

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). *Quarterly 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

2 Objetivos

Este trabalho tem como objetivo principal apresentar a solução encontrada para construir o modelo de inteligência artificial proposto em (Beckenkamp & Engel, 1995). Naquele trabalho, o problema da classificação harmônica é apresentado e a teoria envolvida é detalhada. São apresentadas, ainda, possíveis soluções que possibilitariam a construção do modelo através do uso da inteligência artificial.

O modelo construído deve explorar as técnicas da inteligência artificial, envolvendo dois paradigmas da inteligência artificial: o simbólico e o conexionista. A solução da classificação harmônica foi buscada preferencialmente em sistemas conexionistas com o objetivo de experimentar e validar o uso desta ferramenta no domínio da cognição musical. Porém, isto somente foi feito quando a solução via conexionismo mostrou-se adequada ao problema a ser resolvido.

3 O Modelo Proposto e suas Soluções

O modelo de (Beckenkamp & Engel, 1995) propunha a busca da reconstrução do sistema cognitivo envolvido na classificação harmônica através da decomposição desse problema em subproblemas. Os subproblemas foram modelados de forma que os juntando-se possa reconstruir a inteligência do sistema.

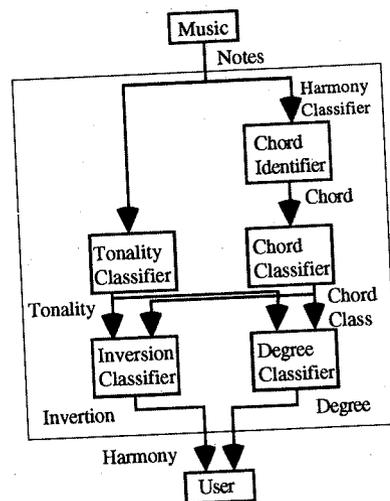


Figura 3.1 Classificação Harmônica

A figura 3.1 acima mostra o modelo proposto para solução do problema da classificação harmônica. A figura mostra o problema, seus subproblemas, bem como o fluxo das informações. Os primeiros subproblemas a serem resolvidos são a identificação e classificação dos acordes. Os acordes devem ser classificados de acordo com sua estrutura intervalar e sua inversão. O modelo deve sugerir uma forma de classificar a tonalidade da música nas 12 tonalidades maiores e nas 12 tonalidades menores o que é fundamental para a classificação dos acordes em graus. Por fim, um classificador de graus deve ser proposto.

No modelo da figura 3.1 acima, a música é fornecida por um usuário, o que pode ser

realizado através de um teclado MIDI ou até mesmo através de um seqüenciador. A música poderá ser fornecida em tempo real ou não, de acordo com a capacidade computacional do ambiente onde for implementado o sistema. As notas da música sendo tocada chegam no classificador de harmonia e são diretamente repassadas ao identificador de acordes e ao classificador de tonalidade. O classificador de tonalidades verifica a tonalidade da música através das notas tocadas. Em qualquer instante de tempo durante a música o classificador deverá ser capaz de fornecer a informação da tonalidade por ele identificada até aquele momento da música. As notas que chegam ao classificador de harmonia também são repassadas diretamente ao identificador de acordes que verifica que conjunto de notas na música formam um acorde. O acorde identificado é passado ao classificador de acordes que fornece em sua saída a classificação encontrada. O classificador de inversões utiliza-se do acorde classificado e da tonalidade da música para classificar as inversões dos acordes. Por fim, o classificador de graus faz uso da tonalidade e dos acordes para classificar em termos de graus harmônicos os acordes anteriormente classificados por tipo no classificador de acordes. Estas informações poderão, então, ser utilizadas pelo usuário.

Existe na literatura alguns trabalhos que buscam soluções para o problema da classificação harmônica e para os subproblemas anteriormente relacionados. Christoph Lischka apresenta um modelo para classificação harmônica funcional através de Máquina de Boltzmann em (De Poli et al, 1991). Uma solução utilizando uma rede neural não-supervisionada (Kohonen Feature Maps - KFM), para a tarefa de classificação de acordes é apresentada por Barucha em (Todd & Loy, 1991). John Maxwell em (Balaban et al, 1992) busca a solução do problema da classificação de acordes fazendo uma análise de intervalos avaliando consonâncias e dissonâncias para determinar o tipo do acorde. Dois modelos conexionistas para a classificação de acordes, um cognitivo e outro psicoacústico, são apresentados por Berenice Ladem et al em (Todd & Loy, 1991). Para a tarefa da classificação da tonalidade da música tem-se o trabalho de Scarborough et al que implementa uma rede neural de perceptrons (Todd & Loy, 1991). Para a classificação tonal tem-se ainda o trabalho de Marc Leman que também utiliza o modelo KFM (Todd & Loy, 1991).

A bibliografia acima citada indica fortemente a possibilidade de solução dos subproblemas da classificação dos acordes e da classificação da tonalidade via modelos conexionistas. Os subproblemas da classificação de inversões e de graus são problemas classificatórios simples que podem ser facilmente resolvidos através da construção de bases de regras. O subproblema da identificação de acordes é menos trivial do que parece inicialmente pois é difícil determinar quando os três tipos de acordes anteriormente citados (bloco, quebrado e arpejado), formam uma estrutura harmônica importante. A seguir serão indicadas maneiras de resolver os subproblemas da classificação harmônica.

3.1 Identificação de acordes

Na literatura sobre harmonia os acordes são as estruturas musicais identificadas para determinar a harmonia. O conceito de acorde diz que três ou mais notas que soam simultaneamente formam um acorde. Baseado na simplicidade do conceito, a identificação do que é um acorde na música seria uma tarefa trivial, porém sabe-se que o que soa simultaneamente pode não ter sido tocado simultaneamente como no caso dos acordes quebrados e arpejados. Isto pode dificultar bastante a identificação dos acordes principalmente se o sistema trabalhar com eventos MIDI, pois uma nota que já foi desligada pode continuar soando. Neste caso, simplesmente a verificação de quais notas estão em *note on* não é suficiente.

Para resolver este problema basta determinar um método de identificar-se, na música, a ocorrência de acordes quebrados e de acordes arpejados. Reconhece-se, no entanto, que isto não é uma tarefa computacionalmente simples pois envolve um problema de explosão combinatória. Para determinar quando notas que são tocadas separadamente formam ou não acordes exige-se combinar todas as notas tocadas num determinado trecho de música a fim de encontrar os possíveis acordes. Um trabalho bastante aprofundado pode ser realizado a fim de determinar que eventos musicais devem ser considerados acordes e quais eventos musicais devem ter importância para a análise harmônica. Este autor não vai apontar nenhuma solução que envolva inteligência artificial para a solução da identificação de acordes, pois este

problema somente foi identificado pouco antes da fase de validação do modelo proposto, não permitindo um estudo mais aprofundado do mesmo. A busca de soluções de Inteligência Artificial para a solução deste problema fica apontado aqui como tema para trabalhos futuros. A solução implementada neste trabalho considera como acordes todo conjunto de três ou mais notas tocadas simultaneamente. Mais detalhes sobre esta solução serão apontados na validação.

3.2 Classificação de Acordes

Um estudo sobre classificação de acordes foi realizado por este autor no trabalho intitulado *Um Modelo Conexionalista para Classificação de Acordes* (Beckenkamp & Engel, 1994). O modelo proposto naquele trabalho utilizava uma rede neural supervisionada e foi utilizada neste trabalho com pequenas modificações que serão detalhadas a seguir.

No caso do uso de redes neurais artificiais, o problema da classificação de acordes consiste em submeter um acorde às entradas da rede neural e esta, através de suas unidades de saída, responder qual é o tipo do acorde. A rede neural proposta neste trabalho deve classificar os 13 tipos de acordes definidos em (Beckenkamp & Engel, 1995), que são: as tríades, os acordes de sétima e os acordes de nona. A escolha destes tipos foi feita levando em consideração a sua vasta utilização na música.

A rede neural utilizada foi a rede *Back-propagation* (BPN) que foi formalizada por Werbos, por Parker, e posteriormente por Rumelhart e McClelland (Rumelhart & McClelland, 1986). A BPN é utilizada na resolução de problemas que requeiram o reconhecimento de padrões complexos, e que necessitem realizar mapeamentos de funções não triviais.

A camada de saída da rede foi aumentada de 3 para 5 unidades. O tipo do acorde é determinado através da verificação dos intervalos formados entre as notas que compõem os acordes. O esquema de codificação dos padrões de saída reflete a estrutura dos intervalos do acorde e foi mantida para as três primeiras unidades de saída. A primeira unidade de saída representa o intervalo entre a fundamental e a terça do acorde (1ª terça), e sua saída deve ser 1 se o intervalo for de terça menor e 0 se for de terça maior. A segunda unidade de saída representa o intervalo entre a terça e a quinta do acorde (2ª terça), sua saída deve ser 1 se o intervalo for de terça menor e 0 se for de terça maior. A terceira unidade de saída representa o intervalo entre a fundamental e a quinta do acorde e sua saída deve ser 1 se o intervalo for de quinta justa e 0 se for de quinta diminuta.

A quarta e quinta unidades foram acrescentadas para modelar os acordes que possuem intervalos de sétima e de nona. A quarta unidade representa o intervalo entre a fundamental e a sétima do acorde. Sua saída será 0 quando o acorde não possuir sétima, 0,3 quando a sétima for diminuta, 0,6 quando for menor, e 1 quando for maior. A quinta unidade de saída representa o intervalo entre a fundamental e a nona do acorde e sua saída deve ser 0 quando o acorde não possuir nona, 0,5 se o intervalo for menor e 1 se for maior.

A rede do trabalho (Beckenkamp & Engel, 1994) classificava apenas tríades, ou seja, acordes formados por 3 notas. Neste trabalho acordes de 4 e 5 notas também foram considerados. Estes acordes são os de sétima (4 notas) e os de nona (5 notas). Desta forma, o número de tipos de acordes que a rede passou a ser treinada aumentou de 4 para 13.

Para o treinamento passou a ser utilizado um conjunto de 156 acordes, doze de cada tipo. O conjunto de 156 acordes foi subdividido em 4 conjuntos de acordo com a estrutura do acorde. O primeiro conjunto contém as tríades, o segundo conjunto contém os acordes de sétima menor, o terceiro os acordes de sétima maior e o quarto os acordes de nona. Esta divisão foi feita para que houvesse um equilíbrio de treinamento entre os acordes que possuem constituição intervalar diferente. A cada ciclo de treinamento um acorde de um grupo diferente é apresentado à rede. Desta forma, por exemplo, nunca dois acordes que possuem intervalo de nona serão apresentados à rede consecutivamente. Internamente os grupos são organizados de forma que dois acordes de mesmo tipo ou de mesma tonalidade não sejam apresentados consecutivamente.

3.3 Classificação da Tonalidade da Música

Sabe-se muito pouco sobre o processo cognitivo humano realizado na identificação da tonalidade.

Mesmo assim, buscou-se modelar o problema a partir do conhecimento cognitivo através de subsídios obtidos do conhecimento teórico de tonalidade. Existem três domínios na organização tonal: notas, acordes e tonalidades. Uma música sempre possui uma ou mais tonalidades. A tonalidade determina um subconjunto de 7 notas que são extraídas das 12 notas da escala cromática e a cada nota dentro de uma escala é associada uma função. Os acordes e notas são ouvidos de acordo com o contexto provido pela estrutura tonal e a tonalidade da música é inferida por um ouvinte através das notas e acordes que ele está ouvindo. A ordem e frequência em que as notas são apresentadas podem influenciar na percepção da tonalidade. Além disso, certas notas da tonalidade são mais importantes para a tonalidade. Deve-se ainda acrescentar que mudanças de tonalidade são rapidamente percebidas por um ouvinte.

O modelo de classificação de tonalidade apresentado por Scarborough em (Todd & Loy, 1991) implementa um modelo bastante simples e eficiente pois programa algumas funções tonais diretamente na rede neural. Este modelo serviu como base na construção do modelo de classificação tonal deste trabalho.

Neste trabalho, implementou-se uma rede neural baseada em perceptrons¹ que mede a frequência com que ocorrem as notas e a importância das notas ocorridas para a percepção da tonalidade. A tonalidade de maior evidência será a que tiver mais notas sendo tocadas e cuja importância for mais significativa. Este processo deve ser dinâmico pois se a tonalidade for modificada durante a música, a rede deverá ser capaz de perceber esta mudança e identificar a nova tonalidade. Esta característica justifica o uso de uma rede neural, pois esta dinâmica é inerente à sua estrutura. A tonalidade com maior ativação será a escolhida, podendo esta ativação mudar de uma tonalidade para outra.

A camada de entrada da rede neural proposta possui 12 neurônios, cada um correspondendo a uma nota da escala cromática. Quando uma nota é tocada, o neurônio correspondente recebe uma ativação. A segunda camada é formada por 24 neurônios correspondendo às doze tríades maiores e às doze tríades menores da música tonal.

As ligações sinápticas entre a camada de entrada e a camada de acordes é determinada pela existência ou não da nota no acorde. Ex.: Os neurônios de entrada referentes às notas Dó-Mi-Sol possuem ligação sináptica válida com o neurônio do acorde Dó maior. Ou seja, o neurônio correspondente a cada acorde receberá ativação somente dos neurônios correspondentes às notas que o formam.

Por último, modelou-se uma camada de 24 neurônios correspondendo às 24 tonalidades da música tonal ocidental, 12 maiores e 12 menores. Os neurônios de tonalidade são ativados pelos neurônios de acordes que são mais frequentes para a tonalidade. Estes acordes são os acordes de tônica, dominante e subdominante. Neste caso, a tonalidade de Dó maior receberá ativação dos neurônios correspondentes aos acordes Dó, Fá e Sol maior.

A distribuição dos pesos na rede foi realizada de forma intuitiva procurando valorizar o que teoricamente é mais importante. A nota fundamental do acorde e o acorde tônico da tonalidade são os mais estáveis de uma tonalidade e indicam a tonalidade da música. As sinapses que ligam as fundamentais e as tônicas da tonalidade receberam peso 2. Ex.: No acorde Dó maior a nota Dó recebeu sinapse com peso 2 e as demais notas, Mi e Sol, receberam peso sináptico igual a 1. Na tonalidade Dó maior o acorde de Dó maior recebeu peso sináptico 2 e os acordes de Fá maior e Sol maior outros pesos sinápticos. As sinapses, tanto de notas quanto de acordes, que não contribuem para a evidência da tonalidade recebem peso 0. Pela experiência, verificou-se que é importante também valorizar as dominantes dos acordes e os acordes de subdominante, para evitar que o sistema não consiga encontrar uma tonalidade no caso de ocorrência de sucessivos acordes de tônica e dominante muito comuns na música. Suas ligações sinápticas foram colocadas em 1,5.

A partir do momento em que uma nota é tocada já se pode verificar o valor das ativações dos neurônios de saída, o neurônio que possui maior ativação determinará a tonalidade. Em qualquer ponto da música a tonalidade pode ser verificada através do neurônio de tonalidade que possui maior ativação, o que faz com que o modelo suporte modulações.

A rede neural percebe a tonalidade a partir das notas que são tocadas. A medida em que a música vai sendo tocada mais notas da mesma tonalidade são tocadas, aumentando a evidência da tonalidade. Porém a música pode possuir modulações, o que determina que notas de outra tonalidade aparecerão e deverão modificar a evidência da tonalidade anterior para a nova tonalidade. Para que a rede neural perceba a

¹ Mais informações sobre perceptrons pode ser obtidas no livro (Freeman & Skapura, 1992).

tonalidade a medida em que as notas vão acontecendo, é preciso que se crie uma forma de memória das notas que foram tocadas. No entanto, esta memória deve ter menos importância para o reconhecimento da tonalidade do que as notas mais recentes de forma que a rede possa rapidamente perceber modulações.

Este sistema de memória do que já foi tocado pode ser realizado através do controle das ativações que as notas geram quando são tocadas. Neste momento, o neurônio correspondente recebe a ativação máxima e permanece com ativação durante algum tempo após ser desacionada. A permanência da ativação após a nota ter sido desligada faz com que as notas que já foram tocadas continuem sendo importantes na escolha da tonalidade. Caso não fosse implementado um sistema de memória, possivelmente uma seqüência de notas resultaria numa seqüência de tonalidades. Este trabalho sugere três maneiras de controlar a ativação dos neurônios de entrada: linear, com decaimento e com ataque e decaimento.

No controle linear, quando uma nota é tocada, o seu neurônio correspondente recebe a ativação máxima (igual a 1). A ativação permanece a mesma durante o tempo em que a nota estiver sendo tocada. Depois que a nota é desligada a ativação também permanece igual durante um tempo que pode ser determinado pelo usuário. Após o tempo estipulado, a ativação cai para zero.

No controle com decaimento, o neurônio recebe ativação máxima durante o tempo em que a nota for tocada. Quando a nota é desacionada, a sua ativação não é imediatamente levada a zero, ela entra em processo de decaimento ao longo do tempo.

No controle com ataque e decaimento a duração da nota influencia na quantidade de ativação aplicada ao neurônio. À medida em que a nota vai sendo sustentada, aumenta o valor da ativação por ela gerada (ataque). Isto faz com que as notas tocadas durante mais tempo influenciem mais na ativação final.

Com a implementação do ataque e do decaimento, as notas mais recentes terão maior influência na ativação da rede, fazendo com que haja a possibilidade de modulações durante a música. Este mecanismo implementa automaticamente um controle de grau de certeza de qual é a tonalidade da música. Quanto mais tempo notas da mesma tonalidade permanecerem sendo ativadas, maior será o grau de certeza de que a tonalidade da música é aquela. Para implementar o ataque e o decaimento basta implementar funções exponenciais na ativação de cada nota.

3.4 Inversões de Acordes

A solução encontrada para resolver o problema das inversões de acordes foi construir um sistema de regras. Para os acordes não simétricos basta verificar a distância do baixo do acorde com as demais notas do acorde. As inversões dos acordes são determinadas por qual nota formadora do acorde é o baixo. Dependendo da nota do acorde que é o baixo, são formados intervalos diferentes entre as notas do acorde. Por exemplo: o acorde dó-mi-sol possui um intervalo de terça maior entre o baixo, a nota dó, e a segunda nota do acorde, a nota mi. Além disso possui um intervalo de quinta justa entre o baixo (dó), e a terceira nota (sol). Caso este acorde estivesse na primeira inversão teria a seguinte estrutura: mi-sol-dó. A distância entre o baixo, agora a nota mi, e a segunda nota do acorde (sol), é de uma terça menor e o intervalo entre o baixo (mi), e a terceira nota (dó), é de uma sexta.

Para ter-se um controle mais preciso destes intervalos deve-se verificar os intervalos na forma de distância em semitons. Isto é necessário pois cada intervalo pode assumir vários tipos (maior, menor, justo, diminuto e aumentado). Deve-se ter cuidado para verificar com exatidão o tipo de intervalo formado para cada tipo de acorde em cada inversão. Por exemplo verificar que tipo de intervalo de sexta tem-se no acorde mi-sol-dó, do exemplo anterior. A Tabela 3.4.1 a seguir, apresenta as distâncias em semitons existentes entre o baixo e as demais notas que formam um acorde do tipo perfeito maior nas três posições possíveis. Os números que aparecem em negrito determinam quais os intervalos que devem ser levados em consideração para que se possa classificar as inversões do acorde.

No modelo proposto neste trabalho o classificador de inversões deve classificar as inversões dos acordes previamente selecionados e classificados pelo classificador de acordes. Tendo-se a informação do tipo do acorde, pode-se verificar os intervalos em semitons formadores do acorde. Para cada tipo de acorde pode-se construir um conjunto de regras que classificam a inversão do acorde. Estas regras podem ser construídas a partir de tabelas como a Tabela 3.4.1 abaixo.

Tabela 3.4.1 Acorde PM

Inversão	1º Intervalo	2º Intervalo
Fundamental	4	7
1ª Inversão	3	8
2ª Inversão	5	9

Analisando-se a Tabela 3.4.1, verifica-se que para acordes perfeitos maiores basta identificar o tipo do intervalo formado entre o baixo e a 1ª nota do acorde. Caso o intervalo seja de 4 semitons sabe-se que o acorde está em posição fundamental, 3 semitons em 1ª inversão e 5 semitons em 2ª inversão. Ou seja, o intervalo entre o baixo e a segunda nota pode ser desprezado. Porém isto não ocorre para todos os tipos de acordes. Na maioria dos casos deve-se verificar mais de um intervalo para que se possa distinguir as inversões. Regras de intervalos para todos os tipos de acorde considerados neste trabalho podem ser facilmente formadas. Um mecanismo de aprendizado pode ser implementado afim de permitir que o conhecimento seja incrementado através da evolução das regras.

Através deste método também não se resolve o problema dos acordes simétricos, pois os intervalos formados entre o baixo e as demais notas do acorde são sempre os mesmos para quaisquer inversões. A solução encontrada foi construir algumas regras levando em consideração o grau em que os acordes simétricos podem ocorrer na harmonia tonal. Estes acordes acontecem somente na tonalidade menor e somente em um grau da escala. O acorde de 5ª aumentada ocorre somente para o terceiro grau da escala menor e o acorde de sétima diminuta somente no sétimo grau da escala menor. Tendo-se a informação de qual escala menor a música está sendo tocada, quando ocorrer um dos acordes simétricos, pode-se verificar facilmente qual é o terceiro ou o sétimo grau da escala. Por exemplo: caso a música esteja em lá menor e acontecer um acorde de 5ª aumentada que este acorde terá grau III. Sendo assim, conclui-se que a nota dó, que é o grau 3 da escala de lá menor, é a nota fundamental do acorde tocado. Verificando-se a distância entre o baixo e a fundamental do acorde, no exemplo a nota dó, pode-se concluir qual a inversão dos acordes simétricos.

3.5 Classificação dos graus

Na literatura sobre harmonia encontram-se discordâncias entre os autores com relação aos graus em que ocorrem cada tipo de acorde. Apesar disto, a classificação dos graus dentro do modelo proposto neste trabalho tornou-se uma tarefa bastante simples, não justificando a busca de soluções complexas como o uso de redes neurais. Neste trabalho buscou-se unificar as informações fornecidas nas várias literaturas consultadas criando-se regras que mapeiam em que graus de cada escala ocorrem os tipos de acordes considerados neste trabalho. Um mecanismo de aprendizado também pode ser implementado para a classificação de graus, permitindo que as regras que definem em que graus cada acorde ocorre possam evoluir com o uso do sistema.

Alguns acordes podem ser mapeados diretamente para um sistema de regras, pois eles ocorrem somente em um grau de determinada tonalidade, como, por exemplo, o acorde de 7ª diminuta que sempre será o grau VII. Porém para aqueles acordes que podem ocorrer em vários graus de uma mesma tonalidade deve-se construir um sistema de regras que leve em consideração informações complementares que possam eliminar a ambigüidade. Por exemplo, o acorde PM pode ser tanto o grau I como o grau IV ou ainda o grau V em tonalidades maiores. Para eliminar-se esta ambigüidade através de regras, pode-se verificar as notas do acorde que somente acontecem em um determinado grau. Por exemplo, se a tonalidade for dó maior e o acorde for dó-mi-sol (PM) sabe-se que este acorde é o grau I, pois a nota mi está presente e ela não ocorre nos acordes de IV e V graus da tonalidade de dó maior (fá-lá-dó e sol-si-ré respectivamente). Pode-se generalizar esta regra para todos os acordes PM em tonalidade maior se considerarmos que a nota mi é o terceiro grau da escala de dó maior. Sabendo-se a tonalidade da música, pode-se saber que nota é cada grau desta escala e com posse das notas que formam o acorde a ser classificado, pode-se verificar se

determinado grau da escala pertence ao acorde.

As regras criadas mapeando em que graus de cada escala ocorrem os acordes podem ser facilmente transformadas num sistema de regras que resolverá a classificação dos graus para as escalas maiores e menores de uma maneira bastante simples. No caso das escalas menores o sistema de regras ficará mais complicado, pois têm-se três tipos de escalas menores, o que aumenta a ambigüidade da classificação. A diferença entre estas três escalas está no sexto e sétimo graus que podem aparecer alterados. Estas alterações determinam que outros tipos de acordes podem ser contruídos nestas escalas. A presença destes graus alterados pode ser utilizada como diferenciadora na classificação dos graus.

4 Validação

A implementação do modelo proposto exigiu a escolha de uma plataforma de hardware e software que permita realizar diferentes tarefas que possuem diferentes necessidades tanto em hardware quanto em software. O hardware deve ter boa velocidade de processamento numérico a fim de facilitar o treinamento das redes neurais. Além disso, o hardware deve possuir interface MIDI o que permite a entrada da música a ser analisada. O software deve ser capaz de trabalhar com MIDI e oferecer ferramentas para a implementação de sistemas de inteligência artificial que envolvam tanto o paradigma conexionista, quanto o simbólico.

Para o desenvolvimento de programas de redes neurais a linguagem C foi escolhida, pois este tipo de programa exige uma linguagem que aproveite ao máximo a velocidade de processamento do hardware disponível. Devido à necessidade de um ambiente de software que facilite a manipulação de MIDI e que também permita o desenvolvimento de programas em linguagem C, o software Max foi escolhido. Além disso, a linguagem Max também permite a criação de programas de Inteligência Artificial baseados em regras, pois possui uma ferramenta chamada Pyrite que implementa uma linguagem baseada em Lisp.

Os subproblemas foram desenvolvidos na forma de objetos Max. Programas Max foram desenvolvidos para testá-los separadamente a fim de se verificar o seu desempenho para a realização do subproblema a que foi destinado. Da mesma forma, foi criado um programa Max para avaliar o desempenho do modelo proposto, onde os objetos Max que implementam os subproblemas, interagiram a fim de realizar a classificação harmônica. Abaixo os resultados dos testes das implementações dos subproblemas individualmente e, por último, o teste do modelo proposto.

Tabela 4.1 Resultados do modelo de Laden

Rede	Unidades Intermediárias	Arquitetura	η	α	Codificação da Saída	Porcentual de acerto	Iterações
6	3	Adjacente ²	0,6	0,6	Intervalar	44	10000
7	25	Adjacente	0,8	0,8	Intervalar	94	3100
8	25	Adjacente	0,9	0,2/0,9	Intervalar	58	5000

O subproblema da classificação de graus depende muito fortemente dos dados fornecidos pelos outros subproblemas. Em consequência disto, torna-se difícil testá-lo individualmente. Testes foram realizados somente com o intuito de verificar se a programação das regras foram corretamente implementadas. O subproblema da classificação de inversão de acordes também requereu apenas um teste, a fim de verificar a correta implementação das regras.

Já o classificador de acordes mereceu atenção especial em sua validação. Para ter-se uma noção correta do desempenho do classificador de acordes, foram comparados os resultados da rede neural proposta com os melhores resultados obtidos pelo modelo conexionista cognitivo apresentado por Berenice Laden em (Todd & Loy, 1991), que foram transpostos para a Tabela 4.1 acima. Na Tabela 4.2 abaixo é mostrado o melhor resultado obtido no treinamento da rede proposta.

² A arquitetura chamada de adjacente por Laden é mais conhecida por *feed-forward*.

Tabela 4.2 Resultados do modelo proposto

Rede	Unidades Intermediárias	Arquitetura	η	α	Codificação da Saída	Porcentual de acerto	Iterações
1	25	Adjacente	0,9	0,25	Intervalar	100	146133

A rede número 1 da Tabela 4.2 apresentou um resultado plenamente satisfatório bastante superior aos resultados encontrados na bibliografia pesquisada. A taxa de aprendizado (η) utilizada para o treinamento foi de 0,25 e o momentum (α) foi de 0,9.

O sucesso alcançado no treinamento do modelo provou que a classificação de acordes via modelo conexionista cognitivo por intervalos é viável. O modelo proposto é capaz de classificar todos os intervalos possíveis para acordes da música tonal considerados no estudo de harmonia.

Para validar o classificador de tonalidade e o classificador de harmonia, foram selecionados no livro (Kostka & Payne, 1984), trechos de música previamente classificada por seus autores. Desta forma, basta confrontar os resultados do sistema com a classificação dada pelos autores. Os trechos de música escolhidos abrangem uma boa parcela da teoria envolvida na classificação harmônica considerada por este trabalho. São eles: Exemplo 7-7, Exemplo 7-10 e Exemplo 7-11. Abaixo, a Figura 4.1 mostra o trecho 7-7 com as classificações do livro.



Figura 4.1 Beethoven, Minuet

Para que o sistema classifique os trechos de música é necessário que estes sejam fornecidos ao sistema na forma de eventos MIDI. Os trechos de música podem ser tocados no controlador MIDI A-30 que é um gerador de eventos MIDI. Os eventos gerados pelo A-30 são enviados para o módulo de som SC-55 que os repassa para o macintosh. Este procedimento de teste é cansativo pois toda vez que se quiser testar o programa, deve-se tocar o trecho de música. Uma alternativa é seqüenciar o trecho de música num software apropriado. O seqüenciador deve ser conectado ao programa de classificação de harmonia de maneira que os eventos MIDI previamente seqüenciados possam ser avaliados em tempo real, quantas vezes se deseje e sem a necessidade de tocá-los.

Os trechos de música foram seqüenciados no Cakewalk para Windows. O PC 486 Dx4 do laboratório de música e computação possui uma placa MIDI. Através da saída MIDI da placa os eventos seqüenciados são passados para a entrada MIDI do módulo de som SC-55, que por sua vez está conectado ao computador macintosh. Como já foi explicado anteriormente, estes dados MIDI chegam no macintosh e são interceptados pelo programa Max. O programa desenvolvido na linguagem Max pode então manipular os eventos MIDI livremente.

A classificação da tonalidade de todos os trechos de música testados foi correta. Os controladores de ataque e decaimento foram utilizados de várias maneiras para verificar o melhor resultado para cada trecho analisado. As características de ritmo e andamento de cada trecho de música determinam que parâmetros são os mais adequados para a classificação da tonalidade. Com alguma experiência, facilmente se identifica quais parâmetros são os mais adequados para cada música. Normalmente o classificador de tonalidade demora aproximadamente um compasso para começar a classificar corretamente a tonalidade. O tempo em que o classificador leva para perceber a modulação também fica em torno de um compasso.

A eficiência do classificador de tonalidade depende sensivelmente dos pesos da rede e dos controles de

ataque e decaimento. O modelo percebe corretamente a tonalidade em vários contextos musicais como escalas, acordes, cadências, arpejos, etc; e classifica rapidamente, com grau de acerto aceitável e consistente com relação a modulações. A percepção de algumas regras musicais que também dão pistas para a tonalidade poderiam ser consideradas pelo modelo. Ex.: intervalos de segunda menor ascendente resolvem na fundamental.

O objeto classificador de harmonia classificou satisfatoriamente os três trechos de música selecionados. Os dois primeiros trechos testados foram corretamente classificados e apenas dois erros foram cometidos no terceiro trecho devido a aparição de acordes incompletos que não formam classificados corretamente. No trecho da Figura 4.1, os dois primeiros compassos foram classificados com grau I. No compasso 3, o primeiro acorde é classificado como grau IV. Porém, assim que ocorre a nota fá, a classificação muda para o grau ii na segunda inversão, como indicado no livro. O mesmo ocorre no quinto compasso, onde aparece um acorde quebrado. Somente quando todas as notas formadoras do acorde são tocadas é que o sistema chega à classificação harmônica definitiva, que é o grau V. Como se pode verificar, o sistema classificou corretamente acordes importantes, como os acordes de grau I, V e ii.

Problemas como os acima citados já foram exaustivamente analisados pelos estudos de análise harmônica e por certo podem ser facilmente isolados. O modelo deste trabalho possui flexibilidade suficiente para poder incorporar soluções para estes problemas quer seja via redes neurais, como via soluções híbridas.

5 Conclusão

Os resultados obtidos são positivos e motivadores, porque mostram que o problema da classificação harmônica tonal pode ser resolvido via modelos de inteligência artificial. A solução da classificação harmônica tonal foi buscada através de uma modelagem que agrupasse soluções para determinadas partes do problema. O paradigma conexionista apresentou-se como solução natural e adequada para problemas cognitivos devido as suas propriedades de tolerância a falhas e generalização. A solução dos problemas de inversões de acordes e classificação de graus através de bases de regras, mostrou-se bem dimensionada diante da complexidade destes problemas. O modelo criado proporcionou uma solução rápida e eficiente para a classificação harmônica.

A implementação do modelo foi possível devido à acertada opção pelo seu desenvolvimento através da linguagem Max que permitiu o desenvolvimento dos programas de redes neurais e bases de regras, facilitando também o trabalho com música na forma de eventos MIDI. O uso de um computador Macintosh Quadra é necessário para que o classificador de harmonia tenha um desempenho que permita realizar sua tarefa em tempo real. Os resultados obtidos e a flexibilidade de programação da linguagem Max permitem que os objetos criados para a solução dos subproblemas da classificação harmônica e o próprio programa de classificação harmônica possam ser utilizados em novos projetos e programas que busquem resolver outros problemas da música.

O modelo desenvolvido neste trabalho fornece resultados analíticos, pois realiza automaticamente a classificação da harmonia tonal. Estes resultados podem ser utilizados em sistemas que requeiram o conhecimento da harmonia, como sistemas de acompanhamento ou composição automática. Estes sistemas auxiliam o músico no aperfeiçoamento de seus conhecimentos e de seu trabalho. Eles também possibilitam a criação de interfaces inteligentes entre o computador e o músico, permitem novas formas de estudo, organização e utilização do material musical, e reduzem a complexidade do estudo e do aprendizado da música.

É importante salientar que o estudo do problema não encerra aqui, pois o modelo criado implementa parte do conhecimento sobre harmonia, podendo evoluir através da inclusão de mais conhecimento, a fim de aumentar a precisão e rapidez da classificação harmônica. Este conhecimento pode continuar a ser buscado na teoria e nos especialistas, mas também pode ser buscado através de soluções que modelem partes do conhecimento, como este trabalho. O modelo proposto mostrou ser abrangente e conciso, formando um alicerce bastante forte para trabalhos futuros.

A teoria que envolve a análise harmônica certamente é bastante ampla, não podendo ser totalmente considerada neste trabalho. No entanto, ele aponta caminhos que ainda não foram totalmente explorados e que, certamente, podem ajudar a entender e a resolver muitos dos desafios da análise harmônica. Espera-se que estes caminhos possam continuar a ser trilhados tanto por pesquisadores da música, como da ciência da

computação.

Bibliografia

- Balaban, Mira; Ebcioğlu, Kemal; Laske Otto (1992). *Understanding Music and IA: perspectives on music cognition*. MIT Press, Cambridge, MA, 1992.
- Beckenkamp, Fábio Ghignatti; Engel, Paulo Martins (1994). A Connectionist Model for Chord Classification. *Proceedings of I Simpósio Brasileiro de Computação e Música*. Caxambu, MG, Brazil, August 1994.
- Beckenkamp, Fábio Ghignatti; Engel, Paulo Martins (1995). A Proposal of a Harmonic Classification Artificial Intelligence Model. *Proceedings of International Joint Conference on Artificial Intelligence 1995*. Montréal, Québec, Canada, August 1995.
- De Poli, Geovanni; Piccialli, Aldo and Roads, Curtis (1991). *Representations of Music Signals*. MIT Press, London, England.
- Freeman, J.; Skapura (1992), D. *Neural Networks: Algorithms, Applications, and Programming Technics*. Addison-Wesley Publishing Company, p. 89-124.
- Kostka, Stefan; Payne, Dorothy (1984). *Tonal Harmony*, with an introduction to the twentieth-century music. Alfred A. Knopf Inc, New York, NY, v. 1, p. vii-x, 17-21, 39-42, 44-46, 54-57, 60-62, 241, 421-422.
- Rumelhart, David; McClelland, James (1986). *Parallel Distributed Processing*. MIT Press, Cambridge, MA, v.1, p. 318-334.
- Todd, Peter; Loy D. Gareth (1991). *Music and Connectionism..* MIT Press, Cambridge, MA.