A neural model to segment musical pieces

Otávio Augusto Salgado Carpinteiro
School of Cognitive and Computing Sciences
University of Sussex
Falmer, East Sussex, BN1 9QH, England
otavioc@cogs.sussex.ac.uk

Abstract

This paper proposes a knowledge representation of rhythmic patterns, and a neural model to segment musical pieces in accordance with three cases of rhythmic segmentation. The neural model has a topology which is identical to that of NETtalk (Sejnowski & Rosenberg, 1987). It is trained on sets of contrived patterns, and evaluated on two two-part inventions, two three-part inventions, and two fugues of Bach (Bach, 1970, 1989).

1 Introduction

Due to memory constraints, it is believed that listeners do not grasp a musical piece in its entirety, but on the contrary, they segment it into parts which can be analysed, and then later related to each other (Drake & Palmer, 1993). Studies of segmentation of nonmusical sound sequences (Gabrielsson, 1973; Garner & Gottwald, 1968) as well as of musical sequences (Drake & Palmer, 1993; Lerdahl & Jackendoff, 1983; Kirkpatrick, 1984) suggest that the Gestalt principles of proximity and similarity may be the basis on which listeners segment music. Based on such principles, several researchers have proposed three cases of rhythmic segmentation: longer durations (Drake & Palmer, 1993; Lerdahl & Jackendoff, 1983), pauses (Drake & Palmer, 1993; Lerdahl & Jackendoff, 1983), and breaks of similarity (Lerdahl & Jackendoff, 1983; Kirkpatrick, 1984).

A neural model is proposed here to segment musical pieces in accordance with the three cases of segmentation. The topology of the model is identical to that of NETtalk (Sejnowski & Rosenberg, 1987). In the next sections, details are given of the three cases of segmentation. A novel knowledge representation for rhythmic sequences is introduced, along with details of the model. Finally, results of three experiments are presented — one for each case of segmentation.

2 Three cases of rhythmic segmentation

We have considered as an example of *longer durations* the case when given four notes $n_1n_2n_3n_4$, the duration of n_2 is greater than the durations of n_1 and n_3. Again, given four notes $n_1n_2n_3n_4$, we have considered as an example of segmentation given by a *pause* when there is at least one pause between n_2 and n_3. Finally, we have considered as an example of *breaks of similarity* the case when given eight notes $n_1n_2n_3n_4n_5n_6n_7n_8$, the durations of n_1, n_2, n_3, n_4 are identical, the durations of n_5, n_6, n_7, n_8 are also identical, and the durations of the first four notes are different from the last four. Figure 1 illustrates the three cases.

3 Knowledge representation for rhythmic sequences

If we set the small figure in a musical sequence (or in a whole piece) to be the *time interval (TI)*, all other figures become multiples of TI. For example, if TI is an eighth note, a quarter note lasts two TIs, a

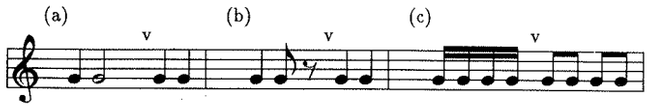

Figure 1: (a) longer durations; (b) pauses; (c) breaks of similarity;

half note lasts four TIs, and so on. We may also define a counter, *time interval counter (TIC)*. One TIC lasts one TI. The TIC is the unit in which the musical sequence is measured. Therefore, at each TIC, either there is a pause, or a note is played, or a note is sounded. Figure 2 illustrates the representation. It shows a sequence which lasts nine TICs, and whose TI is an eighth note. In the experiments, each of the three events was represented by a pair of neural input units. A pause was represented by (00). Note sounded was represented by (10), and note played by (11).

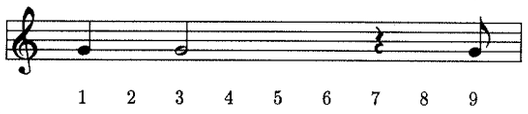

Figure 2: A musical sequence lasting nine TICs

4 The model

The model presented here in brief was developed by Sejnowski and Rosenberg (1987). It is shown in figure 3. The input layer holds a number of pairs of units which make up a window. Each pair represents one of the three events mentioned above (note sounded, note played, pause). The activations of the pairs of units in the window represent a rhythmic pattern. For instance, let TI be an eighth note, and the size of the window be nine pairs of units. Figure 4 would then represent the example in figure 2.

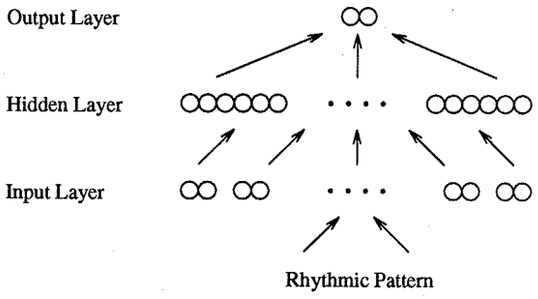

Figure 3: The model

The activation a_i of each hidden unit i is given by the sigmoid function

$$a_i = \frac{1}{1 + e^{-net_i}} \tag{1}$$

net_i is given by

11 10 11 10 10 10 00 00 11

Figure 4: Representation for the sequence in figure 2

$$net_i = \sum_j w_{ij} a_j + bias_i \tag{2}$$

where w_{ij} is the weight from input unit j to hidden unit i, a_j is the activation of input unit j, and $bias_i$ is a special weight which adjusts the values of net_i to make an efficient use of the treshold of the sigmoid.

The activation a_i of each output unit i is given by

$$a_i \doteq net_i = \sum_j w_{ij} a_j + bias_i \tag{3}$$

where w_{ij} is the weight from hidden unit j to output unit i, a_j is the activation of hidden unit j, and $bias_i$ is again a special weight[1].

The weights are updated according to the generalized delta rule (Rumelhart, Hinton, & McClelland, 1986),

$$\Delta w_{ij}(p) = \alpha \delta_i a_j + \beta \Delta w_{ij}(p-1) \tag{4}$$

where $\alpha, \beta \in (0,1)$ are the learning rate and momentum respectively. The subscript p indexes the pattern number, and the learning takes place on a pattern-by-pattern basis. Both the learning rate and momentum are modified at the end of each epoch . The learning rate is reduced when the total error increases, and increased when the error decreases. The momentum is disabled until the end of training if the total error increases. The error signal δ_i, for an output unit i is given by

$$\delta_i = t_i - a_i \tag{5}$$

where t_i is the desired activation value and a_i is the activation obtained. For a hidden unit i, δ_i is given by

$$\delta_i = a_i(1 - a_i) \sum_k \delta_k w_{ki} \tag{6}$$

where w_{ki} is the weight from hidden unit i to output unit k.

We have used two output units in all experiments. We have trained the model to display activation values (10) in these units when the window in the input layer is representing a *negative pattern*, that means, a rhythmic pattern which is not a case of segmentation. We have also trained it to display values (01) when the window is representing a *positive pattern*, a rhythmic pattern which is a case of segmentation.

5 First experiment

The first experiment was on recognizing cases of segmentation given by pauses. We have randomly generated three sets of patterns, each set containing 2000 patterns. The first set was the training set. Every 20 epochs, training was halted, and the model was tested on the second set. When the total error stopped decreasing, training was ended, and the model was tested on the third set. We could thus evaluate different net configurations to find the optimum number of hidden units.

The best performance was given by a configuration with 20 hidden units. The window in the input layer held 10 pairs of units. The initial weights were set randomly, and it was trained for 80 epochs.

A fourth set containing 8 positive and 73 negative patterns was also randomly generated. Principal component analysis (PCA) was performed on the activations of the hidden units given by each pattern in the set. Figures 5 and 6 plot the two first principal components for the negative and positive patterns respectively. We can verify that there is no correlation between the negative and positive patterns, for

[1] As the output units are now linear, the existence of the bias is no longer necessary, although we have decided to keep them.

they are placed in different clusters. Therefore, the internal representations in the hidden units make a distinction between the two types of patterns.

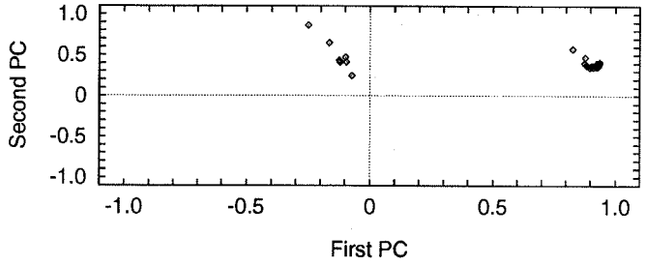

Figure 5: The two first principal components of the negative patterns in the first experiment

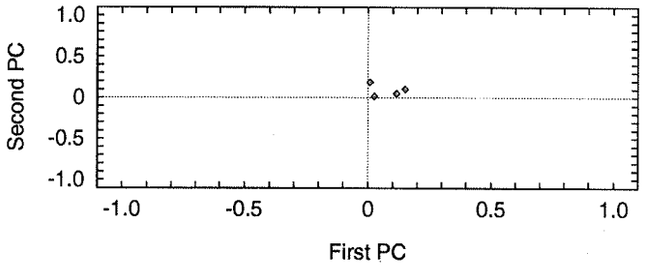

Figure 6: The two first principal components of the positive patterns in the first experiment

The model was evaluated on six musical pieces from Bach. The first two pieces were the nineth and thirteenth two-part inventions in F minor and A minor (Bach, 1970). The third and fourth were the third and fourteenth three-part inventions in D major and B flat major (Bach, 1970). The last two pieces were the fourth and seventeenth fugues in C sharp minor and A flat major of the Well-Tempered Clavier (Bach, 1989). Each part[2] of each piece was input separately. The size of the window was not wide enough to cover all instances of segmentation present in the pieces, yet it was wide enough to cover most of them. The results are displayed in table 1. The percentage of misclassifications is low. We think that these misclassifications can be reduced or even avoided by increasing the number of patterns in the training set.

6 Second experiment

The second experiment was on recognizing cases of segmentation given by longer durations. Again, we have randomly generated three sets of patterns, each set containing 3000 patterns. The training method was identical to that followed in the first experiment.

The best performance was given by a configuration with 10 hidden units. The window in the input layer held 19 pairs of units. Initial weights were set randomly, with training lasting for 240 epochs.

A fourth set containing 110 positive and 440 negative patterns was also randomly generated. Again, PCA was applied to the activations of the hidden units given by each pattern in the set. Figures 7 and 8 plot the two first principal components for the negative and positive patterns respectively. As in the first experiment, we verified that there is no correlation between the negative and positive patterns.

[2] Also known as voice.

Table 1: Results of the first experiment — #NP: number of negative patterns; #PP: number of positive patterns; %NM: percentage of misclassified negative patterns; %PM: percentage of misclassified positive patterns;

Pieces	#NP	#PP	%NM	%PM
1	814	2	0	0
2	784	16	0	0
3	1188	12	0	0
4	2293	10	0	60
5	4556	44	0	14
6	2204	36	0	0

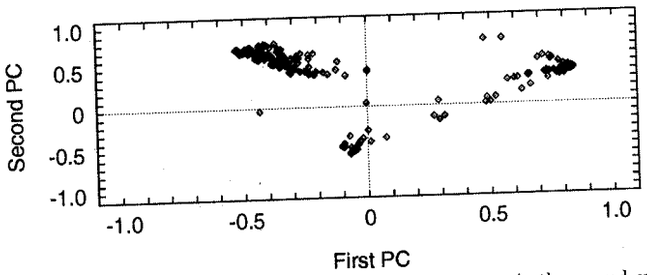

Figure 7: The two first principal components of the negative patterns in the second experiment

The model was evaluated on the same musical pieces as in the first experiment. Each part of each piece was input separately, and the window size was also wide enough to cover most instances of segmentation present in the pieces. As displayed in table 2, there is a low percentage of misclassifications. Once again, we believe that these misclassifications can be reduced by increasing the number of patterns in the training set.

7 Third experiment

The third experiment was on recognizing cases of segmentation given by breaks of similarity. Once again, we have randomly generated three sets of patterns, each set containing 3000 patterns. The training method was identical to that followed in the first two experiments.

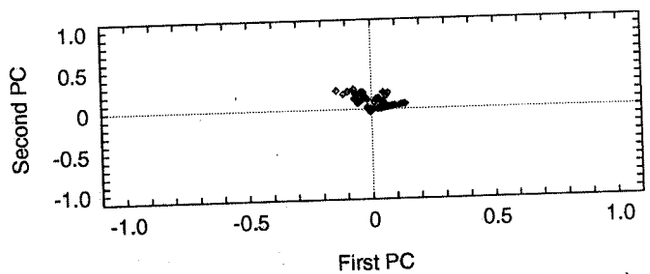

Figure 8: The two first principal components of the positive patterns in the second experiment

Table 2: Results of the second experiment — #NP: number of negative patterns; #PP: number of positive patterns; %NM: percentage of misclassified negative patterns; %PM: percentage of misclassified positive patterns;

Pieces	#NP	#PP	%NM	%PM
1	764	52	2	0
2	790	10	0	0
3	1171	25	10	0
4	2253	39	18	8
5	4483	98	26	15
6	2185	48	22	6

The best performance was given by a configuration with 5 hidden units. The window in the input layer held 16 pairs of units. Random initial weights were trained for 180 epochs.

A fourth set containing 2 positive and 287 negative patterns was also generated randomly. PCA was performed on the activations of the hidden units given by each pattern in the set. Figures 9 and 10 plot the two first principal components for the negative and positive patterns respectively. Unlike the previous experiments, the positive and negative patterns are not completely separated into two different clusters.

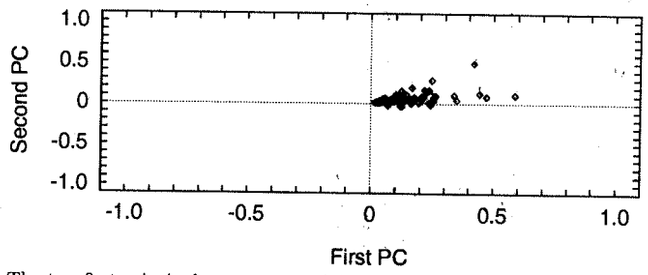

Figure 9: The two first principal components of the negative patterns in the third experiment

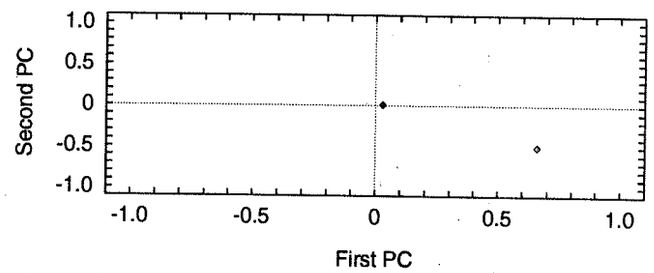

Figure 10: The two first principal components of the positive patterns in the third experiment

The model was evaluated on the same musical pieces as in the former experiments. Each part of each piece was input separately, and the window size was wide enough to cover most instances of segmentation present in the pieces. As displayed in table 3, there is a low percentage of misclassifications. As in the former experiments, we also think that these misclassifications can be reduced by increasing the number of patterns in the training set.

Table 3: Results of the third experiment — #NP: number of negative patterns; #PP: number of positive patterns; %NM: percentage of misclassified negative patterns; %PM: percentage of misclassified positive patterns;

Pieces	#NP	#PP	%NM	%PM
1	813	3	3	0
2	772	28	6	0
3	1196	4	5	0
4	2302	1	0	0
5	4589	8	2	0
6	2233	7	3	0

8 Conclusion

A neural model with an identical topology to that of NETtalk is proposed to segment musical pieces according to three cases of rhythmic segmentation. The model was successfully applied to six musical pieces from Bach. The results presented here suggest that musical segmentation can be accomplished by a neural model with supervised learning.

Acknowledgements

This research was fully supported by CAPES, Brazil.

References

Bach, J. S. (1970). *Inventionen und Sinfonien.* BWV 772–801. Bärenreiter Kassel, Basel, Germany.

Bach, J. S. (1989). *Das Wohltemperierte Klavier.* Vol. 1. BWV 846–869. Bärenreiter Kassel, Basel, Germany.

Drake, C., & Palmer, C. (1993). Accent structures in music performance. *Music Perception, 10*(3), 343–378.

Gabrielsson, A. (1973). Similarity ratings and dimension analyses of auditory rhythm patterns. *Scandinavian Journal of Psychology, 14*, 138–160.

Garner, W. R., & Gottwald, R. L. (1968). The perception and learning of temporal patterns (part 2). *Quaterly Journal of Experimental Psychology, 20*, 97–109.

Kirkpatrick, R. (1984). *Interpreting Bach's Well-Tempered Clavier, A Performer's Discourse of Method.* Yale University Press, London, UK.

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

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). A general framework for parallel distributed processing. In Rumelhart, D. E., McClelland, J. L., & the PDP Research Group (Eds.), *Parallel Distributed Processing*, Vol. 1, chap. 2, pp. 45–76. The MIT Press, Cambridge, MA.

Sejnowski, T. J., & Rosenberg, C. R. (1987). Parallel networks that learn to pronounce english text. *Complex Systems, 1*, 145–168.

Um Modelo Inteligente para Classificação Harmônica Tonal

FÁBIO GHIGNATTI BECKENKAMP*

PAULO MARTINS ENGEL**

CPGCC, INSTITUTO DE INFORMÁTICA, UFRGS
Caixa Postal 15064
91501-970 Porto Alegre - RS- Brazil

ABSTRACT

This work presents an artificial intelligence solution for the harmonic classification problem. The proposed artificial intelligence model divides the harmonic classification problem in subproblems. Intelligent solutions are indicated for each subproblem. The subproblem's solutions interact into the model in the way to find the solution for the harmonic classification problem. The subproblems found are: chord identification, chord classification, chord inversion classification, music tonality classification and harmonic degree's classification. The model indicates connectionist solutions for the chord and tonality classification subproblems, and indicates symbolic solutions for the chord inversions and harmonic degree's classification subproblems. The chord identification problem is partially solved by an algorithm solution. The model was implemented in an appropriately software and hardware that allowed connectionist and symbolic solutions, and the utilization of MIDI interface as music source. The model validation was performed using musical parts from great erudite composers. The model performed an acceptable classification of these music parts showing that cognitive musical problems can be solved by Artificial Intelligence solutions.

1 Introdução

A classificação harmônica consiste em gerar a descrição de uma estrutura audível, formada por um conjunto de tons e suas relações melódicas, rítmicas e métricas que evidenciam uma estrutura de estilo específico. A relação entre acordes e tonalidade é bastante estreita pois é a tonalidade que faz a sonoridade de um acorde com uma função, e ao mesmo tempo, são os acordes, com suas seqüências e relações, que criam a tonalidade.

Dado um conjunto de notas e uma tonalidade específica, pode-se analisar harmonicamente estas notas. Por exemplo: um acorde formado pelas notas dó, mi e sol, possui o grau harmônico I na tonalidade dó maior e possui o grau harmônico V na tonalidade de fá maior. Este é um problema de classificação e a inteligência artificial oferece ferramentas que permitem resolver este tipo de problema como o conexionismo e a abordagem simbólica.

* Student of Master Degree in Computer Science at Universidade Federal do Rio Grande do Sul - UFRGS, E-mail: fgb@inf.ufrgs.br

** Professor at Universidade Federal do Rio Grande do Sul - UFRGS E-mail: engel@inf.ufrgs.br