**UNIVERSIDADE FEDERAL DE PERNAMBUCO**

**CENTRO DE INFORMÁTICA**

**CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO**

**Livisghton Kleber da Silva Araújo**

**Análise do Uso de Chromagrams na Classificação Automática de Acordes Musicais**

**RECIFE**

**2019**

**UNIVERSIDADE FEDERAL DE PERNAMBUCO**

**CENTRO DE INFORMÁTICA**

**CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO**

**Livisghton Kleber da Silva Araújo**

**Detecção de acordes em áudio utilizando Redes Neurais**

Monografia apresentada ao Centro de Informática (CIN) da Universidade Federal de Pernambuco (UFPE), como requisito parcial para conclusão do Curso de Engenharia da Computação, orientada pelo professor Carlos Alexandre Barros de Mello.

**RECIFE**

**2019**

**UNIVERSIDADE FEDERAL DE PERNAMBUCO**

**CENTRO DE INFORMÁTICA**

**CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO**

**Livisghton Kleber da Silva Araújo**

**Detecção de acordes em áudio utilizando Redes Neurais**

Monografia submetida ao corpo docente da Universidade Federal de Pernambuco, defendida e aprovada em 02 de dezembro de 2019.

Banca Examinadora:

Orientador

Carlos Alexandre Barros de Mello

Doutor

Examinador

Tsang Ing Ren

Doutor

Dedicatória (Opcional).

**AGRADECIMENTOS**

sdsfdfdfdffgfgfg

“A visão é o mais avançado dos nossos sentidos, de forma que não é de surpreender que as imagens exerçam o papel mais importante na percepção humana.”

Rafael C. Gonzalez

**RESUMO**

gfgfgfgfgf.

**Palavras-chave**: 3 a 5 palavras

**ABSTRACT**

Resumo em ingles.

**Keywords**: 3 to 5 keywords.

Sumário

[1. Introdução 15](#_Toc23798613)

[1.1. Objetivos 15](#_Toc23798614)

[1.2. Objetivos Específicos 16](#_Toc23798615)

[2. Fundamentação teórica 17](#_Toc23798616)

[2.1. Conceitos Musicais 17](#_Toc23798617)

[2.2. Conceitos de Processamento de Sinais 19](#_Toc23798618)

[2.3. Conceitos de Chromagram 20](#_Toc23798619)

[2.4. Conceitos de Redes Neurais 22](#_Toc23798620)

[3. Trabalhos Relacionados (Estado da Arte) 25](#_Toc23798621)

[3.1. Algoritmo de Mello 25](#_Toc23798622)

[3.2. Algoritmo de Mello 25](#_Toc23798623)

[4 Metologia do Estudo 26](#_Toc23798624)

[4.1 Visão Geral 26](#_Toc23798625)

[4.2 Passo 1 26](#_Toc23798626)

[4.3 Passo 2 26](#_Toc23798627)

[5 Experimentos e Análise 27](#_Toc23798628)

[5.1 Experimento 1 27](#_Toc23798629)

[5.2 Experimento 2 27](#_Toc23798630)

[5.3 Análise 27](#_Toc23798631)

[6 Conclusões e Trabalhos Futuros 28](#_Toc23798632)

[6.1 Contribuições 28](#_Toc23798633)

[6.2 Trabalhos Futuros 28](#_Toc23798634)

[Referências 29](#_Toc23798635)

**Lista de Figuras**

[Figura 1: Escala cromática crescente, ilustrando as notas (DÓ, DÓ#, RÉ, RÉ#, MI, FÁ, FÁ#, SOL, SOL#, LÁ, LÁ#, SÍ), nesta ordem. Fonte: próprio autor. 17](#_Toc23798636)

[Figura 2: Escala cromática decrescente, ilustrando as notas (SÍ, SIb, LÁ, LAb, SOL, SOLb, FÁ, FÁb, MI, MIb, RÉ, RÉb, DÓ), nesta ordem. Fonte: próprio autor. 17](#_Toc23798637)

[Figura 3: Representação de acordes em partituras 18](#_Toc23798638)

[Figura 4: Representação de acordes na música *Let It Be* dos Beatles. Imagem capturada do site cifra Club. 18](#_Toc23798639)

[Figura 5: Variações de acordes de tétrades. A letra T na última coluna representa a tónica do acorde, ou seja, a nota de 1º grau.Fonte: http://aguitarra.com.br 19](#_Toc23798640)

[Figura 6: Passos para construção de um chromagram 21](#_Toc23798641)

[Figura 8: Representação gráfica de uma estrutura do perceptron. 23](#_Toc23798642)

[Figura 9: Arquitetura de uma Rede Neural MLP. Fonte: encurtador.com.br/lmpY1 24](#_Toc23798643)

**LISTA DE TABELAS**

**Nenhuma entrada de índice de ilustrações foi encontrada.**

**TABELA DE SIGLAS**

|  |  |  |
| --- | --- | --- |
| **Sigla** | **Significado** | **Página** |
| C | Dó | 19 |
| D | Ré | 23 |
| E | Mi | 37 |
| F | Fá | 37 |
| G | Sol |  |
| A | Lá |  |
| B | Sí |  |
| M ou maj | Acorde maior |  |
| m | Acorde menor |  |
| dim | Acorde diminuto |  |
| # | Sustenido |  |
| B | Bemol |  |
| Hz | Hertz |  |

# Introdução

A cifra é uma notação musical utilizada para indicar os acordes que serão executados por um instrumento (como piano, violão ou ukulele, por exemplo). Ela é bastante utilizada acima das letras, tablaturas ou partituras, indicando o acorde a ser tocado em conjunto a melodia (1). É bastante comum que músicos amadores ou não profissionais não possuam a habilidade de distinguir os acordes pelos seus próprios ouvidos, fazendo-os recorrerem a sites de cifras como: cifra Club (2), cifras (3) ou pega cifra (4) por exemplo. No entanto, existem várias músicas que não estão cadastradas na base de dados dessas ferramentas, dificultando assim, o aprendizado dessas canções por parte dos músicos menos experientes.

Segundo VELOSO e FEITOSA (5), “...a capacidade de identificar, nomear ou produzir a frequência de um estímulo tonal sem o auxílio de um tom de referência”, caracterizam músicos que possuem “ouvido absoluto”. Similarmente, SANTOS JUNIOR (6) fala sobre “ouvido relativo” em músicos que consiste em; “...não se limita a identificar as notas, mas também as características intervalares que permitem entender a música e as relações entre as notas, acordes e melodias”. Em virtude disto, esses tipos de músicos não apresentam problemas ou dificuldades em identificar notas musicais e reproduzi-las. Por outro lado, a ocorrência de ter um ouvido absoluto é rara, de acordo com VELOSO e FEITOSA, 1 para cada 10.000 pessoas, considerando-se a população da Europa e América do Norte, sendo mais comumente observada em músicos (cerca de 20%) (5). Já para possuir um ouvido relativo requer treino e isso leva tempo para aperfeiçoar.

Com base nisto, este trabalho tem como objetivo construir um modelo computacional que seja capaz de estimar acordes musicais dado um arquivo de música. Para isso, foram estudados modelos baseados em redes neurais artificiais e técnicas de processamento de áudio como o chroma feature para estimar os acordes em músicas. Com isso, espera-se colaborar no auxílio do aprendizado de músicos iniciantes e amadores que não apresentam nenhuma das características mencionadas anteriormente.

## Objetivos

O objetivo deste trabalho, é criar um modelo computacional que seja capaz de estimar acordes musicais dado um arquivo de música. Para isso, foram estudados modelos baseados em redes neurais artificiais e técnicas de processamento de áudio como o chroma feature realçar as características dos acordes dentro de uma composição musical. Com isso, espera-se colaborar no auxílio do aprendizado de músicos iniciantes e amadores que não tem a habilidade de identificar os acordes musicais.

## Objetivos Específicos

* Aplicar a técnica de chroma feature sobre os arquivos de áudio.
* Realizar o Treinamento da máquina utilizando dois algoritmos de aprendizagem: *Multilayer Perceptrons* (MLP).
* Analisar os resultados por meio de tabelas.

# Fundamentação teórica

Neste capítulo, são introduzidos alguns conceitos básicos necessários para o entendimento deste trabalho. Inicialmente, são explicados conceitos básicos de notas e acordes musicais. Em seguida, são apresentadas teorias computacionais de processamento de sinais e redes neurais.

## Conceitos Musicais

O elemento mínimo sonoro dentro de uma composição musical é chamado de nota musical. Este elemento apresenta um único modo de vibração do ar e está relacionado a uma frequência (medida em Hz), onde descreve em termos físicos se ela é mais grave ou mais aguda [1].

O nosso sistema musical, ocidental, é composto por doze notas musicais diferentes, sendo sete as notas naturais (Dó, Ré, Mi, Fá, Sol, Lá e Si) e cinco as acidentadas que, dependendo da escala, podem ser (Dó#, Ré#, Fá#, Sol# e Lá#), representando as notas sustenidas ou (Réb, Mib, Solb, Láb e Sib), representando as notas bemóis. Estas doze notas podem ser vistas graficamente nas Figuras 1 e 2.



Figura 1: Escala cromática crescente, ilustrando as notas (DÓ, DÓ#, RÉ, RÉ#, MI, FÁ, FÁ#, SOL, SOL#, LÁ, LÁ#, SÍ), nesta ordem. Fonte: próprio autor.

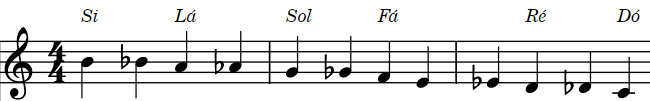


Figura 2: Escala cromática decrescente, ilustrando as notas (SÍ, SIb, LÁ, LAb, SOL, SOLb, FÁ, FÁb, MI, MIb, RÉ, RÉb, DÓ), nesta ordem. Fonte: próprio autor.

Já um acorde pode ser formado por meio de uma combinação de, pelo menos, três ou mais notas tocadas simultaneamente. A representação gráfica de um acorde em uma partitura ou cifra, pode ser vista nas Figuras 3 e 4, respectivamente.

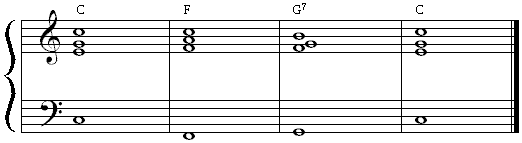


Figura 3: Representação de acordes em partituras

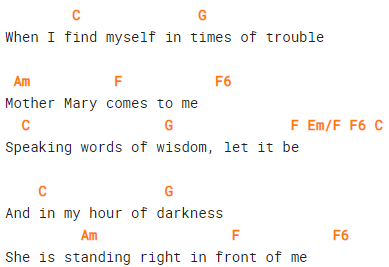


Figura 4: Representação de acordes na música *Let It Be* dos Beatles. Imagem capturada do site cifra Club.

Os acordes formados por três notas são chamados de tríades, onde, na maioria das vezes, são constituídos pelas notas de 1º, 3º e 5º grau. Por exemplo: um acorde de Dó maior é formado pelas notas Dó, Mi e Sol que representa o 1º, 3º e 5º graus respectivamente. A variação das notas de 3º e 5º graus geram novos acordes, onde os mais conhecidos são: acordes maiores, menores, com quartas (SUS4), quinta aumentada e diminuta[2], por exemplo: C, Cm, Csus4, C(#5), C(b5), respectivamente.

Já os acordes formados por quatro notas são chamados de tétrade, onde aparecem as notas de 7º grau. Exemplo: um acorde de Dó com sétima é formado pelas notas Dó, Mi, Sol e Sib, na qual representa 1º, 3º, 5º e 7º graus respectivamente. Similarmente aos acordes formados por tríades, a variação da nota do 7º grau também gera novos acordes, como podemos ver na Figura 5.

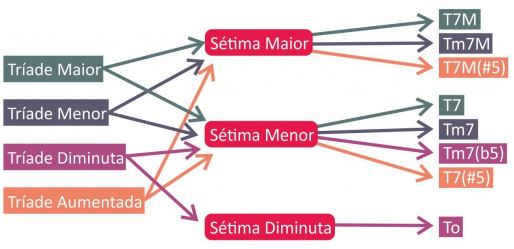


Figura 5: Variações de acordes de tétrades. A letra T na última coluna representa a tónica do acorde, ou seja, a nota de 1º grau.Fonte: <http://aguitarra.com.br>

## Conceitos de Processamento de Sinais

O processamento de sinais está ligado à natureza do sinal e a aplicação, onde normalmente consiste na análise e/ou modificação de sinais, utilizando teoria fundamental, aplicações e algoritmos de forma a obter informações ou apropriando o mesmo para uma aplicação específica. Existem duas formas para fazer o processamento de sinais, uma é a analógica e a outra é a digital[9]. Neste trabalho, é utilizado o sinal no formato digital.

Para poder trabalhar com um sinal digital, partindo de um sinal analógico, é preciso discretizá-lo. Para isso, o sinal analógico passará por duas fases que são amostragem e quantização (geralmente os CDs – Compact Discs - utilizam 16 bit/amostra)(citar tcc solução computacional).

Na fase de amostragem, é calculada a quantidade de amostras em um dado período de tempo, ou seja, haverá uma frequência (taxa de amostragem ou taxa de Nyquist) associada a um período de tempo , que resulta um conjunto finito de amostras para um intervalo de tempo. A expressão matemática que representa esta fase é dada por:

(Eq. 1)

Segundo o teorema de amostragem de Nyquist mostrado na Equação 2, a taxa de amostragem deve ser pelo menos duas vezes a maior frequência . O ouvido humano consegue diferenciar sons entre a faixa de frequência de 20 Hz e 22.050 Hz. Portanto, a máxima frequência do ouvido humano é de 22.050 Hz; então a frequência de Nyquist deve ser 44.100 Hz. Dessa forma, são captadas 44.100 amostras de áudio a cada segundo.

(Eq. 2)

A transformada de Fourier é uma das ferramentas mais fundamentais deste processo, pois ela consegue a representar um sinal que está no domínio do tempo no domínio da frequência. Com foi mostrado na Seção 2.1, cada nota está associada a uma frequência, fazendo com que a nota seja grave para frequências baixas e agudas para frequências altas. A representação da transformada discreta de Fourier (*Discrete Fourier Transform -* DFT)[9] para N pontos é dada por:

|  |
| --- |
| (Eq.3) |

onde: e .

Já a transformada discreta inversa de Fourier é dada por:

|  |
| --- |
| (Eq.4) |

onde: .

## Conceitos de *Chromagram*

O *chroma* (representado em *chromagrams*) é uma ferramenta poderosa de extração de características no contexto de reconhecimento de acordes musicais [10] e foi introduzida por Fujishima [11]. Ele pode ser utilizado para extrair características de harmonia [12] ou melodia [13] de uma composição musical. Geralmente, o *croma* é representado por um vetor de 12 posições, onde cada posição do vetor pode ser interpretada como uma nota musical, por exemplo: {Dó, Do#, Ré, Re#, Mi, Fá, Fá#, Sol, Sol#, Lá, Lá#, Sí}. Neste trabalho, o *chroma* foi utilizado para obter informações apenas da harmonia da música.

Existem várias formas de representar um acorde em um vetor de *chroma.* Por exemplo:

* Atribuir valor 1 quando a nota pertence ao acorde e 0 caso contrário. Exemplo, o acorde de dó é formado pelas notas dó, mi e sol, então o vetor de *chroma*, usando esta abordagem, seria {1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0}.
* Atribuir uma probabilidade para cada nota, onde 1 seria a probabilidade máxima da nota pertencer ao acorde e 0 caso contrário. Exemplo, usando como base o acorde de dó, novamente, um possível vetor gerado seria {1, 0.01, 0.01, 0.01, 0.9, 0, 0, 0.9, 0.01, 0.01, 0.01, 0.01}.

As principais etapas de construção de um *chromagram* podem ser vistas na Figura 6.

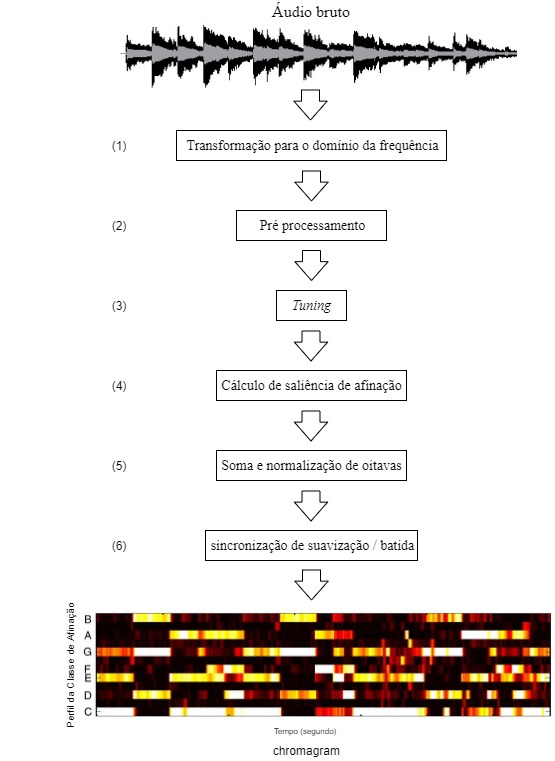


Figura 6: Passos para construção de um *chromagram*

Partindo de um sinal bruto discretizado no domínio do tempo, o primeiro passo para construir um *chromagram* é converter o sinal para domínio da frequência e calcular sua magnitude. Existem algumas técnicas para representar o sinal no domínio da frequência de forma simples como DFT. No entanto, a DFT traz descrição global das frequências presentes no áudio e, de acordo com [10], os pesquisadores que estudam a estimação automática de acordes (*Automatic Chord Estimation* - ACE) estão interessados nas variações harmônicas locais e, por isso, geralmente é utilizada a Transformada de Fourier de tempo curto (*Short-Time Fourier Transform* - STFT) que calcula as magnitudes de frequência em uma janela deslizante através do sinal.

O segundo passo, pré-processamento, tenta eliminar parte inútil do espectro como o espectro de fundo[17][18] e alguns harmônicos que podem facilmente confundir técnicas de extração de características [19][20].

De acordo com Sheh e Ellis, existem algumas faixas de músicas que não estão afinadas no tom padrão A4 = 440 Hz[21]. Então, o terceiro passo, *tuning*, aplica algoritmos como [22][23] para ajustar a afinação dessas músicas.

O quarto passo, cálculo de saliência de afinação, captura a saliência da classe de afinação fazendo o mapeamento do espectro obtido nos passos anteriores com a saliência de uma nota.

No estágio final do cálculo do *chromagram*, soma e normalização de oitavas, é feita a soma de todas as saliências pertencentes a uma mesma classe. Essa soma é normalizada para produzir uma matriz de recurso do *chromagram* que captura a evolução da afinação do áudio ao longo do tempo. Por fim, suavização /sincronização de batida, é uma etapa de pós processamento para minimizar as frequentes mudanças de acordes e ruído.

## Conceitos de Redes Neurais

Para compreender como funciona o modelo *Multilayer Perceptrons* (MLP), primeiramente é necessário entender o funcionamento de um *perceptron*. Semelhante ao cérebro humano que consiste em uma rede de neurônios, uma rede neural consiste em uma rede de neurônios artificiais, chamados de *perceptrons*.

O *perceptron* foi criado por Frank Rosenblatt por volta dos anos 1957. A estrutura do perceptron pode ser vista na Figura 8, onde um conjunto de entradas combinadas aos seus respectivos pesos são classificadas em 0 ou 1, ou -1 e 1, dependendo da função de ativação que for usada. A Equação 5 mostra a representação matemática de um perceptron levando em consideração o seu *bias*, já a Equação 6 mostra a função de ativação de um neurônio (14).

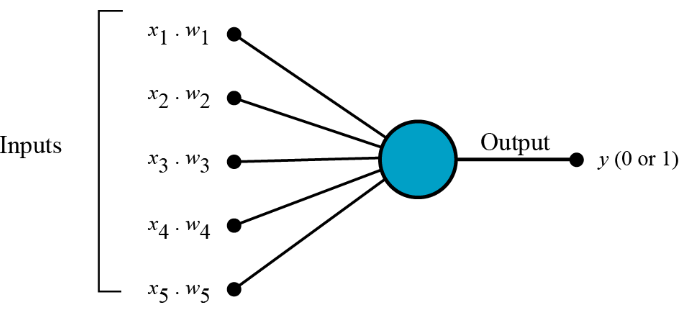


Figura 7: Representação gráfica de uma estrutura do perceptron.

(Eq.5)

(Eq.6)

O grande problema desse modelo, é que ele se limita em resolver problemas onde o conjunto de dados seja linearmente separável. Para resolver problemas mais complexos e semelhante ao cérebro humano que é composto por vários neurônios, existem modelos que são compostos por vários perceptrons, sendo um deles chamado de chamado de Perceptron multicamadas (*Multi Layer Perceptron* - MLP).

A MLP consiste em uma estrutura em camadas (uma camada de entrada, uma ou mais camadas escondidas/intermediárias e uma camada de saída), cada uma pode conter um ou mais perceptron. Além disso, o sinal de entrada é propagado para frente conforme os neurônios de cada camada. Por isso, as MLPs são consideradas redes neurais do tipo *feedforward* [15].

O treinamento de uma MLP é feito de forma supervisionada utilizando o algoritmo gradiente descendente (*backpropagation*). Este algoritmo é dividido em duas etapas, a primeira é a fase de ida, onde um padrão é apresentado a camada de entrada e, a partir desta camada as unidades calculam sua resposta e produz um conjunto de saída. A segunda fase é a de volta, onde partir da saída gerada na fase anterior é calculado um erro, este erro é propagado a partir da camada de saída até a camada de entrada, e os pesos das conexões das camadas internas vão sendo justados [15].

Por fim, a Figura 9 é um exemplo de arquitetura de uma MLP, neste exemplo na camada de entrada apresenta dois neurônios, existem duas camadas intermediárias com quatro neurônios e na camada de dois neurônios.

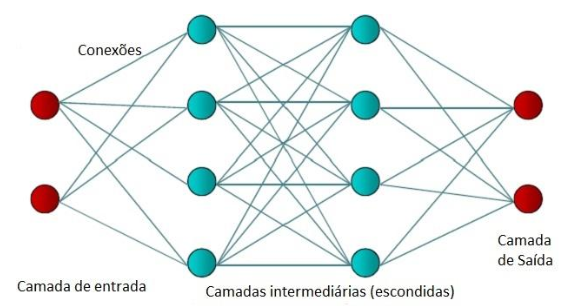


Figura 8: Arquitetura de uma Rede Neural MLP. Fonte: [encurtador.com.br/lmpY1](https://www.researchgate.net/publication/265552060_O_USO_DE_REDES_NEURAIS_PARA_A_ANALISE_E_CONCESSAO_DE_CREDITO_FLAVIO_IZO_-flavioflavioizocom_INSTITUTO_FEDERAL_DO_ESPIRITO_SANTO_-IFES)

# Trabalhos Relacionados (Estado da Arte)

Resumir os principais trabalhos relacionados ao seu objeto de estudo. Lembre de citar fragilidades deles (se não têm falha, são perfeitos, para que estudar um novo?).

# Algoritmo de Mello

scnbdnbndbbdkb

# Algoritmo de Mello

scnbdnbndbbdkb

# Metologia do Estudo

Este capítulo explica

Este capítulo trata do objeto do estudo.

## Visão Geral

Este capítulo trata do objeto do estudo.

## Passo 1

Este capítulo trata do objeto do estudo.

## Passo 2

Este capítulo trata do objeto do estudo.

# Experimentos e Análise

Este capítulo apresenta os experimentos feitos. Sempre é preciso comparar com o que os outros fizeram.

## Experimento 1

dfvbvbvbv.

## Experimento 2

dfvbvbvbv.

## Análise

dfvbvbvbv.

# Conclusões e Trabalhos Futuros

Este capítulo apresenta as conclusões do seu trabalho. Resuma seus resultados e análise e apresente sugestões de trabalhos futuros. Apresente suas contribuições.

## Contribuições

sdfdfdfdfd.

## Trabalhos Futuros

ssddfdgfgfgfg.

x

x

# Referências

1. Wikipédia. [Online]: <https://pt.wikipedia.org/wiki/Nota_musical>. (Acessado em 13/10/2019)
2. descomplicandoamusica. [Online]: <https://www.descomplicandoamusica.com/triades/>. (Acessado em 13/10/2019)
3. wikipedia. [Online]: <https://pt.wikipedia.org/wiki/Cifra_(música)>. (Acessado em 01/10/2019)
4. cifraclub. [Online]: <https://www.cifraclub.com.br/>. (Acessado em 01/10/2019)
5. cifras. [Online]: <https://www.cifras.com.br/>. (Acessado em 01/10/2019)
6. pegacifra. [Online]: <https://www.pegacifra.com.br/>. (Acessado em 01/10/2019)
7. F.Veloso, M.A.Feitosa. "O Ouvido Absoluto: bases neurocognitivas e perspectivas". 2013, p. 357 - 362.
8. P.J. dos Santos Junior. "Ouvido absoluto e ouvido relativo: sua natureza e relevância para a educação musical". In XVI Encontro Regional Sul da ABEM; 2014.
9. A.Oppenheim, R.W.Schafer. "Discrete-Time Signal Processing": Prentice-Hall; 1989.
10. M.McVicar, R.Santos-Rodriguez, Y.Ni, T.De Bie. "Automatic Chord Estimation from Audio: A Review of the State of the Art". IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2014, p. 556-575.
11. T.Fujishima. "Realtime chord recognition of musical sound: A system using Common Lisp Music", Proc. Int. Comput. Music Conf., 1999, p. 464–467.
12. M.Muller e S.Ewert. "Towards Timbre-Invariant Audio Features for Harmony-Based Music", IEEE Transactions on Audio, Speech, and Language Processing, 2010.
13. H.J.Kallman. "Tone chroma is functional in melody recognition", Perception & Psychophysics, 1979, 26.
14. L.C.Akshay. towardsdatascience: https://medium.com/ensina-ai/redes-neurais-perceptron-multicamadas-e-o-algoritmo-backpropagation-eaf89778f5b8.
15. S.Haykin. "Neural Computation : A Comprehensive Foundation". 2ª Edição, Editora Prentice Hall, 1998.
16. deeplearningbook. [Online]: <http://deeplearningbook.com.br/algoritmo-backpropagation-parte-2-treinamento-de-redes-neurais/>. (Acessado em 01/11/2019)
17. N.Ono, K.Miyamoto, J.Le Roux, H.Kameoka, e S.Sagayama. “Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram”, 16th European Signal Processing Conference, 2008.
18. J. Reed, Y. Ueda, S. Siniscalchi, Y. Uchiyama, S. Sagayama, and C. Lee, “Minimum classification error training to improve isolated chord recognition,” in Proc. 10th Int. Soc. Music Inf. Retrieval, 2009, pp. 609–614.
19. S.Pauws. “Musical key extraction from audio”, Proc. 5th Int. Soc. Music Inf. Retrieval. 2004: p. 66–69.
20. M.Mauch, e S.Dixon. “Simultaneous Estimation of Chords and Musical Context From Audio”, IEEE Transactions on Audio, Speech, and Language Processing, 2010, p. 1280-1289.
21. A. Sheh and D. Ellis, “Chord segmentation and recognition using em-trained Hidden Markov Models,” in Proc. 4th Int. Soc. Music Inf. Retrieval, 2003, pp. 183–189.
22. C. Harte and M. Sandler, “Automatic chord identification using a quantised chromagram,” in Proc. Audio Eng. Soc., 2005, pp. 291–301.
23. C. Harte, M. Sandler, and M. Gasser, “Detecting harmonic change in musical audio,” in Proc. 1st Workshop Audio Music Comput. Multimedia, 2006, pp. 21–26.