A Connectionist Model for Chord Classification

FABIO GHIGNATTI BECKENKAMP*
PAULO MARTINS ENGEL**

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

ABSTRACT

This work is a contribution to the unification of the worlds of computer science and music. This is a first step in the direction of opportunities offered by connectionism on music research. This paper studies the application of a connectionist model on an actual problem of music domain and tries to demonstrate the efficiency of using connectionism in the chord classification problem. This work specifies a neural network model based on Backpropagation architecture to recognize four types of chords: major, minor, diminished and augmented. Simulations results have shown that the network recognized the four chord types for twelve major and twelve minor tonalities.

1 Introduction

Music is a human expression that is difficult to study and to model because it evolves great creativity and esthetic concepts. The problems encountered on such study have challenged researchers interested in different aspects, reaching from those that investigate music applying traditional methods to others that apply artificial intelligence (Todd, 1991).

The connectionist paradigm offers a new and unified method of music investigation due to its inherent aspects such as learning, generalization and feature abstraction. Neural networks offer powerful tools to be used by researchers to solve actual music problems such as: human voice and instruments sound perception, performance interpretation, composition process understanding, musical education, etc. Neural networks and music research might provide benefits to musicians by providing new tools to help their work on music.

The chord classification is a relatively easy task for a musician but it is not easy to model the musician knowledge used to do this task. Some efforts to model the chord classification task were done. Pitts and McCulloch in 1947 (Pitts & McCulloch, 1947) created a model based on the cortex tonototopic structure, where equal intervals on the frequency axis were assumed to map into equal cortical distances. Berenice Laden (Laden, 1989) used this interval approach to create two artificial neural models for chord classification. The first model was based on cognitive representation of musical pitch and the second on psychoacustic representation.

2 Goals

This work has two main goals:

- The proposal of a chord classification neural model, and
- the experimentation and validation of neural network applied to music cognition domain.

---

* 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

Chords are more than two simultaneously played notes. The chords considered on this work are triads, i.e., chords of three notes. The proposed neural network must classify four chord types: major, minor, augmented and diminished. Those types were chosen because they are the basic triads for more sophisticated chords that evolve other intervals as seven, ninth, etc.

The chord classification problem consists of presenting a chord to the input of the network. This input will activate the network producing a network output. The output signals indicate what kind of chord was presented. The network used to model this problem was the Backpropagation network.

3 Methodology

Chord classification has been investigated by two points of view: psychoacustic and cognitive. Psychoacustic researchers have been interested in low-level pitch representations like some physic parameters, as frequency. Cognitive researchers may be interested in more abstract levels such as rhythm and tonality. The neural network researchers might be interested in both levels (Laden 1989). In this work only the cognitive approach will be considered.

In this work we implemented the Backpropagation algorithm based on Rumelhart (Rumelhart & McClelland, 1986). The Backpropagation network is used to solve problems that require complex pattern recognition and that need to map nontrivial functions. The proposed neural network shown on figure 1 was modeled with three layers: the input -, the hidden -, and the output layer. The input layer represents the twelve notes on chromatic scale. A hidden layer maps the input patterns into interval patterns. The output layer indicates what kind of chord was presented.

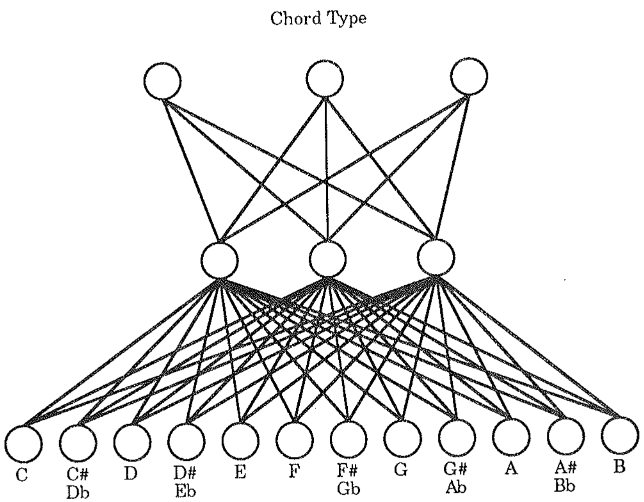

Figure 1 - Neural Network Architecture

The number of neurons on input layer is determined by the pitch representation. The chosen pitch representation is simple: 12 input neurons were stipulated where each neuron is associated to one note of the natural scale. Each input neuron is activated when the respective note is played. The octave of the played note is not considered. All notes are transposed to only one octave. If the same note is played at the same time in different octaves, only one note will be considered. The activation values of the inputs can be 0 or 1, where 0 means a not played note and 1 a played one.

The ideal number of hidden neurons will be determined by inspection during the neural net training. The higher the number of hidden neurons has the neural network, the better will be the convergence possibility. In

this case, a large computational effort will be necessary due to the increase of connections number among network layers.

The output layer has three units that were modeled according to the existent intervals among triad notes. The chord classification can be determined by verification of those intervals. The first output neuron represents the interval between the root and the third of the chord. The second output neuron represents the interval between the third and the fifth of the chord and the third output neuron represents the interval between the root and the fifth of the chord.

In the first and in the second neuron the output must be 1 if the interval is minor and it must be 0 if it is major. In the third neuron, the output must be 0 if the interval is diminished, 0.5 if it is perfect and 1 if it is augmented. The expected output patterns for the four types of chords are presented on table 1. The utilization of the second neuron might be unnecessary because it is not important to know the interval between the third and the fifth of the chord, but it was considered to model all existing chord intervals.

Table 1 - Output Patterns

1ª neuron	2ª neuron	3ª neuron	Chord
0	1	0.5	major
1	0	0.5	minor
1	1	0.0	diminished
0	0	1	augmented

For network training, 48 chords were used; twelve of each type. One example chord is presented to the network at each training cycle. One different chord is selected at each interaction from the chord example set. To assure that the network training will not privilege one chord type, two chords of the same type are never successively presented to the network.

4 Result Analysis

In spite of the fact that the network weights were initialized with random values, each simulation of a BPN can have different performances. This initialization affects directly the number of interactions that the network needs to learn all training examples. Due to this fact a great number of simulations were realized with each network configuration and on next tables only best results are shown.

The learning rate parameter (η), is very important for the training process. With a large η the network training will need a small number of interactions to learn the examples but it can cause the training error to go toward a local minimum. If it happens the network can not find an acceptable solution. On the other hand, a small η ensures that the network will settle on an acceptable solution but it will need more training interactions.

Table 2 shows initial simulation results. These simulations used a small η = 0.25 so that exhaustive training was performed to allow to the network to settle on a solution. Another objective of these simulations was to find an appropriate number of hidden units. The network outputs were observed during the learning phase to verify if they had been modified to settle on a solution.

Table 2 - Short η Training

Network	Number of Hidden Units	η	α	Correct Chords	Interactions
1	36	0.25	0.9	48	41520
2	25	0.25	0.9	48	35760
3	9	0.25	0.9	48	56352
4	3	0.25	0.9	0	300000

The simulations of table 2 correctly classified 100 percent of trained chords but the number of interactions needed for training was larger than expected. The network that had the best performance was number 2, that was trained with 25 hidden units. The network 4, that had only three hidden units, did not recognized any chord after 300000 interactions.

Another way to try to reduce the interaction number is to control the momentum (α). Setting it to 0.2 for the first 40 interactions and to 0.9 for the other ones will reduce the network probability of reaching a local minimum (Franzini, 1988). Table 3 shows a comparative performance between network training using one and two α values. It shows the difference of performance between small and large η too. Network 2 is repeated to an easier comparison.

Table 3 - Two α Training

Network	Number of Hidden Units	η	α	Correct Chords	Interactions
2	25	0.25	0.9	48	35760
5	25	0.25	0.2/0.9	48	37584
6	25	0.9	0.9	48	18096
7	25	0.9	0.2/0.9	48	36336

The use of two α during the training did not increase the network performance as expected. The network 6, which was trained with large η and large α, got the best training result.

The interaction number needed for training is still very large. This fact causes training set adjusts. In the selection of the example set it is important to cover all the problem domain and not to favour one pattern. The number of examples used for training is the same for all patterns but the problem domain is not covered. The input of the network was modeled with 12 binary inputs that means $2^{12} = 4096$ possible input examples. Considering only triads, it is a combination of $C^3_{12} = 220$ possible input triads.

The domain of the problem (48 chords), is smaller than the input domain so that the network is not mapping all possible input patterns. Besides this, it is easy to verify that the 48 chords cover all possible examples for the 4 types of chords. Simulations were realized with less than 48 examples but the network was not able to generalize in the way to classify not trained patterns.

To solve this input modeling problem the network was trained with the 48 known triads plus some examples from the rest possible triads. Those triads were presented to the network as unknown patterns and were called "random triads" because they were randomly searched from the rest of possible input triads. These random triads does not necessarily have musical meaning. During the training, one random triad is searched and presented to the network repeatedly after a certain number of training cicles (n). The trained output pattern for these random triads has zeros for the three output neurons.

Simulations were done to verify if the use of random triads helps to decrease the number of training interactions and to find the best value for n.

Table 4 - Training with Random Chords

Network	Number of Hidden Units	η	α	n	Correct Chords	Interactions
8	25	0.9	0.9	3	48	19536
9	25	0.9	0.9	4	48	21312
10	25	0.9	0.9	8	48	13968
11	25	0.9	0.9	12	48	15360

The results on table 4 show that the use of random examples really decreases the number of network training interactions. The network 10 had the best performance, where 48 chords were correctly recognized after 13968 interactions.

5 Conclusions

This paper demonstrated the use of connectionism in the solution of a musical problem and shows the capacity of using such unconventional musical investigation technique. This work presented a cognitive connectionist model that solved the chord classification problem. It also made investigated network learning parameters, training examples and network architectures that provided an increase of the network performance.

The same model has been tested to classify the 4 triads types plus seven chords. The unique model modification is the increase of one output neuron to indicate the presence of a seven interval in the input chord. The new model has a very similar performance to the older one, indicating that other chord classes may be considered in the future. In further studies this model can be used for more complex chord classification, where tonality and chord inversions may be considered.

Bibliography

Franzini, M. A. Learning to recognize Spoken Words: a study in connectionist speech recognition. In: *Conectionist Models Summer School.* Proceedings... San Mateo: Morgan Kaufmann, 1988, p. 407-416.

Laden, Berenice; Keefe, Douglas H. The Representation of Pitch in a Neural Net of Chord Classification. *Computer Music Journal*, v.13, n. 4, Winter 1989.

Pitts, W.; McCulloch, M.C. How We know Universals: the perception of auditory and visual forms. *Bull. Math. Biophysics*, v.9, p. 127-147, 1947.

Rumelhart, David; McClelland, James. *Parallel Distributed Processing*, MIT Press, Cambridge, MA, 1986, v.1, p. 318-334.

Todd, Peter; LOY D. Gareth. *Music and Connectionism.* MIT Press, Cambridge, MA, 1991, p. ix-xi, 3-19, 31-33, 39-41, 64-85.