Hillsdale, NJ: Erlbaum.

Hinton, G.E. (1988) Representing part-whole hierarchies in connectionist networks. *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pp. 48–54.

Hinton, G.E., McClelland, J.L. & Rumelhart, D.E. (1986) Distributed representations. In D.E. Rumelhart & J.L. McClelland (Eds), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol. I: *Foundations*, pp. 77–109. Cambridge, MA: MIT Press/Bradford Books.

Jones, K. (1981) Compositional applications of stochastic processes. *Computer Music Journal*, **5**, 45–61.

Jones, M.R. & Boltz, M. (1989) Dynamic attending and responses to time. *Psychological Review*, **96**, 459–491.

Kohonen, T. (1989) A self-learning musical grammar, or "Associative memory of the second kind". In *Proceedings of the 1989 International Joint Conference on Neural Networks*, pp. 1–5.

Kohonen, T., Laine, P., Tiits, K. & Torkkola, K. (1991) A nonheuristic automatic composing method. In P.M. Todd & D.G. Loy (Eds), *Music and Connectionism*, pp. 229–242. Cambridge, MA: MIT Press/Bradford Books.

Krumhansl, C.L. (1990) *Cognitive Foundations of Musical Pitch*. New York: Oxford University Press.

Krumhansl, C.L. & Kessler, E.J. (1982) Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review*, **89**, 334–368.

Krumhansl, C.L., Bharucha, J.J. & Kessler, E.J. (1982) Perceived harmonic structure of chords in three related musical keys. *Journal of Experimental Psychology: Human Perception and Performance*, **8**, 24–36.

Laden, B. & Keefe, D.H. (1989) The representation of pitch in a neural net model of chord classification. *Computer Music Journal*, **13**, 12–26.

Lerdahl, F. & Jackendoff, R. (1983) *A Generative Theory of Tonal Music*. Cambridge, MA: MIT Press.

Lewis, J.P. (1991) Creation by refinement and the problem of algorithmic music composition. In P.M. Todd & D.G. Loy (Eds), *Music and Connectionism*, pp. 212–228. Cambridge, MA: MIT Press/Bradford Books.

Longuet-Higgins, H.C. (1976) Perception of melodies. *Nature*, **263**, 646–653.

Longuet-Higgins, H.C. (1979) The perception of music (Review Lecture). *Proceedings of the Royal Society of London*, **205B**, 307–332.

Lorrain, D. (1980) A panoply of stochastic 'cannons'. *Computer Music Journal*, 3, 48–55.

Loy, D.G. (1991) Connectionism and musiconomy. In P.M. Todd & D.G. Loy (Eds), *Music and Connectionism*, pp. 20–36. Cambridge, MA: MIT Press/Bradford Books.

McAuley, J.D. (1993) Finding metrical structure in time. In M.C. Mozer, P. Smolensky, D.S. Touretzky, J.E. Elman & A.S. Weigend (Eds), *Proceedings of the 1993 Connectionist Models Summer School*, pp. 219–227. Hillsdale, NJ: Erlbaum Associates.

McClelland, J.L. (1979) On the time relations of mental processes: an examination of systems of processes in cascade. *Psychological Review*, **86**, 287–330.

Mozer, M.C. (1987) RAMBOT: a connectionist expert system that learns by example. In M. Caudill & C. Butler (Eds), *Proceedings of the IEEE First Annual International Conference on Neural Networks*, pp. 693–700. San Diego, CA: IEEE Publishing Services.

Mozer, M.C. (1989) A focused back-propagation algorithm for temporal pattern recognition. *Complex Systems*, **3**, 349–381.

Mozer, M.C. (1992) The induction of multiscale temporal structure. In J.E. Moody, S.J. Hanson & R.P. Lippman (Eds), *Advances in Neural Information Processing Systems IV*, pp. 275–282. San Mateo, CA: Morgan Kaufmann.

Mozer, M.C. (1993) Neural network architectures for temporal pattern processing. In A. Weigend & N. Gershenfeld (Eds), *Time Series Prediction: Forecasting the Future and Understanding the Past*, pp. 243–264. Redwood City, CA: Addison-Wesley Publishing.

Mozer, M.C. & Das, S. (1993) A connectionist symbol manipulator that induces the structure of context-free languages. In S.J. Hanson, J.D. Cowan & C.L. Giles (Eds), *Advances in Neural Information Processing Systems V*, pp. 863–870. San Mateo, CA: Morgan Kaufmann.

Myers, C. (1990) *Learning with Delayed Reinforcement Through Attention-driven Buffering*. Technical Report. London: Neural Systems Engineering Group, Department of Electrical Engineering, Imperial College of Science, Technology, and Medicine.

Pearlmutter, B.A. (1989) Learning state space trajectories in recurrent neural networks. *Neural Computation*, **1**, 263–269.

Pineda, F. (1987) Generalization of back propagation to recurrent neural networks. *Physical Review Letters*, **19**, 2229–2232.

Pressing, J. (1983) Cognitive isomorphisms between pitch and rhythm in world musics: West Africa, the Balkans, and Western tonality. *Studies in Music*, **17**, 38–61.

Principe, J.C., Hsu, H.-H. & Kuo, J.-M. (1994) Analysis of short-term neural memory structures for nonlinear prediction. In J.D. Cowan, G. Tesauro & J. Alspector (Eds), *Advances in Neural Information Processing Systems VI*. San Mateo, CA: Morgan Kaufmann Publishers.

Ring, M. (1993) Learning sequential tasks by incrementally adding higher orders. In C.L. Giles, S.J. Hanson & J.D. Cowan (Eds), *Advances in Neural Information Processing Systems V*, pp. 115–122. San Mateo, CA: Morgan Kaufmann.

Rumelhart, D.E. (in press) Connectionist processing and learning as statistical inference. In Y. Chauvin & D.E. Rumelhart (Eds), *Backpropagation: Theory, Architectures, and Applications*. Hillsdale, NJ: Erlbaum.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986) Learning internal representations by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. I: Foundations*, pp. 318–362. Cambridge, MA: MIT Press/Bradford Books.

Schmidhuber, J. (1992) Learning unambiguous reduced sequence descriptions. In J.E. Moody, S.J. Hanson & R.P. Lippman (Eds), *Advances in Neural Information Processing Systems IV*, pp. 291–298. San Mateo, CA: Morgan Kaufmann.

Schmidhuber, J.H., Mozer, M.C. & Prelinger, D. (1993) Continuous history compression. In H. Huening, S. Neuhauser, M. Raus & W. Ritschel (Eds), *Proceedings of the International Workshop on Neural Networks, RWTH Aachen*, pp. 87–95. Augustinus.

Shepard, R.N. (1982) Geometrical approximations to the structure of musical pitch. *Psychological Review*, **89**, 305–333.

Shepard, R.N. (1987) Toward a universal law of generalization for psychological science. *Science*, **237**, 1317–1323.

Smolensky, P. (1988) On the proper treatment of connectionism. *Behavioral & Brain Sciences*, **11**, 1–74.

Stevens, C. & Wiles, J. (1993) Representations of tonal music: a case study in the development of temporal relationships. In M.C. Mozer, P. Smolensky, D.S. Touretzky, J.E. Elman & A.S. Weigend (Eds), *Proceedings of the 1993 Connectionist Models Summer School*, pp. 228–235. Hillsdale, NJ: Erlbaum Associates.

Todd, P.M. (1989) A connectionist approach to algorithmic composition. *Computer Music Journal*, **13**, 27–43.

Todd, P.M. (1991) A connectionist approach to algorithmic composition. In P.M. Todd & D.G. Loy (Eds), *Music and Connectionism*, pp. 173–194. Cambridge, MA: MIT Press/Bradford Books.

Tsung, F.-S. & Cottrell, G.W. (1993) *Phase-space Learning in Recurrent Networks*. Technical Report CS93-285. La Jolla, CA: Department of Computer Science and Engineering, University of California, San Diego.

Widrow, B. & Stearns, S.D. (1985) *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall.

Williams, R.J. & Zipser, D. (in press) Gradient-based learning algorithms for recurrent connectionist networks. In Y. Chauvin & D.E. Rumelhart (Eds), *Backpropagation: Theory, Architectures, and Applications*. Hillsdale, NJ: Erlbaum.