A Connectionist Approach to Driving Chord Progressions Using Tension

Andrés F Melo*
*Department of Computing
City University, London EC1V 0HB
andres@soi.city.ac.uk

Geraint Wiggins
†Department of Computing
City University, London EC1V 0HB
geraint@soi.city.ac.uk

Abstract

This paper describes how a backpropagation neural network learnt and reproduced harmonic patterns. These patterns were related in learning and production to musical tension. The neural network thus learnt to produce the next chord in a sequence such that listeners perceive a given degree of musical tension. During this research, it was confirmed that musical tension is measurable and that harmony conveys tension. Ultimately, the results pragmatically show that harmonic tension patterns exist, can be found and reproduced within the style defined by adequate pieces of music.

1 Introduction

The original intention of this project was as a step to produce music in real time—achieving a balance between the unity and variety in the output. The idea is that a neural network provides the unity by learning the harmonic context from a given piece of music. This harmonic context is further related to a measure of musical tension. By varying the tension level in the input, the output would be expected to stay within the harmonic context of the original (providing unity), and the tension would drive the harmony, thus becoming the source of variety.

1.1 Tension in music

Tension can be seen as the incompleteness of the music at any one point. More intuitively, the degree of tension can be related to how *unfinished* the piece of music would sound if it stopped at the point. Tension thus changes dynamically as the piece of music progresses (Krumhansl, 1997).

Tension has been studied for different aspects of pieces of music. Significantly, Narmour (1990) approaches the effect of melody in tension has been approached analytically. The concept of tension is opposed to that of relaxation, and derives from certain patterns in the melodic lines that produce or solve expectancy.

As for harmony, Krumhansl (1997) measured tension and found that the predictors of chord distance defined by Lerdhal (1988) related to the tension curves well for a piano sonata of Mozart. This suggests that harmony has an effect on tension that is composed with the tension resulting from the melodic lines and presumably from other aspects of the music.

1.2 Machine learning

Machine learning has been previously used to learn from a set of examples to predict the next chords in a sequence (Bharucha and Todd, 1989; Berger and Gang, 1997). This work extends the concepts by testing whether a backpropagation neural network can learn patterns of tension in harmony and use them to generate sound sequences with new, specific tension curves.

Machine learning requires examples of the behaviour the system is expected to learn. It is in fact ideal when examples are easier to find than rules (Dolson, 1989). Examples of real pieces of music are of course available in various formats, and the choice of a representation will have an effect on the learning curve. The representation used should favour harmony over other aspects of the music used.

Tension curves that run in parallel to the music examples are required as well in this case, as they are part of the behaviour that the network should learn. In this case, the data needs to come from human listeners (rather than from theoretical models), as part of the interest in this work is in letting the network find patterns in the *raw* tension curves rather than reproducing rules that were stated by the models in the first place.

The results of neural networks tend to be pragmatical, and so they tend to be considered to be black boxes. The primary analysis of the results is thus whether the behaviour wanted—the production of a sequence of chords that produces a given tension curve—can be reproduced by the network. However, backpropagation networks allow analysis and rule extraction to some degree. This is achieved by analysing the weights of the links, comparing the performance of different architectures and repre-

*The authors were affiliated to the Department of Artificial Intelligence of the University of Edinburgh when developing the work presented here.

sentations, and by simulation of basic cases. In this case, the information provided by the network will suggest the tension patterns present in the music used in the training example.

1.3 Procedure

This paper concentrates on the results obtained with a musical piece: the fourth movement of Prokofiev's Classical Symphony. This musical piece was selected because the rhythmical and melodic patterns are relatively uniform, and the chords change frequently (often every other bar) and its harmony appears to drive the tension. Its duration (about four minutes) is substantial, but without compromising seriously the attention of the listeners.

This work consisted of three distinct stages:

1. elicitation of tension curves,
2. development and training of the neural network, and
3. evaluation of the output of the neural network.

The following sections address each stage separately. Melo (1998) presents further detail of the first two stages.

2 Stage 1: Eliciting tension curves

To obtain tension curves for the musical piece, ten Western listeners were asked to indicate the tension they perceived while listening to it. They indicated the tension by rotating a sprung wheel (Figure 1). The reaction of the spring provides feedback on the position of the wheel, so the listeners have to apply more force to indicate higher tension, or no force for no tension. Before the Prokofiev piece of music, listeners used the equipment to record the tension in one to four pieces of music (according to random settings of the test), to make sure that they felt comfortable with the equipment and task.

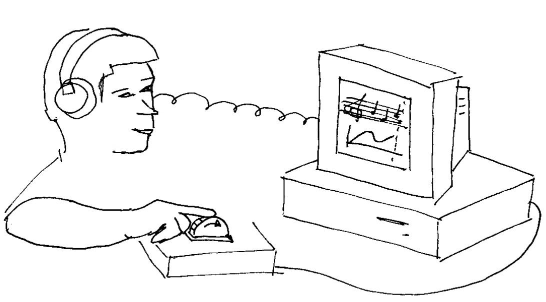

Figure 1: Settings for measuring tension

Before starting the measurement, the concept of tension was explained to the listeners as the uneasy—as opposed to relaxed—sound of the music, and as how "unfinished" the piece would sound if it stopped at that precise moment. Some of the listeners expressed having initial doubts about their ability to perform the task, but felt confident that they could do it after the first trial.

The music used was a quantised MIDI file in which all the instruments were set to piano and the dynamic was set to be constant. This insures that changes in volume, colour and interpretation are not factors in the perceived tension.

2.1 Raw tension curves

The tension curves given by the listeners were different from each other in that some used the whole range of the wheel whereas others would indicate changes in tension with very slight movements. This was to be expected, as the measure of tension is not defined in absolute terms. Still, the listeners seemed to react at the same points and in the same ways, reacting is the same points, increasing or decreasing the tension at the same time. Visually, the peaks and valleys in the curves appear to coincide (see Figure 2).

The agreement among the curves was confirmed using using Friedmans test of rank correlation. Friedman's test compares a number of data series for monotonic behaviour, that is, tests whether the series increase (or decrease) values at the same points (Brownlee, 1965; Hollander and Wolfe, 1973). This test differs from correlation analysis in that it is *non-parametric*: it does not assume that the data series have the same distribution. This allows for the fact that the tension curves provided by different listeners may be expressing their perception of tension in different ways—that is, not necessarily proportionally. Friedman's test showed that the possibility of the curves not being correlated is under 10^{-6}.

2.2 Obtaining a single tension curve

The ten tension curves obtained have to be condensed into one to be used for training. Condensing the data is expected to make up for differences in the "style" of the curves provided by different listeners, as well as for errors in the curve because of a listener losing attention.

The median is used to condense the data series. The argument is the same as when choosing Friedman's test: that the interest is to show that the curves are monotonic. The median is non-parametric because it does not assume a data distribution in the series. To apply it, however, it was found appropriate to normalise the ten curves obtained by scaling them between 0 and 100.

Figure 2 shows the way the median represents the ten curves in a section of the piece of music. This figure illustrates how the median represents the general tendency of the ten curves and how each one of the curves contributes to the resulting curve in the measure that it agrees with the majority of the other curves. The median for the whole piece is shown in Figure 3. The curve reflects the underlying structure of the piece: $AABA'$. A goes ap-

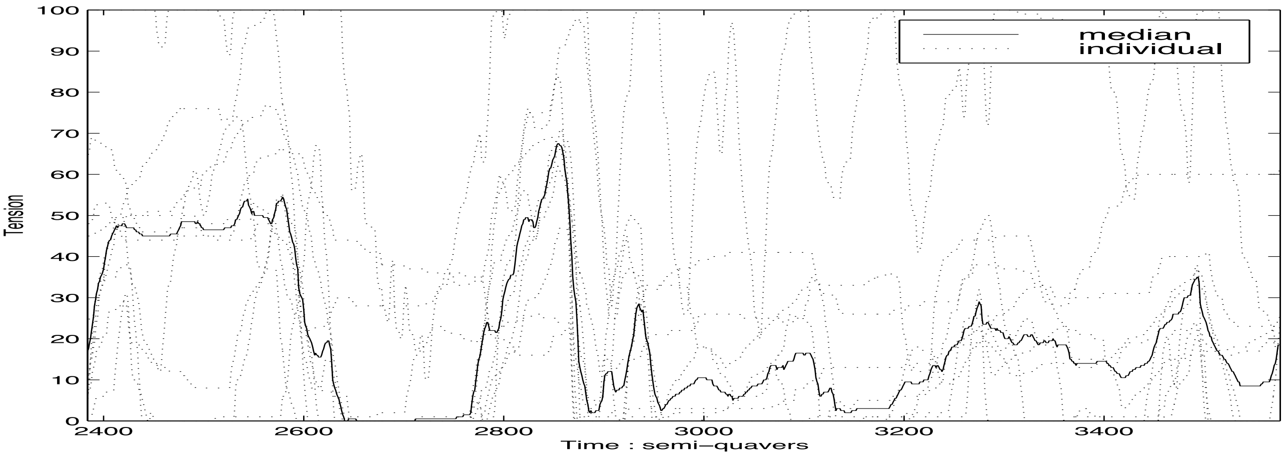

Figure 2: Ten tension curves and their median

proximately from bar 0 to bar 90, then repeats from 90 to 180, and then its variation A' begins at approximately bar 240. This and the statistical agreement in the responses of the listeners give confidence in the curve as a representative indication of the development of the piece.

3 Stage 2: Training neural networks to produce chords based on given tension curves

The concept of this system is that it should produce the chord that continues a sequence to convey a given degree of tension. The input of the neural network is thus the set of preceding chords, and the degree of tension. With the piece of music chosen, time slices of one bar are appropriate, as in most cases in it, the chords take integer numbers of bars. The set of chords used in the input is then the *n* chords in the preceding *n* bars (where *n* is the size of the time window).

Besides the structure of the network itself and the learning parameters, the main variables in the definition of the system were the representation of the chords and the window size. Different representations of chords and window sizes (*n*) were tried, to find those with which the network could produce the best results.

The chord representations tried were based on the occurrence of each one of the twelve tones in each bar. Some variations in the representation included the specific identification of the lowest tones in the bar and the consideration of the percentage of time that each tone occurred in the bar.

In each case, the architecture of the neural network was optimised. The training procedure was standard: one fifth of the data from the musical piece was chosen at random and used for testing the network (and excluded from the data used for training). Each datum contained an output chord, its average tension, and the *n* preceding chords. Training data was presented to the network at random. Training stopped when the accuracy of the network on the testing data started to decrease—which means that the network is starting to memorise more than it is generalising.

3.1 Parameters of the best network built

The representation found to be the best indicated the fraction of time that a given tone would sound during the bar. Doubling was considered in the representation, so if, for example, the same tone is being played simultaneously in two different octaves, that would count twice. Each chord would thus be represented by twelve parameters, one for each of the 12 tones.

The ideal window size for this representation was of 13 bars. It was found to be ideal because the median of the error with the test data with three different random seeds was lower than for other window sizes. These two properties—especially the window size—relate directly to characteristics of the piece of music used, as will be discussed later.

3.2 Analysis of the trained network

Neural networks behave like black boxes in the sense that the knowledge is distributed rather than localised, so rules cannot be extracted by looking at the network itself. However, it is possible to gain some knowledge about the behaviour learnt by the network by 1) training different networks with different information and representations, and comparing their performance; 2) analysing the weights of the links in a trained network to find out the relative importance of the input factors; and 3) testing the trained network for different kinds of input and finding the sensitivity of the outputs. All these functions can be seen as a statistical analysis of the input data; because what the network does during learning is generalising from the training data.

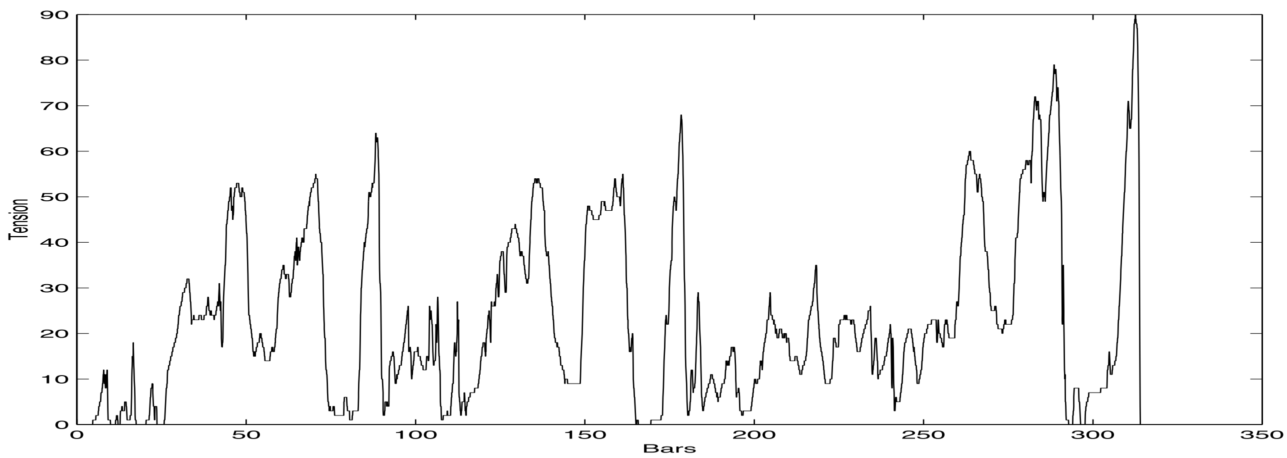

Figure 3: Tension curve for the fourth movement of Prokofiev's Classical Symphony

In this case, the training data was harmonic information from Prokofiev's Classical Symphony, and so the analysis of the properties of the network relate to this piece and do not necessarily apply to other pieces of music.

A close-to-best performance was obtained by the representation that included information about doubling, but that the addition of the bass line does not produce significant improvements. Including doubling information makes the identity of the chord clearer, as this would be expected to scale down the relative occurrence of auxiliary notes. This may also suggest that the density of the sound is a general factor that affects the chord sequence; *i.e.*, some chords sequences appear to be reserved for denser parts of the piece of music.

It is possible to perform sensitivity analysis on the network to determine the effect of changing the inputs. This was done by recording the differences in each output node after varying each input datum for each example by 5 and -5 percent of the range of values the corresponding input node can take. The results for all the examples are averaged, producing values representing the sensitivity of each output to changes in each input.

Figure 4 shows the average of the absolute values of the effects of each input chord to the output chord. It indicates the relative relevance each bar has in the chord the net outputs. The chord in the last bar (column 13 in the figure), for example, has the strongest effect. This is reasonable considering that the output is its immediate continuation, probably similar in density and often continuing the same chord. It is significant that the even numbered columns tend to have a stronger effect than their neighbours, as this suggests a harmonic rhythm.

By considering the effect of tension on each one of the tones of the output chord (Figure 5), some insight can be gained as to the way harmonic tension behaves in the piece. In the figure, column 1 relates to the sensitivity of the output c to tension, column 2 relates to $c\sharp$, column 3 to d, and so on.

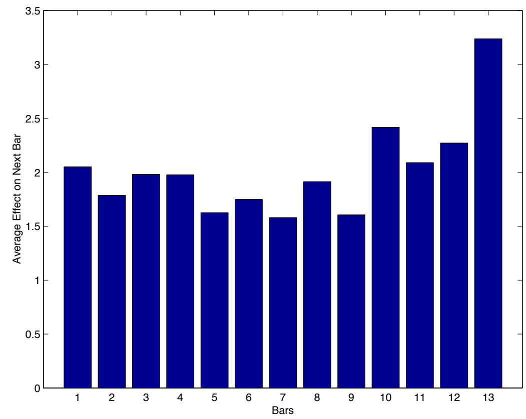

Figure 4: Average absolute sensitivity of the output chord to the input chords

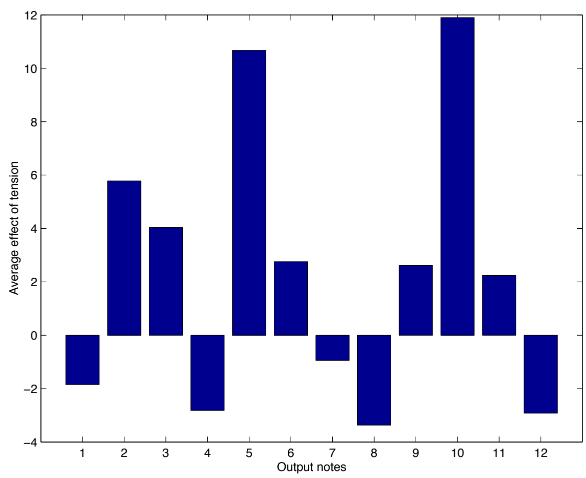

Figure 5: Sensitivity to the tones in the output chord to tension

As can be appreciated in the sensitivity values, tension has an important influence in the output of the net. Considering these values, it becomes apparent that high tension biases the output towards the A major (columns 2, 5 and 10). This is not surprising, since A major is the dominant of the main key. Low tension is more ambiguous, as its main effect is biasing to g, b and $d\sharp$ (tritone). This suggests that the subdominant (G major) is produced for low tension effects. It is even more surprising, however, that the tonic, D major, seems not to be related to low tension.

High tension in the dominant, ambiguity in the tonic and low tension in the subdominant appears to be counter-intuitive. In this context the piece would appear to be written in G major (rather than D major), as it makes more sense to consider that low tension is associated to the tonic, some ambiguity in tension to the dominant, and high tension to the dominant of the dominant.

This result has an explanation: many phrases of the piece of music used for training are—on first listening—harmonically ambiguous. It frequently uses plagal cadences (progressing from the subdominant to the tonic), which would seem to have been interpreted as half cadences (in the subdominant key) by the listeners [1].

4 Stage 3: Testing whether the network's output fits the given tension curves

What would be expected at this point is that by giving 13 preceding bar-chords to the network and a desired degree of tension to the network, it will output the chord for the next bar. The chords for the 13 preceding bars are interpreted in the context of the musical piece used for training, and the chord output by the net would be expected to produce the degree of tension wanted. In this way, it may be possible to obtain a sequence of chords that is within the harmonic context of the training piece and that has the wanted tension curve. The next stage is to test whether the tension obtained is actually the one given to the network.

To test this, random initial chords and two tension curves were used to initialise the system. The objective is to measure the tension in these sequences (using the same procedure as in Stage 1) and compare it to the two tension curves used. In this way, it can be found whether the sequences produced by the network actually produce the wanted tension curves. However, the procedure described in Stage 1 requires a piano-roll (or a similar representation).

[1] This ambiguity was reportedly introduced intentionally, as it was frequently exploited by Haydn, who was one of the models Prokofiev used for the Classical Symphony.

4.1 Obtaining a piano-roll version of the network's output

A second backpropagation neural network was used to produce a piano-roll version of each bar output by the network. Each column in the piano-roll contains 70 tones (the range of the original piece) and has a duration of one quaver (the minimum duration in the piece). Each tone in the column will be either on or not.

The input of this second network is the chord of the current bar, four piano-roll columns (*i.e.*, a time window with the previous four quavers), and the desired tension. Its output is the following two piano-roll columns.

A small time window was chosen because the purpose of this network is to convert chords into piano-roll columns rather than to memorise melodies or the piece of music. The output contains two columns to exploit the fact that the first quaver is on the beat while the second is not.

This network was trained using data from the original piece. The output nodes represent notes in the piano-roll columns; but their values are real numbers between 0 and 1—not binary. The original piece was used to determine the thresholds for each output node that maximise the accuracy of its output.

The whole system is illustrated in Figure 6. The main network ("bar-scale net")is used at the beginning of each bar to generate the chord (the "bar suggestion") that will produce the degree of tension in the input. The piano-roll network is used every two quavers to produce the following two piano-roll columns ("next two quavers"), based on the chord, the four previous piano-roll columns and the tension.

At the end of the bar, after eight piano-roll columns have been generated, the chord equivalent to these eight columns (rather than the chord output by the main network) is used to update the time window at the input of the main network. This is to allow for the fact that the piano-roll network may distort the chords in the main network output.

For testing purposes, two sets of pieces were created using this system. All sound sequences were under 80 seconds long. The first set consisted of two pieces generated to produce a section of the tension curve of the original piece. The second set consisted of eight pieces generated using a skewed sinusoidal wave as tension curve. The chord and piano-roll windows were initialised with random data for each one of the pieces. Importantly, the random seeds and the resulting sound sequences were not evaluated or selected before the experiment.

As expected, the piano-rolls produced by the system described are not melodic, and listeners expressed that they sounded almost like random music, while admitting that some "order" and even "style" could be noticed in them.

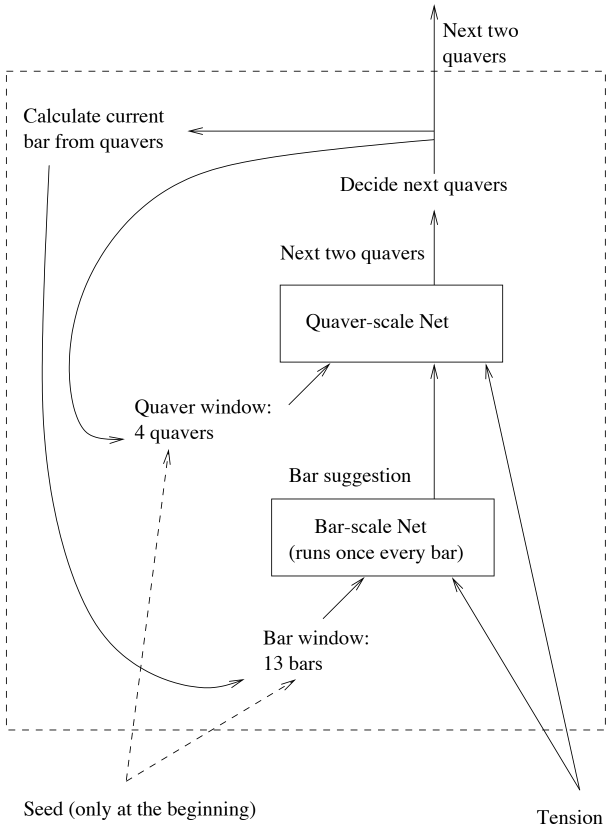

Figure 6: System to produce piano-rolls

4.2 Test procedure

At this stage, 16 Western subjects listened to the original piece of music and recorded the tension they perceived using the sprung wheel—both to let them get used to the experimental settings and to find differences between their response and the original tension curve used to train the network. After this, each participant recorded the tension for the two pieces in the first set and for four of the pieces in the second set. Listeners heard the sound sequences generated by the system in random order.

The data obtained was normalised and processed in the same way as in Stage 1, scaling the individual curves from 0 to 100 and extracting the median.

4.3 Results and analysis

Figures 7, 8 and 9 illustrate the results of Stage 3. Figure 7 shows the tension curve obtained for sequence A of the first set. In this case, the fitness of the resulting curve to the input tension is quite clear.

Figure 8 shows the tension curve obtained for sequence B of the first set. In this case, even though at some points the resulting curve moves in the same direction as the input tension, their fitness is poor. The problem in this case is that the random seed used to initialise the piano-roll produced very dense columns which were unconfortable to the listeners and allowed no noticeable changes in the chords or piano-roll.

Figure 9 shows the input tension curve used for the second set and the average of the tension curves obtained for its eight sequences. This curve has a good correlation in the first half and a tendency to a steady state in the second half. This curve is an average, and reflects the results obtained from the eight sequences in the set. Individual pieces fall into one of the following categories:

- Good fitness, with an obvious relation between the input and the resulting curves.
- Good fitness at the beginning, but tension falls into a steady state by the end.
- Delayed response: the resulting curve is an offset of the input tension.

The system can enter a stable state of saturation in which the inputs and outputs maintain high values, and this clearly depends on the random data used to initialise the input piano roll columns and chords.

4.4 Statistical analysis of the results

To test the fitness of the tension curves, they were compared to the tension curves used to generate the sound

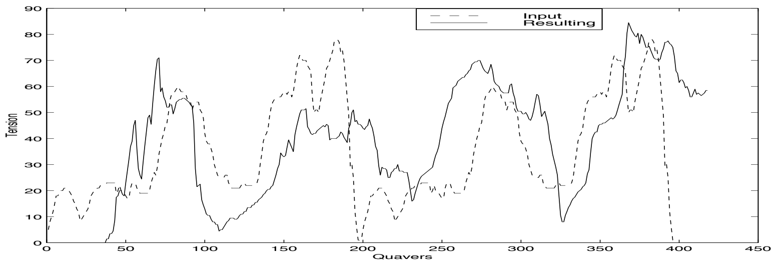

Figure 7: Input tension and perceived tension for sequence A, first set

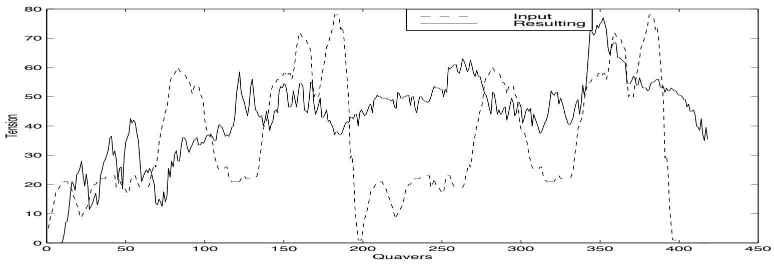

Figure 8: Input tension and perceived tension for sequence B, first set

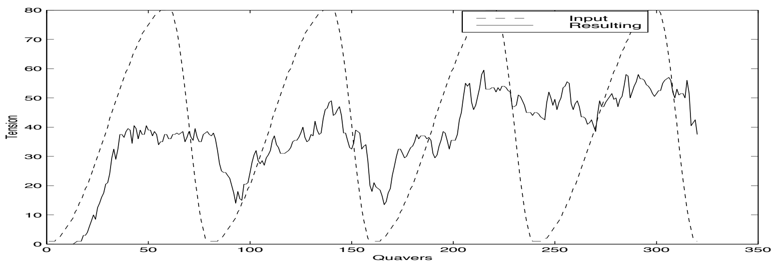

Figure 9: Input tension and average perceived tension for the second set

sequences. The comparison was done using the test of rank correlation (which is equivalent to applying Friedman's test to two data series only). The choice of a non-parametric test is once again justified by the fact that a distribution of tension cannot be assumed.

The rank correlation between the input tension curves and the tension indicated by the listeners was statistically significant in all the cases, but suspected not to be as strong as that found among the listeners in the original test.

The tension curves recorded for the original piece of music can be used as reference to test whether these differences could have originated from the different set of listeners that participated. The procedure for this comparison involves calculating the rank correlation for the tension curves obtained in Stage 1 and Stage 3 for the original piece. This value is then compared to each one of the correlations, considering the degrees of freedom in the data seried. The results indicate that the correlation between the two sets of listeners is better by far—so the perception of tension of the listeners that participated in Stage 3 is similar enough to that of the listeners in Stage 1.

In most cases, the correlation between the input tension and the perceived tension has high statistical significance nonetheless. The results are encouraging in that they show that the network learnt to reproduce aspects of the harmony to produce a given tension. The network interpreted the random input data in terms of the harmonic context it learnt, and produced the sound sequence based on it. In some cases, however, the initial random data forced the system into a stable state of saturation.

5 Discussion

The backpropagation network appears to have learned patterns that exist between harmonic information and the tension measured, within the harmonic context of the musical piece used. In some cases it is able to produce sound sequences that produced responses in the users equivalent to those measured for the original musical piece. These "successful" sequences are new as they are clearly distinct from the original musical piece. However, they are copies in the sense that they are based on the harmonic context of the original. What was not learnt from the original was the result of random factors that affect the neural network or of external inputs.

The random factors in this case are the data that is removed for testing purposes and the order in which the training data is presented to the network. These factors affect the way it contains the patterns harmony-tension patterns. Once training has finished, the network can be considered to be deterministic, but still its structure and outcomes will be affected by the random factors in its training.

The chords used for initialisation and the input tension curve are inputs to the system and not necessarily random. They affect the resulting sequence, probably at the same level that the musical piece used for training does, as they represent the material that goes through the harmonic context learnt by the network. The novelty of the sequence produced by the network could thus be traced back to the random factors during learning and the interaction of the system inputs and the systems complexity.

At a more basic level, the results showed that listeners tend to agree in what they understood as musical tension. The tension they perceived in the musical piece used was found by the neural network to correlate to the harmony, to the point that it could produce chords appropriate for the context and desired tension. The resulting structure of the backpropagation network reflects important features of the way harmony is perceived by listeners. The results ultimately suggest that the concept of tension can be used to drive the production of musical sequences.

References

J. Berger and D. Gang. A neural network modle of metric perception and cognition in the audition of functional tonal music. In *International Conference for Music Perception and Cognition*, pages 23–26. International Computer Music Association and Program of Psychoacoustics of IPSA, ICMA and IPSA, 1997.

J.J. Bharucha and P.M. Todd. Modeling the perception of tonal structure with neural nets. In P.M. Todd and D.G. Loy, editors, *Music and Connectionism*, pages 128–137. The MIT Press, Cambridge, MA, USA, 1989.

K.A. Brownlee. *Statistical Theory and Methodology in Science and Engineering*. John Wiley & Sons, New York, NY, USA, 1965.

M. Dolson. Machine tongues xii: Neural networks. In P.M. Todd and D.G. Loy, editors, *Music and Connectionism*, pages 3–19. The MIT Press, Cambridge, MA, USA, 1989.

M. Hollander and D.A. Wolfe. *Non-parametric Statistical Methods*. John Wiley & Sons, New York, NY, USA, 1973.

C.L. Krumhansl. Musical tension: cognitive, motional and emotional aspects. In *Proceedings from the Third Triennal ESCOM Conference*, pages 3–12. European Society for the Cognitive Sciences of Music, 1997.

F. Lerdhal. Tonal pitch space. *Music Perception*, 5, 1988.

A.F. Melo. A connectionist model of tension in chord progressions. Master's thesis, Department of Artificial Intelligence, University of Edinburgh, 1998.

E. Narmour. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realisation Model*. University of Chicago Press, Chicago, IL, USA, 1990.