Connectionism, Music and Fourier Phase Spaces

by

Arturo Pérez

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Psychology
University of Alberta

Abstract

How does cognition represent musical properties? Even with our growing understanding of the cognitive neuroscience of music (Abbott, 2002; Peretz and Zatorre, 2003; Peretz and Zatorre, 2005; Zatorre and McGill, 2005), the answer to this question remains unclear. One method for conceiving possible representations is to use artificial neural networks, which can provide biologically plausible models of cognition (Rumelhart and McClelland, 1986; Bechtel and Abrahamsen, 2002; Enquist and Ghirlanda, 2005). One could train networks to solve musical problems, (Todd and Loy, 1991; Griffith and Todd, 1999) and then study how these networks encode musical properties. However, researchers rarely conduct detailed examinations of network structure(Dawson, 2009, 2013, 2018) because networks are difficult to interpret, and because it is assumed that networks capture informal or subsymbolic properties (Smolensky, 1988; McCloskey, 1991; Bharucha, 1999). Within this thesis, we explore the relations between network connection weights and discrete Fourier phase spaces used to represent musical sets (Amiot, 2016; Callender, 2007; Quinn, 2006, 2007; Yust, 2016), and how these networks use Fourier components to differentiate between different sets of musical entities. That networks discover Fourier phase spaces indicates that these spaces have an important role to play outside of formal music theory.

Preface

Some of the research conducted for this thesis forms part of the collaboration between lab members at the BCP, with Professor Michael R.W. Dawson at the lead.

Chapter 2 of this thesis has been published as Dawson, M.R.W., Perez, A., & Sylvestre, S. (2020). Artificial neural networks solve musical problems with Fourier phase spaces. Scientific Reports, 10(1), 7151. doi: 10.1038/s41598-020-64229-4.

A version of chapter 3 is currently under editorial review: Perez, A.I., Ma, H.L., and Dawson M.R.W. (Under editorial review). Artificial Neural Networks Use 'Sympathetic Vibration' To Classify Musical Triads. Cognitive Science, under review (10,796 words, submitted June 14, 2021)

Table of Contents

Abstract ii

Preface iii

List Of tables vii

List of figures viii

Chapter 1:

Introduction 1

Elements of Music Theory 2

Musical Set Theory 5

Artificial Neural Networks 7

Interpreting Musical Networks 10

Fourier Components and Strange Circles 13

Conclusion 17

References 18

Chapter 2

Abstract 20

Introduction 21

An Example Network 22

Results From Other Networks 26

Discussion 28

References 30

Acknowledgments 32

Chapter 3

Abstract 33

Introduction 34

Musical Sets and Fourier Phase Spaces 36

Rationale For Current Research 40

Method 43

Task 43

Training 45

Results 46

Discovering Phase Space Projections 46

Hidden Unit Activity As 'Sympathetic Vibration' 49

Coarse Coding In An Example Network 52

Discussion 62

References 66

Chapter 4

Introduction..71

Directions For Future Research 74

References 79

Bibliography 81

List Of tables

Table 1-1. 4

Table 1-2 5

Table 3-1 44

Table 3-2 47

Table 3-3 48

Table 3-4 52

Table 3-5 54

Table 3-6 56

Table 3-7 59

Table 3-8 60

List of figures

Figure 1-1 Top: Keyboard highlighting the 12 notes of the western tonal system.. Bottom: Keyboard showing multiple instances of a note across octaves 3

Figure 1-2 Graphical representation of the multi-layered perceptron used. At the bottom we have input units for notes between C and B. Middle layer is the Hidden Unit Layer. At the top we have the output units, one for each kind of pattern presented to the network 7

Figure 1-3 Top: Keyboard highlighting notes C and E, which are 4 semitones apart. 11

Figure 1-4 Interval cycles that form from different starting points within the octave, if one advances along the keyboard in intervals of 4 semitones. 11

Figure 1-5 Weights from input units representing notes, and a hidden unit within the network. Note how a cycle repeats: first a small weight with a positive sign (A#), then a large weight of a positive sign(B). Then a weight similar size as A#, but with an opposit..................................... 12

Figure 1-6 A complex signal (blue) decomposed into two simpler periodic functions by Fourier analysis (red and green). 14

Figure 1-7 Yust's (2016) six different Fourier phase spaces. 15

Figure 1-8 Circular averaging of the C major triad as performed by Yust (2016) 16

Figure 1-9 Phase space projections are determined from the vectors that result from circular averaging.. 16

Figure 2-1 (A) An ANN for measuring distances between pairs of pitch-classes. Stimuli are presented by activating two input units; 1A illustrates presenting C and G. The ANN learns to activate the output units that measure (as in (B)) the distances between the two pitch-classes by processing signals from four hidden units. All output and hidden units employ a Gaussian activation function $f(net) = e^{-\pi(net-\mu)2}$ that ranges between 0 and 1; net is the processor's incoming signal, and μ is the function's mean. Weights are modified by a variant of the generalized delta rule (Dawson and Schopflocher, 1992). The bar plots (1C-1F) depict the weights of connections between input units and each hidden unit after training ends. 23

Figure 2-2 The six Fourier phase spaces, Ph1, Ph2, and so on used by Yust (2016). Each space represents the twelve pitch-classes on a clock face. (A) Phase space Ph1. The arrow, pointing at 0 o'clock, indicates a phase of 0. Solid lines denote projections from the pitch-class locations on the clock to the arrow. (B) Distances between each projection in (A) and the center of the phase space. Distances towards the arrowhead are positive. (C, D) Ph2 and its projection distances. (E, F) Ph3 and its projection distances. (G, H) Ph4 and its projection distances. (I, J) Ph5 and its projection distances. (K, L) Ph6 and its projection distances. The projection distance graphs show that each phase space represents a cosine function, and that the difference between phase spaces is the frequency of this function. Ph1 has a frequency of 1, Ph2 has a frequency of 2, and so on. Musical sets can be reconstructed by adjusting the phase of each component, weighting them and summing them together. 25

Figure 2-3. The results of fitting Fourier phase spaces to hidden unit weights for the Figure 2-1 network. (A) The weights for processor H1 are best fit by Ph3 (see Figure 2-2) with a phase of 2.47 (r = 0.97). (B) The relationship between H1 connection weights and projected distances after scaling the distances by 1.33. Projection distances are measured as described in Figure 2-2. (C) H2 weights are best fit by Ph6 with a phase of 3.87 (r = 0.97). (D) H2 weights and projected distances scaled by 2.02. (E) H3 weights are best fit by Ph5 with a phase of 1.43 (r = 0.99). (F) The relationship between H3 weights and projected distances scaled by 0.774. (G) H4 weights are best fit by Ph1 with a phase of 7.63 (r = 0.98). (H) H4 weights and projected distances scaled by 1.25. 26

Figure 2-4. The mean correlations between hidden unit weights and Fourier phase space projections, with standard errors. The 'All Problems' data is based upon 240 different hidden units. The 'Interval', 'Mode', and 'Triad' data is based upon 80, 100, and 60 hidden units respectively. For each hidden unit, six different correlations were calculated, one for each phase space. The correlations were then ordered from the best fit to the worst fit, as is reflected in the six different bars presented for each problem type. 28

Figure 3-1**.** An example multilayer perceptron for classifying triads. 35

Figure 3-2 components.. 38

Figure 3-3. 39

Figure 3-4. Two examples of different activation functions used by processors in multilayer perceptrons. (A) The bell-shaped activation function of a value unit, defined by the Gaussian equation. (B) the sigmoid-shaped activation function of an integration device, defined by the logistic equation. 41

Figure 3-5. An illustration of the 'sympathetic vibration' hypothesis. Each circle represents the best fitting phase space for one of a network's eight hidden units, and the dashed arrow represents the unit's preferred phase for the space. The solid arrow represents the phase of the F major triad in the same space. The angle between the two arrows is represented by θ. The net input computed for each hidden unit, and the cosine of θ, are presented beneath each circle. 50

Figure 3-6**.** The phases for different types of triads in the two phase spaces (Ph3 and Ph4) to which hidden units in the example network are most sensitive. Phases are indicated by the directions of the solid arrows. When more than one arrow is present, different subsets of triads have different phases in the same space. The triads belonging to different subsets are detailed later in Table 3-8 55

Figure 3-7. Coarse coding in the example network. Each oval contains all triads that produce a Table 3-6 value of activity in a hidden unit. Triads are indicated in terms of type and subtype. For instance, 'Aug 1' indicates augmented triads of subtype 1 in Ph3, and 'Min 1' indicates minor triads of subtype 1 in Ph4, as detailed in Table 3-8. The intersection of the ovals indicates that the only triads which produce these eight values of hidden unit activities are minor triads of subtype 1 in Ph4. 58

Figure 4-1**.** Yust Phase spaces, as shown in Chapter 1. 72

Figure 4-2 Plot of mean correlation between a Fourier phase space projections and hidden unit weights, as shown on chapter 2. Note how for every problem tackled by the network on that chapter, there is a phase space that is highly correlated with the weights. 73

Figure 4-3 Plot representing the similarity relationship between each scale, on a three-dimensional space. 76

Figure 4-4**.** Circle of fifths 77

Figure 4-5**.** Three-dimensional map of the activity of our network, for the Dorian Scales. 77

Chapter 1 : Exploring Music Through Artificial Neural Networks and Fourier Phase Spaces

Introduction

Musical experience is complex because it is composed of many different interconnected characteristics. Krumhansl (1990) notes musical experience combines objective musical properties (such as frequency, interval structure) with subjective properties (the listener's perception of pitch and perception). As a result, to study musical cognition one must decide what materials to use (e.g., Western tonal music or other forms of tonality), what aspects of musical experience to study (e.g., harmony, rhythm, pitch perception) and what methodology to employ (e.g., isolated stimuli vs complete musical works, human participants vs computational models, task of pitch discrimination vs pitch fit in a context).

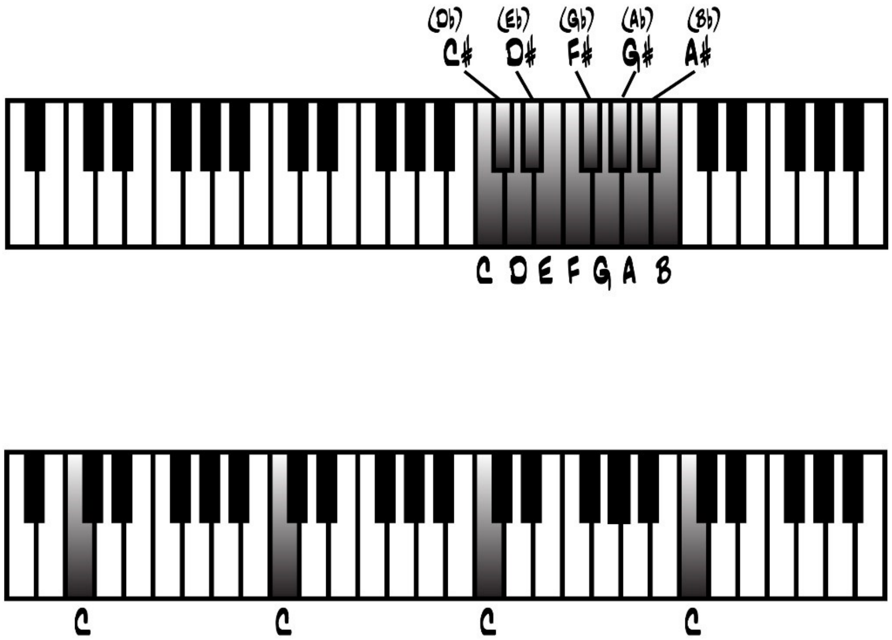
This thesis uses artificial neural networks to study the identification and classification of musical entities from Western tonal music. Artificial neural networks (ANNs) have been used to study a wide variety of problems in many different fields. The use of ANNs to study a particular topic, musical cognition, is growing in popularity. Connectionist cognitive scientists interested in music train artificial neural networks to solve musical tasks in order to understand the "[...]processes and representations involved in music perception, production, comprehension and composition" and to "capture musical behaviour in an artificial system" (Griffith & Todd, 1999)**.** Additionally, connectionists often believe networks capture *informal* aspects of musical creation and experience, properties not explained by formal theories of Western tonality (Todd & Loy, 1991).

Many kinds of networks have been trained to perform a variety of musical tasks. These tasks include pitch and harmony perception, detecting musical meter and rhythm, and composing new music (ibid.). However, because the internal structure of these networks is rarely examined, little is known about *how* networks process stimuli, or how networks can provide insight into the algorithms or brain structures underlying human musical cognition (Dawson, 2018).

In *Connectionist representations of tonal music,* Dawson (2018) extends the connectionist exploration of music. He analyzes the internal structure of trained networks to understand how they convert stimuli into responses. Dawson uses networks to study several different musical tasks. These tasks all use different collections of the twelve Western pitch-classes. To understand Dawson's approach, I must first introduce some basic elements of music theory to describe such tasks. I must then introduce some basic properties of ANNs. Finally, I must relate properties of ANNs to ideas from a mathematical approach to music, musical set theory. To begin, I introduce basic elements of music theory.

Elements of Music Theory

This thesis involves training ANNs on tasks related to Western tonal music. Musical pitch provides the foundation for Western tonal music. More complex musical entities, like scales or chords, are created by combining different musical pitches. In Western music, each pitch corresponds to a different musical note, as illustrated in Figure 1-1 provided below



1-1 Top: Keyboard highlighting the 12 notes of the western tonal system.. Bottom: Keyboard showing multiple instances of a note across octaves

A traditional piano has 88 keys. Each key, when pressed, plays a note. Each note is associated with a sound wave oscillating at a specific frequency, measured in cycles per second or Hertz (Hz). This is the note's fundamental frequency, and determines how high or low the note's pitch is experienced. As the fundamental frequency increases, so too does our experience of how high the note is. For instance, the A below middle C on the piano (the note A3) has a frequency of 220hz. The A that is an octave higher (A4) has a frequency of 440 Hz.

In Western music, pitches that are nearest neighbours are separated by a musical distance of one semitone. For example, in Figure 1-1, the keys E and F are adjacent to one another, so F is one semitone higher than E. Similarly, C# is one semitone higher than C. On a modern piano, when one note is a semitone higher than another, then the difference between the frequencies of the two notes is 27 Hz. For instance, middle C on a piano has a frequency of 262 Hz, while the note a semitone higher (C#) has a frequency of 289 Hz.

The distance between two particular pitches, measured in semitones, is called an interval. Western music is based on twelve different intervals. The smallest is 0 semitones (the distance between a pitch and itself), and the largest is 12 semitones. Each interval is also given a name. For instance, an interval of 0 semitones is called perfect unison, while an interval of 12 semitones is called a perfect octave. The basic intervals of Western music are provided in Table 1-1:

Semitones Between Pitches	Interval Name
0	Perfect Unison
1	Minor Second
2	Major Second
3	Minor Third
4	Major Third
5	Perfect Fourth
6	Tritone
7	Perfect Fifth
8	Minor Sixth
9	Major Sixth
10	Minor Seventh
11	Major Seventh
12	Perfect Octave

Table 1-1The thirteen possible distances between pitches that can be used to create interval cycles.

In Figure 1-1 above, there are 12 different note names given to the keys in the upper part of the figure. These names are repeated to name the next 12 keys on the piano. Notes that are an octave apart (piano keys separated by 12 semitones) are given the same name. This is illustrated in the lower part of the figure, which shows different keys separated by an octave all being given the name C. Giving the same name to pitches separated by an octave is known as *octave equivalence*. Octave equivalence assigns different pitches to the same *pitch-class*. For instance, all of the different Cs in the figure (C2, C3, C4, etc) belong to the pitch-class C. Western music uses only 12 different pitch-classes: C, C#, D, D#, E, F, F#, G, G#, A, A# and B. When music theory is applied to Western tonal music, it expresses formal regularities in terms of pitch-classes.

We can now define a more complex musical entity: the *chord*. A chord is a collection of pitch-classes separated by particular intervals. One basic chord is called a triad, and is composed of three different pitch-classes. The relationship between these pitch-classes determines triad type. For instance, one type is the major triad. The lowest pitch-class in a major triad is the root, or starting note. The next pitch-class in a major triad is a major third above the root (four semitones higher). The last pitch-class in a major triad is a perfect fifth (seven semitones) higher than the root. For example, the C major triad consists of the pitch-classes C, E and G.

A more complex musical entity is a *scale*. A scale is a subset (typically 7) of the 12 different pitch-classes of the Western system. When ordered from lowest to highest in pitch, we refer to each element of a scale by its position. So the first pitch-class is the first degree, or root, the second pitch-class is the second degree, the third is the third degree and so on.

There are different types of scales. Each type is defined the interval distances between adjacent scale degrees. For example, the C major scale has a semitone between its third and fourth degrees, while C minor has semitones between its second and third, as well as between its fifth and sixth, degrees., as shown in Table 1-2 below.

	Scale Degree						
	I	**II**	**III**	**IV**	**V**	**VI**	**VII**
Scale		←Tone→		←Tone→		←Tone→	
C Major	C	D	E	F	G	A	B
	←Tone→		← Semitone→		←Tone→		
		← Semitone→		←Tone→		←Tone→	
C Minor	C	D	D#	F	G	G#	A#
	←Tone→		←Tone→		← Semitone→		

Table 1-2 Table shows two scales, C Major (top) and C Minor(bottom). Besides having different collections of notes, these scales have different interval structures.

Musical Set Theory

The previous section introduced examples of musical entities like chords and scales. In general, any musical entity can be described as a collection of pitch-classes. A powerful description of any such collection is called a *pitch-class set* or PC set (Forte, 1973). PC sets are mathematical sets which formally describe musical entities.

A PC set represents a collection of pitch-classes as a set of integers. Forte uses a standard convention to translate the twelve different pitch-classes into twelve different integers. C is coded as 0, C# as 1, D as 2 and so on. Alternatively, a musical entity can be represented as a different kind of set called a characteristic function (Amiot, 2016). A characteristic function is a set of twelve integers. Position in a characteristic function corresponds to pitch-class (i.e., position 0 corresponds to C, position 1 corresponds to C#, etc.). The integer in each position indicates how many instances of a pitch-class are present in a musical entity. For example, for Forte the C major triad (C,E,G) becomes the pc-set (0,4,7) or the characteristic function (1,0,0,0,1,0,0,1,0,0,0,0). Forte performs mathematical operations on pitch-class sets to reveal similarities between musical objects that seem at first glance quite different. For instance, consider the characteristic function (1,0,0,0,1,0,0,1,0,0,0,0) and a different characteristic function (1,0,0,0,0,1,0,0,0,1,0,0). While the two sets have some similarities (e.g., each has one instance of three different pitch-classes), they have many differences. For example, there are differences in the pitch-classes that are present in both sets, and there are differences in how many absent pitch-classes exist between values of 1 in the set. Forte's operations, however, can translate either of these vectors into an identical form. This is because the two characteristic functions represent different major triads – the C major and the F major triad, respectively. The sets are different because they represent different triads. However, they can be translated into one another because they belong to the same class (major triad). Pitch-class set representations permit regularities to be discovered between sets that appear to be very different.

Forte’s approach provides a language to think about classifying musical stimuli as solving set theoretical problems. Forte solves such problems using basic operations from mathematical set theory. Importantly, such set theoretic problems can also be solved by ANNs. The next section introduces ANNs, relating them to musical problems studied by set theorists.

Artificial Neural Networks

What is an ANN, and how can an ANN learn to classify musical sets? Qualitatively, an ANN can be defined as a set of processors, connected by weights, which learns to respond to a stimulus. Processors called *input units* represent stimuli, while processors called *output units* represent responses. This thesis is concerned with a particular kind of ANN called a *multilayer perceptron*, which is illustrated in Figure 1-2 below. This network includes a layer of intermediate processors called *hidden units*. Hidden units detect complex or higher-order regularities in input units; the features detected by hidden units then determine how output units will respond.

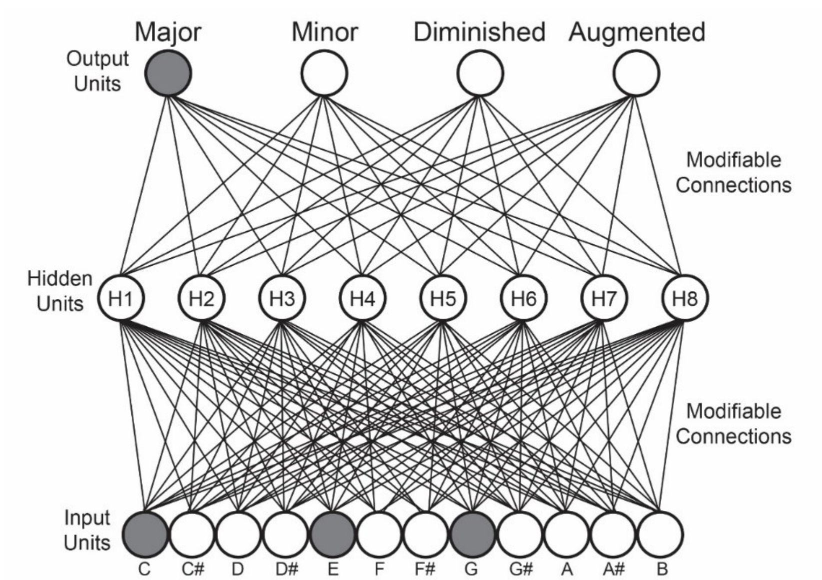


Figure 1-2 Graphical representation of the multi-layered perceptron used. At the bottom we have input units for notes between C and B. Middle layer is the Hidden Unit Layer. At the top we have the output units, one for each kind of pattern presented to the network

A stimulus is presented to a network by activating each of its input units. We call the pattern of input unit activities the input vector. When activated, input units send signals to hidden units. Each signal sent is equal to the activity of the input unit multiplied by the weight of the connection through which the signal is sent.

Presenting an input vector to a multilayer perceptron causes the hidden units to activate. Hidden units behave as follows: First, a hidden unit determines the total signal being received, called the *net input*. Net input is the sum of the scaled signals being received from the input units. Mathematically, net input is the dot product between the input vector and the vector representing the hidden unit's connection weights. Second, a hidden unit converts net input into internal activity; activity in our networks ranges between 0 and 1. Activity is determined by passing net input into a nonlinear *activation function*. Later chapters describe networks which employ one of two such functions: a bell-shaped Gaussian activation function, or a sigmoid-shaped logistic activation function.

When hidden units activate, signals are sent through another layer of weighted connections to the output units. Each output unit computes net input (the dot product between the vector of hidden unit activities and the vector of connection weights), and converts net input into activity with a nonlinear activation function.

The stimulus-response mapping made by the Figure 1-2 network classifies stimuli into triad types. The network is presented a triad by activating three of its input units with a value of 1 (indicating the three pitch-classes present in the triad) and by activating the other input units with a value of 0 (indicating the pitch-classes absent from the triad). For example, the C major triad, comprised of the notes C,E and G, is presented to the network as the input vector (1,0,0,0,1,0,0,1,0,0,0,0). To respond correctly to the stimulus, the Figure 1-2 network turns one input unit on (activity near 1) and turns the remaining output units off (activity near 0). When presented the C major triad, the network should turn the output unit representing major triads on, and the other three output units off, as Figure 1-2 illustrates. How does an ANN like Figure 1-2 respond correctly to different stimuli? Multilayer perceptrons *learn* to perform a desired stimulus-response mapping. The connections in the Figure 1-2 network begin as small, random values, and are then modified by learning. Learning involves teaching the network with a training set, a collection of stimulus-response pairs. During learning, the network is presented a stimulus, causing the output units to activate. Error is computed by taking the difference between observed responses and the desired responses associated with the stimulus in the training set. Error is then used to modify the network's connection weights to decrease network error (Rumelhart, Hinton, & Williams, 1986). Repeatedly presenting the training set, and using error to modify connection weights, reduces errors to each stimulus as much as possible. Learning is important because networks can discover new representations for stimulus-response mappings. Researchers do not insert ideas about representation into networks; networks uncover surprising representations on their own.

What is the relationship between a network like Figure 1-2 and musical set theory? First, the input vector used to represent a network's stimulus is identical to a characteristic function. For instance, to present the C major triad to the Figure 1-2 network, the C major triad's characteristic function defines the input vector. Second, after training, the Figure 1-2 network will produce the same response to different stimuli. For example, it will turn the major triad output unit on when presented either the C major triad or the F major triad, even though there are many differences between the input vectors for these two stimuli as was briefly noted above. This means the Figure

1-2 network can be described as performing the same task accomplished by Forte when he uses set theoretical operations: translating different pitch-class sets into a common category.

Interpreting Musical Networks

How do ANNs solve problems studied by musical set theory? To answer this question, one must study the internal structure of ANNs trained to solve such problems. Dawson (2018)investigates the structure of his networks in order to discover how ANNs represent musical properties. Dawson reports his discovery of many regularities in his musical networks, and argues these regularities, while formal, are surprising. He proposes that ANNs solve musical problems in a very different manner than the operations of traditional theories: "An artificial neural network can discover a completely different method – an alien or novel music theory – that generates the same input/output relationships as are defined by Western music theory" (Dawson, 2018, p. 4).

One such novel musical representation is what Dawson calls a *strange circle*. A strange circle occurs when different pitches or different pitch-classes are treated as being identical by a hidden unit in a network. Furthermore, there is a specific relationship – defined by musical intervals – determining which pitch-classes belong to the same strange circle.

Dawson's notion of a strange circle begins with a basic notion from traditional music theory, the interval cycle (Susanni & Antokoletz, 2012) An interval cycle is a cyclic representation of pitch-classes in which adjacent pitch-classes are a specific interval apart. One example is illustrated in Figure 1-3. In the top part of the figure, the distance between C and E is four semitones or a major third. The bottom part of the figure shows that the same distance is found from E to G#. However, if one moves a major third from G#, then one returns to the start of the cycle (C). In other words, C, E, and G# all belong to the same interval cycle of a major third.

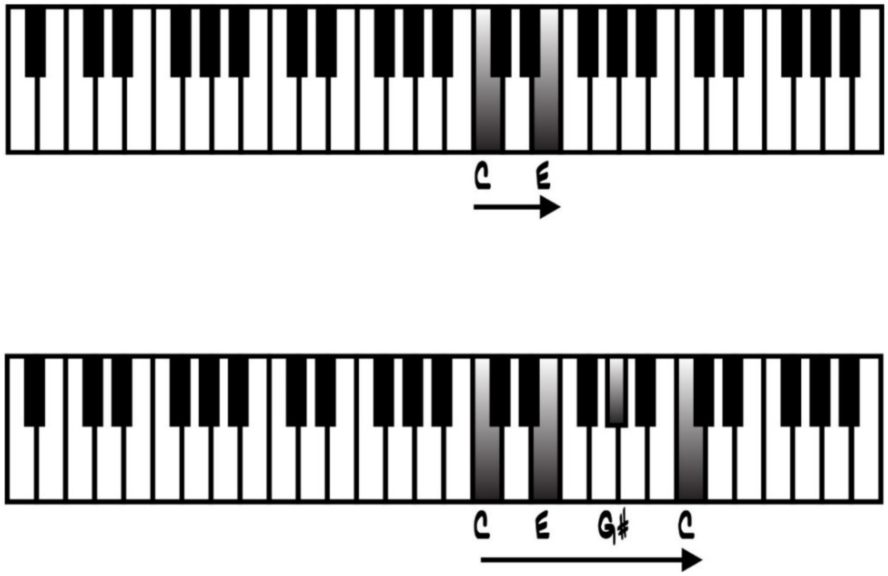


Figure 1-3 Top: Keyboard highlighting notes C and E, which are 4 semitones apart.

There are three other interval cycles based on the interval of a major third; each is created by starting on a different piano key (C#, or D, or D#) and moving to the next piano key a major third to the left. The four different interval cycles of a major third are illustrated in Figure 1-4 below.

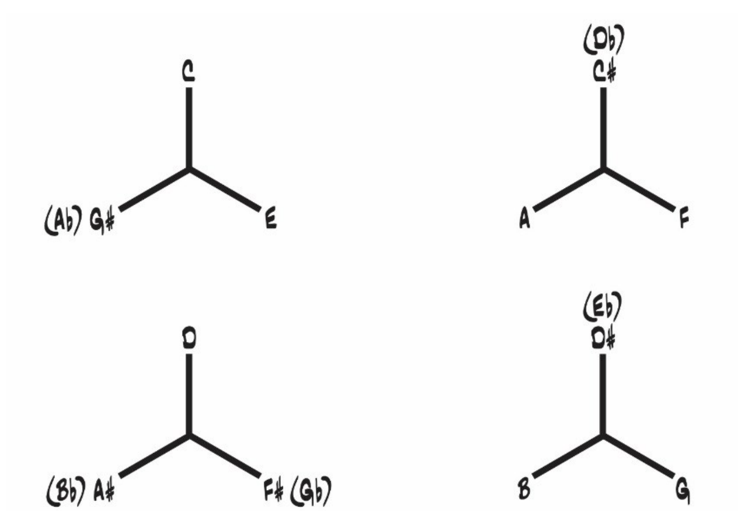


Figure 1-4 Interval cycles that form from different starting points within the octave, if one advances along the keyboard in intervals of 4 semitones.

Dawson's strange circles arise from his discovery that hidden units in a musical ANN assign the same connection weight to pitch-classes that belong to the same interval cycle. Figure 5 illustrates an example set of connection weights (Dawson, 2018, p. 187); Dawson interprets these weights in terms of strange circles based upon the major third. The bars in this figure represent the weights from 12 different input units (like those from Figure 1-2 earlier) to one hidden unit. Note that pitch-classes belonging to the same interval cycle in Figure 1-4 are assigned nearly identical connection weights in Figure 5. For Dawson, this means the hidden units treats pitch-classes from the same Figure 1- 4 interval cycle as belonging to the same class.

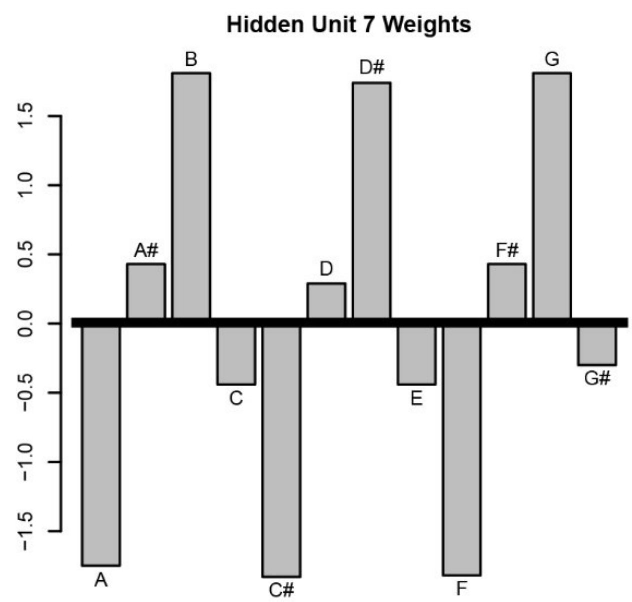


Figure 1-5 Weights from input units representing notes, and a hidden unit within the network. Note how a cycle repeats: first a small weight with a positive sign (A#), then a large weight of a positive sign(B). Then a weight similar size as A#, but with an opposit

Dawson (2018) interprets many ANNs trained on different musical tasks, and discovers hidden units using strange circle representations in almost all of them. These strange circles are not limited to the interval of a major third. Dawson also discovers strange circles based on intervals of a major second, a minor third, and a tritone. Dawson considers each strange circle representation as representing an alternative to traditional Western music theory. He adopts this position because Western music theory is founded upon the assumption that there exist twelve distinct pitch-classes as was discussed earlier. A hidden unit that encodes a strange circle behaves as if there are fewer basic pitch-classes. For example, a hidden unit that represents strange circles of major thirds (Figure 1-5) behaves as if there are only four different basic pitch-classes.

However, a different approach to musical set theory seems strongly related to the strange circles reported by Dawson (2018). There is a growing interest in performing the discrete Fourier transform (DFT) on pitch-class sets, and then using components of the DFT to explore relationships between different musical entities. The next section introduces this approach in musical set theory, and shows there is a strong relationship between it and the strange circles reported by Dawson In other words, Dawson's networks might be more strongly related to Western music theory – and less alien -- than he expected.

Fourier Components and Strange Circles

There are many accounts of the Fourier transform in the context of musical set theory (Amiot, 2016; Callender, 2007; Lewin, 2001; Quinn, 2006, 2007) but one of the more comprehensible is Yust's explanation using phase spaces (Yust, 2016). Fourier decomposition is a mathematical technique used to describe periodic functions in terms of simpler components (Tolstov, 1962). Looking at Figure 1- 6 we can see a complex wave in black, and, in colors, the simpler components (cosine functions) that when summed yield that complex wave.

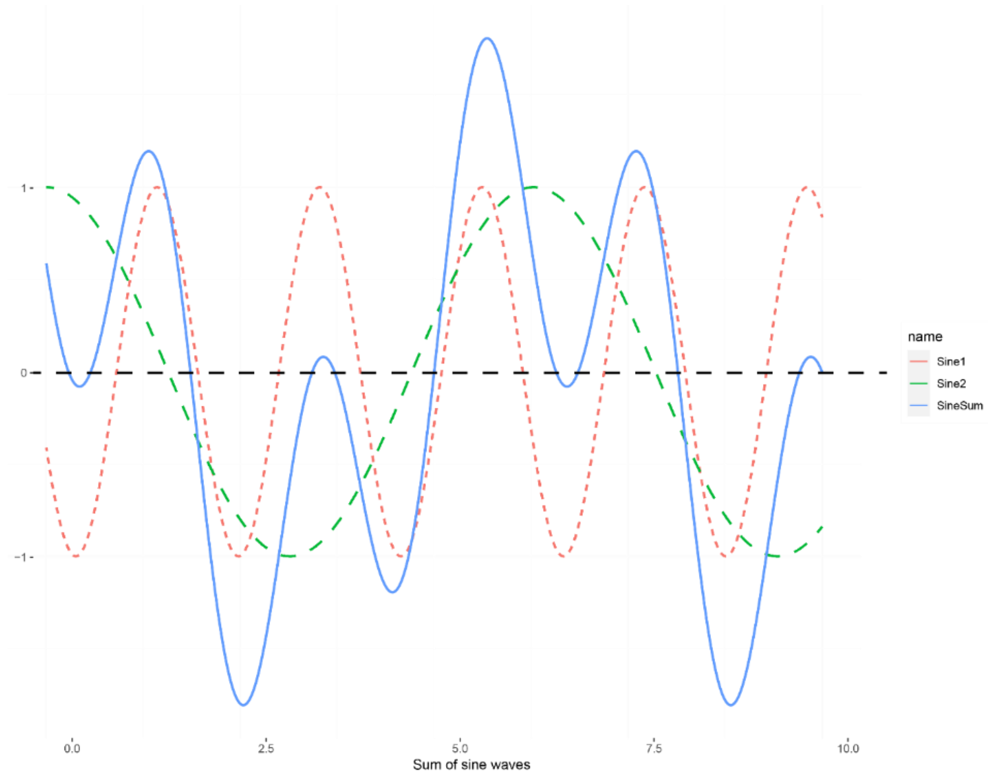


Figure 1-6 A complex signal (blue) decomposed into two simpler periodic functions by Fourier analysis (red and green).

The Fourier transform is extremely useful for analyzing the structure of musical sets (Amiot, 2016; Callender, 2007). Yust (2016) uses one dimensional phase spaces to serve as the simpler components into which more complex signals (characteristic functions) are decomposed. Each phase space is generated by arranging pitch-classes in particular locations around a clock as illustrated in **Figure 1- 7.**

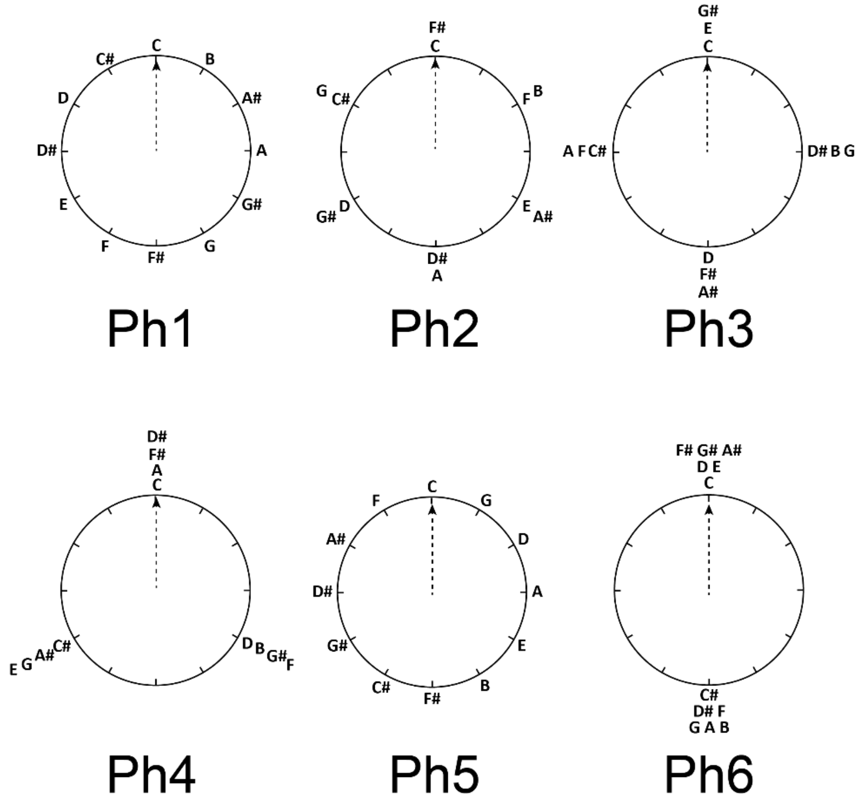


Figure 1-7 Yust's (2016) six different Fourier phase spaces.

How are phase spaces used to obtain the Discrete Fourier Transform (DFT) of a pitch-class set? One performs what Yust (2016) calls *circular averaging*. In circular averaging, one first extends a vector from the center of the circle to each of the location of a pitch-class on the circle for each pitch-class in the set. Figure 1-8 illustrates this for the C major triad (C,E,G) in each of the six phase spaces. That is on each phase space there are three vectors that have been created (one each for C, E and G). Then one sums these vectors together by placing them end to end. The dashed lines in Figure 1-8 represent the sum of the three vectors for each phase space. The orientation of a vector sum is the position of the pitch-class set on a phase space (i.e., the phase of the pitch class set in that space), while the length of the vector is its magnitude (i.e., the amount that the phase is to be weighted when used to reconstruct signals).

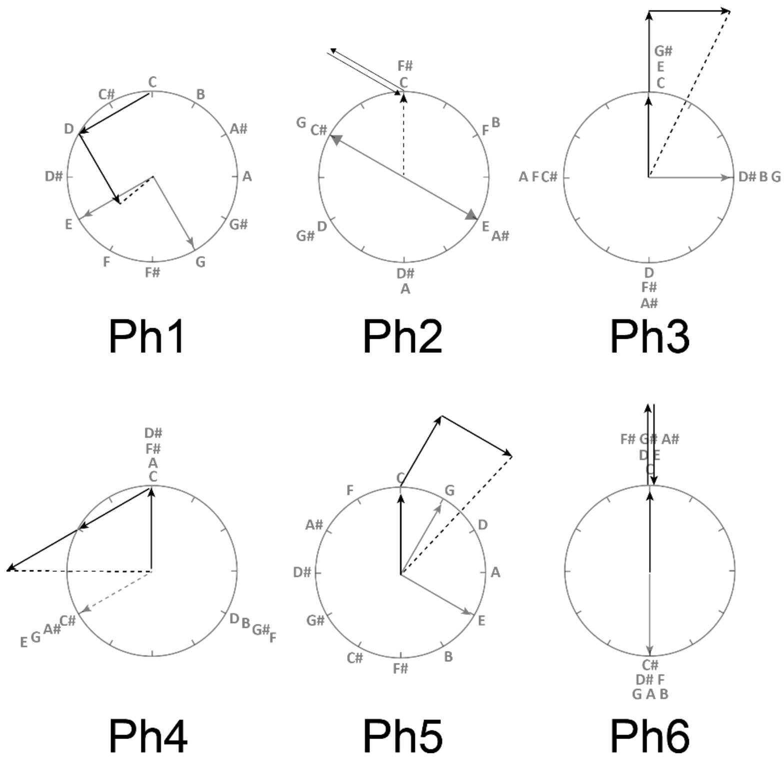


Figure 1-8 Circular averaging of the C major triad as performed by Yust (2016)

To fully recover a pitch-class set from phase spaces, an axis is drawn in corresponding phase for that phase space, and then project each pitch-class onto that axis to obtain the cosine function for that phase space (Figure 1-9). Cosine functions of each phase space are then weighted by their magnitude and added.

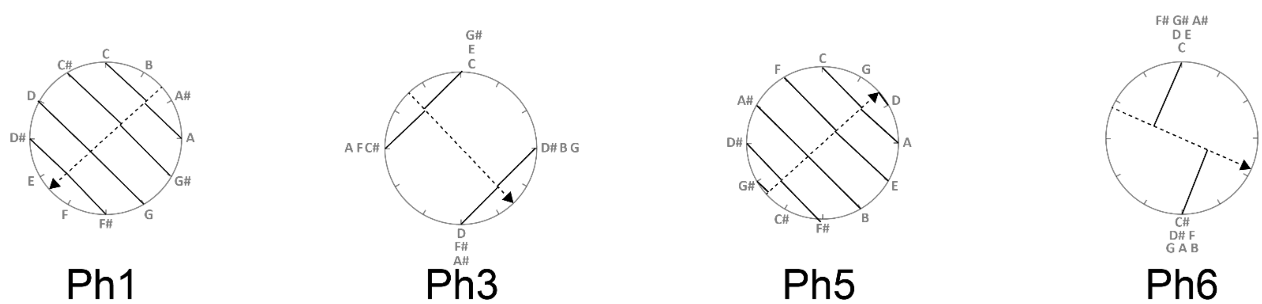


Figure 1-9 Phase space projections are determined from the vectors that result from circular averaging.

The relationship between Fourier theories of music and Dawson's strange circles raises a few questions: Are hidden unit spaces finding phase spaces? If so, are the phase spaces being used to distinguish families of pitch-class sets? How are they being employed and how does the network arrive at that solution? and what is the relation between phase spaces and pitch-class set families? How many phase spaces would be needed to distinguish pitch-class set families? The questions are explored in this thesis, which investigates the relationship between network structure and Fourier components.

Conclusion

The aim of this thesis is to study how ANNs classify musical entities. The basic methodology involves training networks to solve musical problems, and then interpreting their internal structure in order to understand how these networks work. This thesis extends Dawson's (2018) notion of strange circles by exploring whether network structures can be described in terms of Fourier components related to the DFT of musical sets. If such descriptions are possible, then a strong relationship between ANNs and musical set theory will be established. Furthermore, if such descriptions are possible, then Fourier components like Yust's phase space projections become potential, biologically plausible, representations for musical cognition. Chapter 2 of this thesis examines 3 different multilayered perceptrons with Gaussian activation functions (value unit networks), each trained on a particular musical classification problem: a triad classifying, interval measure and scale mode classifying. Even though these networks differ in their task or stimulus encoding, we find that there is a correspondence between network structures and Fourier phase spaces.

Having established that Fourier phase spaces are being used by value unit networks, Chapter 3 tests if these phase spaces are also captured by networks whose processors employ a different activation function. Several multilayered perceptrons with logistic activation functions were trained to classify triads. Chapter 3 also shows how one property of the logistic activation function can be used to describe ‘sympathetic vibration’ in these networks. For these networks, sympathetic vibration occurs when the similarity between the Fourier structure of a hidden unit and a stimulus increases, causing increased hidden unit activity. We describe how such sympathetic vibration is used by ANNs to generate correct responses. Chapter 4 provides closing remarks and discusses further directions in which to take this research. I argue that in order to inform cognitive science, researchers need to look inside the models that are being built. While many researchers are interested in studying music with artificial neural networks (Griffith & Todd, 1999; Todd & Loy, 1991) they do so expecting to capture informal properties of music (Bharucha, 1999). As a result, they rarely look inside their networks. In our exploration of the internal structure of networks, we will find some very formal properties of music. Because these formal properties are found in ANNs, they are plausible candidates for representations used in musical cognition. ANNs become a medium to provide rich links between musical cognition and modern music theory.

References

Amiot, E. (2016). *Music Through Fourier Space: Discrete Fourier Transform in Music Theory*. Cham: Springer International Publishing : Imprint: Springer.

Bharucha, J. J. (1999). Neural nets, temporal composites, and tonality. In D. Deutsch (Ed.), *The Psychology Of Music* (2nd ed., pp. 413-440). San Diego, CA: Academic Press.

Callender, C. (2007). Continuous harmonic spaces. *Journal of Music Theory, 51*(2), 277-332. doi: 10.1215/00222909-2009-004

Dawson, M. R. W. (2018). *Connectionist Representations of Tonal Music: Discovering Musical Patterns By Interpreting Artificial Neural Networks*. Edmonton, AB: Athabasca University Press.

Forte, A. (1973). *The Structure Of Atonal Music*. New Haven: Yale University Press.

Griffith, N., & Todd, P. M. (1999). *Musical Networks: Parallel Distributed Perception And Performace*. Cambridge, Mass.: MIT Press.

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

Lewin, D. (2001). Special cases of the interval function between pitch-class sets X and Y. *Journal of Music Theory, 45*(1), 1-30. doi: 10.2307/3090647

Quinn, I. (2006). General equal-tempered harmony: Introduction and Part 1. *Perspectives of New Music, 44*, 114-158.

Quinn, I. (2007). General equal-tempered harmony: Parts 2 and 3 *Perspectives of New Music, 45*, 4-63.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533-536.

Susanni, P., & Antokoletz, E. (2012). *Music and twentieth-century tonality: Harmonic progression based on modality and the interval cycles*: Routledge.

Todd, P. M., & Loy, D. G. (1991). *Music And Connectionism*. Cambridge, Mass.: MIT Press.

Yust, J. (2016). Special collections: Renewing set theory. *Journal of Music Theory, 60*(2), 213-262. doi: 10.1215/00222909-3651886.

Chapter 2 : Artificial Neural Networks Solve Musical Problems With Fourier Phase Spaces

Note: A version of this chapter has been published: Dawson, M.R.W., Perez, A., & Sylvestre, S. (2020). Artificial neural networks solve musical problems with Fourier phase spaces. *Scientific Reports, 10(1), 7151*. doi: 10.1038/s41598-020-64229-4.

Abstract

How does the brain represent musical properties? Even with our growing understanding of the cognitive neuroscience of music (Abbott, 2002; Peretz and Zatorre, 2003; Peretz and Zatorre, 2005; Zatorre and McGill, 2005), the answer to this question remains unclear. One method for conceiving possible representations is to use artificial neural networks, which can provide biologically plausible models of cognition (Rumelhart and McClelland, 1986; Bechtel and Abrahamsen, 2002; Enquist and Ghirlanda, 2005). One could train networks to solve musical problems, (Todd and Loy, 1991; Griffith and Todd, 1999) and then study how these networks encode musical properties. However, researchers rarely conduct detailed examinations of network structure(Dawson, 2009, 2013, 2018) because networks are difficult to interpret, and because it is assumed that networks capture informal or subsymbolic properties (Smolensky, 1988; McCloskey, 1991; Bharucha, 1999). Here we report very high correlations between network connection weights and discrete Fourier phase spaces used to represent musical sets (Amiot, 2016; Callender, 2007; Quinn, 2006, 2007; Yust, 2016). This is remarkable because there is no clear mathematical relationship between network learning rules and discrete Fourier analysis (Rumelhart, Hinton et al., 1986; Dawson and Schopflocher, 1992; Amiot, 2016). That networks discover Fourier phase spaces indicates that these spaces have an important role to play outside of formal music theory.

Finding phase spaces in networks raises the strong possibility that Fourier components are possible codes for musical cognition.

Introduction

A main goal of studying musical cognition is identifying musical representations. Researchers seek musical representations by conducting psychological experiments, (Sloboda, 1985; Krumhansl, 1990; Deutsch, 1999) or by using the methods of cognitive neuroscience (Abbott, 2002; Peretz and Zatorre, 2003; Peretz and Zatorre, 2005; Zatorre and McGill, 2005). However, the nature of musical representations is still unclear (Zatorre and McGill, 2005).

A different method for generating new ideas about musical representations involves training artificial neural networks (ANNs). An ANN is a brain-like system of processors (analogous to neurons). Processors convert incoming signals into activity that is then sent to other processors via weighted connections (analogous to synapses). ANNs learn to convert stimuli into correct responses by modifying connection weights (Rumelhart, Hinton et al., 1986; Dawson and Schopflocher, 1992). Many researchers train ANNs to solve musical problems (Todd and Loy, 1991; Griffith and Todd, 1999; Dawson, 2018). While one could examine network structures to discover new musical representations (Dawson, 2018), this approach is rarely taken. It is widely assumed that ANNs are difficult to interpret because they capture subsymbolic properties (Smolensky, 1988; McCloskey, 1991; Bharucha, 1999). Indeed, some music researchers are attracted to ANNs because they assume their networks capture important properties that *cannot* be formalized (Bharucha, 1999).

Musical set theory provides a quite different, formal, approach to studying music (Forte, 1973; Lewin, 2007; Schuijer, 2008). It converts the twelve different Western pitch-classes (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) into integers using the convention C = 0, C# = 1, and so on. Musical entities such as intervals, scales, or triads are combinations of pitch-classes that can be represented as musical sets. For instance, using this scheme the C major triad (C, E and G) becomes the set (0, 4, 7). Mathematical operations on musical sets uncover striking similarities between musical entities that otherwise seem quite dissimilar(Forte, 1973). Musical objects can also be encoded as ordered sets of twelve integers; the integer code of pitch-classes defines the order, while set members represent the number of each pitch-class present. Thus, the C major triad becomes (1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0). Theorists find important musical regularities by decomposing such sets with the discrete Fourier transform (DFT) (Quinn, 2006; Callender, 2007; Quinn, 2007; Amiot, 2016; Yust, 2016).

Schuijer notes that critics worry that musical set theory is too formal to capture important informal characteristics (Schuijer, 2008). If ANNs capture informal musical properties (Bharucha, 1999), then we would expect that they are unrelated to musical set theory. Nevertheless, here we show striking relationships between networks and the discrete Fourier analysis of musical sets.

An Example Network

Figure 2-1A depicts an ANN for measuring the interval distance, in semitones, between pairs of pitch-classes. For instance, the pitch-classes C and G are separated by either 4 or 5 semitones depending on whether one measures distance clockwise or counterclockwise after arranging pitch-classes in a circle (Figure 2-1B). The network is trained on 66 different pairs of pitch-classes using standard procedures (Dawson, 2018). To learn to respond correctly to each stimulus, the ANN requires four hidden units, each using a Gaussian activation function, to detect higher-order stimulus properties. Four hidden units is the minimum number that is required for the network to solve this problem. Previous research has shown that using the Gaussian activation function, and focusing on units that are limited to the minimum number of hidden units required, increases the likelihood that the internal structure of a trained network can be understood (Dawson, 2009; Dawson, 2018). A trained network's connection weights from the input units to each hidden unit are plotted in Figures 1C-1F.

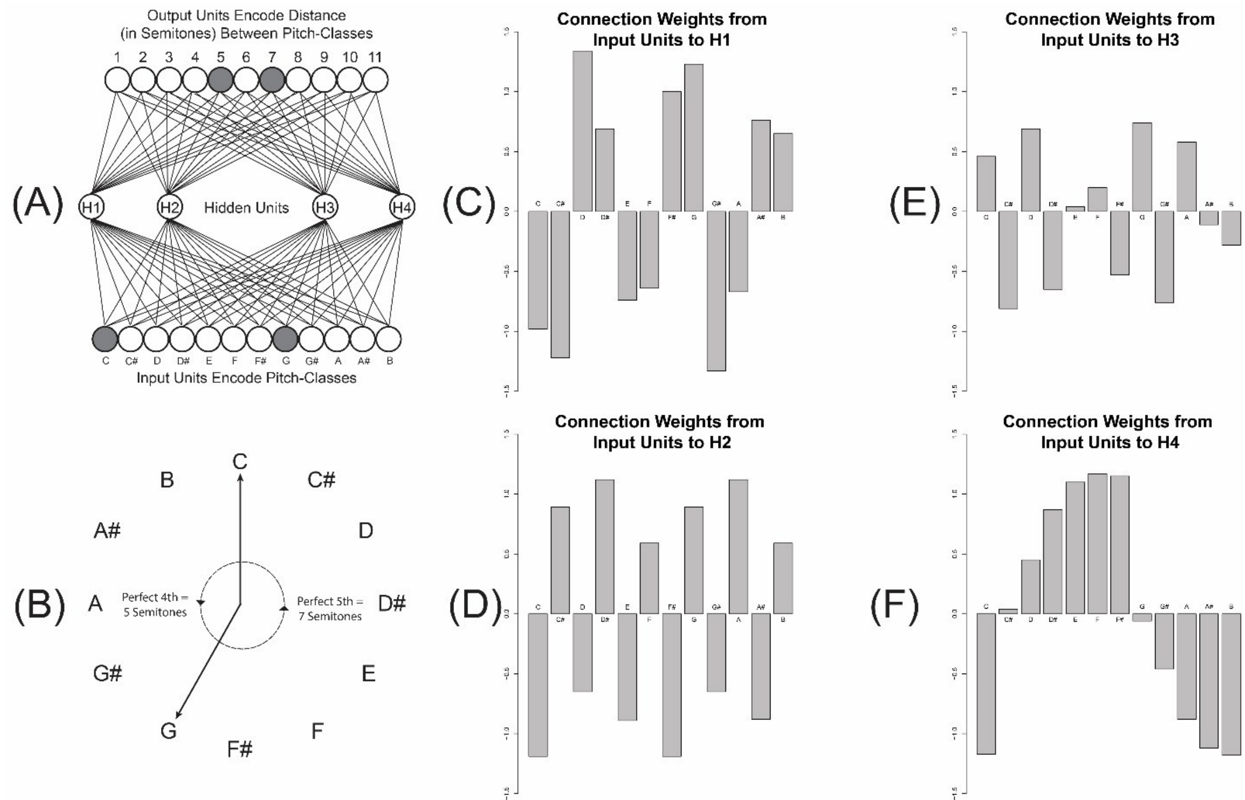


Figure 2-1 (A) An ANN for measuring distances between pairs of pitch-classes. Stimuli are presented by activating two input units; 1A illustrates presenting C and G. The ANN learns to activate the output units that measure (as in (B)) the distances between the two pitch-classes by processing signals from four hidden units. All output and hidden units employ a Gaussian activation function $f(net) = e^{-\pi(net-\mu)^2}$ *that ranges between 0 and 1; net is the processor's incoming signal, and μ is the function's mean. Weights are modified by a variant of the generalized delta rule (Dawson and Schopflocher, 1992). The bar plots (1C-1F) depict the weights of connections between input units and each hidden unit after training ends.*

These connection weights show formal, but qualitative, musical properties (Dawson, 2018). For example, Processor H1's weights (Figure 2-1C) reveal *tritone balance*: two pitch-classes separated by a tritone (i.e., six semitones, such as C and F#, C# and G, etc.) are assigned weights equal in magnitude, but opposite in sign. Thus, the second six bars in Figure 2-1C are produced by inverting the graph's first six bars. Tritone balance is also true of H3 and H4 (Figures 1E, 1F). H2 reveals *tritone equivalence*, in which pitch-classes a tritone apart are assigned the same connection weights (Figure 2-1C). H2 also divides pitch-classes into two interval cycles in which adjacent members are separated by two semitones (e.g., once cycle is C, D, E, F#, G# and B). Members of one cycle are assigned a negative connection weight, while members of the other are assigned a positive weight.

Crucially, Figures 1C-1F also suggest that weights are related to musical DFT components. Figure 2-2 presents the six Fourier phase spaces used to analyze musical sets by Yust (2016). Each phase space places the twelve pitch-classes at particular positions around a clock face whose center is (0, 0). If one plots the y-position on the clock of each pitch-class in the order C, C#, D, and so on, a cosine function is produced. Phase spaces differ from one another with respect to cosine frequency. The first phase space has a frequency of 1, the second has a frequency of 2, and so on. Yust uses these phase spaces to perform a Fourier decomposition of a musical set as follows: For each phase space, vectors are drawn from the phase space origin to each pitch-class that is present in the set. These vectors are then added. The orientation of the resultant vector represents the phase of the phase space's cosine function; the length of the resultant vector represents the magnitude of the cosine function. To generate Fourier components for reconstructing a musical set, an oriented arrow is drawn to bisect the phase space; the orientation is that of the resultant vector for that phase space. Yust then projects each pitch-class onto the arrow and measures the distance from the projection to the phase space's origin (see the Figure 2-2 bar graphs). When these projections are weighted by the magnitudes of the resultant vectors, the original musical set can be reconstructed.

Importantly, the qualitative structure of the phase space projections in Figure 2-2 is similar to Figures 1C – 1F. For example, tritone equivalence is true of the bar graphs for phase spaces Ph2, Ph4, and Ph6, while tritone balance is true of Ph1, Ph3, and Ph5.

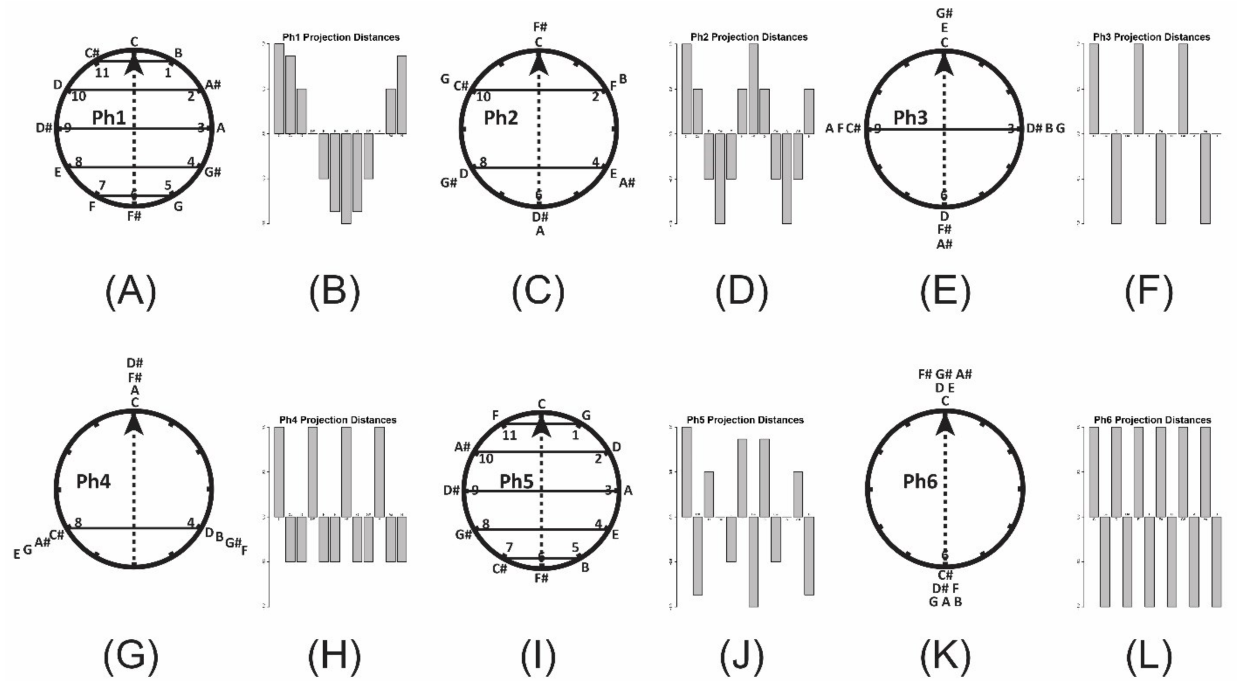


Figure 2-2 The six Fourier phase spaces, Ph1, Ph2, and so on used by Yust (2016). Each space represents the twelve pitch-classes on a clock face. (A) Phase space Ph1. The arrow, pointing at 0 o'clock, indicates a phase of 0. Solid lines denote projections from the pitch-class locations on the clock to the arrow. (B) Distances between each projection in (A) and the center of the phase space. Distances towards the arrowhead are positive. (C, D) Ph2 and its projection distances. (E, F) Ph3 and its projection distances. (G, H) Ph4 and its projection distances. (I, J) Ph5 and its projection distances. (K, L) Ph6 and its projection distances. The projection distance graphs show that each phase space represents a cosine function, and that the difference between phase spaces is the frequency of this function. Ph1 has a frequency of 1, Ph2 has a frequency of 2, and so on. Musical sets can be reconstructed by adjusting the phase of each component, weighting them and summing them together.

Is there a stronger, quantitative, relationship between hidden unit weights and Fourier phase spaces? To answer this question, we fit each set of hidden unit weights to each of six Fourier phase spaces. We rotated each phase space's arrow to the position yielding the highest correlation between weights and projection distances. After finding the best-fitting phase, we used regression to predict weights from the distances in order to scale distances to match weights most closely.

Figure 2-3 illustrates the results of this analysis for the Figure 2-1 ANN; there is a high correlation between each set of weights and a phase space, with *r* ranging from 0.97 to 0.99.

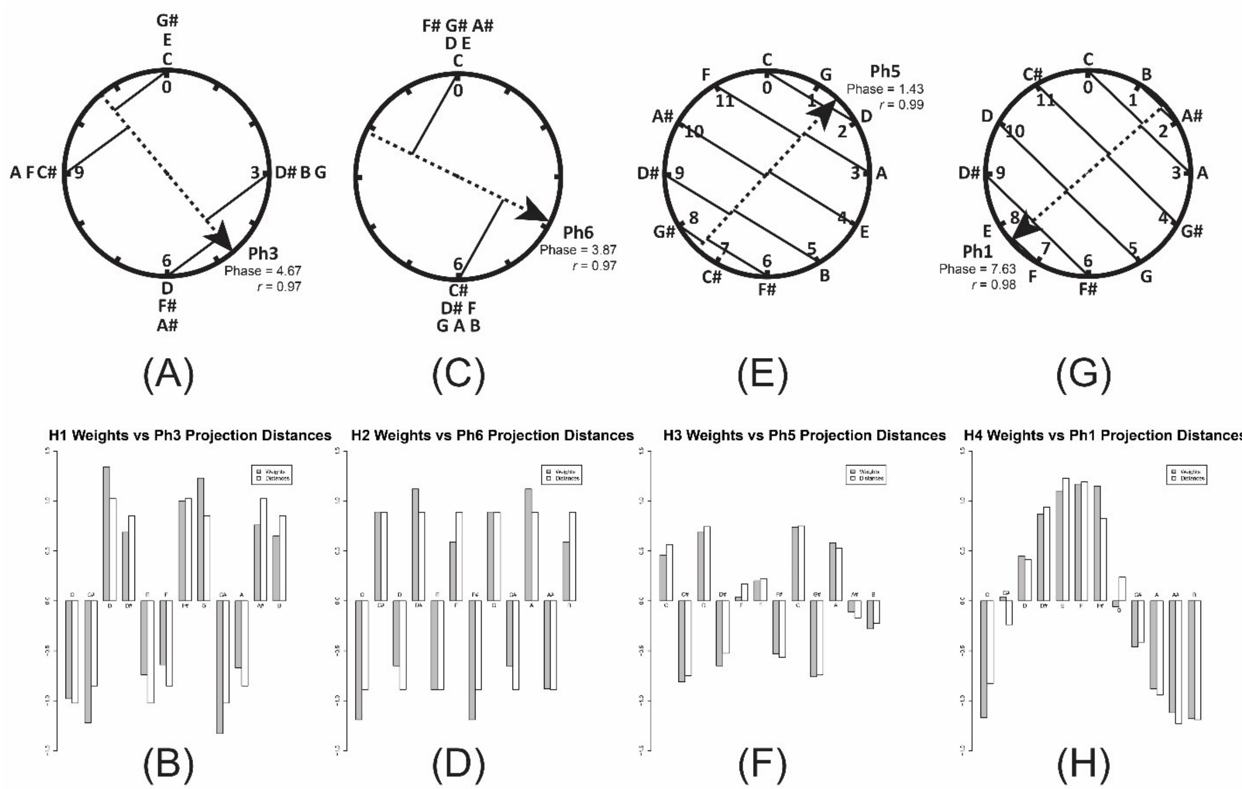


Figure 2-3. The results of fitting Fourier phase spaces to hidden unit weights for the Figure 2-1 network. (A) The weights for processor H1 are best fit by Ph3 (see Figure 2-2) with a phase of 2.47 (r = 0.97). (B) The relationship between H1 connection weights and projected distances after scaling the distances by 1.33. Projection distances are measured as described in Figure 2-2. (C) H2 weights are best fit by Ph6 with a phase of 3.87 (r = 0.97). (D) H2 weights and projected distances scaled by 2.02. (E) H3 weights are best fit by Ph5 with a phase of 1.43 (r = 0.99). (F) The relationship between H3 weights and projected distances scaled by 0.774. (G) H4 weights are best fit by Ph1 with a phase of 7.63 (r = 0.98). (H) H4 weights and projected distances scaled by 1.25.

Results From Other Networks

This result is replicated by other networks that learn to solve the same problem. We trained 20 different ANNs to measure intervals, with each network starting from a random weight configuration. Across 80 different hidden units (i.e., the 4 hidden units for each network), the average correlation between weights and the best-fitting phase space was 0.945 (S.D. = 0.056). In a second set of simulations, we trained 20 different ANNs to classify scales (seven different pitch-classes) as belonging to one of seven different scale modes. Networks required five hidden units to classify 84 stimuli. Again, we found a very high correlation between one phase space and the connection weights of each of the 100 sets of hidden unit weights of these networks (mean = 0.881, S.D. = 0.131). In a third set of simulations, we trained 20 different networks to classify triplets of pitch-classes into four different triad types. These networks required three hidden units to classify 48 stimuli. Each of the 60 hidden units from these networks had a high correlation between their connection weights and a single phase space.

Figure 2-4 presents the mean correlation between connection weights and phase spaces for all of the simulations described above. For each condition, six different mean correlations were computed: the average correlation with the best fitting phase space, the average correlation with the next best fitting phase space, and so on. Figure 2-4 reveals that on average a hidden unit's connection weights are highly related to a single Fourier phase space. The average correlation between the weights and the best fitting phase space is striking higher than is the average correlation between the weights and the second best fitting phase space. That the standard errors of adjacent bars in Figure 2-4 do not overlap indicates that the difference between them is statistically different. Dependent t-tests confirm this result for all comparisons. For example, when all problems are combined the mean correlation for the best fitting phase space (0.91) is significantly higher than the mean correlation for the next best fitting phase space (mean = 0.28; t = 33.819, DF = 239, $p < 2.2e\text{-}16$). Similarly, the means for the same two bars for mode network problems (0.88 vs 0.31) are significantly different t = 15.123, DF = 99, $p < 2.2e\text{-}16$).

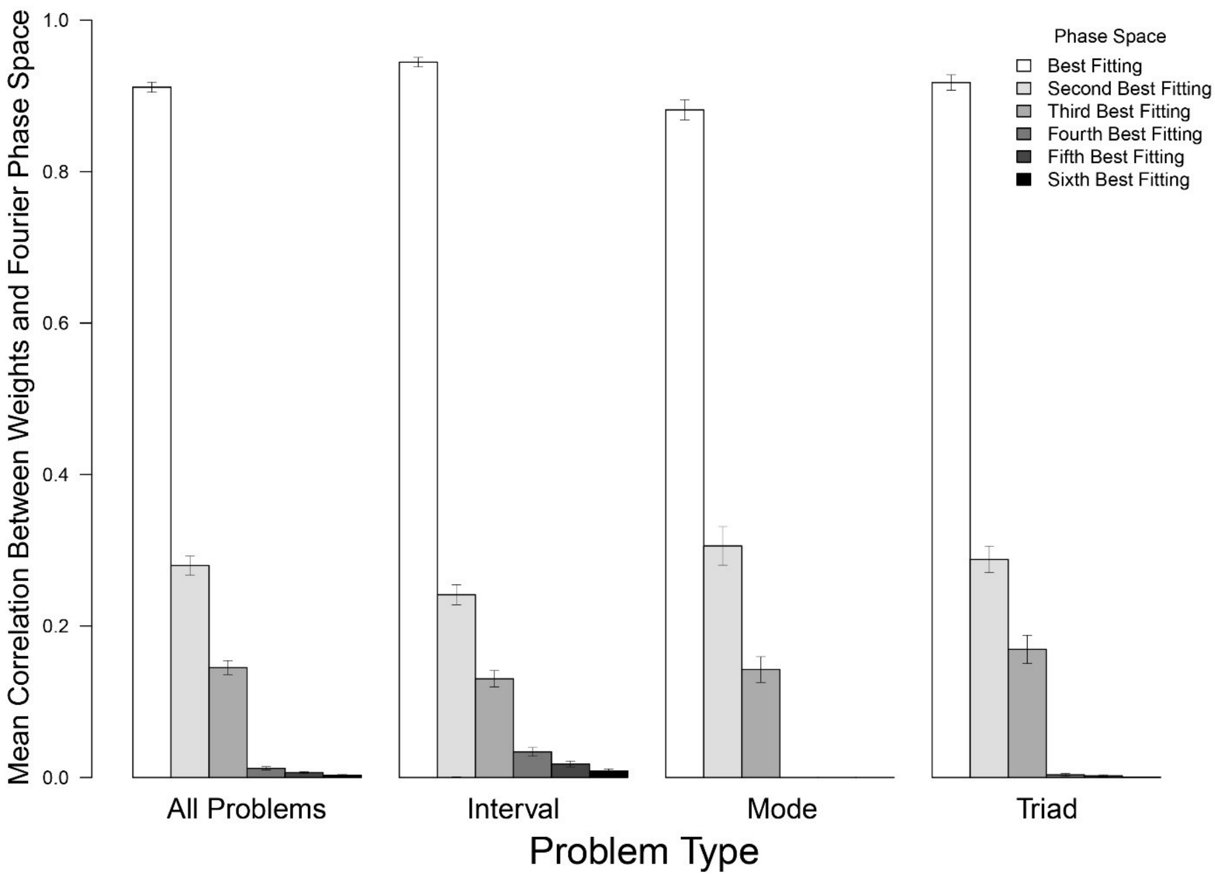


Figure 2-4. The mean correlations between hidden unit weights and Fourier phase space projections, with standard errors. The 'All Problems' data is based upon 240 different hidden units. The 'Interval', 'Mode', and 'Triad' data is based upon 80, 100, and 60 hidden units respectively. For each hidden unit, six different correlations were calculated, one for each phase space. The correlations were then ordered from the best fit to the worst fit, as is reflected in the six different bars presented for each problem type.

Discussion

How do ANNs solve musical problems with the phase spaces they discover? Networks create a hidden unit space that codes each stimulus as a point whose coordinates are the hidden unit activities that it causes (Dawson, 2018). Gaussian output units carve this hidden unit space into decision regions with two parallel, narrowly separated, planes. To turn an output unit on, a point must lie between its planes. The network learns to position similar stimuli between the same planes, permitting an output unit to respond to them (and not to others). In music theory, Fourier components represent rich properties of musical objects, such as the complete set of musical intervals spanned by a set of pitch-classes (Quinn, 2006; Callender, 2007; Quinn, 2007; Amiot, 2016; Yust, 2016). Hidden unit activity represents the similarity between a phase space and a stimulus. Thus, when musical stimuli are arranged in hidden unit space, their positions are determined by their intervallic structure, which is explicitly encoded by Fourier phase spaces: the hidden unit weights.

That ANNs discover discrete Fourier phase spaces is surprising because there is no obvious mathematical relationship between network training and computing the DFT. For instance, the DFT of a musical set requires correlating it with a various cosine and sine functions of different frequencies (Amiot, 2016). In contrast, ANNs are modified by small, iterative weight changes designed to reduce network response error (Rumelhart, Hinton et al., 1986; Dawson and Schopflocher, 1992). Furthermore, the DFT is computed for individual musical entities. In contrast, our ANNs are trained with multiple stimuli; the phase spaces they discover are applied to *every* stimulus.

The discovery of discrete Fourier phase spaces in musical ANNs emphasizes the importance of conducting detailed examinations of networks (Dawson, 2018). It also reveals that these spaces have an important role to play beyond formal music theory. ANNs can generate new, biologically plausible, proposals about mental representation (Rumelhart and McClelland, 1986; Smolensky, 1988; Bechtel and Abrahamsen, 2002; Dawson, 2004). If basic learning rules for ANNs discover Fourier phase spaces, then this strongly suggests that such spaces are possible representations for musical cognition. Finding these representations in ANNs raises the possibility that Fourier phase spaces can bridge the cognitive psychology of music with its cognitive neuroscience (Tillmann, Bharucha et al., 2003)

References

Abbott, A. (2002). "Neurobiology: Music, maestro, please!" *Nature* **416**(6876): 12-14.

Amiot, E. (2016). *Music Through Fourier Space: Discrete Fourier Transform in Music Theory*. Cham, Springer International Publishing : Imprint: Springer.

Bechtel, W. and A. A. Abrahamsen (2002). *Connectionism And The Mind : Parallel Processing, Dynamics, And Evolution In Networks*. Malden, MA, Blackwell.

Bharucha, J. J. (1999). Neural nets, temporal composites, and tonality. *The Psychology Of Music*. D. Deutsch. San Diego, CA, Academic Press**:** 413-440.

Callender, C. (2007). "Continuous harmonic spaces." *Journal of Music Theory* **51**(2): 277-332.

Dawson, M. R. W. (2004). *Minds And Machines: Connectionism And Psychological Modeling*. Malden, MA, Blackwell Pub.

Dawson, M. R. W. (2009). Computation, cognition – and connectionism. *Cognition, Computation, and Pylyshyn*. D. Dedrick and L. Trick. Cambridge, MA, MIT Press**:** 175-199.

Dawson, M. R. W. (2013). *Mind, Body, World: Foundations Of Cognitive Science*. Edmonton, AB, Athabasca University Press.

Dawson, M. R. W. (2018). *Connectionist Representations of Tonal Music: Discovering Musical Patterns By Interpreting Artificial Neural Networks*. Edmonton, AB, Athabasca University Press.

Dawson, M. R. W. and D. P. Schopflocher (1992). "Modifying the generalized delta rule to train networks of nonmonotonic processors for pattern classification." *Connection Science* **4**: 19-31.

Deutsch, D. (1999). *The Psychology Of Music*. San Diego, Academic Press.

Enquist, M. and S. Ghirlanda (2005). *Neural Networks and Animal Behavior*. Princeton, Princeton University Press.

Forte, A. (1973). *The Structure Of Atonal Music*. New Haven, Yale University Press.

Griffith, N. and P. M. Todd (1999). *Musical Networks: Parallel Distributed Perception And Performace*. Cambridge, Mass., MIT Press.

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

Lewin, D. (2007). *Generalized Musical Intervals And Transformations*. New York, Oxford University Press.

McCloskey, M. (1991). "Networks and theories: The place of connectionism in cognitive science." *Psychological science* **2**: 387-395.

Peretz, I. and R. J. Zatorre (2003). *The Cognitive Neuroscience Of Music*. Oxford ; New York, Oxford University Press.

Peretz, I. and R. J. Zatorre (2005). "Brain organization for music processing." *Annual Review of Psychology* **56**: 1-26.

Quinn, I. (2006). "General equal-tempered harmony: Introduction and Part 1." *Perspectives of New Music* **44**: 114-158.

Quinn, I. (2007). "General equal-tempered harmony: Parts 2 and 3 " *Perspectives of New Music* **45**: 4-63.

Rumelhart, D. E., G. E. Hinton, et al. (1986). "Learning representations by back-propagating errors." *Nature* **323**(6088): 533-536.

Rumelhart, D. E. and J. L. McClelland (1986). *Parallel Distributed Processing, V.1*. Cambridge, MA, MIT Press.

Schuijer, M. (2008). *Analyzing atonal music: Pitch-class set theory and its contexts*. Rochester, NY, University of Rochester Press.

Sloboda, J. A. (1985). *The Musical Mind: The Cognitive Psychology Of Music*. Oxford, Oxford University Press.

Smolensky, P. (1988). "On the proper treatment of connectionism." *Behavioral and Brain Sciences* **11**: 1-74.

Tillmann, B., J. Bharucha, et al. (2003). Learning and perceiving musical structures: Further insights from artificial neural networks. *The Cognitive Neuroscience of Music*. I. Peretz and R. Zatorre. Oxford, Oxford University Press**:** 109-123.

Todd, P. M. and D. G. Loy (1991). *Music And Connectionism*. Cambridge, Mass., MIT Press.

Yust, J. (2016). "Special collections: Renewing set theory." *Journal of Music Theory* **60**(2): 213-262.

Zatorre, R. and J. McGill (2005). "Music, the food of neuroscience?" *Nature* **434**(7031): 312-315.

Acknowledgments

We would like to thank Brad Gersh of Princeton University for first bringing to our attention the possible relationship between our ANN weights and Fourier phase spaces. We would also like to thank Jeremy Caplan, Clayton Dickson, Michael Frishkopf, Lauren Guillette and Dana Hayward for their suggestions on this project. All authors contributed equally to this manuscript. There is no funding to declare, and authors declare no competing interests.

Chapter 3 : Artificial Neural Networks Use 'Sympathetic Vibration' To Classify Musical Triads

Note: A version of this chapter is currently under editorial review: Perez, A.I., Ma, H.L., and Dawson M.R.W. (Under editorial review). Artificial Neural Networks Use 'Sympathetic Vibration' To Classify Musical Triads. *Cognitive Science*, under review (10,796 words, submitted June 14, 2021).

Abstract

How does the brain represent musical properties? One method for conceiving representations is to use artificial neural networks, which can provide biologically plausible models of cognition. Our previous work studied networks whose processors used the Gaussian activation function, and discovered very high correlations between network connection weights and discrete Fourier phase spaces used to represent musical sets. The current research studies triad classification in networks whose processors use the more typical logistic activation function. Again, we discover Fourier components represented by connection weights. We then take advantage of a property of the logistic monotonicity, to determine hidden unit activity reflects the similarity between the Fourier structure of the hidden unit and of a stimulus. Metaphorically, hidden unit activity reflects 'sympathetic vibration' between network and stimulus structures. Finally, we examine an example network to determine how 'sympathetic vibrations' are used to classify triads. We discover the network uses coarse coding in which individual hidden units are poor triad classifiers, but provide some evidence in favor of one classification over another. Output units combine and weight this evidence to determine a correct response to each stimulus. Our results indicate musical networks may often represent Fourier properties of music, as such representations are evident in networks which use radically different activation functions. Network use of Fourier phase spaces indicates

these spaces have an important role to play beyond formal music theory. Finding phase spaces in networks indicates Fourier components are possible codes for musical cognition.

Introduction

Musical cognition aims to identify musical representations. Many cognitive psychologists seek musical representations by experimenting on human listeners (Deutsch, 1999; Krumhansl, 1990; Sloboda, 1985). Cognitive neuroscience methods have also been employed (Abbott, 2002; Peretz & Zatorre, 2003, 2005; Zatorre & McGill, 2005). The cognitive neuroscience of music raises a new question: how does the brain represent musical properties? Answers to this question remain elusive (Zatorre & McGill, 2005).

One approach to discover how the brain might represent musical properties uses artificial neural networks (ANNs). An ANN is a brain-like computer model of processors (analogous to neurons). ANNs transform stimuli into responses (Rumelhart & McClelland, 1986). Processors convert incoming signals into activity, and send activity to other processors via weighted connections (analogous to synapses). A growing number of researchers explore musical cognition using ANNs (Dawson, 2018; Griffith & Todd, 1999; Todd & Loy, 1991).

In this manuscript, we investigate musical representations by studying the ANN illustrated in Figure 3-1. This network, a *multilayer perceptron* (Rumelhart, Hinton, & Williams, 1986), classifies stimuli (musical triads) into four triad types. A triad is presented by activating (turning on or off) the ANN's input units. Input units send activities to hidden units; connection weights scale each signal. Hidden units sum their received signals (i.e., calculate *net input*), and then convert net input into internal activity. Hidden unit activities represent higher-order stimulus features and send signals about stimulus features to output units. Output units convert hidden unit signals into activity, again by transforming net inputs into activities. Figure 3-1 network classifies a triad by turning on one output unit and turning off the other three.

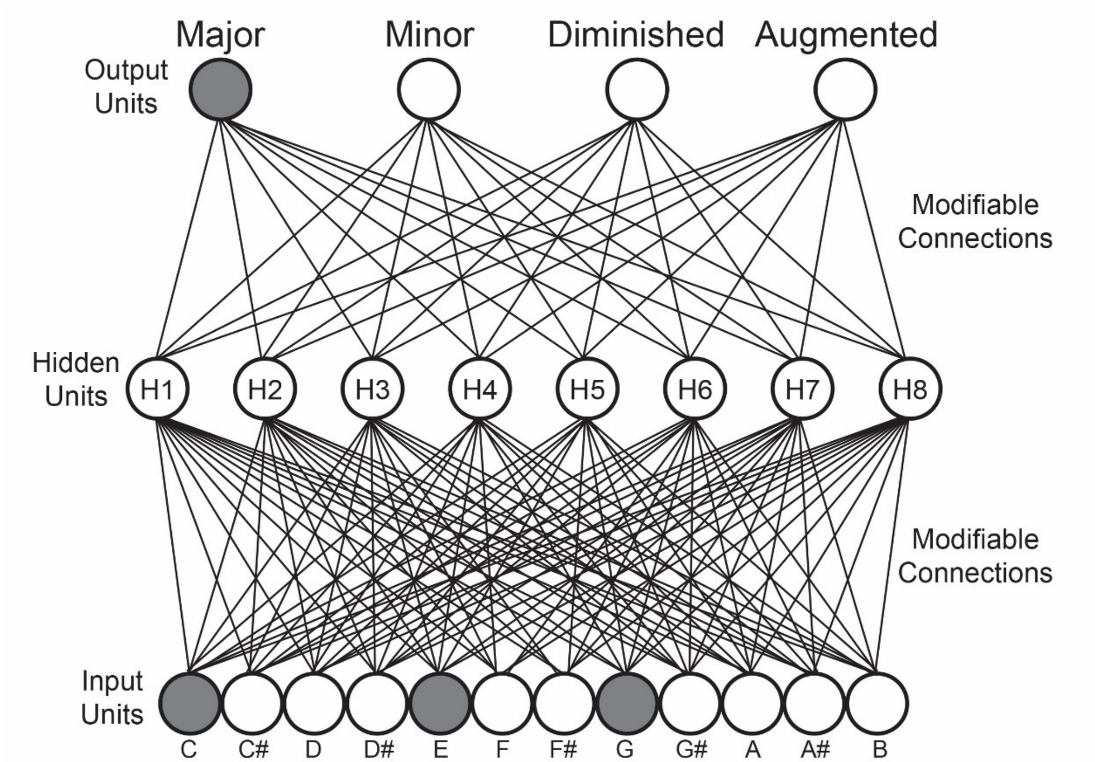


Figure 3-1. An example multilayer perceptron for classifying triads.

Many researchers favor using ANNs because they *learn* to perform a desired stimulus-response mapping. The connections in the Figure 3-1 network begin as small, random values, and are then modified by learning. Learning involves teaching the network with a training set, a collection of stimulus-response pairs. During learning, the network is presented a stimulus, causing the output units to activate. Error is computed by taking the difference between observed responses and the desired responses associated with the stimulus in the training set. Error is then used to modify the network's connection weights to decrease network error (Rumelhart et al., 1986). Repeatedly presenting the training set, and using error to modify connection weights, reduces errors to each stimulus as much as possible. Crucially, networks can learn new representations for cognitive phenomena. Researchers do not insert ideas about representation into networks; networks uncover surprising representations on their own.

To understand musical representations discovered by ANNs, one must examine the networks' inner structures – connection weights, features detected by hidden units, and so on (Dawson, 2018). Without understanding network structure, ANNs *cannot* inform cognitive theory, yet researchers do not often interpret musical ANNs (McCloskey, 1991; Seidenberg, 1993). ANNs are popular in musical cognition because many researchers believe networks capture *informal* musical properties (Bharucha, 1999), and have little interest in investigating whether ANNs represent formal properties.

However, musical ANNs discover rich, surprising, formal representations (Dawson, 2018; Dawson, Perez, & Sylvestre, 2020). For example, connection weights from input units to hidden units (Figure 3-1) often represent musical properties using discrete Fourier transform (DFT) components. In this paper, we determine whether this result extends to different networks, and investigate how such representations mediate musical decisions. Before describing our research, we introduce using the DFT to describe musical sets.

Musical Sets and Fourier Phase Spaces

One formal approach for studying music represents musical entities as mathematical sets (Forte, 1973; Lewin, 2007; Schuijer, 2008). Set theorists convert the twelve different Western pitch-classes (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) into integers using the convention C = 0, C# = 1, and so on. For example, using this scheme the C major triad (C, E and G) becomes the set (0, 4, 7). An alternate C major set representation is (1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0), a representation called a pitch-class set or *pc-set*. A pc-set is an ordered set; the position in the set corresponds to each pitch-class's integer code of pitch-classes (C in position 1, C# in position 2, etc.); set members represent the number of each pitch-class present in an entity. Mathematical operations on musical sets uncover striking similarities between musical entities that otherwise seem dissimilar.

One such operation is the discrete Fourier transform (DFT). The DFT converts a complex signal into a set of simpler components. Each component is a cosine of a particular frequency, magnitude, and phase. Performing DFT on a pc-set delivers twelve different cosines, each represented by a complex number (Amiot, 2016). The real and imaginary parts of the complex number define the cosine's magnitude and phase. Adding together the cosines delivered by the DFT recreates the original pc-set. Fourier representations permit numerous musical relationships to be formalized (Amiot, 2016; Quinn, 2006, 2007; Yust, 2016, 2017a, 2019).

This paper employs Yust's method to determine a pc-set's DFT (Yust, 2016). Yust uses six different phase spaces (Ph1, Ph2, etc.). Ph1 represents a cosine with 1 Hz frequency, Ph2 represents a cosine with 2 Hz frequency, and so on. A clock face represents each phase space; pitch-classes are arranged around the clock according to the space's frequency (see Figure 3-2). The arrangements of pitch-classes causes each phase space to represent different harmonic qualities. Hanson describes our experience of tone combinations as being sensitive to a small number of combination types. The members of each type consist of sounds predominantly composed of one basic interval (e.g., mostly perfect fifths, mostly minor thirds, etc.) (Hanson, 1960). Each category proposed by Hanson is most closely associated with one of the six phase spaces (Quinn, 2006, 2007). For instance, musical entities consisting mostly of perfect fifths are best described by Ph5.

Yust (2016) describes the DFT as follows: Each pitch-class in a pc-set defines a vector from the clock's center to the pitch-class position in a phase space. Figure 3-2 illustrates this for the C major triad; each phase space contains three vectors (dashed arrows) representing the triad's three components (C, E, and G). The solid arrows in Figure 3-2 represent the sum of the three vectors.

A vector sum's direction indicates a cosine's phase; a vector sum's length indicates the cosine's magnitude. Yust's method is formally equivalent to Amiot's (Yust, 2016, 2017b). For instance, the vector sum's *x* and *y* coordinates correspond, respectively, to the imaginary and real parts of the complex number representing a Fourier component. Similarly, each phase space is used twice when during set reconstruction, because each space is identical to two components delivered by Amiot's DFT procedure.

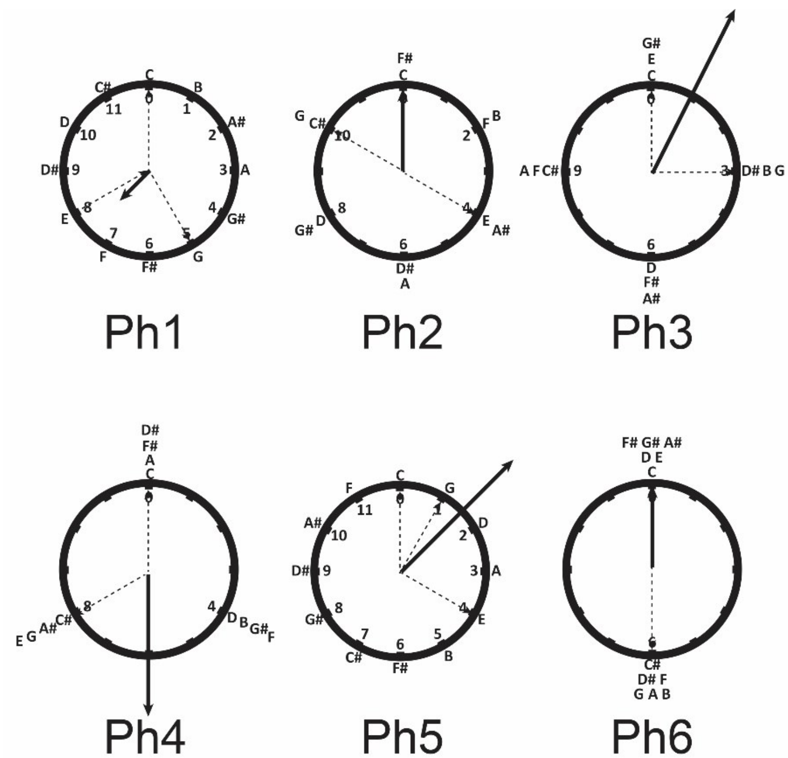


Figure 3-2 Using Yust's (2016) phase spaces to perform the DFT of the C major triad. Dashed arrows represent vector components of the triad. Solid arrows represent the vector sums of triad components.

Yust (2016) uses vector sums (see Figure 3-2) to determine what cosine components to combine to reproduce the original pc-set. Yust constructs a diameter through a phase space (dashed arrows in Figure 3); the diameter's orientation is identical to the vector sum's direction in the space. Yust then projects each pitch-class from its position on the clock face to the diameter (solid lines in

Figure 3-3). Yust then determines the distance from the clock's center, along the diameter, to each projection. A positive distance is from the center in the direction of the arrow on the dashed line; otherwise, the distance is negative (see Figure 3-3). The twelve distances define the *phase space projections* for a phase space. The phase space projections plotted in Figure 3-3 represent discretely sampled cosines. The original pc-set is reconstructed by summing the six sets of phase space projections together, after weighting them.

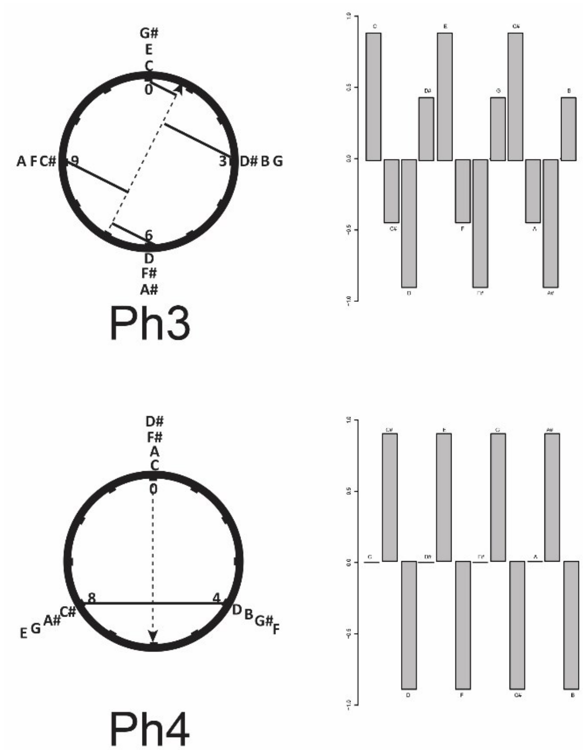


Figure 3-3 Two examples of converting vector sums into phase space projections. The two dashed arrows are diameters pointing in directions of Figure 3-2 vector sums. The solid lines represent projections of pitch-classes to the diameter. The histograms indicate phase space projections: distances from the center of the circle along the diameter.

Our paper focuses on phase space projections. Our previous research trained multilayer perceptrons to judge pc-sets, and examined the connections from input units to hidden units in these networks. We discovered connection weights represent phase space projections (Dawson et al., 2020). We now examine whether different kinds of networks also represent phase space projections, and determine how phase space projections can mediate musical judgments, like classifying triads.

Rationale For Current Research

Figure 3-1 illustrates a general architecture, the multilayer perceptron. However, different kinds of multilayer perceptrons use different processing units. Processing units convert net input into internal activity by using a mathematical equation called an *activation function*. Different processing units use different activation functions; different activation functions cause dramatic changes in processor behavior. ANNs constructed from different processors often discover qualitatively different representations for converting stimuli into responses (Dawson, Berkeley, Medler, & Schopflocher, 1994; Shamanski & Dawson, 1994).

One kind of processor is the *value unit* (Ballard, 1986). A value unit activates to a very narrow range of net inputs; if net input falls outside this range (too high or too low), then a value unit does not respond. Retinal receptors, tuned to a narrow range of light wavelengths, provide biological analogue. In an ANN, a value unit uses the bell-shaped Gaussian equation as the activation function (Dawson, 1990). In the equation, *net* represents net input, and μ is the mean of the Gaussian. When net input equals μ, the equation generates a maximum activity of 1. In Figure 3-4A μ equals 0.

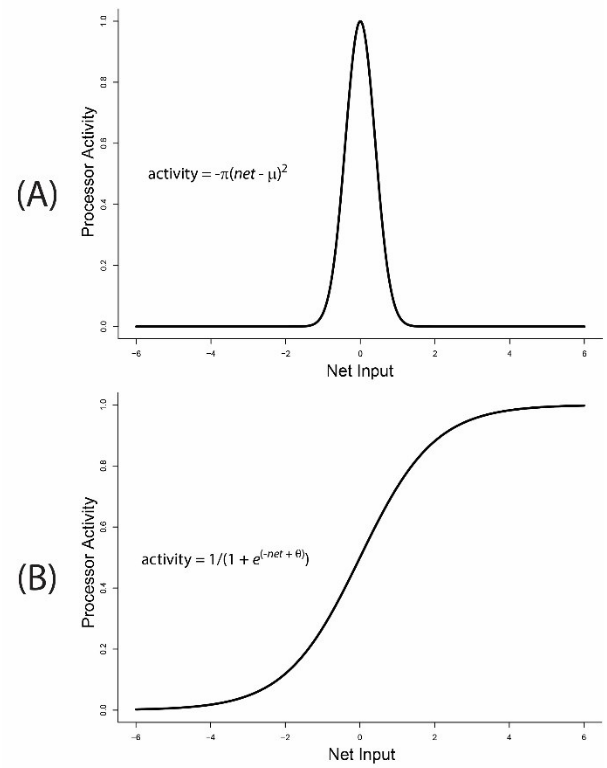


Figure 3-4. Two examples of different activation functions used by processors in multilayer perceptrons. (A) The bell-shaped activation function of a value unit, defined by the Gaussian equation. (B) the sigmoid-shaped activation function of an integration device, defined by the logistic equation.

An *integration device* uses a different activation function (Ballard, 1986). An integration device is not tuned to a narrow net input range, like a value unit, but instead responds monotonically: increased net input causes increased activity. Ballard describes motor neurons as example integration devices. While the relationship between integration device activity and net input is monotonic, it is not linear. Typically, integration devices ‘squash’ net input into a range between 0 and 1 by using a sigmoid-shaped activation function. One popular activation function for an integration device is the logistic equation (Rumelhart et al., 1986). The logistic function is illustrated in Figure 3-4B, along with its equation. In the equation, *net* represents net input, and θ is the bias of the equation. Bias is analogous to a neuron’s threshold. When net input equals θ, the equation generates activity of 0.5. In Figure 3-4B θ is equal to 0.

Our research interprets ANN structure. Value unit networks offer multiple advantages for this purpose. Value unit networks usually require fewer hidden units to solve classification problems (Dawson, 1990, 2004); network interpretation is also aided by some emergent properties not exhibited by integration devices (Berkeley, Dawson, Medler, Schopflocher, & Hornsby, 1995). For these reasons, our previous work on interpreting musical ANNs used value unit networks (Dawson, 2018). However, researchers who use ANNs to study musical cognition rarely use value units; integration device networks are the norm (Griffith & Todd, 1999; Todd & Loy, 1991). To connect our methodology to the broader musical ANN literature, we must also interpret integration device networks.

In particular, we need to determine whether integration device networks also represent phase space projections. If phase space projections are only discovered by value units, then such a representation must be due to the Gaussian activation function. However, if integration devices also discover phase space projections, then these projections are likely not processor dependent, but instead reflect important stimulus properties.

Furthermore, while our previous research identified phase space projections in value unit networks, we did not investigate how networks use such projections to make musical judgments. The monotonic nature of an integration device's activation function offers advantages to such an investigation, provided integration device networks also represent phase space projections. A monotonic activation function like the logistic means hidden unit activity can be interpreted as measuring similarity between hidden unit and stimulus structure. If hidden units represent phase space projections, it stands to reason the similarity being measured is between the Fourier structures represented by hidden unit weights and similar stimulus properties.

To foreshadow our results, we train networks (Figure 3-1) to classify musical triads, using integration devices as hidden and output processors. After training, we examine connection weights, and discover integration devices also represent phase space projections. We then use the logistic function's monotonicity to describe hidden unit activity as 'sympathetic vibration': a stimulus with Fourier structure more similar to the Fourier structure represented by a hidden unit's weights produces higher hidden unit activity. Finally, we determine how output units use 'sympathetic vibration' to classify musical triads.

Method

Task

We know, from previous research, that value unit networks discover phase space projections when learning to classify musical stimuli. We wish to determine if the same is true for integration device networks by training integration device networks on a problem previously studied using value unit networks: classifying triads. In the triad classification task, we present a network a triad – a stimulus comprised of three different pitch-classes. The network learns to classify the stimulus as belonging to one of four different triad types: major, minor, diminished, or augmented.

Network Architecture

We train multilayer perceptrons to classify triads (Figure 3-1). Each network has twelve input units to represent a stimulus; each input unit represents the presence of a pitch-class. We present a stimulus to the network by turning on the three input units representing the three pitch-classes included in a triad; all other input units are turned off. Each network uses eight hidden units to detect higher-order stimulus properties to permit the network to classify each stimulus. Each hidden unit is an integration device. Each network uses four output units to represent triad type; again, all output units are integration devices.

We train a network to generate the correct response to each triad in the training set. When a triad is presented, the network should turn on the output unit representing the triad's type, and should also turn off the other three output units. This is illustrated in Figure 3-1. The grey shading indicates the input units representing the pitch-classes C, E and G have been presented (i.e., the C major triad). It also indicates the network correctly activates the output unit representing a major triad, and correctly turns off the other three output units.

We chose the Figure 3-1 architecture after conducting numerous pilot simulations, following our standard methodology (Dawson, 2018). Pilot simulations determine various parameter values (e.g., learning rate) as well as the minimum number of hidden units required to reliably train networks to classify triads. Our pilot simulations indicated the simplest integration device network required eight hidden units to learn the triad classification task.

Training Set

The training set consists of 48 different triads: 12 different major triads, 12 different minor triads, 12 different diminished triads, and 12 different augmented triads. We create these stimuli by taking each of the 12 possible pitch-classes in Western music as a tonic for each of the four different triad types.

Table 3-1 shows the 48 stimuli. The numbers in the pitch-class columns of Table 3-1 indicate whether an input unit turned on (activity = 1) or off (activity = 0) when the triad is presented to a network. Also, we use enharmonic notation to represent the tonic and other components of each triad, as is standard practice in musical set theory.

Table 3-1

Table 3-1. The representation of each triad in the training set. Each pitch-class representation column is associated with an input unit of the Figure 3-1 network. In each row provides input unit activity. 1 indicates an input unit is 'on' and 0 indicates an input unit is 'off'.

		Pitch-Class Representation											
Type	**Tonic**	**C**	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**
Augmented	C	1	0	0	0	1	0	0	0	1	0	0	0
	E	1	0	0	0	1	0	0	0	1	0	0	0
	G#	1	0	0	0	1	0	0	0	1	0	0	0
	B	0	0	0	1	0	0	0	1	0	0	0	1

	D#	0	0	0	1	0	0	0	1	0	0	0	1
	G	0	0	0	1	0	0	0	1	0	0	0	1
	A#	0	0	1	0	0	0	1	0	0	0	1	0
	D	0	0	1	0	0	0	1	0	0	0	1	0
	F#	0	0	1	0	0	0	1	0	0	0	1	0
	A	0	1	0	0	0	1	0	0	0	1	0	0
	C#	0	1	0	0	0	1	0	0	0	1	0	0
	F	0	1	0	0	0	1	0	0	0	1	0	0
Diminished	A	1	0	0	1	0	0	0	0	0	1	0	0
	C	1	0	0	1	0	0	1	0	0	0	0	0
	D#	0	0	0	1	0	0	1	0	0	1	0	0
	F#	1	0	0	0	0	0	1	0	0	1	0	0
	A#	0	1	0	0	1	0	0	0	0	0	1	0
	C#	0	1	0	0	1	0	0	1	0	0	0	0
	E	0	0	0	0	1	0	0	1	0	0	1	0
	G	0	1	0	0	0	0	0	1	0	0	1	0
	B	0	0	1	0	0	1	0	0	0	0	0	1
	D	0	0	1	0	0	1	0	0	1	0	0	0
	F	0	0	0	0	0	1	0	0	1	0	0	1
	G#	0	0	1	0	0	0	0	0	1	0	0	1
Major	A#	0	0	1	0	0	1	0	0	0	0	1	0
	C#	0	1	0	0	0	1	0	0	1	0	0	0
	E	0	0	0	0	1	0	0	0	1	0	0	1
	G	0	0	1	0	0	0	0	1	0	0	0	1
	B	0	0	0	1	0	0	1	0	0	0	0	1
	D	0	0	1	0	0	0	1	0	0	1	0	0
	F	1	0	0	0	0	1	0	0	0	1	0	0
	G#	1	0	0	1	0	0	0	0	1	0	0	0
	A	0	1	0	0	1	0	0	0	0	1	0	0
	C	1	0	0	0	1	0	0	1	0	0	0	0
	D#	0	0	0	1	0	0	0	1	0	0	1	0
	F#	0	1	0	0	0	0	1	0	0	0	1	0
Minor	A#	0	1	0	0	0	1	0	0	0	0	1	0
	C#	0	1	0	0	1	0	0	0	1	0	0	0
	E	0	0	0	0	1	0	0	1	0	0	0	1
	G	0	0	1	0	0	0	0	1	0	0	1	0
	B	0	0	1	0	0	0	1	0	0	0	0	1
	D	0	0	1	0	0	1	0	0	0	1	0	0
	F	1	0	0	0	0	1	0	0	1	0	0	0
	G#	0	0	0	1	0	0	0	0	1	0	0	1
	A	1	0	0	0	1	0	0	0	0	1	0	0
	C	1	0	0	1	0	0	0	1	0	0	0	0
	D#	0	0	0	1	0	0	1	0	0	0	1	0
	F#	0	1	0	0	0	0	1	0	0	1	0	0

Table 3-1 reveals the relationship between our stimuli and musical set theory. The numbers in Table 3-1 indicate whether input units are turned on or off. However, each row can also be interpreted as a triad's pc-set. Later we use the pc-set for each stimulus to determine its Fourier structure, and link this structure to like structure represented by hidden units.

Training

We use the generalized delta rule to train networks to solve the triad classification problem (Rumelhart et al., 1986). We initialize all connection weights in a network to a random value between -0.1 and 0.1 before training began. The bias of each processor – the value of θ in the logistic equation – is held at 0 throughout training. We use a 0.1 learning rate to train networks. Our pilot simulations indicated the settings we adopted would train our networks to classify triads.

Training proceeds as follows: we present a stimulus, causing a network to respond. We use the generalized delta rule to modify connection weights based on response error (Rumelhart et al., 1986). We then repeat this procedure for the next stimulus. We present stimuli in an epoch-by-epoch fashion, as is our standard practice (Dawson, 2004, 2005). One epoch involves training a network once on each stimulus in the training set. We randomize stimulus order before each epoch. We train a network until it converges upon a solution to the triad classification problem. We define a converged network as one generating a 'hit' for each output unit for each stimulus in the training set. We define a 'hit' as output activity 0.9 or higher when the desired response is 1, or as output activity 0.1 or lower when the desired response is 0.

We train 20 different networks; because connection weights are randomized prior to training, each network is a different 'subject' in our study. At issue is whether each network discovers phase space projections. Each network solves the triad classification problem. On average, convergence occurs after 16,5302.2 epochs. The slowest network required 39,509 epochs; the fastest network required 4,888 epochs.

Results

We present our results as follows: First, we report all trained networks represent phase space projections in hidden unit connection weights. Second, we describe hidden unit activity as 'sympathetic vibration' between the Fourier structures of the unit and a stimulus. We then interpret an example network to determine how sets of 'sympathetic vibrations' are used to classify triads.

Discovering Phase Space Projections

Do our integration device networks represent phase space projections in order to classify triads? To answer this question, we fit phase space projections from each phase space to each hidden unit's connection weights, a method used in our previous research (Dawson et al., 2020). For each phase space, we begin with a diameter pointed to 0 o'clock. We determine the phase space projections to this diameter, and correlate them with a hidden unit's weights. We then rotate the diameter clockwise by one degree, and repeat the process. We use this procedure to find the diameter providing the best fit (i.e., the highest correlation) with the weights. We repeat this process to find the best fit between the projections of the next phase space to the same connection weights. In other words, we find the best fitting phase space projections from each of the six phases spaces to one hidden unit's connections. We then repeat this process for the next hidden unit.

Table 3-2 summarizes our results for all hidden units from all networks. We find a very high correlation between each hidden unit's connection weights and phase space projections from one phase space. Across all 160 hidden units in our networks, correlations involving the best-fitting phase space range from a maximum of 1.00 to a minimum of 0.49. The average correlation involving the phase space projections from the next best fitting phase space is much lower, ranging from a maximum of 0.64 to a minimum of 0.01. The Table 3-2 results are quite similar to previous results obtained for value unit networks (e.g., Dawson et al., 2020, Figure 3-4). Clearly, integration device networks discover phase space projections.

Table 3-2

The mean values of the best fits between hidden unit weights and phase space projections for six phase spaces, with standard deviations provided in parentheses. Fit is the correlation between phase space projections and connection weights.

Best Fit	**2nd Best Fit**	**3rd Best Fit**	**4th Best Fit**	**5th Best Fit**	**Worst Fit**
0.83 (0.13)	0.35 (0.17)	0.25 (0.13)	0.16 (0.11)	0.11 (0.08)	0.07 (0.07)

We also perform the DFT on each set of connection weights to be compared to our fitted phase spaces. A DFT of connection weights provides information not provided by our fitting methodology (i.e., real and imaginary components) which we use later. Table 3 summarizes our results; it is analogous to Table 3-2, but replaces the notion of 'best fitting phase space' with 'phase space of highest magnitude'. The magnitudes in Table 3-3 are strikingly similar to fit values in Table 3-2: just as the average best fit in Table 3-2 is roughly twice the value of the next best fit, the average highest magnitude in Table 3-3 is roughly twice the value of the next highest magnitude.

Table 3-3

The mean magnitudes of the Fourier components for six phase spaces, with standard deviations provided in parentheses.

Highest Magnitude	**2nd Highest Magnitude**	**3rd Highest Magnitude**	**4th Highest Magnitude**	**5th Highest Magnitude**	**Lowest Magnitude**
55.87 (11.05)	23.53 (13.00)	16.36 (9.96)	10.77 (8.23)	7.41 (6.74)	4.88 (5.62)

The similarities between Tables 2 and 3 suggest our fitting methodology produces similar phase space projections to the components delivered by the DFT of connection weights. We can test this hypothesis by comparing the differently obtained phase space projections. For all 160 hidden units, we correlate the phase space projections obtained by our fitting procedure with the corresponding DFT phase space projections. The mean correlation is nearly perfect ($r = 0.97$, SD = 0.04). Similar results are obtained when we correlate the six fit values from our fitting procedure to the corresponding magnitudes delivered by the DFT (mean $r = 0.98$, SD = 0.03). The excellent fits between phase space projections and hidden unit weights (Table 3-2), as well as the strong relationship between these fitted projections and the fitted projections obtained from a DFT of the connection weights, clearly indicate the hidden units in our integration device networks represent Fourier properties.

Discovering phase space projections in integration device networks permits us to explore issues neglected in our previous research. How do integration device networks use phase space projections to classify triads? We answer this question in two stages. First, we describe hidden unit activity as 'sympathetic vibration' due to similarity between the Fourier structure of a hidden unit and a presented triad. Second, we explore how the output units use sets of 'sympathetic vibrations' to determine triad type.

Hidden Unit Activity As 'Sympathetic Vibration'

External vibrations can cause a passive body to vibrate. Such sympathetic vibration arises from harmonic similarity between the body and the stimulus. Here we propose a similar, metaphorical, account of our networks. A hidden unit represents phase space projections with its connection weights. We propose hidden unit activity reflects the similarity between the unit's phase space structure and the stimulus's phase space structure. As this similarity increases, so too does hidden unit activity, due to the monotonic nature of the logistic equation. Metaphorically, activity represents a hidden unit's 'sympathetic vibration' to a stimulus.

A hidden unit's net input is the sum of the weighted signals received from input units. Mathematically, net input is the inner product between a vector of weights and a vector of input unit activities. In linear algebra, the cosine of the angle between two vectors is their inner product divided by each vector's length. We hypothesize a hidden unit's 'sympathetic vibration' is related to such similarity between hidden unit and stimulus phases. As this angle becomes smaller, phases of the unit and the stimulus point in more similar directions, causing higher inner products. In turn, the logistic equation's monotonicity means higher inner products produce higher hidden unit activities.

Figure 3-5 uses the F major triad to illustrate our hypothesis. In this figure, each clock face illustrates the highest magnitude phase space for one hidden unit in a network, as determined from the DFT of the unit's connection weights. The dashed arrow represents the phase represented by the connection weights. The solid arrow represents the phase for the F major triad in the same space. We present two values beneath each clock face. One is the net input: the inner product between connection weights and input unit activities. The other is the cosine of the angle between the solid and dashed arrows (the angle indicated by θ in each clock). When the two arrows point in similar directions, both the net input and the cosine are high and positive. When the two arrows point in quite different directions, both the net input and the cosine are high and negative. For this example, the correlation between the eight net inputs and the corresponding cosines in Figure 3-5 is 0.98.

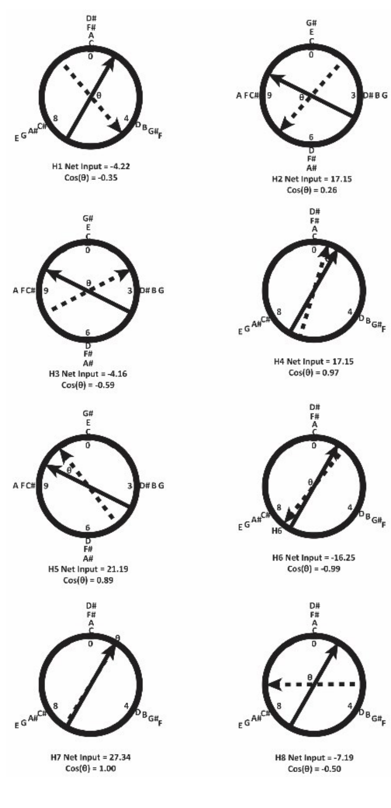


Figure 3-5. An illustration of the 'sympathetic vibration' hypothesis. Each circle represents the best fitting phase space for one of a network's eight hidden units, and the dashed arrow represents the unit's preferred phase for the space. The solid arrow represents the phase of the F major triad in the same space. The angle between the two arrows is represented by θ. The net input computed for each hidden unit, and the cosine of θ, are presented beneath each circle.

However, Figure 3-5 idealizes the relationship between a hidden unit and a stimulus. It assumes the relationship only involves one phase space. However, defining ‘sympathetic vibration’ with only one phase space causes problems. For instance, when triads have zero magnitudes in a phase space, such as augmented triads in Ph4, the comparison depicted in Figure 3-5 is impossible, because the solid arrow has zero length. Furthermore, while hidden unit weights typically have a high magnitude for one phase space, they also have nonzero magnitudes for other phase spaces (see Table 3-3). Therefore, hidden unit weights represent more complex harmonic structure involving multiple phase spaces. A better account of ‘sympathetic vibration’ must consider all phase spaces when measuring the similarity between hidden units and stimuli.

To make this more complex comparison, we first perform Yust’s (2016) DFT on each triad. We then create a measure of similarity between hidden units and stimuli which considers all six phase spaces from the DFTs. As noted earlier, the DFT delivers the real and imaginary components of a complex number for each phase space. These two values provide a phase space’s phase and magnitude. We define the similarity between two DFTs as the correlation between their respective real and imaginary components. That is, we correlate the real and imaginary components (twelve numbers, two for each phase space) of a hidden unit’s DFT to the corresponding components of a triad’s DFT.

We compute the similarity between each triad and each hidden unit for each network, producing 7,680 comparisons. According to our ‘sympathetic vibration’ hypothesis, the higher the similarity, the higher the unit’s activity. To evaluate this hypothesis across all networks, we correlate our 7,680 similarity measures with the corresponding activities produced in a hidden unit by a stimulus. The resulting correlation is 0.69. This supports our ‘sympathetic vibration’ hypothesis. As the similarity between phase space structures increases, so too does hidden unit activity.

It is remarkable the correlation between DFT similarity and hidden unit activity is as high as 0.69. A network's basic operation computes inner products between weights and input activities. The inner product most strongly relates to an operation involving phase space projections (Figure 3-3) due to the strong relationship between projections and weights (Tables 3-2 and 3-3). Our use of real and imaginary components to measure similarity is far removed from operations involving phase space projections. Hidden unit activity is also removed from an inner product involving phase space projections, because activity is a logistic transformation of the inner product. Nevertheless, a strong relationship exists between DFT similarity and hidden unit activity.

We have now established two important results. First, integration device networks represent phase space projections. Second, hidden unit activities reflect the similarity between Fourier components of hidden units and stimuli. We now examine how hidden unit activities are used to classify triads by exploring how a network's output units convert hidden unit signals into correct responses.

Coarse Coding In An Example Network

Artificial neural networks can provide new ideas about representation. *Coarse coding*, detailed below, exemplifies a network-inspired representation. Below, we illustrate coarse coding in one example network to explain how the network converts 'sympathetic vibrations' into triad classifications.

Our example network solved the triad classification problem after 5116 training epochs. Table 3-4 provides the network's connection weights from input units to hidden units.

Table 3-4

The connection weights from each input unit to each hidden unit for the network being analyzed.

	Input Unit											
Hidden Unit	**C**	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**
H1	-9.25	-1.25	13.48	-9.41	-1.19	13.80	-9.02	-1.13	14.15	-8.77	-1.10	13.88
H2	-0.36	8.05	11.03	-2.50	-8.02	11.05	6.39	0.99	-1.88	6.46	8.92	-5.48
H3	6.67	-3.53	0.20	7.96	6.47	-5.55	2.36	6.84	7.90	-5.28	-1.78	9.91
H4	11.76	-5.76	-0.87	11.37	-6.44	-1.14	10.04	-6.74	-1.25	10.99	-5.70	-1.18

H5	6.76	6.68	-2.73	1.09	5.71	8.64	-2.61	-1.26	6.12	5.80	-2.42	1.40
H6	-7.57	12.81	-0.67	-7.34	12.28	-0.61	-8.31	11.60	-0.65	-8.07	13.06	-0.52
H7	11.38	-3.40	4.46	12.18	-4.53	4.18	12.13	-2.87	4.70	11.77	-4.08	4.45
H8	-0.32	5.87	-6.58	-0.32	5.85	-6.55	-0.33	5.68	-6.63	-0.32	5.76	-6.25

The connection weights in Table 3-4 determine hidden unit activity. When a stimulus is presented, each hidden unit calculates net input by computing the inner product of its connection weights (represented as a vector) with the input unit activities (represented as another vector). The hidden unit then transforms net input into activity using the logistic equation. Hidden unit activity becomes a signal to send onwards to output units.

To illustrate, imagine presenting the C augmented triad to the network. For this stimulus, Hidden Unit 1's net input is the inner product between the first row of Table 3-1 and the first row of Table 3-4. Algebraically, this inner product is $((-9.25 \cdot 1) + (-1.25 \cdot 0) + (13.48 \cdot 0) + (-9.41 \cdot 0) + (-1.19 \cdot 1) + (13.80 \cdot 0) + (-9.02 \cdot 0) + (-1.13 \cdot 0) + (14.15 \cdot 1) + (-8.77 \cdot 0) + (-1.10 \cdot 0) + (13.88 \cdot 0)) = ((-9.25 \cdot 1) + (-1.19 \cdot 1) + (14.15 \cdot 1)) = 3.71$. Hidden Unit 1's logistic equation transforms the net input, 3.71, into activity equal to 0.978.

One could mechanically describe network behavior with equations for net input and for unit activity. In contrast, we aim to provide a more meaningful network description. What musical interpretation might we give to a hidden unit's net input or activity? How does the network use this musical interpretation to classify triads?

The connection weights in Table 3-4 make such a musical interpretation possible because they represent Fourier phase space projections. Table 3-5 provides the phase space from the DFT of each connection weight set (Table 3-4). Table 3-5 reveals five hidden units have the strongest relationship – the highest magnitude -- to phase space projections from Ph4 (H1, H4, H6, H7, H8). The other three have the strongest relationship to with phase space projections from Ph3 (H2, H3, H5). Though different hidden units most strongly related to the same phase space, this relationship involves different phases (represented as clock settings), as shown by the phase values in Table 3-5. The reason that different hidden units are most sensitive to the same phase space, but at different phases, will be detailed below.

Table 3-5

The Fourier phase space analysis for the network being interpreted. Each fit value is the correlation between a hidden unit's connection weights and the best fitting phase for a phase space. The phase value is the clock setting for the best fitting phase. Cells highlighted in grey indicate the best fitting phase space for each hidden unit.

Hidden Unit		Ph1	Ph2	Ph3	Ph4	Ph5	Ph6
H1	Magnitude	1.50	0.17	0.57	80.70	0.51	0.08
	Phase	3.72	4.47	10.32	4.67	5.65	3.80
H2	Magnitude	0.66	11.06	48.97	4.75	17.56	2.50
	Phase	3.05	9.77	7.38	3.98	11.30	8.52
H3	Magnitude	1.10	4.30	44.01	4.14	8.08	11.47
	Phase	10.42	11.07	2.08	2.30	5.97	11.47
H4	Magnitude	3.14	0.85	0.87	61.26	1.89	0.00
	Phase	11.65	5.68	10.45	0.55	10.70	11.45
H5	Magnitude	2.69	4.24	33.00	4.08	3.51	11.51
	Phase	10.12	2.23	10.77	3.02	9.15	7.55
H6	Magnitude	2.34	1.28	0.38	71.15	1.99	0.25
	Phase	11.50	4.37	8.87	7.32	9.00	1.15
H7	Magnitude	1.06	2.52	1.53	54.01	0.75	2.26
	Phase	4.87	9.35	4.27	1.05	7.83	3.17
H8	Magnitude	0.42	0.45	0.11	42.58	0.46	0.35
	Phase	11.63	2.23	2.28	9.00	5.25	8.27

In the example network, each hidden unit generates a 'sympathetic vibration' to a stimulus. 'Sympathetic vibration' – realized as hidden unit activity -- represents the similarity between the Fourier components represented by a hidden unit and like components for describing the stimulus. Importantly, such 'sympathetic vibration' predicts an individual hidden unit will not accurately discriminate different kinds of triads from one another. Two reasons account for the poor discrimination of individual hidden units.

First, triads belonging to the same type do *not* always share the same phase in a phase space, as illustrated in Figure 3-6. Figure 3-6 represents the phases of the 48 triads in the two highest magnitude phase spaces for our example network, Ph3 and Ph4. It reveals four different subtypes of augmented triads; each subtype has a different Ph3 phase. Similarly, major triads have three different subtypes; each subtype has a different Ph4 phase. Under 'sympathetic vibration, two triads belonging to the *same type*, but belonging to *different subtypes* must produce *different* activities in the *same* hidden unit, because they must have different similarities to the hidden unit's phase structure.

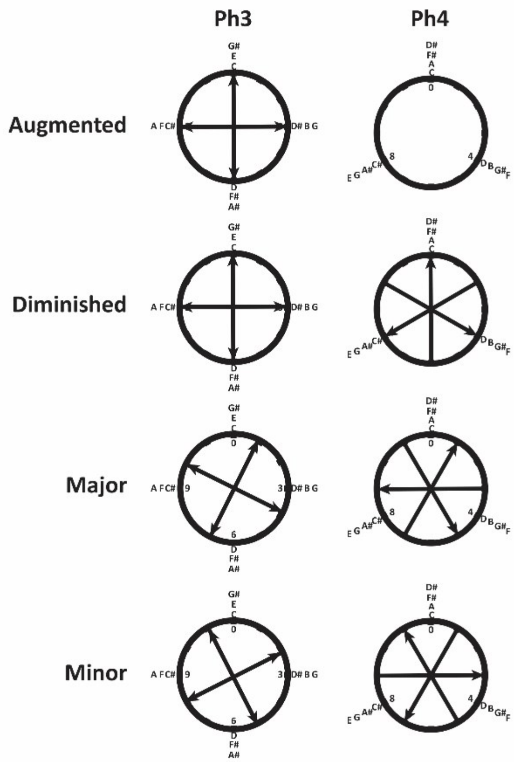


Figure 3-6. The phases for different types of triads in the two phase spaces (Ph3 and Ph4) to which hidden units in the example network are most sensitive. Phases are indicated by the directions of the solid arrows. When more than one arrow is present, different subsets of triads have different phases in the same space. The triads belonging to different subsets are detailed later in Table 3-8

Second, Figure 3-6 also reveals *different types* of triads can produce *similar activity* in the same hidden unit because different triad types may have similar phase space structures. For example, different augmented triad subtypes and diminished triad subtypes have identical Ph3 phase components. As a result, some augmented triads produce similar activity in one hidden unit to the activity produced by some diminished triads.

In short, there are two reasons to expect an *individual* hidden unit's activity cannot be used to discriminate triad type. We confirm this in Table 3-6, which summarizes hidden unit activities produced by different triads in the example network. We classify hidden unit activity as high if it is greater than 0.7, as low if it is less than 0.3, and as medium otherwise. We count the times each triad type produces such responses in each hidden unit.

Table 3-6

The number of times each type of triad produces high, medium, or low activity in each hidden unit in the example network.

Activity Type	**Triad Type**	**Hidden Unit**							
		H1	**H2**	**H3**	**H4**	**H5**	**H6**	**H7**	**H8**
High Activity	**Aug**	12	6	6	12	9	12	12	0
	Dim	4	12	12	4	12	4	8	4
	Maj	4	8	6	4	8	8	12	4
	Min	8	7	9	8	9	4	8	8
Medium Activity	**Aug**	0	0	3	0	0	0	0	0
	Dim	0	0	0	0	0	0	0	0
	Maj	0	0	0	0	2	0	0	0
	Min	0	0	0	0	0	0	0	0
Low Activity	**Aug**	0	6	3	0	3	0	0	12
	Dim	8	0	0	8	0	8	4	8
	Maj	8	4	6	8	2	4	0	8
	Min	4	5	3	4	3	8	4	4

Table 3-6 demonstrates each hidden unit's activity is, by itself, a poor predictor of triad type. For every hidden unit, high activity is produced by at least three different kinds of triads. Furthermore, different instances of the same triad type produce markedly different activities (i.e., high vs low) in the same hidden unit. For example, six major triads produce high activity in H3, but the other six major triads produce low activity in the same hidden unit. How does the network use these poor detectors to correctly respond? It does so by employing coarse coding.

Coarse coding is a distributed representation (Hinton, 1986; Van Gelder, 1991). A distributed representation uses activity from several network components to represent a single concept.

Artificial neural networks use distributed representations when individual hidden unit activities do not accurately distinguish one concept from another (Churchland & Sejnowski, 1992; Hinton, McClelland, & Rumelhart, 1986). Even when individual hidden units are poor discriminators, their combined activities can accurately represent one concept when each hidden unit represents information from different perspectives.

We illustrate the network's coarse coding with a Venn diagram (Figure 3-7). Each oval in this figure represents a hidden unit's state of activity (along the lines of Table 3-6). For example, in Figure 3-7 all triads in one oval produce high activity in H1, all triads in another reflects low activity in H2, and so on. Within each oval, we place the type (and subtype) of triads producing the oval's activity state in the hidden unit. (Specific details about triad subtypes are provided in Table 3-8.) For instance, high activity in H1 is produced by Ph4 subtype 1 diminished triads, by Ph3 subtypes 1, 2, and 3 augmented triads, by Ph4 subtype 1 major triads, and by Ph4 subtypes 1 and 3 minor triads.

The ovals intersect one another in the middle of Figure 3-7. The intersection contains any triads found every oval. In Figure 3-7, we only find Ph4 subtype 1 in this intersection. Thus, combining the activity of each hidden unit uniquely identifies triad type, in this case particular minor triads. We can create analogous accounts for other patterns of hidden unit activities. In each account, the intersection reveals only one triad type.

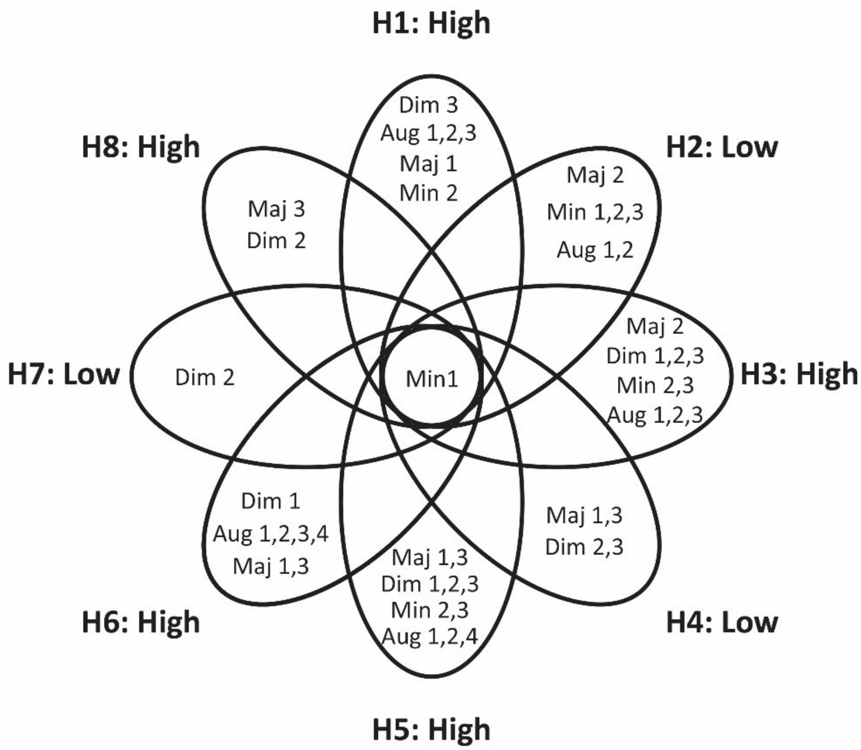


Figure 3-7. Coarse coding in the example network. Each oval contains all triads that produce a Table 3-6 value of activity in a hidden unit. Triads are indicated in terms of type and subtype. For instance, 'Aug 1' indicates augmented triads of subtype 1 in Ph3, and 'Min 1' indicates minor triads of subtype 1 in Ph4, as detailed in Table 3-8. The intersection of the ovals indicates that the only triads which produce these eight values of hidden unit activities are minor triads of subtype 1 in Ph4.

Figure 3-7 illustrates the essence of this network's coarse coding. Each hidden unit is a poor triad discriminator, because different triad types cause similar activity in the unit, as shown by the diversity of triads belonging to each oval. However, combining all activities (i.e., intersecting the sets) reveals one triad type.

Importantly, identifying the correct triad type using this method requires each hidden unit to have a different perspective on stimuli. To be more particular, for the intersection in Figure 3-7 to pick out one triad type, there must be different stimuli in different ovals.

In our network, each hidden unit's different perspective is reflected in its phase structure. Why are several hidden units in the network highly sensitive to the same phase space, but at a different phase (Table 3-5)? Such phase differences provide the different perspectives required by the coarse coding illustrated in Figure 3-7. Different phases, for the same phase space, cause different activities to be produced in different hidden units by the same stimulus.

Figure 3-7 represents a network's response as arising from intersecting subsets. The network uses a different procedure to accomplish the same result. In this procedure, each output unit computes its own net input using the inner product between connection weights and hidden unit activities. Net input is then converted into output activity using the logistic activation function. In this network, the connection weight values convert patterns of hidden unit activity into correct output unit responses. In other words, when the network converts hidden unit activities into output unit activities, it does so via connection weights that deliver a result analogous to the set intersection illustrated in Figure 3-7. Let us briefly explore how the net inputs for the output units are functionally equivalent to intersecting sets.

Table 3-7 presents the connection weights from each hidden unit to each output unit in our example network. When hidden units activate, they send a signal to each output unit, which then determine their net input. Net input is the inner product between an output unit's weights (a unit's row in Table 3-7) and the vector representing the eight hidden unit activities. Net input is then converted to output unit activity using the logistic equation. With the connection weights in Table 3-7, presenting a triad will produce high output activity in one output unit, and low output activity in the other three.

Table 3-7

Connection weights from hidden units to output units.

	Connection Weights From Hidden Units To Each Output Unit							
	H1	**H2**	**H3**	**H4**	**H5**	**H6**	**H7**	**H8**
Augmented	1.92	-6.67	-6.63	7.22	-5.22	7.31	-0.50	-1.51
Major	-9.44	-3.13	-3.36	-5.99	-1.86	4.99	13.54	-7.18
Minor	7.79	-1.12	-1.04	4.93	-1.16	-9.51	-7.01	8.09
Diminished	-6.14	6.09	6.02	-8.01	4.06	-7.23	-4.73	-1.96

A positive connection weight makes a signal to an output unit excitatory, and increases net input. In contrast, a negative weight makes a signal to an output unit inhibitory, and decreases net input. With these two observations in mind, one can inspect Table 3-7 to determine an output unit's 'ideal stimulus' (Dawson, Kremer, & Gannon, 1994). The ideal stimulus produces the maximum net input for an output unit. To do so, the ideal stimulus requires the smallest activity in hidden units associated with negative connection weights, and the highest activities in hidden units associated with positive connection weights. For example, according to Table 3-7 the ideal major triad produces high activity in H6 and H7 and produces low activity in the other six hidden units. Similarly, the ideal minor triad produces high activity in H1, H4 and H8 and produces low activity in the other five hidden units.

However, not every stimulus produces the ideal pattern of activity in the eight hidden units. We see this in Table 3-8, which presents the hidden unit activities produced by each triad subtype. Consider major triads belonging to Ph4 subtype 1. They produce undesirable, high activity in H1, while other major triad subtypes do not. Similarly, major triads belonging to Ph4 subtype 1 fail to produce desirable, high activity in H6, while the other two subtypes do. Similar observations can be made about the other triads.

Table 3-8

The activity produced in each hidden unit by each triad subtype, along with the net input these produce in each output unit. Net inputs that cause output units to turn on are highlighted in grey.

				Activity Produced In Each Hidden Unit								Net Input To Each Output Unit			
Triad Type	**Ph3 Subtype**	**Ph4 Subtype**	**Tonics**	**H1**	**H2**	**H3**	**H4**	**H5**	**H6**	**H7**	**H8**	**Aug**	**Maj**	**Min**	**Dim**
Aug	1		C, E, G#	0.98	0.00	1.00	0.98	1.00	0.98	1.00	0.25	3.44	-3.67	-4.08	-16.10
Aug	2		B, D#, G	0.97	0.00	1.00	0.97	0.77	0.98	1.00	0.29	4.39	-3.39	-3.58	-16.88
Aug	3		A#, D, F#	0.97	1.00	0.69	0.97	0.00	0.98	1.00	0.24	3.96	-3.67	-3.92	-15.77
Aug	4		A, C#, F	0.98	1.00	0.00	0.98	1.00	0.98	1.00	0.27	3.38	-3.60	-4.00	-16.09
Dim	1	1	A, C, D#, F#	0.00	0.97	1.00	1.00	1.00	0.00	1.00	0.28	-12.03	-2.70	-3.13	2.74
Dim	1	2	A#, C#, E, G	0.03	1.00	0.76	0.00	1.00	1.00	0.00	1.00	-11.07	-10.00	-4.26	5.38
Dim	1	3	B, D, F, G#	1.00	1.00	0.99	0.04	1.00	0.14	1.00	0.00	-15.69	-3.75	-3.68	3.90
Maj	3	1	A#, C#, E, G	1.00	0.00	1.00	0.00	1.00	1.00	0.99	0.00	-3.10	3.73	-10.84	-7.96

Maj	2	2	B, D, F, G#	0.01	0.01	1.00	1.00	1.00	0.00	1.00	0.00	-5.17	2.19	-4.19	-2.66
Maj	4	3	A, C, D#, F#	0.00	0.00	1.00	0.20	1.00	1.00	0.98	1.00	-5.13	4.71	-9.53	-5.31
Min	1	1	A#, C#, E, G	1.00	1.00	0.00	0.00	1.00	1.00	0.04	0.99	-4.17	-16.09	3.80	-5.33
Min	1	2	B, D, F, G#	1.00	1.00	1.00	1.00	0.02	0.00	1.00	0.00	-4.76	-8.42	3.53	-6.68
Min	1	3	A, C, D#, F#	0.00	0.13	1.00	1.00	1.00	0.03	1.00	1.00	-7.24	-5.05	3.32	-4.06

Each output unit has an ideal pattern of hidden unit activities, and produces a maximum response to this pattern. However, because triads will generally not produce the ideal pattern, an output unit's net input measures the similarity between the ideal pattern and hidden unit activities. With sufficiently high similarity, the output unit will turn on. The weights resulting from training (Tables 6 and 9) ensure a stimulus will always be most similar to the appropriate output unit's ideal pattern, and more dissimilar to the other three ideal patterns. As a result, a stimulus will produce activity in the eight hidden units that produces positive net input for one output unit, and negative net input for the other three (see Table 3-8). The activity produced by a stimulus in each hidden unit is a piece of evidence used to determine triad type. Single pieces of evidence are noisy (see our earlier discussion of Table 3-6). However, output unit responses depend on considering all the weighted evidence. Net input represents the accumulated, weighted evidence.

We provide details about weighing the evidence in Table 3-8. Each row represents a type and subtype of triad; triads belonging to the same type and subtype have identical effects on hidden units. Beside each row are the net inputs produced by hidden unit activities. Each net input is the inner product of the row's hidden unit activities with the appropriate set of connection weights from Table 3-8. For each row, net input is positive for the correct output unit, and is negative for the other three output units. When the logistic transforms each net input, the correct output unit will turn on while the other three will turn off.

Three general points to make about Table 3-8. First, the table illustrates that triad type emerges from a representation distributed over all eight hidden units. Output units turn on or off using the signals from all eight hidden units. No one hidden unit perfectly represents triad type.

Second, calculating net input is functionally equivalent to intersecting the Venn diagram components in Figure 3-7. Each oval in Figure 3-7 asserts different triads produce particular levels of activity (i.e., values from Table 3-8) in each hidden unit. Figure 3-7 illustrates correct triad type as the intersection of these subsets. Analogously, each oval represents a hidden unit activity to be modified by a connection weight. Four different sets of weights, one for each output unit, are available. The weighted activities' sum will identify the same triad type discovered at the intersection of the ovals.

Third, Table 3-8 reveals not all same-type triads are equal, at least from the perspective of our network. For instance, while every augmented triad activates the augmented output unit, some augmented triads produce higher net inputs than others. For this network, augmented triads with the tonics of B, D# or G produce the highest net input in comparison to other augmented triads, and therefore are better examples of augmented triads. The other three triad types behave similarly. For this network, some triads are better examples than others because a) triads of the same type have different phases in the same phase space (Figure 3-6) and b) net input reflects the similarities between triad Fourier structures and hidden unit Fourier structures. In other words, differences in phases between same type produce differences in 'sympathetic vibration' in the same hidden units.

Discussion

How might brain-like systems represent music? Our previous work discovered value unit networks represent Fourier phase space projections. Our current goal was establishing whether such representations also exist in networks whose units employ the logistic activation function.

We trained 20 integration device networks to classify triads, and discovered phase space projections in their hidden unit weights. Phase space representations are not unique to value unit networks.

A second goal was determining how ANNs use Fourier phase spaces to classify musical entities. We first determined hidden unit activities in musical integration device networks represent ‘sympathetic vibrations’: as similarity between phase spaces represented by a hidden unit and a stimulus increases, so too does net input and hidden unit activity. We then analyzed an example network to determine how it used ‘sympathetic vibrations’ to respond. We discovered coarse coding in which individual hidden units are poor triad classifiers, but correct classifications result when hidden unit activities are combined as weighted evidence. Triad classification involves comparing hidden unit activities to an ideal pattern, which occurs when output units compute net input. Coarse coding succeeds because even when different hidden units are, most sensitive to the same phase space, phase space projections represented different phases of the space. Hidden units most sensitive to the same phase space, but at different phases, generate different activities to the same stimulus providing coarse coding’s foundation.

Our results establish ANN phase space projections do not depend upon a particular activation function, the Gaussian. Discovering phase space projections in integration device networks demonstrates ANNs comprised of markedly different processors (Figure 3-4) use musical representations related to the DFT. Discovering similar representations in different network types indicates musical networks prefer to capture stimulus regularities describable as DFT properties.

Our results indicate musical Fourier representations have important roles beyond musical set theory. Historically, musical set theory has been criticized for being too abstract and too technical to account for musical experience (Schuijer, 2008). Lewin’s introduction of the DFT to musical set theory (Lewin, 1959) only sketched his mathematical reasoning, "anticipating an outraged reaction at the introduction of 'high-level' mathematics to a music journal" (Amiot, 2016, p. VI). Recent research makes clear the power of the DFT to capture important relationships between musical entities, indicating its importance to music theory (Amiot, 2016; Quinn, 2006, 2007; Yust, 2016, 2017a, 2019). Unsurprisingly, music theorists hope the importance of the DFT extends to theories of musical cognition. For example, Amiot (2016, p. 179) argues Fourier space "is closest to our perception of music […] since Fourier qualities seem to mirror exactly musical features processed by the human brain."

Amiot's (2016) intriguing claim that human brains process Fourier qualities of music requires empirical support. We believe discovering phase space projections in ANNs provides crucial evidence supporting his claim. Using ANNs to study cognition exploded in popularity because researchers believed networks provided biologically inspired or neuronally plausible models (McClelland & Rumelhart, 1986; Rumelhart & McClelland, 1986; Schneider, 1987). If the human brain processes Fourier qualities, then a biologically plausible representation of such qualities must exist. Finding phase space projections in ANNs raises the possibility that Fourier phase spaces can bridge the cognitive psychology of music with cognitive neuroscience (Tillmann, Bharucha, & Bigand, 2003).

Still, ANNs' ability to discover phase space projections is surprising. There is no clear mathematical relationship between network training and computing the DFT. Determining the DFT of a musical set requires correlating it with different frequencies of cosine and sine functions (Amiot, 2016). In contrast, small, iterative weight changes modify ANNs, reducing response error (Dawson & Schopflocher, 1992; Rumelhart et al., 1986). Remarkably, network training offers a unique technique for extracting Fourier properties from musical sets.

Discoveries made by music theorists guided our search for Fourier representations in ANNs. Had the importance of the DFT not already been established in music theory, we would probably not have sought Fourier phase spaces in our networks. However, we hope our networks might also inform music theory. For example, music theorists are interested in using Fourier components to create mathematical spaces which position similar musical entities closer to one another than dissimilar musical entities (Amiot, 2016; Quinn, 2006, 2007; Tymoczko, 2011; Yust, 2016). Our networks offer a new candidate for exploring musical spaces, because hidden unit activities produced by stimuli can be interpreted as locating a stimulus in a multidimensional space. In the networks described, a point represents each triad in an 8-dimensional space that reflects the triad's relationship to eight different phase spaces. Distance relationships between triads in this space can be determined, and can be compared to distance relationships in other musical spaces that have been proposed.

Critically, the potential implications emerging from discovering discrete Fourier phase spaces in musical ANNs are only possible when researchers make the effort to examine and interpret network structure (Dawson, 2018). Unfortunately, researchers who study ANNs rarely determine how networks represent solutions to problems (Dawson, 2009). Even though many researchers believe ANNs are important for capturing informal properties of music (Bharucha, 1999), it is important to realize networks can discover surprising, formal, musical regularities. By discovering Fourier phase spaces in our musical networks, we see rich links between formal music theory and musical cognition. However, to establish such links, one must examine the internal structure of trained networks.

References

Abbott, A. (2002). Neurobiology: Music, maestro, please! *Nature, 416*(6876), 12-14. doi: 10.1038/416012a.

Amiot, E. (2016). *Music Through Fourier Space: Discrete Fourier Transform in Music Theory*. Cham: Springer International Publishing : Imprint: Springer.

Ballard, D. (1986). Cortical structures and parallel processing: Structure and function. *The Behavioral And Brain Sciences, 9*, 67-120.

Berkeley, I.S.N., Dawson, M.R.W., Medler, D.A., Schopflocher, D.P., & Hornsby, L. (1995). Density plots of hidden value unit activations reveal interpretable bands. *Connection Science, 7*, 167-186.

Bharucha, J.J. (1999). Neural nets, temporal composites, and tonality. In D. Deutsch (Ed.), *The Psychology Of Music* (2nd ed., pp. 413-440). San Diego, CA: Academic Press.

Churchland, P.S., & Sejnowski, T.J. (1992). *The Computational Brain*. Cambridge, MA: MIT Press.

Dawson, M.R.W. (1990). Training networks of value units: Learning in PDP systems with nonmonotonicactivation functions. *Canadian Psychology, 31*(4), 391.

Dawson, M.R.W. (2004). *Minds And Machines: Connectionism And Psychological Modeling*. Malden, MA: Blackwell Pub.

Dawson, M.R.W. (2005). *Connectionism : A Hands-on Approach* (1st ed.). Oxford, UK ; Malden, MA: Blackwell Pub.

Dawson, M.R.W. (2009). Computation, cognition – and connectionism. In D. Dedrick & L. Trick (Eds.), *Cognition, Computation, and Pylyshyn* (pp. 175-199). Cambridge, MA: MIT Press.

Dawson, M.R.W. (2018). *Connectionist Representations of Tonal Music: Discovering Musical Patterns By Interpreting Artificial Neural Networks*. Edmonton, AB: Athabasca University Press.

Dawson, M.R.W., Berkeley, I.S.N., Medler, D.A., & Schopflocher, D.P. (1994). Density plots of hidden value unit activations reveal interpretable bands and microbands.

Dawson, M.R.W., Kremer, S., & Gannon, T. (1994). Identifying the trigger features for hidden units in a PDP model of the early visual pathway. In R. Elio (Ed.), *Tenth Canadian Conference On Artificial Intelligence* (pp. 115-119). San Francisco, CA: Morgan Kaufmann.

Dawson, M.R.W., Perez, A., & Sylvestre, S. (2020). Artificial neural networks solve musical problems with Fourier phase spaces. *Scientific Reports, 10*(1), 7151. doi: 10.1038/s41598-020-64229-4

Dawson, M.R.W., & Schopflocher, D.P. (1992). Modifying the generalized delta rule to train networks of nonmonotonic processors for pattern classification. *Connection Science, 4*, 19-31.

Deutsch, D. (1999). *The Psychology Of Music* (2nd ed.). San Diego: Academic Press.

Forte, A. (1973). *The Structure Of Atonal Music*. New Haven: Yale University Press.

Griffith, N., & Todd, P.M. (1999). *Musical Networks: Parallel Distributed Perception And Performace*. Cambridge, Mass.: MIT Press.

Hanson, H. (1960). *Harmonic materials of modern music; resources of the tempered scale*. New York,: Appleton-Century-Crofts.

Hinton, G.E. (1986). *Learning distributed representations of concepts.* Paper presented at the The 8th Annual Meeting of the Cognitive Science Society, Ann Arbor, MI.

Hinton, G.E., McClelland, J., & Rumelhart, D. (1986). Distributed representations. In D. Rumelhart & J. McClelland (Eds.), *Parallel Distributed Processing* (Vol. 1, pp. 77-109). Cambridge, MA: MIT Press.

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

Lewin, D. (1959). Re: Intervallic relations between two collections of notes. *Journal of Music Theory, 3*(2), 298-301.

Lewin, D. (2007). *Generalized Musical Intervals And Transformations*. New York: Oxford University Press.

McClelland, J.L., & Rumelhart, D.E. (1986). *Parallel Distributed Processing, V.2*. Cambridge, MA: MIT Press.

McCloskey, M. (1991). Networks and theories: The place of connectionism in cognitive science. *Psychological Science, 2*, 387-395.

Peretz, I., & Zatorre, R.J. (2003). *The Cognitive Neuroscience Of Music*. Oxford ; New York: Oxford University Press.

Peretz, I., & Zatorre, R.J. (2005). Brain organization for music processing. *Annual Review of Psychology, 56*, 1-26.

Quinn, I. (2006). General equal-tempered harmony: Introduction and Part 1. *Perspectives of New Music, 44*, 114-158.

Quinn, I. (2007). General equal-tempered harmony: Parts 2 and 3 *Perspectives of New Music, 45*, 4-63.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533-536.

Rumelhart, D.E., & McClelland, J.L. (1986). *Parallel Distributed Processing, V.1*. Cambridge, MA: MIT Press.

Schneider, W. (1987). Connectionism: Is it a paradigm shift for psychology. *Behavior Research Methods Instruments & Computers, 19*(2), 73-83. doi: 10.3758/bf03203762

Schuijer, M. (2008). *Analyzing atonal music: Pitch-class set theory and its contexts*. Rochester, NY: University of Rochester Press.

Seidenberg, M. (1993). Connectionist models and cognitive theory. *Psychological Science, 4*, 228-235.

Shamanski, K.S., & Dawson, M.R.W. (1994). *Problem type by network type interactions in the speed and transfer of connectionist learning.* Paper presented at the Machine Learning Workshop at AI/GI/VI'94, Banff, AB.

Sloboda, J.A. (1985). *The Musical Mind: The Cognitive Psychology Of Music*. Oxford: Oxford University Press.

Tillmann, B., Bharucha, J., & Bigand, E. (2003). Learning and perceiving musical structures: Further insights from artificial neural networks. In I. Peretz & R. Zatorre (Eds.), *The Cognitive Neuroscience of Music* (pp. 109-123). Oxford: Oxford University Press.

Todd, P.M., & Loy, D.G. (1991). *Music And Connectionism*. Cambridge, Mass.: MIT Press.

Tymoczko, D. (2011). *A Geometry Of Music: Harmony And Counterpoint In The Extended Common Practice* (E-pub ed.). New York: Oxford University Press.

Van Gelder, T. (1991). What is the "D" in "PDP"? A survey of the concept of distribution. In W. Ramsey, S. P. Stich & D. E. Rumelhart (Eds.), *Philosophy And Connectionist Theory* (pp. 33-59). Hillsdale, NJ: Lawrence Erlbaum Associates.

Yust, J. (2016). Special collections: Renewing set theory. *Journal of Music Theory, 60*(2), 213-262. doi: 10.1215/00222909-3651886

Yust, J. (2017a). Harmonic qualities in Debussy's "Les sons et les parfums tournent dans l'air du soir". *Journal of Mathematics and Music, 11*(2-3), 155-173. doi: 10.1080/17459737.2018.1450457

Yust, J. (2017b). Review of Emmanuel Amiot, *Music through Fourier Space: Discrete Fourier Transform in Music Theory* (Springer, 2016). *Music Theory Online, 23*(3).

Yust, J. (2019). Stylistic information in pitch-class distributions. *Journal of New Music Research, 48*(3), 217-231. doi: 10.1080/09298215.2019.1606833

Zatorre, R., & McGill, J. (2005). Music, the food of neuroscience? *Nature, 434*(7031), 312-315. doi: 10.1038/434312a.

Chapter 4 : Understanding music by understanding the networks

Introduction

This thesis used artificial neural networks to explore musical sets. Even though we do not usually experience music as musical sets, ANN's can provide us some insight as to how certain properties of these sets might be represented or encoded by the brain.

While many researchers use artificial neural networks to explore music, these researchers rarely focus on their networks' internal structure, because they expect networks capture informal properties of music. But in order to inform cognitive science, we need to look inside networks to see what the networks are doing, their decision processes and internal representations. Dawson's (2018) finding of strange circles serve as a good example of musical properties that might have gone unnoticed if he did not look inside his networks. More importantly, those strange circles were an important piece of evidence to support the claim that networks do also find and capture very formal properties of music. However, Dawson's approach did not go far enough in exploring this particular discovery. With the aid of musical set theory, such as the one pioneered by Forte, we can further understand the formal properties of musical networks.

What kind of formal musical properties do networks represent? In Chapter 1 we introduced some basic elements of formal music theory. We introduced notes, chords and scales, and different ways to group them in sets, according to musical rules. Music theory reveals the interval relation between the elements of musical sets is a key element in defining each type of musical entity.

Music set theorist use set theory operations that allow them to uncover novel relations between musical elements. The Discrete Fourier Transform (DFT) in particular has been used to analyze interval structure and similarity between musical sets, because of how it allows us to explore periodic phenomena. In particular, we used Yust's approach which represented musical sets on phase spaces, that could be visualized as clocks.

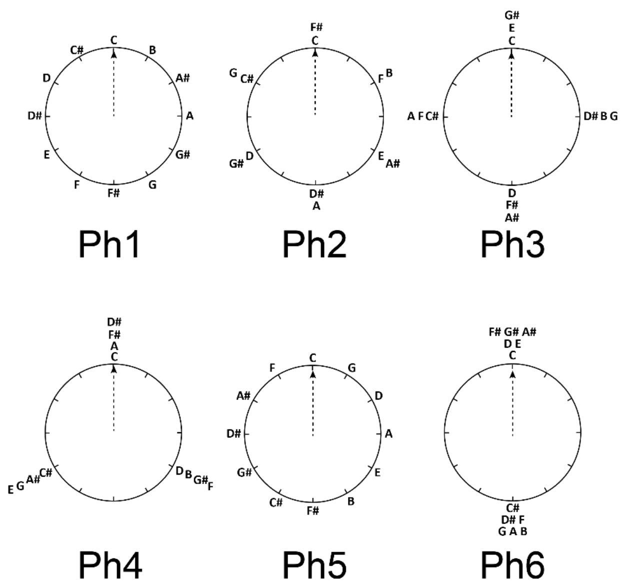


Figure 4-1. Yust Phase spaces, as shown in Chapter 1.

Taking into account the striking resemblance between hidden unit structure in the form of strange circles and Yust's phase spaces (see Chapters 2 and 3), we proceeded to explore the hypothesis that may be networks are using Fourier phase spaces in their decisions. Chapter 2 explored three value unit networks, trained on different musical tasks: interval classification, scale mode classification, and triad classification.

Having already identified the qualitative similarities between our hidden unit structure and Fourier phase spaces, we asked whether there is a quantitative relation between those 2.

To test this, we fitted our weights to projections of pitch classes on phase spaces, a procedure detailed on chapter 2. We found that for each hidden unit, there was a particular phase space, at a particular phase, with a nearly perfect fit to the hidden unit structure. The same was true for our other two tasks. For all networks trained, across the different tasks, we found near perfect fits between hidden unit weights and Fourier phase spaces.

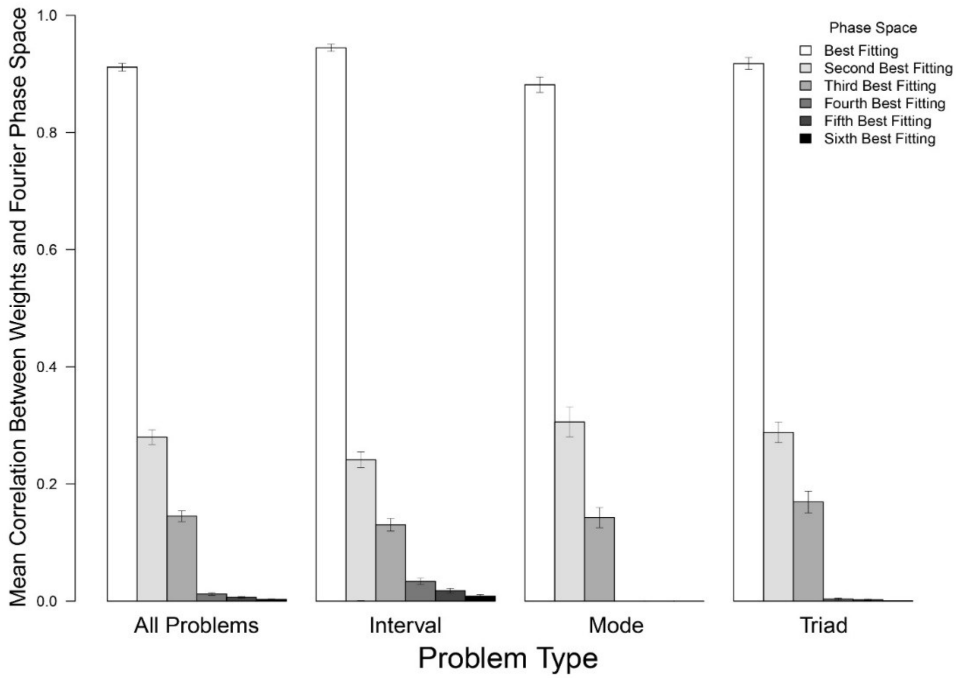


Figure 4-2 Plot of mean correlation between a Fourier phase space projections and hidden unit weights, as shown on chapter 2. Note how for every problem tackled by the network on that chapter, there is a phase space that is highly correlated with the weights.

Chapter 2 results demonstrate value unit networks use Fourier phase spaces to classify our musical sets. Is this true for networks using a different activation function? Moreover, how exactly are networks using these Fourier properties to make the decisions? To explore this, Chapter 3 focused on musical networks that use a logistic activation function. If we find Fourier phase spaces in a network so fundamentally different, this strengthens our argument for the importance of Fourier phase spaces for music and musical cognition.

Chapter 3 described training logistic networks to classify musical sets into four different kinds of triads. Once trained, we again fit Fourier phase spaces to hidden unit weights. Again, we found a high correlation between phase spaces and hidden unit structure, and so we confirmed that logistic unit networks, just as Gaussian networks, find Fourier phase spaces and use them to perform musical classifications

On chapter 3 we also present a further exploration into *how* networks used phase spaces to perform classify triads. Here, we noticed that hidden units were turning on to types of triads. But the activity elicited by a triad type was not always the same on a hidden unit. A closer look to HU activity by triad type, showed us that certain triads had very high activity on one unit, but they also had low activity on others, sometimes of opposite sign. This indicates logistic networks use coarse coding to classify triads. Closer examination of HU activities showed us that the network was using all its HU activities to make its decision.

Directions For Future Research

Overall, the results of Chapters 2 and 3 show, in different kinds of networks, and in different musical tasks, musical networks use Fourier phase spaces to encode musical properties and to classify musical stimuli. Chapters 2 and 3 illustrate how formal music theory can guide network interpretation: knowing ideas developed in music theory (i.e., Fourier phase spaces), we search for similar types of representations in musical networks. However, now we have discovered Fourier phases spaces within our networks, we can now plan future research in which network interpretations can guide formal music theory, as well as inspire new research on musical cognition.

Network interpretations can offer new ideas to both music theory and musical cognition because our networks do not use Fourier phase spaces as they are used in musical set theory. In musical set theory, we compute a different DFT for each musical set, and we can then compare different musical entities by comparing their DFTs. However, our networks use the same Fourier phase space – the same hidden unit connection weights – to transform *every* musical stimulus. Indeed, because networks apply the same 'Fourier analysis' to each stimulus, networks discover novel ways to compare stimuli (e.g., the coarse coding detailed in Chapter 3). Our new understanding of how networks use Fourier phase spaces inspires new approaches for comparing different musical entities, offering new ideas to music theory. Furthermore, because networks apply Fourier phase spaces in new ways, different similarity relationships between musical entities emerge, relationships which we can test with human participants. In general, if human musical cognition uses representations similar to those discovered in our networks, then we expect to predict behavioral regularities – for instance, judgments of similarity between stimuli – from network properties. To provide a concrete example of how network interpretations inform music theory and musical cognition, let us briefly consider one important topic in music theory: geometric relations between different musical entities.

Many music theorists develop theories which place different musical entities in a geometric space (Krumhansl, 1990, 2005; Rings, 2011; Tymoczko, 2006, 2011, 2012). Similar entities are located nearer to one another, while dissimilar entities are located further apart. Musical geometries differ from one another in how similarity is defined. For instance, one theory might measure similarity by counting the number of pitch-classes shared by two different musical entities. A different theory could measure similarity by comparing the DFTs of two different musical entities.

After defining a similarity measure, we can create a map to illustrate the spatial arrangement of musical entities. One approach to creating a map is to create a similarity matrix in which each row corresponds to a musical entity, each column also corresponds to a musical entity, and each entry in the matrix provides the measured similarity between two entities. One can then convert the similarity matrix into a map by analyzing the matrix with multidimensional scaling (MDS).

We can illustrate this approach using different scale modes (one of the tasks reported in Chapter 2) as musical entities. We measure the similarity between two different scale modes as the number of pitch-classes shared by the musical sets representing two different scales. After measuring similarity between all possible pairs of scale modes, an MDS analysis of the similarity matrix produces a map like the one illustrated in Figure 4-3

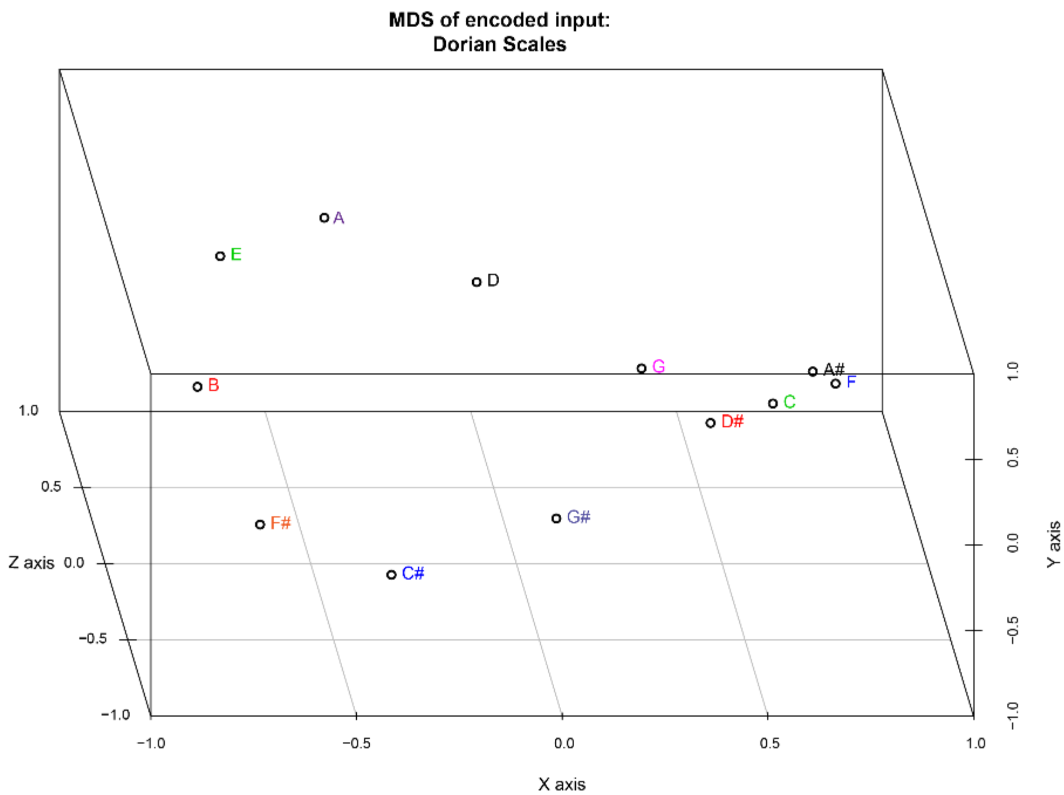


Figure 4-3 Plot representing the similarity relationship between each scale, on a three-dimensional space.

The MDS map in Figure 4-3 plots all of the Dorian mode scales; each label corresponds to the root of a scale. Note the positions of the scales in the MDS plot arranges the roots around a circle of fifths. Dorian scales whose roots are closer together in the circle of fifths (Figure 4-4) are also closer together in the MDS plot.

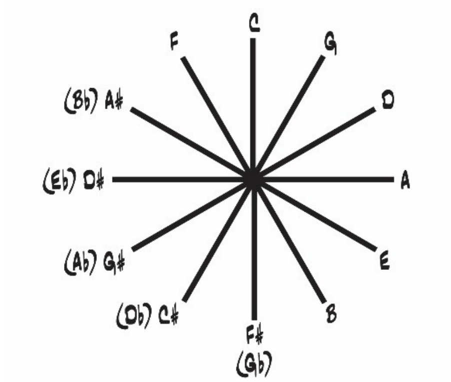


Figure 4-4. Circle of fifths

Importantly, our network interpretations provide a new way for measuring similarity between musical entities: the similarity of hidden unit activities produced in the network by two different musical entities. We can define similarity as the Euclidean distance between the vector of hidden unit activities produced by one entity and the vector of hidden unit activities produced by another. When MDS is applied to a similarity matrix based on this new matrix, a very different map of scales is produced (Figure 4-5).

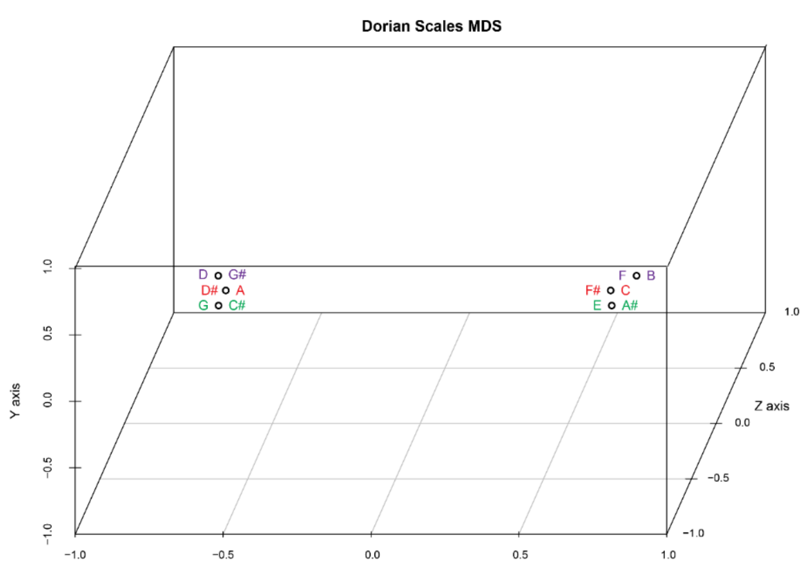


Figure 4-5. Three-dimensional map of the activity of our network, for the Dorian Scales.

The organization of the scales in Figure 4-5 differs strikingly from the organization of the same scales in Figure 4-3. First, the Figure 4-5 scales are not arranged around the circle of fifths. Second, the organization of Figure 4-5 symmetric: the left side of the graph mirrors the right side. Third, pairs of scales occupy the same position in the graph: for example, F and B, or F# and C. Indeed, scales whose roots are a tritone apart are placed in identical positions by the MDS. Clearly, when similarity is measured in terms of hidden unit activities – which in turn are related to Fourier phase spaces – a very different musical space results when compared to a space based on shared pitch-classes.

By using network interpretations, we can create a new musical geometry for formal music theory. This theory raises new ideas to be studied by music theorists. For example, how might we formally express how two scale modes, with roots separated by a tritone, are equivalent? How might we formally describe the symmetric relationship between scales revealed in Figure 4-5? Answering such questions will be the focus of future research; we expect the answers will emerge from understanding how networks exploit Fourier phase spaces to classify musical stimuli.

Another implication emerging from Figure 4-5, and its dramatic difference from Figure 4-3, concerns musical cognition. The two figures illustrate how different ideas about musical similarity lead to very different musical spaces. By hypothesis, we propose humans represent music in such a space (Krumhansl, 1990). But what space provides the foundation for human musical cognition? To answer this question, we need to collect data from human participants to generate similarity matrices (e.g., by having participants judge the similarity of pairs of musical entities). By performing MDS on human data, and by comparing the results to potential musical geometries (e.g., Figures 4-3 and 4-5), we can determine whether human musical cognition uses representations similar to the Fourier phase spaces we discovered in our networks in Chapters 2 and 3.

Exploring musical geometry is not new, nor is using artificial neural networks to study musical tasks. However, we rarely see these two activities combined, because doing so requires one missing element: interpreting the structure of trained network. It is only when we look inside networks, and discover entities like Fourier phase spaces, that we can investigate new musical geometries inspired by network structure.

Indeed, the main point of this thesis is to reveal the rich musical properties represented in artificial neural networks trained to solve musical problems. In Chapter 1 we saw most network studies of music are inspired by the goal of capturing musical properties which we cannot express formally. This goal, in turn, means researchers rarely explore the structure of musical networks, and certainly do not seek *formal* structures in such networks. However, Chapters 2 and 3 revealed musical networks build representations describable in the same formal terms used to describe the Fourier structure of musical sets. In addition, how networks use (formal) Fourier structure differs from how musical set theorists do (in particular, see the distributed representations detailed in Chapter 3). Thus, musical networks provide a bridge between formal mechanisms in musical set theory and potential representations for musical cognition. Importantly, though, this bridge only becomes available when we explore and detail the internal structure of trained musical networks.

References

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

Krumhansl, C.L. (2005). The geometry of musical structure: A brief introduction and history. *ACM Computers In Entertainment, 3*(4), 1-14.

Rings, S. (2011). *Tonality and transformation*. New York: Oxford University Press.

Tymoczko, D. (2006). The geometry of musical chords. *Science, 313*(5783), 72-74.

Tymoczko, D. (2011). *A Geometry Of Music: Harmony And Counterpoint In The Extended Common Practice* (E-pub ed.). New York: Oxford University Press.

Tymoczko, D. (2012). The generalized Tonnetz. *Journal of Music Theory, 56*(1), 1-52. doi: 10.1215/00222909-1546958

Bibliography

Abbott, A. (2002). Neurobiology: Music, maestro, please! *Nature, 416*(6876), 12-14. doi: 10.1038/416012a.

Amiot, E. (2016). *Music Through Fourier Space: Discrete Fourier Transform in Music Theory*. Cham: Springer International Publishing : Imprint: Springer.

Ballard, D. (1986). Cortical structures and parallel processing: Structure and function. *The Behavioral And Brain Sciences, 9*, 67-120.

Bechtel, W. and A. A. Abrahamsen (2002). *Connectionism And The Mind : Parallel Processing, Dynamics, And Evolution In Networks*. Malden, MA, Blackwell.

Berkeley, I.S.N., Dawson, M.R.W., Medler, D.A., Schopflocher, D.P., & Hornsby, L. (1995). Density plots of hidden value unit activations reveal interpretable bands. *Connection Science, 7*, 167-186.

Bharucha, J.J. (1999). Neural nets, temporal composites, and tonality. In D. Deutsch (Ed.), *The Psychology Of Music* (2nd ed., pp. 413-440). San Diego, CA: Academic Press.

Callender, C. (2007). Continuous harmonic spaces. *Journal of Music Theory, 51*(2), 277-332. doi: 10.1215/00222909-2009-004

Churchland, P.S., & Sejnowski, T.J. (1992). *The Computational Brain*. Cambridge, MA: MIT Press.

Dawson, M.R.W. (1990). Training networks of value units: Learning in PDP systems with nonmonotonicactivation functions. *Canadian Psychology, 31*(4), 391.

Dawson, M.R.W. (2004). *Minds And Machines: Connectionism And Psychological Modeling*. Malden, MA: Blackwell Pub.

Dawson, M.R.W. (2005). *Connectionism : A Hands-on Approach* (1st ed.). Oxford, UK ; Malden, MA: Blackwell Pub.

Dawson, M.R.W. (2009). Computation, cognition – and connectionism. In D. Dedrick & L. Trick (Eds.), *Cognition, Computation, and Pylyshyn* (pp. 175-199). Cambridge, MA: MIT Press.

Dawson, M.R.W. (2018). *Connectionist Representations of Tonal Music: Discovering Musical Patterns By Interpreting Artificial Neural Networks*. Edmonton, AB: Athabasca University Press.

Dawson, M.R.W., Berkeley, I.S.N., Medler, D.A., & Schopflocher, D.P. (1994). Density plots of hidden value unit activations reveal interpretable bands and microbands.

Dawson, M.R.W., Kremer, S., & Gannon, T. (1994). Identifying the trigger features for hidden units in a PDP model of the early visual pathway. In R. Elio (Ed.), *Tenth Canadian Conference On Artificial Intelligence* (pp. 115-119). San Francisco, CA: Morgan Kaufmann.

Dawson, M.R.W., Perez, A., & Sylvestre, S. (2020). Artificial neural networks solve musical problems with Fourier phase spaces. *Scientific Reports, 10*(1), 7151. doi: 10.1038/s41598-020-64229-4

Dawson, M.R.W., & Schopflocher, D.P. (1992). Modifying the generalized delta rule to train networks of nonmonotonic processors for pattern classification. *Connection Science, 4*, 19-31.

Deutsch, D. (1999). *The Psychology Of Music* (2nd ed.). San Diego: Academic Press.

Enquist, M. and S. Ghirlanda (2005). *Neural Networks and Animal Behavior*. Princeton, Princeton University Press.

Forte, A. (1973). *The Structure Of Atonal Music*. New Haven: Yale University Press.

Griffith, N., & Todd, P.M. (1999). *Musical Networks: Parallel Distributed Perception And Performace*. Cambridge, Mass.: MIT Press.

Hanson, H. (1960). *Harmonic materials of modern music; resources of the tempered scale*. New York,: Appleton-Century-Crofts.

Hinton, G.E. (1986). *Learning distributed representations of concepts.* Paper presented at the The 8th Annual Meeting of the Cognitive Science Society, Ann Arbor, MI.

Hinton, G.E., McClelland, J., & Rumelhart, D. (1986). Distributed representations. In D. Rumelhart & J. McClelland (Eds.), *Parallel Distributed Processing* (Vol. 1, pp. 77-109). Cambridge, MA: MIT Press.

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

Krumhansl, C.L. (2005). The geometry of musical structure: A brief introduction and history. *ACM Computers In Entertainment, 3*(4), 1-14.

Lewin, D. (1959). Re: Intervallic relations between two collections of notes. *Journal of Music Theory, 3*(2), 298-301.

Lewin, D. (2001). Special cases of the interval function between pitch-class sets X and Y. *Journal of Music Theory, 45*(1), 1-30. doi: 10.2307/3090647

Lewin, D. (2007). *Generalized Musical Intervals And Transformations*. New York: Oxford University Press.

McClelland, J.L., & Rumelhart, D.E. (1986). *Parallel Distributed Processing, V.2*. Cambridge, MA: MIT Press.

McCloskey, M. (1991). "Networks and theories: The place of connectionism in cognitive science." *Psychological science* **2**: 387-395.

McCloskey, M. (1991). Networks and theories: The place of connectionism in cognitive science. *Psychological Science, 2*, 387-395.

Peretz, I., & Zatorre, R.J. (2003). *The Cognitive Neuroscience Of Music*. Oxford ; New York: Oxford University Press.

Peretz, I., & Zatorre, R.J. (2005). Brain organization for music processing. *Annual Review of Psychology, 56*, 1-26.

Quinn, I. (2006). General equal-tempered harmony: Introduction and Part 1. *Perspectives of New Music, 44*, 114-158.

Quinn, I. (2007). General equal-tempered harmony: Parts 2 and 3 *Perspectives of New Music, 45*, 4-63.

Rings, S. (2011). *Tonality and transformation*. New York: Oxford University Press.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533-536.

Rumelhart, D.E., & McClelland, J.L. (1986). *Parallel Distributed Processing, V.1*. Cambridge, MA: MIT Press.

Schneider, W. (1987). Connectionism: Is it a paradigm shift for psychology. *Behavior Research Methods Instruments & Computers, 19*(2), 73-83. doi: 10.3758/bf03203762

Schuijer, M. (2008). *Analyzing atonal music: Pitch-class set theory and its contexts*. Rochester, NY: University of Rochester Press.

Seidenberg, M. (1993). Connectionist models and cognitive theory. *Psychological Science, 4*, 228-235.

Shamanski, K.S., & Dawson, M.R.W. (1994). *Problem type by network type interactions in the speed and transfer of connectionist learning.* Paper presented at the Machine Learning Workshop at AI/GI/VI'94, Banff, AB.

Sloboda, J.A. (1985). *The Musical Mind: The Cognitive Psychology Of Music*. Oxford: Oxford University Press.

Smolensky, P. (1988). "On the proper treatment of connectionism." *Behavioral and Brain Sciences* **11**: 1-74.

Susanni, P., & Antokoletz, E. (2012). *Music and twentieth-century tonality: Harmonic progression based on modality and the interval cycles*: Routledge.

Tillmann, B., Bharucha, J., & Bigand, E. (2003). Learning and perceiving musical structures: Further insights from artificial neural networks. In I. Peretz & R. Zatorre (Eds.), *The Cognitive Neuroscience of Music* (pp. 109-123). Oxford: Oxford University Press.

Todd, P.M., & Loy, D.G. (1991). *Music And Connectionism*. Cambridge, Mass.: MIT Press.

Tymoczko, D. (2006). The geometry of musical chords. *Science, 313*(5783), 72-74.

Tymoczko, D. (2011). *A Geometry Of Music: Harmony And Counterpoint In The Extended Common Practice* (E-pub ed.). New York: Oxford University Press.

Tymoczko, D. (2012). The generalized Tonnetz. *Journal of Music Theory, 56*(1), 1-52. doi: 10.1215/00222909-1546958

Van Gelder, T. (1991). What is the "D" in "PDP"? A survey of the concept of distribution. In W. Ramsey, S. P. Stich & D. E. Rumelhart (Eds.), *Philosophy And Connectionist Theory* (pp. 33-59). Hillsdale, NJ: Lawrence Erlbaum Associates.

Yust, J. (2016). Special collections: Renewing set theory. *Journal of Music Theory, 60*(2), 213-262. doi: 10.1215/00222909-3651886

Yust, J. (2017a). Harmonic qualities in Debussy's "Les sons et les parfums tournent dans l'air du soir". *Journal of Mathematics and Music, 11*(2-3), 155-173. doi: 10.1080/17459737.2018.1450457

Yust, J. (2017b). Review of Emmanuel Amiot, *Music through Fourier Space: Discrete Fourier Transform in Music Theory* (Springer, 2016). *Music Theory Online, 23*(3).

Yust, J. (2019). Stylistic information in pitch-class distributions. *Journal of New Music Research, 48*(3), 217-231. doi: 10.1080/09298215.2019.1606833

Zatorre, R., & McGill, J. (2005). Music, the food of neuroscience? *Nature, 434*(7031), 312-315. doi: 10.1038/4343