UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

MODELOS DE *DEEP LEARNING* PARA ESTIMATIVA DE TEMPO EM MÚSICAS

MILA SOARES DE OLIVEIRA DE SOUZA

Orientador
PEDRO NUNO DE SOUZA MOURA

RIO DE JANEIRO, RJ – BRASIL
FEVEREIRO DE 2020

Catalogação informatizada pelo(a) autor(a)

S719 Souza, Mila Soares de Oliveira de
Modelos de deep learning para estimativa de tempo em músicas / Mila Soares de Oliveira de Souza. -- Rio de Janeiro, 2020.
67 f.

Orientador: Pedro Nuno de Souza Moura.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal do Estado do Rio de Janeiro, Graduação em Sistemas de Informação, 2020.

1. Deep learning. 2. Music Information Retrieval. 3. Estimativa de tempo. 4. B-RNN. 5. Comparação de performance. I. Moura, Pedro Nuno de Souza, orient. II. Título.

MODELOS DE *DEEP LEARNING* PARA ESTIMATIVA DE TEMPO EM MÚSICAS

MILA SOARES DE OLIVEIRA DE SOUZA

Projeto de Graduação apresentado à Escola de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado por:

_____

Prof. Pedro Nuno de Souza Moura, D.Sc. (UNIRIO)

_____

Profa. Geiza Maria Hamazaki da Silva, D.Sc. (UNIRIO)

_____

Prof. Jean-Pierre Briot, Ph.D. (CNRS/LIP6/UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

FEVEREIRO DE 2020

Agradecimentos

Ao meu pai Pedro Francisco de Souza Filho e minha mãe Michela Soares de Oliveira de Souza. Caberiam aqui inúmeras palavras, mas gostaria de pontuar o esforço quase sobre-humano que vocês sempre fizeram para que eu e minha irmã tivéssemos o melhor possível, principalmente na educação. Obrigada por sempre terem me incentivado a fazer o que eu bem quisesse.

Gostaria também de incluir minha família inteira. Por vocês, nunca me faltou amor e carinho. Obrigada por acreditarem em mim mais do que eu mesma acredito.

Ao Leonardo Giucci, pela paciência comigo e pelo carinho em todos os momentos de dificuldade durante a execução desse trabalho.

Ao Yan Gonçalves, pois sem seu apoio emocional eu não estaria nem aqui fisicamente.

Ao meu orientador e professor Pedro Nuno de Souza Moura, por sempre ter sido solícito e compreensivo comigo nos meus inúmeros momentos de muita dificuldade enquanto aluna, e de dúvidas enquanto orientanda. Sua presença e preocupação constante foi essencial para que eu pudesse concluir esse trabalho.

Ao professor Jean-Pierre Briot, por ter introduzido a mim enquanto aluna o incrível universo de *deep learning* para música, área que engloba duas das minhas grandes paixões, e ser muito solícito quanto às minhas dúvidas.

Ao Paulo Fontana, por ter auxiliado no início do processo deste trabalho e, principalmente, pela ideia de estudo de ritmo.

Ao Günter Loch, por ter me assistido com paciência durante as dificuldades para encontrar um ambiente que pudesse executar o experimento deste trabalho.

Ao André Abrantes, por ter me ensinado ao longo de 1.5 ano (e com toda paciência do mundo) muito do que sei sobre desenvolvimento de *software*.

A todos os outros amigos da *Microsoft Advanced Technology Labs*, que também contribuíram para meu conhecimento técnico e me deram muito carinho.

A todos os colegas, docentes e funcionários da UNIRIO, por sempre terem sido solícitos e dispostos a me ajudar a resolver qualquer problema ou dúvida que fosse. A boa vontade deles sempre foi um grande fator de motivação para continuar a graduação.

RESUMO

Este trabalho realiza a modelagem e treinamento, avaliação e comparação de dois modelos de redes neurais, um convolucional (CNN) e outro recorrente bidirecional (B-RNN) que realizam a estimativa de tempo (em bpm) de uma música. A arquitetura da CNN foi reproduzida de (SCHREIBER; MÜLLER, 2018), enquanto a B-RNN foi proposta para este trabalho. A partir da entrada do espectrograma mel de uma peça musical, oriundo de seu sinal de áudio, as redes estimam o valor do tempo. Os *datasets* utilizados totalizam 12.550 músicas, incluindo apenas linhas de percussão, abrangendo diversos gêneros musicais. Os desempenhos das duas redes neurais são registrados, e comparados a outros resultados considerados estado da arte.

Palavras-chave: *deep learning*, redes neurais, música, tempo, estimativa.

ABSTRACT

This project proposes the training, evaluation, and comparison of two neural network models (one CNN and one B-RNN, convolution and recurrent-based) which perform the tempo estimation of musical pieces. The CNN model implementation is reproduced from its original article (SCHREIBER; MÜLLER, 2018), while the B-RNN was implemented in this paper based on the former model. Having a mel spectrogram of a musical piece as the input, the neural networks estimate the tempo (bpm) of said piece. A large and extensive dataset was constructed (12.550 samples), which include a variety of music genres, for conducting our comparative, quantitative and qualitative evaluation. The two trained models' performances are also compared to a state-of-the-art model. This experiment reports about results and its analysis, lessons learned and future prospects.

Keywords: deep learning, neural networks, music, tempo, estimation.

Índice

Índice de Tabela

Índice de Figuras

1 Introdução

1.1 Motivação

O ritmo está presente nas mais diversas formas de manifestação da vida — inclusive na própria Natureza (FLANNERY, 1990). O grande padrão geral da Natureza é de ciclos dentro de ciclos, que estariam, por sua vez, contidos em outros ciclos (MEDAWAR; MEDAWAR, 1983). Alguns exemplos citados por Flattely (1920) incluem comportamentos periódicos em resposta a (ou imposta por) fatores externos, como a alternância entre dia e noite e a recorrência do ciclo das estações. O coração humano apresenta funcionamento rítmico (TEIE, 2016), que pode ser medido em batidas por minuto (bpm) tal qual uma música.

Não obstante, para a convivência e socialização de seres humanos, o ritmo também mostra-se de fundamental importância: manifestando-se na fala (HAUSEN et al., 2013), o ritmo com que uma pessoa dirige seus argumentos e opiniões à outra é capaz de expressar diferentes emoções ao ouvinte, ou mesmo influenciá-lo. Semama (1991) explica que através da linguagem oral — que tem o ritmo como fator fundamental — é exercido o poder, conquistas, consensos teóricos e práticos, sendo possível afirmar que todas as relações sociais se devem à linguagem.

Dentro das mais diversas aplicações do ritmo no contexto humano encontra-se a música. Evidências históricas apontam que era utilizada indispensavelmente em rituais (KUBATZKI, 2016); nos tempos atuais, é usada como plataforma de expressão social, entretenimento e como um produto para o mercado, existindo toda uma indústria baseada na criação, venda e performance musical. Sendo uma atividade participativa, as formas como a música é expressa pode variar de acordo com o contexto social; alguns exemplos foram o movimento rebelde *punk* e o movimento pacifista *hippie*, que expressaram alguns de seus valores pela plataforma musical.

A possibilidade de gravar músicas em uma forma física que pode ser reproduzida em aparelhos de som viabilizou a venda de música como um produto. Popularizada pelos discos de vinil — e, em seguida, gravada em formas de armazenamento mais eficientes como CDs —, atualmente é consumida prioritariamente de forma digital, seja pelo

armazenamento em arquivos de áudio como MP3 e WAV ou por serviços de *streaming* como o Spotify.

Paralelamente, o ramo tecnológico computacional veio evoluindo a todo vapor, tanto no desenvolvimento de *hardwares* quanto *softwares* e linguagens de programação. Na última década, é possível dizer que *machine learning* já vinha apresentando um crescimento em sua exploração; Uma das motivações para popularização do uso de aprendizagem de máquina também se deu após Krizhevsky, Sutskever e Hinton (2012) publicarem um método novo usando *deep learning* que superou todos os demais por uma ampla margem de 15.3%, ultrapassando a marca de 75% de acurácia. Como prova da relevância do ramo para o avanço da comunidade científica, o *Turing Award* de 2018 foi oferecido a Geoffrey Hinton, Yoshua Bengio e Yann LeCun considerando a importância alcançada pelo *deep learning* graça aos esforços dos três pesquisadores e suas equipes.

Deep learning, cujo conceito foi elaborado pela primeira vez por Dechter (1986), é uma categoria de *machine learning* que permite sistemas computacionais a se aprimorar através de dados e experiência (GOODFELLOW; BENGIO; COURVILLE, 2016), utilizando redes neurais para processamento de informações. Dentro deste campo, já existem aplicações com um ótimo nível de sucesso, como a tecnologia de reconhecimento facial (LIU et al., 2017), que é aplicada por empresas como o Facebook.

Dentre as diversas aplicações de *deep learning* encontra-se a possibilidade de incorporá-lo ao estudo da música. Um exemplo em voga é a geração automatizada de peças musicais. Ao utilizar redes neurais para criação e composição de novas músicas, aumenta-se o leque de possibilidades musicais. Isso acontece porque, diferentemente do cérebro humano, a máquina não é influenciada por memórias subconscientes, podendo ser capaz de gerar conteúdo completamente novo com base apenas no que lhe é apresentado (BRIOT, 2018).

Em 2018, foi lançado o primeiro CD com composições inteiramente criadas por Inteligência Artificial — "*Hello World*"[1], por SKYGGE, que já apresenta mais de 5 milhões de execuções de suas faixas. Empresas como a Google também vêm investindo em aplicações do *deep learning* nas artes; Magenta[2] é um de seus projetos *open source*

[1] Disponível em: <https://www.helloworldalbum.net/>. Acesso em: 20 jul. 2019.

[2] Disponível em: <https://ai.google/research/teams/brain/magenta/>. Acesso em: 20 jul. 2019.

de pesquisa que explora tais aplicações como ferramenta em processos criativos. Através do Magenta, são disponibilizadas, por exemplo, ferramentas para auxiliar na criação de padrões musicais novos e para transcrição polifônica de piano.

Além da contribuição no processo criativo da composição de músicas, também há um crescente interesse em pesquisas sobre coleta de informações musicais (*Music Information Retrieval*, ou MIR). Em Janeiro de 2020, a conferência "ISMIR" (*International Society for Music Information Retrieval*), criada para abordar pesquisas de MIR, ocupava a posição de nono lugar no ranking de publicações na subcategoria Multimídia de Engenharia e Ciência da Computação do Google Scholar[3], e primeiro lugar na subcategoria Música e Musicologia na área de Humanidades, Literatura e Artes[4].

Sua atual relevância não se restringe apenas ao campo acadêmico — segundo seu website oficial, o ISMIR 2019[5] recebeu patrocínios consideráveis de empresas como Spotify, Deezer, Google, Sony, Adobe, Facebook e outras, evidenciando um *appeal* econômico atraente para investimentos na indústria. Algumas dessas empresas também realizam seus próprios estudos: Uma pesquisa feita por profissionais da Spotify analisou experimentalmente o que poderia ser considerado um ritmo interessante para uma pessoa com base no *feedback* do público — músicas na plataforma Spotify que são mais ouvidas até o final podem ser um indicador de características mais interessantes para o público geral do que as que são mais puladas (MONTECCHIO; ROY; PACHET, 2019). A partir desse feedback, pode ser possível analisar futuramente que características musicais podem ser mais bem aproveitadas quando for necessário compor uma nova peça para a indústria musical.

Tendo em vista a importância da música não só para o ser humano como um todo como também para o mercado, através dos exemplos supracitados, mostra-se interessante avaliar, estudar e experimentar a utilização das técnicas de *deep learning* para geração e classificação musical, ramos que vem demonstrando ser uma nova tendência em

---

[3] Disponível em: <https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_multimedia>. Acesso em: 18 jan. 2020.

[4] Disponível em: <https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=hum_musicmusicology>. Acesso em: 18 jan. 2020.

[5] Disponível em: <https://ismir2019.ewi.tudelft.nl/>. Acesso em: 3 fev. 2020.

crescimento tanto na indústria musical como no âmbito acadêmico. O estudo do ritmo, por tratar de um dos conceitos mais fundamentais da música, mostra-se um potencial desafio relevante para a comunidade; O estudo de dois dos seus principais conceitos — métrica e tempo — poderiam ser o primeiro passo em direção ao aprofundamento de pesquisas sobre ritmo na computação.

1.2 Problema

Enquanto processos altamente formalizados são facilmente solucionáveis por computadores, processos intuitivos para seres humanos podem se mostrar muito mais complexos de serem reproduzidos pelas máquinas (GOODFELLOW; BENGIO; COURVILLE, 2016). A construção e o entendimento de ritmo, embora intuitivas para o ser humano, são algumas dessas atividades que ainda se tem dificuldade em replicar computacionalmente, uma vez que não existe um consenso em relação a representações de seus conceitos (GOUYON; DIXON, 2004).

Uma das etapas fundamentais para a análise de ritmo é a detecção de tempo (GOUYON; DIXON, 2004), que é o tema de estudo deste trabalho; Tais detecções são o passo fundamental para as mais inúmeras aplicações e projetos maiores que possam vir a ser estudados em música, uma vez que tempo é um dos conceitos-chave do ritmo (BERRY, 1976).

1.3 Objetivos

O objetivo geral do trabalho é realizar um estudo sobre a aplicação de modelos de redes neurais artificiais em análise de tempo musical.

Os objetivos específicos são:

- Modelar redes neurais que sejam capazes de analisar o tempo de um *input* referente ao sinal de áudio;
- Realizar experimentos com redes neurais de modo a abordar o problema em questão;
- Comparar o desempenho das redes neurais utilizadas, analisando os resultados obtidos;

- Verificar se há diferença significativa de precisão quando se analisa uma peça apenas de percussão (ex.: bateria) em relação a análise de peças com mais instrumentos e/ou linhas vocais.

É esperado que os modelos, após processarem uma entrada de áudio, gerem uma saída correta sobre o tempo da música em batidas por minuto.

1.4 Organização do texto

O presente trabalho está estruturado em capítulos e, além desta introdução, será desenvolvido da seguinte forma:

- Capítulo II: Revisão bibliográfica – Apresenta uma breve história do *deep learning* para música, detecção de tempo e trabalhos relacionados.
- Capítulo III: Conceitos preliminares – Explica conceitos musicais, sinais de áudio e redes neurais artificiais.
- Capítulo IV: Experimento – Relata a metodologia, estrutura e resultados do experimento.
- Capítulo V: Conclusões – Reúne as considerações finais, assinala as contribuições da pesquisa e sugere possibilidades de aprofundamento posterior.

2 **Revisão Bibliográfica**

As primeiras evidências sólidas de estudos aplicando redes neurais à música são encontradas em 1988. Lewis (1988) introduziu algumas variações computáveis do paradigma *Creation by Refinement* (CBR, traduzido livremente como "criação por refinamento"), chamadas de *Attentional CBR* e *Genetic CBR*, apresentando ao final do artigo uma simulação de aplicação do paradigma.

> *Creation by Refinement* (CBR) é um paradigma de rede neural desenvolvido especificamente para problemas de criatividade artificial como a composição de música por máquina. O CBR consiste em uma fase de aprendizagem [...], seguido de uma fase de criação [...] (LEWIS, 1988).

Até 1989, a criação prática de música computacional se restringia a algoritmos que pré-estabeleciam cada regra de sua composição (TODD, 1989). O artigo *A Connectionist Approach to Algorithmic Composition*, publicado por Todd (1989), propôs formas de representação dos conceitos de notas musicais e duração de uma nota - conceitos esses que permitiriam que um modelo rede neural fosse capaz de ser treinada para gerar um *output* completamente novo, baseando-se em *inputs* de sequências de notas musicais. Um modelo de representação de notas apresentado foi o de posições relativas - uma nota musical de valor absoluto é fornecida como "ponto de partida", e as notas musicais sucessoras são implicitamente definidas pela variação de tom em relação à nota antecessora. O intervalo entre os tons é definido pelo autor como um semitom (meio tom de diferença entre duas notas). Dessa forma, a melodia A-B-C seria representada como {A, +2, + 1} (onde +2 significa que a nota sucessora de A é 2 semitons acima de A, ou seja, B; e +1 significa que a nota sucessora de B é 1 semitom acima de B). Outro modelo de representação apresentado foi o de valores absolutos definidos explicitamente para cada nota, no qual a melodia A-B-C seria representada como {A, B, C}. O autor enfrentou limitações pela tecnologia disponível na época; ainda assim, foi possível obter como

output novas músicas geradas, levando em consideração apenas as notas musicais e duração de cada uma dela.

Embora o estudo apresentasse inovações promissoras, pesquisas que envolviam redes neurais foram pouco fomentadas nos anos seguintes à publicação em comparação com outras áreas de estudo da computação, o que, consequentemente, não propiciou um ambiente favorável para grandes investimentos na música computacional. Esse cenário permaneceu relativamente estagnado até aproximadamente 2006, quando a terceira e atual "grande onda" de redes neurais, cunhada de *deep learning* começou (GOODFELLOW; BENGIO; COURVILLE, 2016), e continua até o presente momento.

Ainda assim, durante os anos 90 e início dos 00, alguns pesquisadores que acreditavam no potencial das redes neurais continuaram trabalhando no tema. Hochreiter (1991) e Bengio et al. (1994) identificaram dificuldades matemáticas fundamentais na modelagem de sequências longas (GOODFELLOW; BENGIO; COURVILLE, 2016); Hochreiter e Schmidhuber (1997) introduziram a *long short-term memory network (*LSTM) para resolver algumas dessas dificuldades. LSTM é um modelo de rede neural que contém uma célula de memória, frequentemente utilizado na geração de música via *deep learning* pela sua capacidade de manter memória de informações musicais geradas previamente; Isso é interessante, por exemplo, para construir uma música que faça sentido dentro de uma escala, aproximando-se o máximo possível de uma música gerada por humanos.

Dada sua atratividade para a música, não tardou para que estudos musicais com LSTM começassem a aparecer: Logo em 2002, Eck e Schmidhuber propuseram um sistema capaz de gerar improvisos de Blues utilizando-a, em seu artigo *Finding Temporal Structure in Music: Blues Improvisation with LSTM Recurrent Networks.*

É interessante ressaltar que já existiam técnicas para identificação de tempo (e até mesmo geração de música) sem aplicar o *deep learning*. Um exemplo é o uso de *Markov models* (modelos de Markov); Cemgil e Kappen (2003) realizaram testes aplicando cadeias de Markov (especificamente, MCMC, *Monte Carlo Markov Chain*) para percepção de tempo e quantização de ritmo. Briot, Hadjeres e Pachet (2019) citam alguns prós (precedidos por +) e contras (precedidos por –) do uso de redes neurais e modelos de Markov:

+ Modelos de Markov são conceitualmente simples.

> + Modelos de Markov possuem implementação simples e algoritmo de aprendizagem simples, visto que o modelo é uma tabela de probabilidade de transição (as estatísticas são coletadas a partir do *dataset* de exemplos para então computar a probabilidade).
> – Modelos de redes neurais são conceitualmente simples, mas a implementação otimizada de arquiteturas de redes neurais profundas mais atuais pode ser complexa, necessitando de muitos ajustes.
> – Modelos de Markov de ordem 1 (isto é, considerando apenas o estado anterior) não capturam estruturas temporais a longo termo. [...]
> + Redes neurais podem capturar vários tipos de relações, contextos e regularidades. [...]
> + Modelos de Markov não generalizam muito bem.

No que tange à detecção de tempo, em 2004 foi lançado um concurso pela ISMIR cujo objetivo foi comparar algoritmos do estado-da-arte na detecção de tempo a partir de sinais de áudio[6], mas ainda sem a presença de *deep learning*. Gouyon et al. (2006) publicaram um artigo com o resultado, também testando e descrevendo as 12 entradas ao concurso, no qual o algoritmo chamado Klapuri foi o vencedor (apresentando a melhor acurácia). Böck, Krebs e Widmer (2015) fizeram um resumo dos algoritmos participantes, sendo alguns deles:

> O trabalho de Schreirer (1998) foi o primeiro a processar o sinal de áudio continuamente em vez de trabalhar sobre uma série de eventos temporais discretos. Ele propôs o uso de *resonating comb filters*, que são uma das principais técnicas usadas para estimativa de periodicidade desde então. A análise de periodicidade é performada sobre um número de *band pass filtered signals* e os outputs dessa análise são combinadas, e um tempo global é informado. [...]

[6] Disponível em: <http://mtg.upf.edu/ismir2004/contest/tempoContest/>. Acesso em: 16 fev. 2020.

> Klapuri, Eronen e Astola (2006) analisam conjuntamente a peça musical em três escalas de tempo: *tatum*, *tactus* (que corresponde à batida ou tempo) e nível de medida. O sinal é dividido em múltiplas bandas e então combinado em quatro bandas de acento musical antes de serem alimentadas a um banco de *comb filters* similar a (SCHREIRER, 1998). Suas evoluções temporais e a relação de diferentes escalas temporais são modeladas com um *framework* probabilístico para informar a posição final das batidas. O tempo é então calculado como a mediana dos intervalos de batidas durante a segunda metade do sinal.

As primeiras evidências de utilização de redes neurais para estimativa de tempo foram surgindo no final da primeira e início da segunda década do Século XXI. Alguns desses trabalhos foram feitos por Böck (ca. 2010) — que utiliza B-RNN (redes neurais recorrentes bidirecionais) e LSTM (*Long short-term memory*) — Gkiokas, Katsouros e Carayannis (2012), e Böck, Krebs e Widmer (2015). Schreiber e Müller (2018) fazem um pequeno resumo sobre alguns desses trabalhos existentes que utilizam redes neurais para estimativa de música, considerando também o modelo de Böck, Krebs e Widmer (2015) como estado da arte:

> Outra área de pesquisa ativa busca criar um melhor OSS através do uso de redes neurais. Elowsson (2016) utiliza separação de fonte harmônica/percussiva e duas redes neurais feedfoward diferentes para classificar um frame como batida ou não-batida. Böck, Krebs e Widmer (2015) utilizam uma *bidirectional long-short term memory* (BLSTM) *recurrent neural networ*k (RNN) para mapear frames de magnitude espectral e suas diferenças de primeira ordem para valores de ativação de batida. Eles então são processados ainda mais com comb filters. Para suas aplicações de robô dançante, Gkiokas, Katsouros e Carayannis (2012) usam uma rede neural convolucional (CNN) para derivar uma função de ativação de batida (*beat activation function*), que é então utilizada para *beat tracking* e estimativa de tempo.

Conforme novos modelos vão sendo propostos e aperfeiçoados por pesquisadores, novas implementações na música vão sendo apresentadas. Aplicações de GANs (*generative adversarial networks)* (GOODFELLOW et al., 2014), para a criação de música, por exemplo, apresentam resultados satisfatórios, como o MidiNet (YANG; CHOU; YANG, 2017) e o MuseGAN (DONG et al., 2018); Considerando os exemplos de aplicação de *deep learning* e o desenvolvimentos de sistemas que dizem respeito, é possível, portanto, esperar cada vez mais novidades na música computacional para o futuro, tanto na variedade de projetos quanto de formas de implementação.

3 **Conceitos Preliminares**

Neste capítulo, serão introduzidos os conceitos necessários para o entendimento deste trabalho. Os leitores que julgarem já possuir os conhecimentos a serem descritos podem pular este capítulo. Dessa forma, são abordados, portanto, conceitos referentes à teoria musical, sinais de áudio e redes neurais artificiais.

3.1 *Music Information Retrieval* (MIR)

Music Information Retrieval (MIR) é o nome atribuído ao campo de estudo que busca extrair, analisar e fornecer informações de uma música (SCHEDL, 2008), tendo como algumas metodologias básicas o processamento de sinal de áudio, percepção musical, entre outros (GÓMEZ et al., 2016). Alguns exemplos de pesquisa de MIR envolvem a estimativa de tempo (ALONSO; DAVID; RICHARD, 2004) — que é o próprio tema deste trabalho —, identificação de um estilo musical (ORAMAS et al., 2018) e comparação de similaridade entre duas músicas (LOGAN; SALOMON, 2001). Outros exemplos são ilustrados conforme a Figura 1. Sua história é mais antiga do que possa parecer — alguns sistemas já haviam sido desenvolvidos na década de 1960, tendo raízes em *Information Retrieval*, que é um campo que pesquisa sobre coleta de informações), Musicologia e Psicologia da Música (UITDENBOGERD; CHATTARAJ; ZOBEL, 2000). Um cenário fictício de utilização de um sistema para MIR pode ser descrito:

> Imagine um mundo no qual você vai até um computador e canta um trecho de música que está na sua cabeça desde o café da manhã. O computador aceita seu canto desafinado, corrige sua requisição, e prontamente lhe sugere que "*Camptown Races*" é a causa da sua inquietação. Você confirma a sugestão do computador após ouvir um dos vários arquivos MP3 encontrados. Satisfeito, então, você gentilmente recusa a oferta de conferir todas as outras versões da canção, incluindo uma recém-lançada no estilo rap Italiano e outra versão orquestral que inclui um dueto de gaita de fole.
>
> Esse tipo de sistema existe atualmente? Não. Existirá no futuro?

> Sim. Tal sistema será fácil de produzir? Decididamente, não. Inúmeras dificuldades ainda precisam ser superadas antes da criação [...] de sistemas para *Music Information Retrieval* (MIR) robustos e em larga escala se tornar realidade (DOWNIE, 2003).

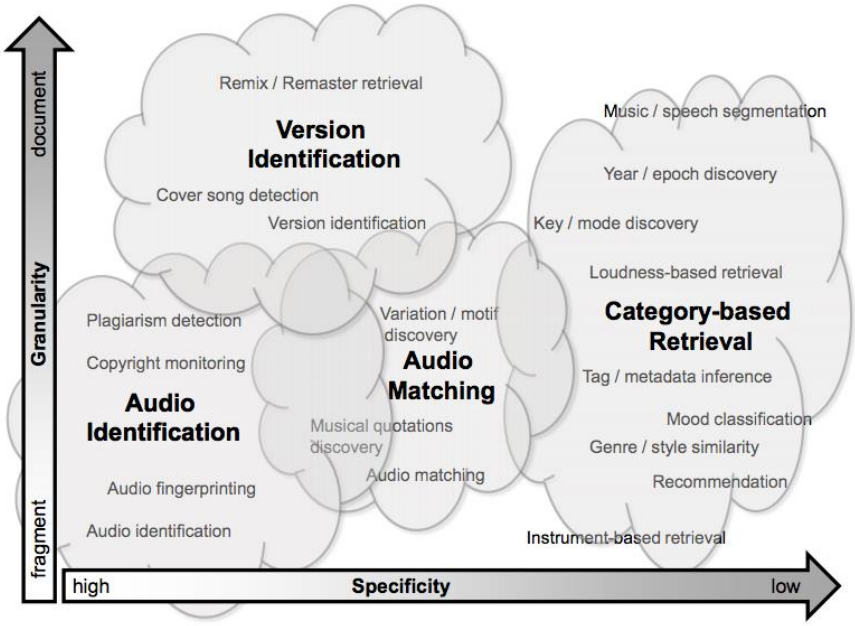


Figura 1: Algumas áreas de estudo de sistemas MIR (extraído de GROSCHE; MÜLLER; SERRÀ, 2012).

A primeira conferência dedicada ao ramo ocorreu em 2000, nos Estados Unidos, chamada "*International Society for Music Information Retrieval*" (ISMIR); segundo o website oficial do evento[7], tópicos de pesquisa abordados incluem a estimativa de similaridade entre melodias, representação e indexação de músicas, construção de *databases* musicais e outros. A motivação para criação do evento foi reunir, discutir e estimular produções e pesquisas na área, que vinha apresentando crescimento.

Os tópicos de pesquisa de informação musical em voga vêm apresentando variações ao longo dos anos (GÓMEZ et al., 2016). Os pesquisadores Gómez et al. (2016) explicam que no início dos anos 2000 houve um foco no conteúdo musical (como análise

[7] Disponível em: <https://ismir2000.ismir.net/>. Acesso em 4 fev. 2020.

de sinal de áudio e partituras). No meio dos anos 2000, em contexto musical (envolvendo *tags* geradas colaborativamente, reputação de um artista etc.). Mais recentemente, uma mudança de foco para designs centrados no usuário (como aspectos psicológicos do usuário, se uma música é considerada "triste" ou "feliz" e que tipo de emoção busca em uma música), junto com uma mudança técnica para o uso de *machine learning*. A crescente popularização da aprendizagem profunda para MIR é evidenciada pelo contínuo aumento do uso de redes neurais em publicações submetidas ao ISMIR, conforme visto pelos dados estatísticos da Figura 2.

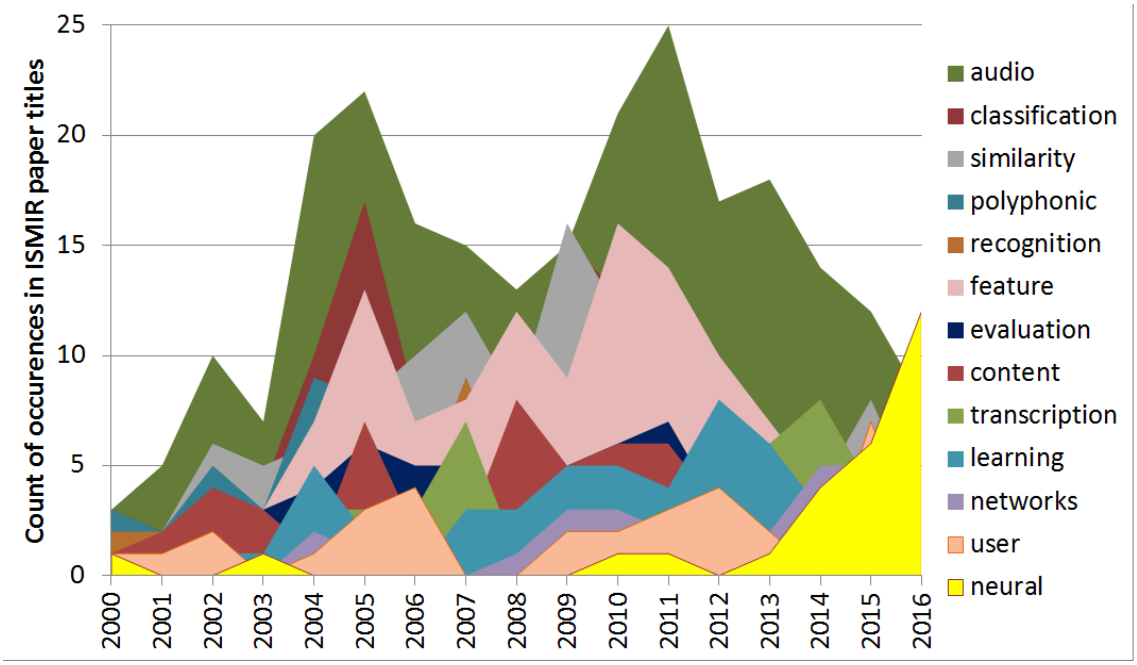


Figura 2: Contagem de ocorrências de palavras nos títulos de artigos para a ISMIR entre 2000 e 2016 (extraído de GÓMEZ et al., 2016).

Conforme novos modelos vão sendo propostos e aperfeiçoados por pesquisadores, novas implementações na música vão sendo apresentadas. Aplicações de GANs (*generative adversarial networks)* (GOODFELLOW et al., 2014), para a criação de música, por exemplo, apresentam resultados satisfatórios, como o MidiNet (YANG; CHOU; YANG, 2017) e o MuseGAN (DONG et al., 2018); Considerando os exemplos de aplicação de deep learning e o crescimento de suas aplicações citados neste capítulo, é possível, portanto, esperar cada vez mais novidades na música computacional para o futuro, tanto na variedade de projetos quanto de formas de implementação.

3.2 Conceitos Musicais

Nesta subseção, serão introduzidos alguns conceitos referentes à música e sua estrutura relevantes a este trabalho.

3.1.1 Tempo

Tempo é uma das características intrínsecas ao ritmo (BERRY, 1976). O tempo de uma música é a quantidade de batidas por minuto contidas na mesma. O número de batidas é definido pelo intervalo entre pulsos dentro do período de um minuto. Berry (1976) apresenta uma forma de definir um pulso:

> Pulso é entendido como a unidade recorrente, que pode ser sentida, pela qual a duração temporal de uma música é medida e cujas divisões são sentidas em algum nível específico — a base para contagem, [...] e para indicações em metrônomos de "tempo".

Em termos gerais, tempo significa a quantidade de eventos no ambiente; um tempo rápido indica uma taxa de eventos rápida (MCAULEY, 2010). Por isso, se ouvimos uma música com bpm tal como 280, é comum a sensação de estar ouvindo uma música rápida, ou ainda animada, bem estimulante ao cérebro; Já ao ouvir uma música de 40 bpm, é comum a sensação de estar ouvindo uma música lenta, possivelmente mais calma, pelo menos comparada à anterior de 280 bpm. Por esse fator, o tempo de uma música pode ser um artefato para compositores transmitirem com mais efetividade mensagens e sentimentos que desejam em sua música.

Tome como exemplo a música "*Like A Stone*" da banda Audioslave; Se você, ao ouvir a música e sentir o ritmo, contar quantas vezes bate o dedo indicador na mesa durante 1 minuto, perceberá que bateu o seu dedo 108 vezes (ou algum múltiplo de 108); Isso porque o BPM da música é 108[8]. Caso tenha contado algum múltiplo de 117, é perfeitamente aceitável — essa questão é abordada na literatura científica de música computacional como "*octave error*", ou "*tempo octave error*".

O "*tempo octave error*" na música computacional se refere à situação em que o tempo detectado pelo algoritmo ou sistema é um múltiplo do tempo considerado verdadeiro por pessoas (HÖRSCHLÄGER et al., 2015). Ainda assim, não é incomum

[8] Disponível em: <https://songbpm.com/@audioslave/like-a-stone>. Acesso em: 16 fev. 2020.

haver discordâncias durante o entendimento do tempo de uma música por parte de seres humanos: É possível, ao ouvir uma música, contar o tempo como múltiplos (dobro, metade etc.) do valor considerado oficialmente pelo autor. Isso pode variar conforme a pessoa que está ouvindo (pessoas diferentes têm percepções diferentes (WU, 2015)); inclusive, uma mesma pessoa pode contar batidas em métricas diferentes para a mesma música dependendo do seu estado mental no momento (GKIOKAS; KATSOUROS; CARAYANNIS, 2012). Muitos estudos interessantes e inclusive comerciais também revelam o impacto causado por músicas de tempos diferentes sobre a psicologia do comportamento e emoções humanas no ato de compras, bebida, exercícios e processos de cura (WU, 2015).

3.1.2 Nota Musical

Uma nota musical representa uma vibração emitida em frequências pré-determinadas. Na notação musical atualmente utilizada, as notas musicais básicas são Dó (C), Ré (D), Mi (E), Fá (F), Sol (G), Lá (A) e Si (B), podendo apresentar variações de tom como sustenido (#) e bemol (b). São definidas oito oitavas, onde cada oitava contém as 7 notas.

As características físicas da primeira oitava, como frequência e comprimento de onda, constam na Tabela 1. Para saber os valores correspondentes das notas de cada oitava seguinte, basta multiplicar o valor das frequências da nota por 2 a cada oitava (e diminuir o comprimento da onda pela metade). Por fim, é importante ressaltar que a primeira oitava tem o som que soa mais grave; as oitavas consecutivas soam mais agudas ao ouvido humano.

Nota	Frequência (Hz)	Comprimento de onda (cm)
C_0	16.35	2109.89
$C^{\#}_0/D^{b}_0$	17.32	1991.47
D_0	18.35	1879.69
$D^{\#}_0/E^{b}_0$	19.45	1774.20
E_0	20.60	1674.62
F_0	21.83	1580.63
$F^{\#}_0/G^{b}_0$	23.12	1491.91
G_0	24.50	1408.18
$G^{\#}_0/A^{b}_0$	25.96	1329.14
A_0	27.50	1254.55
$A^{\#}_0/B^{b}_0$	29.14	1184.13
B_0	30.87	1117.67

Tabela 1: Notas musicais da primeira oitava (adaptado de SUITS, 1998).

Em uma partitura (representação escrita de uma peça de música), as notas musicais são representadas pelas suas durações temporais, mas não possuem uma duração fixa específica — elas possuem valores relativos. Algumas das durações musicais básicas

estão definidas na Tabela 2 abaixo. Por exemplo, uma mínima dura o dobro de uma semínima; já uma colcheia é metade de uma mínima. Suas durações em segundos são calculadas conforme o número de batidas por minuto. Conforme a Tabela 2, a semibreve foi escolhida como o valor de referência (R), e o valor das outras notas são definidas a partir de seu valor. Não é necessário que a semibreve seja sempre o valor inicial de referência — ela foi escolhida na Tabela 2 apenas para facilitar o entendimento do leitor. Na prática, a semínima frequentemente é escolhida como valor referencial em representações musicais.

Nome	**Duração relativa**
Semibreve	R
Mínima	R/2
Semínima	R/4
Colcheia	R/8
Semicolcheia	R/16

Tabela 2: As notas musicais e suas durações relativas.

Mas como é definida em uma música a duração em segundos de uma nota de referência? Na verdade, seu valor depende do tempo da música, um conceito introduzido na subseção anterior. Como citado no parágrafo anterior, a semínima frequentemente é utilizada como valor de referência em partituras e conceitos musicais. Se uma música tem 60 bpm (batidas por minuto), então a duração de uma semínima é tal que caibam 60 semínimas em um minuto. Isso quer dizer que a duração de uma semínima é 1 segundo, já a mínima dura 2 segundos, e assim por diante.

Instrumentos de percussão, como bateria, ou sinais digitais parecidos são frequentemente responsáveis pelo andamento rítmico da música, já que eles são

responsáveis pelas batidas na performance. Assim, é possível tentar estimar o tempo em cima do áudio contendo apenas a linha da percussão.

3.3 Representações de sinais de música

O sinal de uma música geralmente é uma mistura complexa de sons, e que consiste em diversos componentes sonoros distintos. Devido a essa complexidade, extrair informações musicais relevantes de um formato de onda constitui um problema difícil (MÜLLER, 2015).

Assim, para realizar tarefas de extração de informação do sinal, frequentemente são utilizadas abordagens alternativas ao uso apenas do contexto temporal (LI; CHAN; CHUN, 2010). Duas dessas abordagens são a aplicação da Transformada de Fourier de curto termo e a Escala de Frequência Mel, que serão introduzidas nas Subseções .

No que tange ao campo de *Music Information Retrieval*, frequentemente é realizada a conversão do áudio do domínio do tempo para o domínio da frequência (CASEY et al., 2008). A partir daí, são executados procedimentos matemáticos desejados sobre trechos do sinal do áudio ou do espectrograma (representação visual do sinal, que também será introduzida em uma subseção abaixo) a fim de se obter uma informação desejada, que pode variar conforme o autor e sua pesquisa. A extração de *features* de um sinal de áudio não tem uma "receita única", ou seja, pode ser realizada de diversas maneiras, algumas delas sendo esquematizadas pela Figura 3.

Na Figura 3, tem-se, da esquerda para a direita: *log-frequency chromagram*, *Mel-frequency cepstral coefficients*, *linear-frequency chromagram* e *beat tracking*. FFT = Transformada rápida de Fourier; STFT = Transformada de Fourier de curto termo; MEL = Conversão para a escala Mel; LOG = Conversão para a escala logarítmica na intensidade; DCT = Transformada discreta de cosseno.

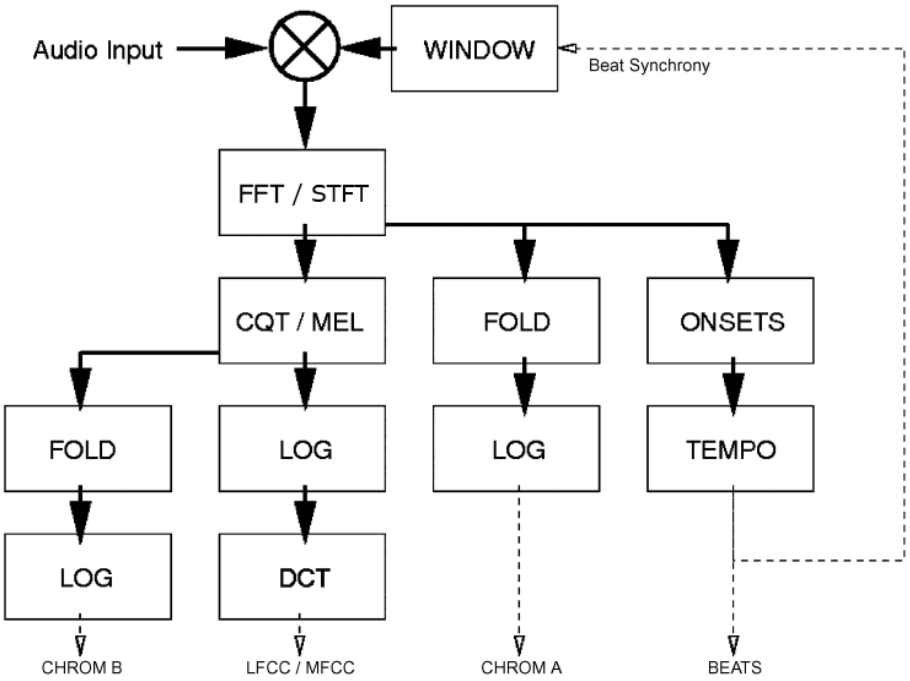


Figura 3: Esquematização de alguns processos de extração de features comuns (adaptado de CASEY et al., 2008).

Para este trabalho, apenas parte do segundo caminho (da esquerda para direita) é considerada relevante, sendo utilizada na metodologia do experimento. O *input* do experimento é o espectrograma Mel em escala logarítmica (também chamado de espectrograma log Mel); A última etapa da aplicação da transformada discreta de cosseno (DCT) não é aplicada.

A Figura 4 representa visualmente um processo de obtenção de espectrograma log Mel. A partir de um sinal de áudio como entrada (representado como *waveform*, no domínio do tempo), aplica-se uma transformada de Fourier como a STFT (representado visualmente como *spectrogram*), para então converter a frequência para a escala Mel, com magnitude medida em decibéis (representado visualmente como *mel-spectrogram*). Os conceitos envolvidos no processo da extração do espectrograma log Mel — transformada de Fourier de curto termo (abreviada como STFT em inglês),

espectrograma, escala Mel e o espectrograma Mel — serão abordados nas Subseções 3.3.1, 3.3.2 e 3.3.3.

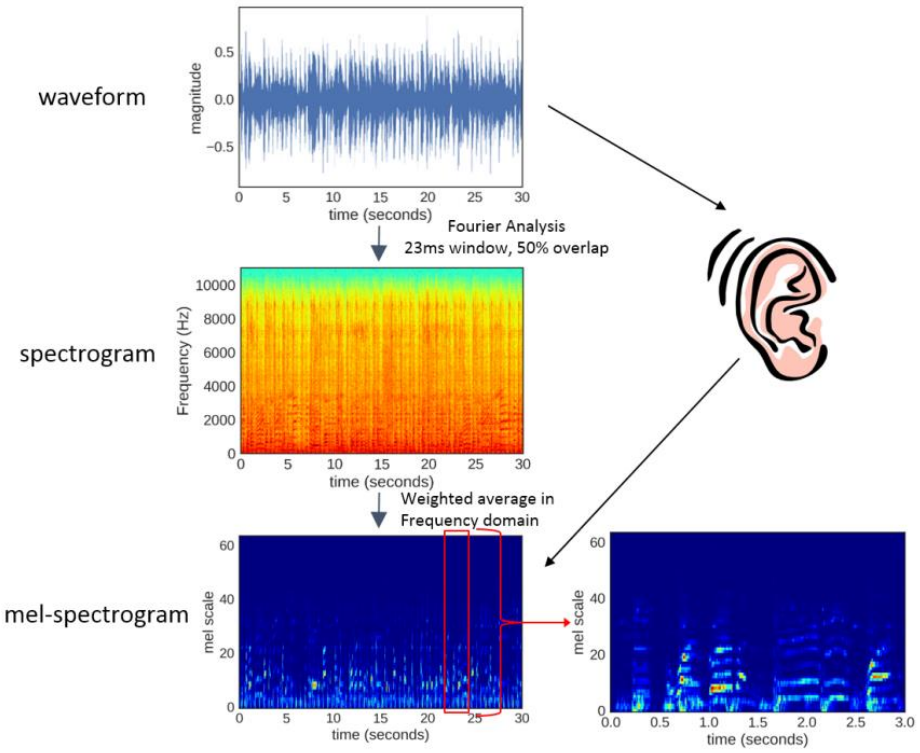


Figura 4: Uma conversão de *waveform* para espectrograma e espectrograma mel (extraído de DONG, 2018).

3.3.1 Transformada de Fourier de curto termo (STFT)

Um sinal de áudio não-estacionário representado espectralmente pela sua amplitude em função do tempo pode não ser suficiente para estudo. A Transformada de Fourier de curto termo — *short-term Fourier transform* ou *short-time Fourier transform*, também abreviada como STFT — foi proposta por Gabor (1946) para analisar o sinal de onda em um curto intervalo de tempo considerando as pequenas variações que acontecem em uma janela temporal. A STFT converte um sinal que depende do tempo para uma representação que depende da frequência (MÜLLER, 2015). Em termos gerais, a conversão para o domínio da frequência permite a obtenção mais informações sobre detalhes do que está "acontecendo" em cada momento na peça musical. Essa mudança

pode ser visualmente compreendida por um espectrograma, conceito que será mais bem introduzido na próxima subseção.

> A transformada de Fourier fornece informações de frequência que são medidas sobre todo o domínio de tempo. Entretanto, a informação sobre quando essas frequências ocorrem estão ocultas na transformada. Para recuperar a informação oculta, [...] em vez de considerar o sinal inteiro, a ideia principal da STFT é considerar apenas uma pequena seção do sinal. Para esse propósito, é fixada a chamada função de enjanelamento, que é uma função não-nula apenas por um curto período de tempo (definindo a seção a ser considerada). O sinal original é, então, multiplicado pela função de enjanelamento para gerar um sinal enjanelado. Para obter informação de frequência em diversas instâncias temporais, desloca-se a função de enjanelamento ao longo do tempo e computa-se uma transformada de Fourier para cada um dos sinais enjanelados resultantes (MÜLLER, 2015).

Choi et al. (2017) também discorrem sobre a eficiência da transformada:

> A STFT fornece uma representação no domínio tempo-frequência [...]. A computação de uma STFT é mais eficiente do que outras representações tempo-frequência graças à transformada rápida de

Fourier (FFT) que reduz o custo $O(N^2)$ para $O(N \log(N))$ no que diz respeito ao número de pontos FFT.

Em definições matemáticas, para retirar uma sub-amostra do sinal, define-se uma função de enjanelamento $w[t, \tau]$ para um intervalo como em (1):

$$w[t, \tau] \rightarrow w[(t - \tau)] \tag{1}$$

A STFT contínua pode ser definida[9] como $X(\tau, \omega)$:

$$STFT\{x(t)\}(\tau, \omega) \equiv X(\tau, \omega) \equiv \int_{-\infty}^{\infty} x(t) w(t - \tau) e^{-i\omega t} dt \tag{2}$$

Sendo em (1) e (2):

$x(t)$	Sinal no domínio do tempo a ser transformado
τ	Tempo (menor que t)
ω	Frequência
$w(t - \tau)$	Função de enjanelamento para um intervalo de tempo de τ até t
$X(\tau, \omega)$	Função complexa representando a fase e magnitude do sinal ao longo do tempo e frequência (Transformada de Fourier)

Um exemplo de aplicação em sinal pode ser visualizado na Figura 5. Na Figura 5, da esquerda para direita, de cima para baixo, respectivamente, estão representadas visualmente: Sinal original não-estacionário; algumas amostras com função de

[9] AHMAZIZADEH, M. **An Introduction to Short-Time Fourier Transform (STFT)**. Disponível em: <http://sharif.edu/~ahmadizadeh/courses/advstdyn/Short-Time%20Fourier%20Transform.pdf>. Acesso em: 12 jan 2020.

enjanelamento do tipo Hanning; resultado após aplicação da transformada a cada amostra; espectrograma em mapa de cores.

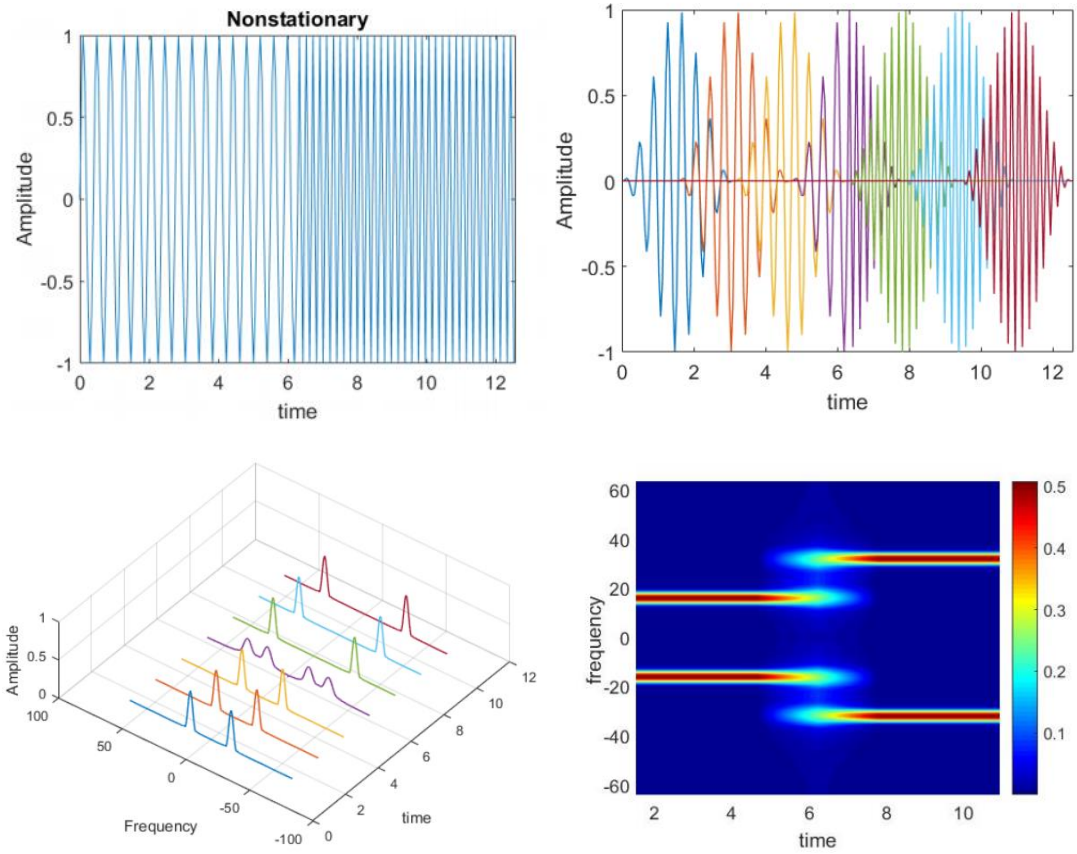


Figura 5: Sinal de áudio e transformações (extraído de INTRODUÇÃO…, 2019)[10].

3.3.2 Espectrograma

O espectrograma fornece uma representação visual bidimensional de um sinal de áudio, na qual o eixo horizontal representa o tempo e o eixo vertical representa a frequência (MÜLLER, 2015), enquanto esquemas de cores podem ser utilizados para representar a intensidade. A Figura 6 é um exemplo visual de um espectrograma. As frequências podem ser representadas em outras escalas, como a escala Mel (utilizada neste trabalho), que será introduzida na Subseção 3.2.2.

[10] **Introdução a análise temporal-espectral**. 2019. Disponível em: <http://lef.mec.puc-rio.br/wp-content/uploads/2019/06/Tranformada-de-Fourier-de-Tempo-Curto.pdf>. Acesso em: 2 fev. 2020.

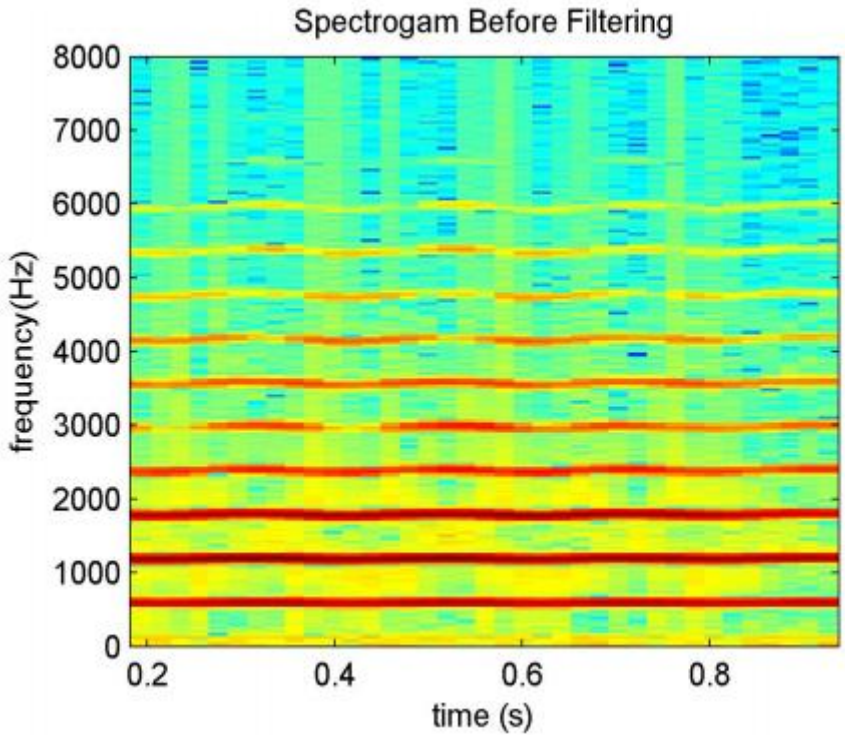


Figura 6: Espectrograma de um *oboe tone* a 596.8 Hz (D5) (extraído de ZHANG; BOCKO; BEAUCHAMP, 2014).

Em termos técnicos, o espectrograma de magnitude $\gamma(\tau, \omega)$ (*magnitude spectrogram*) é uma representação bidimensional da STFT. O espectrograma de potência (*power spectrogram*) é definido como uma representação bidimensional do quadrado da magnitude da STFT $X(\tau, \omega)$ (MÜLLER, 2015, p. 55) como em (3):

$$\gamma(\tau, \omega) := |X(\tau, \omega)|^2. \quad (3)$$

Ainda segundo MÜLLER (2015), "o espectrograma pode ser visualizado através de uma imagem bidimensional, na qual o eixo horizontal representa o tempo e o eixo vertical representa frequência". O espectrograma (especialmente o espectrograma com frequências representadas na escala Mel, que será abordado na Subseção 3.2.3) é muito utilizado como input para análise em MIR (MONTECCHIO; ROY; PACHET, 2019) por conter informações de modo que análises mais precisas possam ser realizadas.

A intensidade de um espectrograma, a princípio, segue uma escala linear; entretanto, é possível realizar uma conversão para escala logarítmica, fazendo com que a unidade de medida seja o decibel (dB). Este procedimento também será abordado novamente na Subseção 3.2.3 sobre escala mel.

3.3.3 Escala mel

A escala mel é uma escala subjetiva para medição de tons (STEVENS; VOLKMANN; NEWMAN, 1937) desenvolvida após experimentações sobre a forma que a audição humana interpreta um tom. Conforme experimentos foram realizados, percebeu-se que a percepção tonal não ocorre de maneira totalmente linear; A escala mel propõe-se a representar, portanto, a interpretação de tons de uma forma mais semelhante à audição humana, com alta precisão para bandas de baixa frequência e baixa precisão para bandas de alta frequência (DONG, 2018). Ela leva em consideração que a percepção de frequências até 1127 Hz funciona com uma responsividade linear; acima disso, a escala é logarítmica.

A escala mel pode ser aproximada para uma forma logarítimica[11]; uma fórmula amplamente utilizada para representá-la está indicada em (4):

[11] **The mel frequency scale and coefficients**. Disponível em: <http://kom.aau.dk/group/04gr742/pdf/MFCC_worksheet.pdf>. Acesso em: 27 jan. 2020.

$$F_{mel} = \frac{1000}{\log(2)}\left[1 + \frac{F_{Hz}}{1000}\right] \quad (4)$$

Sendo F_{mel} a frequência resultante na escala mel (medida em mels) e F_{Hz} a frequência medida em Hertz. A relação resultante está esquematizada na Figura 7.

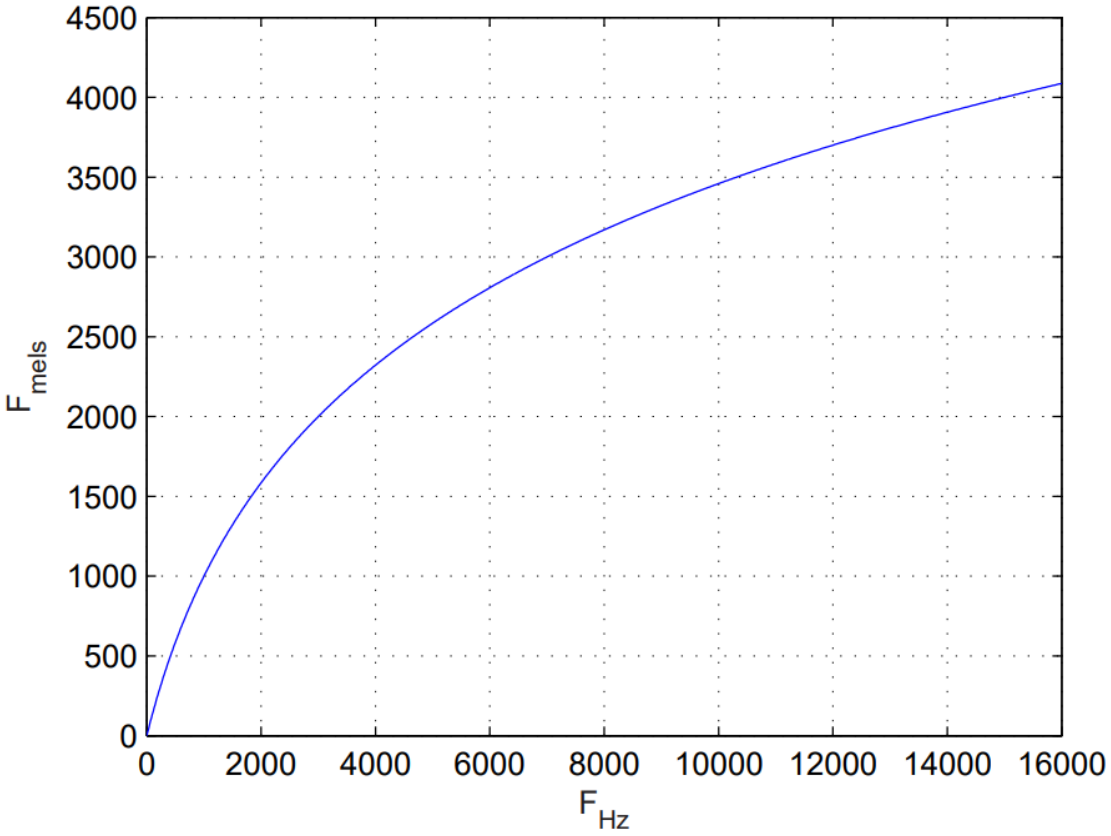
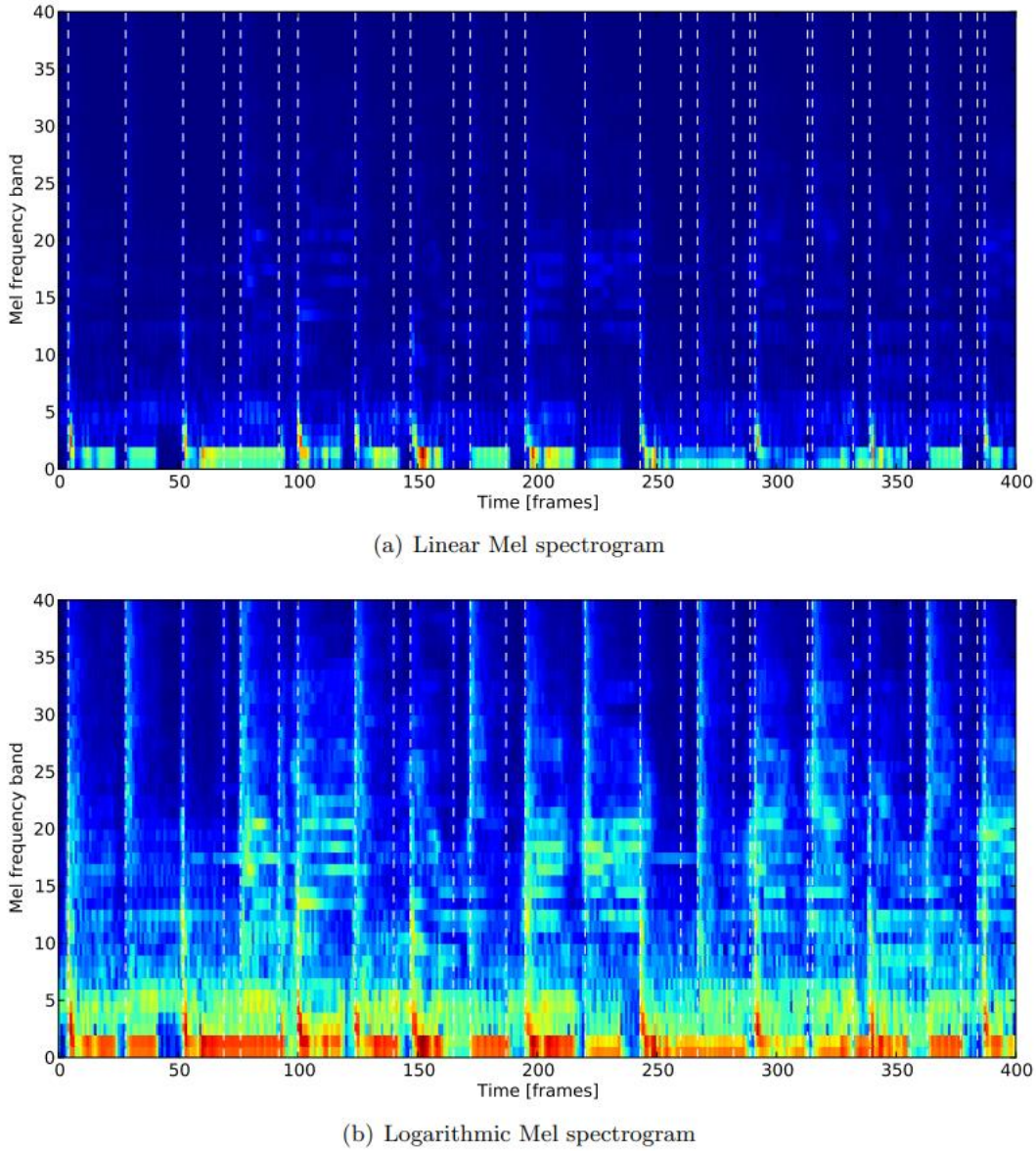


Figura 7: Relação entre a escala de frequência Hertz e a escala mel[10].

Um espectrograma pode ser convertido para utilizar frequências na escala Mel; O espectrograma Mel, ou *Mel-spectrogram*, é "uma representação 2d que é otimizada para a percepção auditiva humana" (CHOI et al., 2017), frequentemente utilizada em pesquisas de MIR (MONTECCHIO; ROY; PACHET, 2019).

Não apenas a audição humana tem uma forma não-linear de percepção de frequências, como também a percepção de intensidade do som também apresenta não-linearidade, e levar este fato em consideração melhora consideravelmente a performance de sistemas MIR (BÖCK, ca. 2010). Assim, converter a escala de intensidade linear do espectrograma de potência para uma escala logarítmica (ou seja, em decibéis, ou dB) é

uma operação freqüentemente realizada. O espectrograma Mel com a escala logarítmica em decibéis costuma fornecer detalhes úteis para performar tarefas que ouvidos humanos também realizam, como estimativa de tempo. A diferença entre a utilização da escala linear e logarítmica de intensidade pode ser visualizada através da Figura 8, na qual as linhas tracejadas verticais representam *onsets* (isto é, o instante de início de uma batida).



(a) Linear Mel spectrogram

(b) Logarithmic Mel spectrogram

Figura 8: Espectrogramas Mel (a) linear e (b) logarítmico (extraído de BÖCK, ca. 2010).

4 **Abordagem ao Problema**

Este estudo realizou o treinamento, a avaliação e a comparação de uma rede neural convolucional e uma rede neural recorrente bidirecional com a finalidade de detectar o tempo em bpm de uma música preferencialmente com tempo constante. Isso porque, caso a música sofra variações de tempo, a precisão da detecção pode ser comprometida.

O problema da detecção de tempo é frequentemente encarado como um problema de regressão. Entretanto, é possível também tratá-lo como um problema de classificação. Schreiber e Müller (2018) propuseram essa abordagem, de modo que seria possível classificar o tempo de uma música como uma classe de tempo, cujo intervalo de números inteiros vai de 30 a 290 bpm (cada bpm correspondendo a uma classe). Essa abordagem foi julgada interessante e eficiente do ponto de vista dos resultados apresentados, sendo selecionada para este trabalho.

O modelo de CNN foi reproduzido de uma publicação já existente (SCHREIBER; MÜLLER, 2018), enquanto a B-RNN foi criada para este trabalho tendo como referência a própria CNN. A performance dos dois modelos foi comparada entre si e com resultados de outros estudos existentes.

As etapas realizadas foram, consecutivamente, seleção de *dataset*, definição dos modelos de redes neurais, treinamento da rede e avaliação. Todas serão abordadas com mais detalhes em Subseções desta Seção. Para o experimento, foram formuladas as seguintes perguntas:

P1: Existe diferença significativa de performance entre os dois modelos?; e
P2: A estimativa de tempo é mais precisa para inputs oriundos de áudios que contêm apenas linha de bateria do que para áudios que contém múltiplos instrumentos?.

Espera-se que a Pergunta 1 (P1) tenha uma resposta afirmativa, uma vez que o modelo de B-RNN foi proposto por este trabalho, como já explicado, e testado apenas uma vez sem realizar ajustes para melhorar a performance. Já a CNN foi reproduzida de um modelo proposto por Schreiber e Müller (2018), que se propuseram a criar um modelo

que superasse a performance de outros já existentes. Essa questão será mais bem discutida na subseção que trata sobre os modelos.

Espera-se, ainda, que a Pergunta 2 (P2) seja rejeitada. Embora seja sabido que a linha de percussão (no caso, bateria) é, frequentemente, o instrumento que guia o ritmo (e, portanto, o tempo) em músicas que a contenham, apenas um *dataset* contém peças contendo somente linhas de bateria, e é um número muito pequeno comparado ao total, como será explicado na Subseção 4.1.

Por fim, alguns obstáculo afetaram a realização dos experimentos, tais como a infraestrutura técnica — a autora não dispõe de uma GPU para realizar ajustes de hiperparâmetros ou testes, o que prejudica a otimização dos resultados — e pouca disponibilidade de dados para treinamento, problema que será melhor abordado nas subseções deste capítulo.

4.1 Representação do *input*

Neste trabalho, a representação para o sinal escolhida foi o espectrograma mel, a fim de aproximar-se da percepção de frequências por seres humanos, e, portanto, aproximando-se também da sua percepção de tempo. A representação logarítmica em decibéis não foi utilizada por limitações técnicas. Dessa forma, seria interessante, para um trabalho futuro, refazer o experimento utilizando a representação logarítmica.

Para este experimento, considera-se de maior relevância a estimativa de tempo de músicas que não apresentem variações temporais, isto é, músicas que não apresentem brusca variação de tempo. Se uma música não apresenta grandes variações de bpm ao longo do eixo do tempo, então é possível fornecer apenas uma parte da peça diretamente, em vez do espectrograma inteiro. Isso apresenta vantagem do ponto de vista de eficiência, uma vez que o *input* se torna muito menor, e também do ponto de vista de número de exemplos para treinamento — "dividindo" o espectrograma por janelas, é possível gerar vários exemplos diferentes para a rede a partir de um só.

A duração de cerca de 10 segundos de uma música é considerada suficiente pra ter uma boa noção do tempo de uma música. Assim, inspirado por Schreiber e Müller (2019), foi escolhido o valor de 256 frames para o *input* do espectrograma, o que é equivalente a aproximadamente 11.9 segundos. É possível, ainda, comprimir o espectrograma completo no eixo do tempo (mantendo o eixo da frequência intocado)

antes de cortá-lo pra 256 frames para aumentar a eficiência, ou esticar caso o sample inteiro tenha menos de 11.9 segundos de duração.

Para o tratamento do sinal e conversão, foi utilizada a biblioteca *librosa*[12]. As dimensões da entrada () e outros valores escolhidos para conversão, como tamanho de janela etc. foram inspirados por Schreiber e Müller (2018). O arquivo de áudio é carregado pelo *librosa*, convertido para mono com uma *sampling rate* de 11.025 Hz, e foram escolhidas janela de *1.024* samples. Isso é equivalente ao frame rate de 21.5 Hz, o que, conforme o teorema de *sampling* de Nyquist-Shannon, é o suficiente para representar tempo até 646 bpm (valor bem acima do que é frequentemente encontrado em músicas) (SCHREIBER; MÜLLER, 2018). Ainda segundo o teorema de *sampling* de Nyquist-Shannon, a frequência máxima de uma música é metade do *sampling rate*. Com isso, são escolhidos 40 *frequency bins* para o *input*, o que cobre até 5,000 Hz (próximo o suficiente da metade do *sampling rate*). Para converter em espectrograma mel, o *librosa* aplica o enjanelamento de Hamming, STFT e um filterbank adequado. Para a geração de *inputs* para rede, o espectrograma é esticado ou comprimido apenas no eixo temporal (mantendo o eixo da frequência original) com um fator aleatório {0.8, 0.84, ..., 1.16, 1.2}, e então, cortado para 256 frames em um *offset* escolhido aleatoriamente. Quando há o processo de corte ou alongamento, a anotação é ajustada adequadamente. Caso o espectrograma completo tenha mais de 11.9 segundos de duração, múltiplos cortes são fornecidos à rede neural para treinamento.

Esse processo de *scale & crop* (escalamento e corte) ocorrerá durante o treinamento da rede neural, logo antes do fornecimento de *input*, e está representado visualmente na Figura 9 (onde é apresentado, de cima pra baixo, o Espectrograma Mel completo, escalamento e corte). No processo, o espectrograma mel é primeiro esticado (ou comprimido) ao longo do eixo do tempo, o que requer um ajuste da anotação original, e então cortado para 256 frames em um *offset* aleatoriamente escolhido.

[12] Disponível em: <https://librosa.github.io/librosa/>. Acesso em: 20 jan 2020.

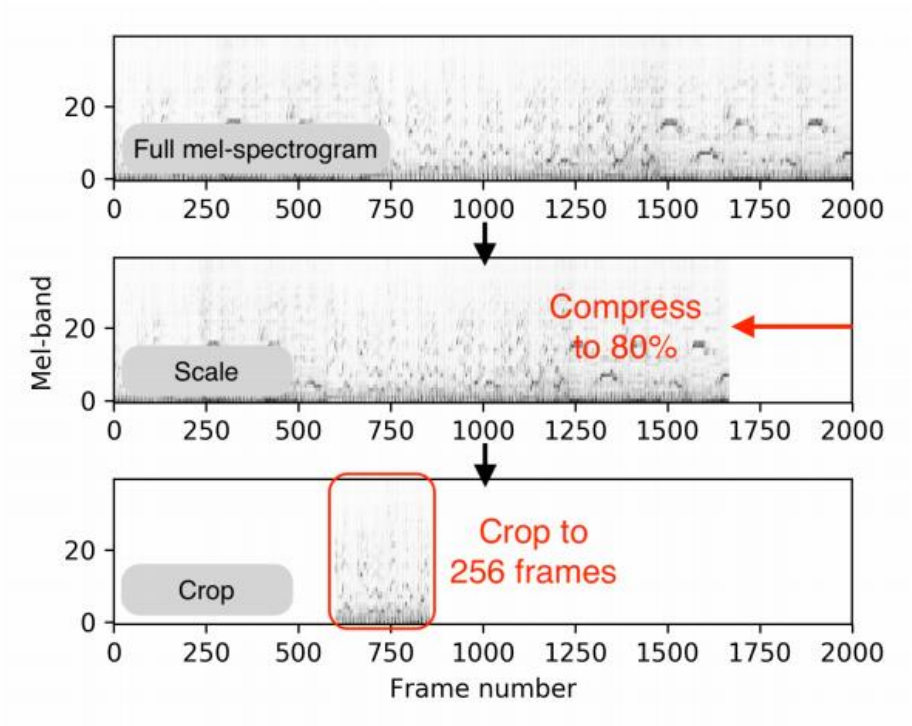


Figura 9: Aumento de dados de *input* por *scale-&-crop* (extraído de SCHREIBER; MÜLLER, 2018).

Em suma, o *input* para a rede neural é um espectrograma mel de dimensões $F_T \times T_T = 40 \times 256$, e o processo para sua obtenção pode ser resumido como por Schreiber e Müller (2019):

> Nós [...] utilizamos espectrogramas mel [...] com as dimensões $F_T \times T_T = 40 \times 256$ como input, sendo F_T o número de *frequency bins* e T_T o número de frames de tempo. F_T cobre o intervalo de frequências entre 20 e 5.000 Hz. A resolução temporal é de 0.46 ms por frame de tempo, ou seja, 256 frames correspondem a 11.9 segundos. [...] Os espectrogramas mel são cortados em um tamanho adequado utilizando um offset aleatoriamente escolhido durante cada época. Para aumentar o *dataset* de treinamento, os espectrogramas são escalados (*scaling*) ao longo do eixo de tempo antes do corte (*cropping*) utilizando fatores {0.8, 0.84, ..., 1.16, 1.2}. As anotações são ajustadas conforme o *scaling*. Após o *cropping* e *scaling*, os espectrogramas são normalizados, garantindo média zero e variância unitária por *sample*.

4.2 Modelos

Nesta seção, os modelos utilizados para o experimento são introduzidos. Escolheu-se utilizar uma CNN e uma B-RNN, que serão explicadas em detalhes nas próximas subseções.

4.2.1 CNN

As Redes Neurais Convolucionais (*Convolutional Neural Networks*, abreviadas como CNN) são, atualmente, um *de-facto standard* para coleta de informações de áudio baseadas em *deep learning* (SCHINDLER; LIDY; BÖCK, 2020). Pela sua reconhecida performance em classificação de imagens, a CNN é um modelo bem-visto em MIR porque muitas análises podem ser feitas sobre o espectrograma, que é uma representação visual bidimensional do sinal de música. Considerando o número de recursos e bibliografias disponíveis para o melhor entendimento de redes neurais convolucionais em processamento de imagens, assim como uma boa probabilidade de desempenho satisfatório, foi decidido que seria o primeiro modelo a ser utilizado no experimento.

O modelo proposto por Schreiber e Müller (2018) foi selecionado. Sua arquitetura pode ser visualizada na Figura 10. Três camadas convolucionais são seguidas por quatro módulos *mf_mod*, que, por sua vez, são seguidas por quatro camadas densas.

O modelo de Schreiber e Müller foi escolhido por ser um trabalho consideravelmente recente, apresentar resultados muito bons, superando outros considerados estado-da-arte (SCHREIBER; MÜLLER, 2018) e boa reprodutibilidade. O artigo explica bem a arquitetura visual e textualmente, de forma a ser consideravelmente simples reconstruí-lo através das informações explicadas no texto. Os autores também disponibilizaram o código para utilizar o modelo treinado e realizar conversões de áudio para espectrograma[13]. Um ponto muito forte a favor da escolha desse modelo foi o fato de não utilizar nenhum outro sistema nem outras etapas associadas à rede neural, algo comum em trabalhos de detecção de tempo, como Elowsson (2016) e Böck, Krebs e Widmer (2015). No trabalho de Schreiber e Müller (2018), apenas a rede neural é suficiente para gerar um *output* de tempo a partir de um *input* de espectrograma de áudio.

[13] Disponível em: <https://github.com/hendriks73/tempo-cnn>. Acesso em: 20 jan 2020.

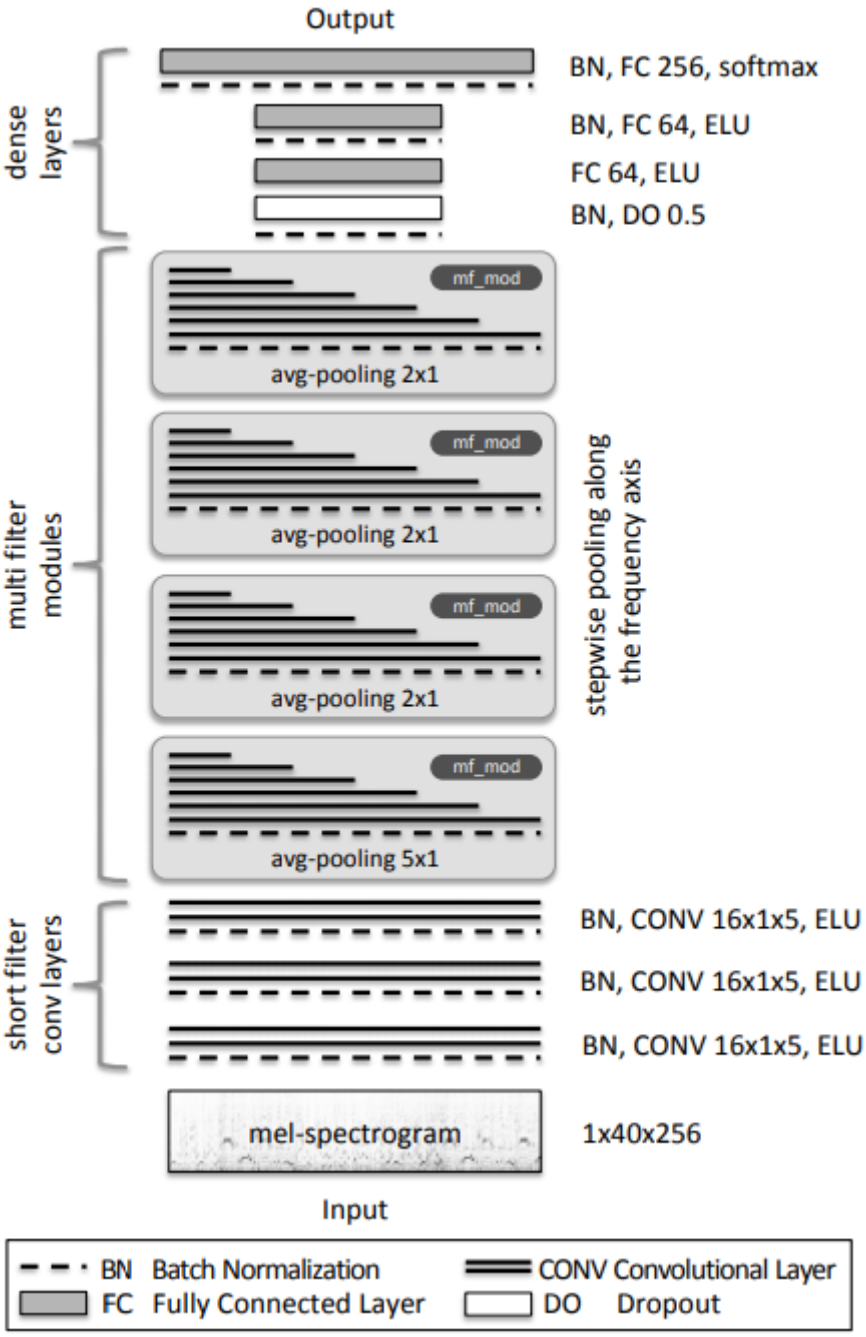


Figura 10: Visão geral da arquitetura da CNN (extraído de SCHREIBER; MÜLLER, 2018).

Conforme os autores, a ideia para a arquitetura foi inspirada pela abordagem tradicional de criar um OSS (*onset strength signals*), que seriam analisados depois por periodicidades. O *input* é processado por três camadas convolucionais (cada uma precedida por *batch normalization*), a fim de detectar *onsets* no sinal, de 16 (1 × 5) filtros cada, ao longo do eixo do tempo com *padding* e *stride* de 1.

O *output* dessas camadas, então, é processado por quatro módulos multifiltro (*mf_mod*), esquematizado conforme a Figura 11. O módulo começa com uma camada de *average pooling* (*m* × 1), passa por *batch normalization* segue para seis camadas convolucionais paralelas de filtros cujo comprimento variam entre (1 × 32) e (1 × 256), após as quais há uma camada de concatenação e uma última camada convolucional "*bottleneck*" para reduzir a dimensionalidade.

Com cada um destes módulos, tentamos atingir dois objetivos:

1) *Pooling* pelo eixo da frequência para sumarizar as *mel-bands*; e
2) *Matching* do sinal com uma variedade de filtros que são capazes de detectar longas dependências temporais. [...] Em um sistema tradicional, isso poderia ser referido como algum tipo de *comb filterbank* (SCHREIBER; MÜLLER, 2018).

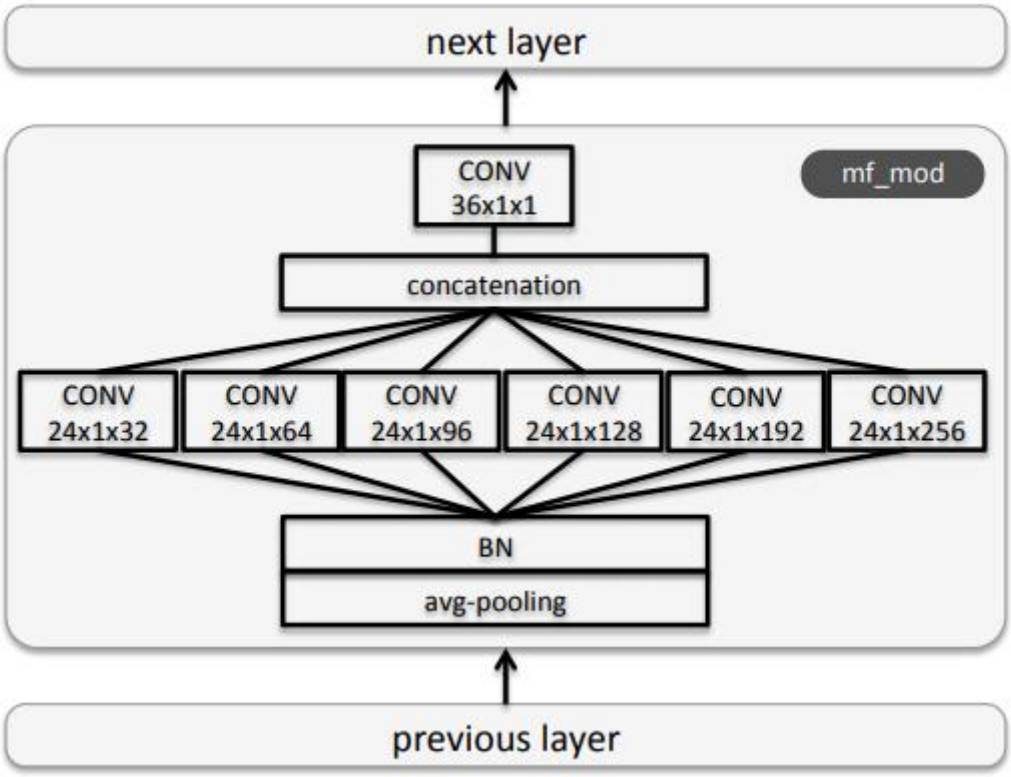


Figura 11: Arquitetura de um módulo multifiltro *mf_mod* (extraído de SCHREIBER; MÜLLER, 2018).

A Figura 11 esquematiza a arquitetura de um módulo multifiltro *mf_mod*. Cada um consiste em uma camada de *average pooling, batch normalization*, seis camadas convolucionais paralelas, uma camada de concatenação e uma camada "*bottleneck*". A função de ativação para todas as camadas convolucionais é ELU.

A camada densa foi feita pelos autores com o propósito de classificar as *features* detectadas pelas camadas convolucionais. Após uma *batch normalization*, é adicionada uma camada de *Dropout* (p = 0.5) para evitar *overfitting*, seguindo para duas camadas densas (cada uma precedida por *batch normalization*). As duas primeiras camadas densas utilizam ELU como função de ativação, enquanto a última utiliza *softmax*. A função de perda utilizada é *categorical cross-entropy*, como costuma ser padrão para problemas de classificação multiclasse. A rede neural convolucional apresenta 2.921.042 parâmetros treináveis.

4.2.2 B-RNN

As redes neurais recorrentes bidirecionais (B-RNN), introduzidas por Schuster e Paliwal (1997), são reconhecidamente um bom modelo para tarefas como reconhecimento de fala (SCHUSTER, 2020), também mostrando-se valiosas para estimativa de tempo. Böck e Schedl (2011) propõem o uso de uma BLSTM (LSTM bidirecional) para estimativa de tempo, sendo tal modelo considerado o estado-da-arte inclusive por Schreiber e Müller (2018). A utilização da bidirecionalidade faz sentido, uma vez que, durante o processamento do *input* pela rede neural, não só o contexto anterior como também o contexto futuro de um momento de uma música pode ser utilizado para determinar o *output* de tempo.

Tendo em vista os fatores citados acima, a rede neural recorrente bidirecional foi escolhida como um tipo de modelo adequado a ser trabalhado neste projeto. A primeira referência para a construção da arquitetura foi a CNN explicitada na subseção anterior; A fim de manter uma certa coerência entre os dois modelos, assim como evitar a utilização de outros sistemas que não a própria rede para a detecção de tempo, suas camadas e o objetivo de cada uma devem ser levadas em conta. Assim, a ideia de base seria substituir as camadas convolucionais por camadas recursivas. A camada densa seria mantida.

O objetivo das camadas recorrentes é identificar *onsets*, analisando as frequências do espectrograma mel, e identificar suas dependências temporais. *Onset* é considerado o momento exatamente início de uma batida, e suas detecções são necessárias para encontrar sua periodicidade. Com a recorrência bilateral, espera-se que seja possível detectar dependências temporais suficientemente longas.

A idéia principal era contar com unidades LSTM para as camadas recorrentes; entretanto, por limitações técnicas, unidades recorrentes simples foram escolhidas. O treinamento de LSTM demandaria consideravelmente mais tempo, e como o número de horas disponíveis da máquina virtual era limitado, acabou-se optando pelas unidades recorrentes simples.

Na arquitetura feita para o experimento, o *input* passa por normalização, e então é enviado para as camadas recorrentes. Foram definidas 3 camadas para cada direção com 25 unidades "*SimpleRNN*" cada (resultando num total de 6 camadas e 150 unidades). A quantidade de camadas e unidades foi inspirada pela BLSTM de Böck e Schedl (2011). Teve-se a ideia de testar mais camadas e unidades, o que não pôde ser concretizado por

limitações técnicas. As camadas recorrentes utilizam função de ativação *tanh* (tangente hiperbólica).

A etapa seguinte é o processamento por camadas densas, conforme mostrado no modelo da subseção CNN, cujo propósito é classificar as *features* detectadas pelas camadas recorrentes. Primeiramente, o *output* das camadas recorrentes passa por um average pooling (5 × 1). Logo após, seguem exatamente as mesmas camadas densas da CNN: após *batch normalization*, é adicionada uma camada de *Dropout* (p = 0.5) para evitar *overfitting* e então uma camada densa, seguindo para mais duas camadas densas consecutivas precedidas por *batch normalization*. As duas primeiras camadas densas utilizam ELU como função de ativação, enquanto a última utiliza *softmax*.

Por se tratar um problema de classificação multiclasses, a função de erro escolhida foi *categorical cross entropy*, assim como para a CNN. O otimizador escolhido para o keras foi SGD (*Stochastic Gradient Descent*) com *clipping* value de 5 para evitar a explosão de gradiente. O valor para *learning* rate é 0.001 (também conforme a CNN), e para *momentum* é 0.9. A rede neural recorrente bidirecional apresenta um total de 6.583.772 parâmetros treináveis.

5 Experimentos Computacionais

5.1 *Dataset*

Para a realização do experimento, a base de dados (*dataset*) deve conter peças musicais completas (ou trechos de peças), as quais podem consistir em apenas uma linha de bateria, sem necessidade de outros instrumentos.

Os *datasets* selecionados foram: *ACM Mirum* (PEETERS; FLOCON-CHLOET, 2012), *Extended Ballroom* (MARCHAND; PEETERS, 2016), *GiantSteps Tempo* (KNEES et al., 2015), *GiantSteps MTG*[14], *Groove*, *GTzan* (TZANETAKIS; COOK, 2002), *Hainsworth* (HAINSWORTH; MACLEOD, 2004), *LMD* (RAFFEL, 2016) e *SMC Mirum* (HOLZAPFEL et al., 2012), totalizando 12.550 peças. Foram escolhidos por sua variedade de músicas (evitando repetição de dados no treinamento, e contendo gêneros musicais distintos) e sua disponibilidade gratuita e livre para uso em estudos acadêmicos. Esses *datasets* também possuem anotações públicas disponíveis na *Web* contendo o tempo de cada peça, o que também foi considerado um fator essencial durante a escolha. Abaixo, uma lista com mais detalhes sobre cada *dataset* e os números de exemplos:

- *ACM Mirum* (1.410 *samples*): Arquivos de música de 30 segundos com extensão *.wav*, com a maior parte das músicas do gênero *pop* e *rock*;

- *Extended Ballroom* (3.826 *samples*): Arquivos de música de 30 segundos com extensão *.mp3*. Contém músicas de 13 tipos diferentes de gêneros considerados pouco representados em *datasets*, como *Chacha*, Salsa, *Foxtrot*, Samba, Valsa e outros. Apresenta músicas com métricas diferentes de 4/4, como 3/4;

- *GiantSteps Tempo* (664 *samples*): Arquivos de música de 2 minutos com extensão *.mp3*. O gênero de todas as faixas é EDM (*Electronic Dance Music*);

[14] Disponível em <https://github.com/GiantSteps/giantsteps-mtg-key-dataset>. Acesso em: 13 fev. 2020.

- *GiantSteps MTG* (1.158 *samples*): Arquivos de música de 2 minutos com extensão *.mp3*. O gênero de todas as faixas é EDM (*Electronic Dance Music*). As anotações de tempo não são provenientes do *dataset* original: foram criadas por (SCHREIBER; MÜLLER, 2018).

- *Groove* (443 *samples*): Arquivos de linha de bateria de aproximadamente 10 segundos com extensão *.wav*. Apresenta linhas populares em diversos gêneros, tais como *punk*, *rock*, *jazz* e *gospel* e em métricas variadas, incluindo 3/4 e 6/8. É possível considerar que o Groove não sofre de viés de estilo, uma vez que as faixas foram performadas por diversos bateristas. Embora o *dataset* original seja extenso, muitas das peças contêm apenas informações que não são relevantes para esse estudo. Foram excluídas do *dataset* as peças que consistiam apenas de viradas de bateria (não constituindo uma linha de música) ou não apresentassem arquivo de áudio no formato .wav.

- *GTzan* (999 *samples*): Arquivos de música de 30 segundos 22050 Hz *Mono* 16-*bit* com extensão *.wav*. Contém músicas de 10 gêneros diferentes (100 samples de cada). As gravações são provenientes de diversas fontes, tais como CDs, rádio e gravações de microfone.

- *Hainsworth* (222 *samples*): Arquivos de música de duração variada, sendo a maioria com duração entre 40 e 60 segundos, com extensão *.wav*. O *dataset* contém músicas de diversos gêneros, tais como *folk*, *jazz*, música clássica etc.

- *LMD* (3.611 *samples*): Arquivos de música de 30 segundos com extensão *.mp3*. A maioria das faixas é dos gêneros *pop* e *rock*, embora também contenha outros estilos como música clássica. As anotações de tempo não são provenientes do *dataset* original: foram criadas por (SCHREIBER; MÜLLER, 2018).

- *SMC* (217 *samples*): Arquivos de música de 40 segundos com extensão *.wav*. Inclui músicas de diversos estilos, tais como *música clássica*, *música Romântica*,

trilhas sonoras de filmes, *blues* etc. Segundo o autor, este *dataset* foi criado com o intuito de ser especialmente desafiador para sistemas de análise de ritmo.

O número de músicas pode parecer pequeno quando comparado a pesquisas em outras áreas envolvendo *deep learning*, como a de Krizhevsky, Sutskever e Hinton (2012) que reúne mais de um milhão de imagens para o *dataset*. Entretanto, por questões como *copyright*, que resulta em baixa disponibilidade de dados públicos, baixa disponibilidade de anotações (MONTECCHIO; ROY; PACHET, 2019), assim como descentralização dos dados (não há uma plataforma única reconhecidamente que hospede e distribua os *datasets* e anotações), ainda não há uma gama mais extensa de dados disponíveis para treinamento no que tange à música. Entende-se, portanto, que há uma carência na literatura de repositórios livres contendo um bom número de peças musicais para treinamentos de modelos, até para servirem como *benchmark* de comparação entre diferentes abordagens.

Como abordado na lista de *datasets*, apenas o *Groove* contém peças que consistem exclusivamente de uma linha de bateria; Isso pode apresentar um obstáculo para determinar se há mais facilidade para detectar corretamente o tempo quando há apenas percussão, uma vez que a quantidade de dados é um fator crucial para a performance da rede neural (GOODFELLOW; BENGIO; COURVILLE, 2016).

5.2 Ambiente Computacional

Os treinamentos da CNN e B-RNN foram executados em uma máquina virtual da *Azure* (utilizando horas de crédito gratuitas fornecidas pelo serviço), sistema operacional *Windows 10*, utilizando uma GPU Tesla K80. A linguagem de desenvolvimento utilizada é o *Python*, utilizando também bibliotecas para desenvolvimento de redes neurais *TensorFlow* e *keras*. A biblioteca *librosa* é utilizada para tratamento do sinal de áudio.

5.3 Treinamento

O código utilizado para treinamento e teste foi publicado por (SCHREIBER; MÜLLER, 2019) e modificado pela autora para se adequar às necessidades do experimento e suas particularidades.

É preciso separar *datasets* para a etapa de treinamento e para a etapa de testes.

Para evitar resultados enviesados, foi decidido que *datasets* que participam do treinamento da rede não participam do teste (e vice-versa), com exceção do *Groove* (por ser o único *dataset* contendo linha de bateria). Também buscou-se ter uma variedade considerável de gêneros para o treinamento da rede.

Com base nesses critérios, a seguinte divisão dos *datasets* apresentados na 4.1 foi feita:

- Os *datasets* selecionados para a etapa treinamento dos modelos foram *Extended Ballroom*, *GiantSteps MTG*, *Hainsworth*, *LMD* e parte do *Groove* (90%); e
- Os *datasets* selecionados para avaliação dos modelos foram *ACM*, *GiantSteps Tempo*, *GTzan*, *SMC* e parte do *Groove* (10%).

Dentre os *datasets* selecionados para treinamento, é necessário selecionar uma parte de cada um para o treinamento da rede propriamente dito e outra parte para a validação da rede, de modo a mensurar o *validation loss*. Nesse experimento, foi escolhida a proporção de 20% de cada base de dados para validação, com exceção do *Groove* — por conter poucas músicas (apenas 443 peças) e ser o único contendo apenas linhas de bateria, de modo que é a única base de dados que fornece músicas tanto para o treinamento quanto para o teste, foi dividido com a proporção 80% treino, 10% validação e 10% teste. As proporções 80-20 e 80-10 (com os 10% restantes entrando na etapa de teste/avaliação) são frequentemente utilizadas em experimentos no geral.

Assim, os números de músicas para treino e avaliação dos modelos foram, respectivamente, 7.410 e 1.805, totalizando 9.215 faixas.

Como abordado na seção de Representação de *input*, não é fornecido o sinal de música inteiro como input para a rede neural. O espectrograma log-Mel é comprimido (ou expandido, se a duração do áudio inteiro for menor que 11.9 s) e então cortado em pequenas janelas. Esse processo aumenta, portanto, o número de exemplos que são fornecidos para a rede durante o treinamento.

Para o treinamento da CNN, foram escolhidos os mesmos critérios de (SCHREIBER; MÜLLER, 2019): Otimizador Adam (KINGMA; BA, 2014), uma *learning rate* de 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, ε = `1e-8`, e um *batch size* de 32. Para

evitar *overfitting*, também há *early stopping* quando a *validation loss* não apresenta decréscimo nos últimos 100 Epochs.

Para a B-RNN, o otimizador escolhido foi SGD (*stochastic gradient descent*, traduzido como gradiente descendente estocástico) com *momentum* 0.9 e *clip value* 5 para evitar explosão do gradiente. A *learning rate* é 0.001, e também há *early stopping* seguindo os mesmos critérios da CNN.

Os treinamentos da CNN e B-RNN duraram um total de 481 Epochs (em 10 horas) e 410 Epochs (em 25 horas), respectivamente, ambos encerrados por *early stopping*. Cada modelo foi treinado apenas uma vez, sem haver retreinamento nem reajustes de hiperparâmetros.

5.4 Avaliação

Como citado anteriormente, os *datasets* para a etapa de avaliação foram *ACM*, *GiantSteps Tempo*, *GTzan*, *SMC* e parte do *Groove* (10%), totalizando 3.335 músicas diferentes.

Como a entrada para a rede é frequentemente de uma dimensão menor do que o espectrograma da faixa completa, para predizer o tempo de uma música completa, é preciso analisar o tempo "localmente" em vários trechos, para, então, determinar o que pode ser o valor do tempo "global". É seguido o método de Schreiber e Müller (2018):

> A fim de estimar o tempo global para uma faixa, nós calculamos múltiplas saídas de ativação usando uma janela deslizante com sobreposição (*overlap*) de metade da janela, i.e., um tamanho de salto de 128 frames ≈ 5.96 s. As ativações são por classe, e então [...] a classe de tempo com a maior ativação é escolhida como o resultado.

É comum que as máquinas acabem interpretando o valor do tempo como 2, 3 vezes maior (ou menor) que o tempo percebido por humanos. Na verdade, os próprios seres humanos por vezes entram em discordância quanto ao valor do tempo percebido por cada um (HÖRSCHLAGER et al., 2015). Uma pessoa pode, por exemplo, interpretar o tempo de uma música como 60 bpm, enquanto outra interpreta como 120 bpm. Sendo assim, é de interesse simular a audição humana e suas imprecisões, não descartando resultados que sejam múltiplos do tempo considerado original de uma música; Para isso,

costumam ser introduzidas mais de um nível de acurácia para a medição (acurácias secundárias) em estudos de estimativas de tempo (HÖRSCHLAGER et al., 2015).

Para a avaliação dos resultados, são utilizados 3 tipos de acurácia diferentes, cujos critérios foram também inspirados por Schreiber e Müller (2018): Acurácia0, que considera diretamente os valores detectados (arredondados para o número inteiro mais próximo) que forem equivalentes ao tempo anotado; Acurácia1, que considera valores detectados com desvio de $\pm$4% do valor anotado do tempo; e Acurácia2, que considera valores detectados duas ou três vezes maiores do que o tempo anotado, também considerando uma margem de $\pm$4%.

A introdução da Acurácia1 é interessante levando em consideração a decisão do trabalho de encarar a estimativa do tempo como um problema de classificação, na qual o tempo de uma peça é classificado como um número inteiro dentro de um intervalo definido previamente. Como o sistema arredonda o tempo para o número inteiro mais próximo, é possível que haja uma diferença sutil de 1 bpm entre a detecção e a anotação, por exemplo. Já acurácias secundárias como a Acurácia2 são frequentemente utilizadas em trabalhos de estimativa de tempo devido ao chamado "*tempo octave error*", introduzido no capítulo 3. Não só os critérios se apresentam interessantes, como também facilita o processo de comparação de resultados.

O artigo de Schreiber e Müller (2018) não fornece resultados para os *datasets* *GiantSteps Tempo* e *Groove*, pois não foram utilizados por eles. Entretanto, a rede neural convolucional treinada resultante do artigo está disponibilizada publicamente no GitHub[15], de forma que foi possível utilizá-la para prever o tempo dos dois *datasets* que faltavam.

Os resultados obtidos pela CNN para as Acurácias 0, 1 e 2 estão exibidas na Tabela 3. Já para a B-RNN, as Acurácias 0, 1 e 2 estão exibidas na Tabela 4. Os resultados de cada modelo para cada acurácia estão indicados na Tabela 5 (Acurácia0), Tabela 6 (Acurácia1) e Tabela 7 (Acurácia2), nas quais a melhor performance por *dataset* encontra-se em negrito.

[15] Disponível em: <https://github.com/hendriks73/tempo-cnn>. Acesso em: 17 fev. 2020.

Dataset	**Acurácia0** (%)	**Acurácia1** (%)	**Acurácia2** (%)
ACM	39.291	73.759	96.525
Groove	60.465	72.093	93.023
GiantSteps Tempo	27.711	82.982	92.470
GTzan	30.531	64.565	91.892
SMC	11.060	27.189	40.553

Tabela 3: Resultados de Acurácia0, Acurácia1 e Acurácia2 pela CNN.

Dataset	**Acurácia0** (%)	**Acurácia1** (%)	**Acurácia2** (%)
ACM	32.979	72.128	93.121
Groove	58.140	76.744	95.349
GiantSteps Tempo	15.663	69.277	86.295
GTzan	25.225	61.962	85.185
SMC	5.991	18.433	30.415

Tabela 4: Resultados de Acurácia0, Acurácia1 e Acurácia2 pela B-RNN.

Dataset	**CNN**	**B-RNN**	**(SCHREIBER; MÜLLER, 2018)**
ACM	39.3	33.0	**40.6**
Groove	**60.6**	58.1	37.2
GiantSteps Tempo	**27.7**	15.7	27.6
GTzan	30.5	25.2	**36.9**
SMC	11.1	6.0	**12.4**

Tabela 5: Comparação de Acurácia0 entre a CNN, B-RNN, e Schreiber e Müller (2018).

Dataset	**CNN**	**B-RNN**	**(SCHREIBER; MÜLLER, 2018)**
ACM	73.8	72.1	**79.5**
Groove	72.1	**76.7**	62.8
GiantSteps Tempo	**83.0**	69.3	64.6
GTzan	64.6	62.0	**69.4**
SMC	27.2	18.4	**33.6**

Tabela 6: Comparação de Acurácia1 entre a CNN, B-RNN, e Schreiber e Müller (2018).

Dataset	**CNN**	**B-RNN**	**(SCHREIBER; MÜLLER, 2018)**
ACM	96.5	93.1	**97.4**
Groove	93.0	**95.4**	86.0
GiantSteps Tempo	**92.5**	86.3	83.1
GTzan	91.9	85.2	**92.6**
SMC	40.5	30.4	**50.2**

Tabela 7: Comparação de Acurácia2 entre a CNN, B-RNN, e Schreiber e Müller (2018).

5.5 Análise dos Resultados

Os resultados tendem a evidenciar que, no geral, há uma diferença considerável de performance entre a CNN e a B-RNN, indicando uma resposta afirmativa à P1 introduzida no capítulo 4 (“Existe diferença significativa de performance entre os dois modelos?”). Era um resultado esperado, uma vez que a B-RNN foi proposta sem passar por nenhum reajuste de arquitetura ou parâmetros, e nem mesmo múltiplos treinamentos para escolher o que tiver o melhor desempenho.

A performance superior do artigo de Schreiber e Müller (2018) em comparação à da CNN deste trabalho não é surpreendente. Embora o modelo e a representação de *input* sejam as mesmas, existem outras diferenças entre o experimento deles e o deste trabalho. Os critérios de treinamento foram diferentes — Schreiber e Müller (2018) tiveram como critério de *early stopping* não haver melhora na acurácia de validação e foram treinados 3 modelos, dos quais o de melhor performance foi selecionado, enquanto neste trabalho considerou-se não haver decréscimo no erro de validação (*validation loss*). Além disso, também deve-se levar em conta que os modelos propostos não passaram por nenhum reajuste de parâmetros ou modificações na arquitetura.

Como a CNN deste trabalho foi inspirada em um artigo publicado e bem testado, também era esperado que sua performance geral fosse inferior à do artigo original. Entretanto, para os *datasets* que não foram abordados pelo artigo original (*GiantSteps Tempo* e *Groove*), a CNN de (SCHREIBER; MÜLLER, 2018) apresentou resultado inferior a algum dos modelos treinados neste experimento (por vezes, inferior a ambos). Isso pode evidenciar que a escolha de *datasets* pode ser decisiva para a performance das redes neurais, e que a falta de maiores quantidades de músicas disponíveis para treinamento faz diferença considerável na performance.

O *dataset* com o menor desempenho apresentado foi o *SMC*. Não é um resultado surpreendente, uma vez que sua intenção, segundo os criadores, é ser desafiador para *beat tracking* (e, como consequência, tempo). Muitas faixas incluem um tempo não muito bem definido, assim como faixas apenas de instrumentos como violão, o que dificulta a estimativa do tempo.

O maior valor de Acurácia 0, após o Groove, é o *ACM*. É um resultado que faz sentido, uma vez que a maior parte das suas músicas são de *pop* e *pop-rock*, que são peças bem produzidas com poucas mudanças de tempo. A Acurácia2 apresentou os maiores valores em todos os casos, o que é esperado, uma vez que é a acurácia secundária menos "rigorosa" em questão de precisão. Entretanto, como seu valor ultrapassou 90% em todos os *datasets* (com exceção do *SMC*), é possível afirmar que ambas as redes neurais do experimento apresentaram resultados satisfatórios, de forma coerente com a audição humana.

Interessantemente, o *dataset Groove* apresentou valores relativamente altos nas Acurácias 0, 1 e 2 por ambos os modelos. De fato, foi o *dataset* que apresentou os maiores valores de todas as Acurácias 0, e o único *dataset* em que a B-RNN superou a CNN em performance, conforme já comentado anteriormente. Por ser o único *dataset* apenas com linhas de bateria e ter apenas 443 faixas — isto é, apenas 3.53% do total de músicas —, era esperado que fosse apresentar um desempenho inferior se comparado aos *datasets* de peças completas, (que são 12.107 faixas, 96.47% do total). Isso poderia apoiar uma resposta afirmativa à P2 (A estimativa de tempo é mais precisa para inputs oriundos de áudios que contêm apenas linha de bateria do que para áudios que contém múltiplos instrumentos?), de forma que a detecção de tempo é mais precisa quando é realizada sobre arquivos de áudio contendo apenas a linha de bateria.

Algumas das imprecisões do Groove foram a estimativa de 145 bpm fornecida para uma música originalmente de 290 bpm, o que pode ter ocorrido por ser um tempo consideravelmente incomum, especialmente no que tange aos *datasets* utilizados. Um erro de estimativa foi a atribuição de 84 bpm a uma música originalmente de 60, mas que possui uma métrica não muito comum para os *datasets* de 6/8, o que pode contribuído para o erro; Não obstante, para outra música em métrica 6/8 a detecção foi precisa o suficiente (uma música de 70 bpm foi classificada como 140 bpm, exatamente o dobro do valor real).

Ainda assim, é preciso considerar também que o *Groove* foi utilizado para o treinamento da rede, e com um número muito pequeno de *inputs* (44 faixas). Mesmo que o *dataset* tenha sido performado por bateristas diferentes e contenha uma boa variedade de métrica, tempos e estilos, isso pode ter sido um fator que influenciou na alta performance para seu *dataset* de teste. Para dar mais força à resposta da pergunta P2, seria ideal realizar mais testes com bancos de dados diferentes quando estes forem disponibilizados de forma ampla.

6 Conclusão

6.1 Considerações Finais

Este trabalho realizou a modelagem, treinamento e teste de dois modelos de redes neurais (CNN e B-RNN) que, ao fornecer um *input* de espectrograma mel (gerado a partir de uma música), retornam uma estimativa de seu tempo em bpm (batidas por minuto).

Ambos os modelos apresentaram acurácias satisfatórias (ultrapassando 90% para todos os *datasets* de teste, com exceção de um, se permitidos valores múltiplos do tempo original). Entretanto, a CNN apresentou uma performance geral melhor do que a B-RNN, o que era esperado, uma vez que o modelo da CNN foi reproduzido de Schreiber e Müller (2018) que já apresentava alto desempenho, mesmo que houvesse diferenças entre hiperparâmetros, *datasets* e *inputs* deste experimento para o deles.

Não obstante, a performance da B-RNN sobre o *dataset Groove* (que contém apenas áudios com linhas de bateria) foi superior à da CNN deste experimento, um resultado que não era esperado: de três tipos de acurácia estabelecidos, a B-RNN apresentou o melhor resultado em dois. Além disso, o Groove foi o *dataset* que apresentou as melhores acurácias pelos modelos deste experimento. Evidentemente, deve ser considerada a possibilidade de viés pelo fato de que parte do *dataset* foi utilizada para treinamento e a outra parte para a avaliação em si; Entretanto, considerando que o *dataset* foi composto de peças tocadas por vários bateristas diferentes (evitando um forte viés de estilo de performance), e que o número de inputs de linha de bateria foi muito menor (apenas 3.53% do *dataset*) do que o de músicas com diversos instrumentos, tende-se a considerar que a estimativa de tempo é mais precisa para peças apenas de linhas de percussão.

Embora fosse esperado que a rede neural treinada de Schreiber e Müller (2018) sempre apresentasse desempenho superior à CNN deste artigo, ela não apresentou desempenho superior para os *datasets GiantSteps Tempo* e *Groove*, que não foram abordados pelo trabalho. A rede neural deles não foi treinada com áudios apenas de percussão, e o número total de músicas para treinamento foi inferior ao deste experimento, o que pode evidenciar que a escolha de *datasets* de treino e quantidade de músicas fazem

diferença significativa no desempenho de uma rede neural.

Assim, a resposta evidenciada para a P1 (Existe diferença significativa de performance entre os dois modelos?) e P2 (A estimativa de tempo é mais precisa para inputs oriundos de áudios que contêm apenas linha de bateria do que para áudios que contém múltiplos instrumentos?) neste experimento, com base nas evidências dos resultados, tende a ser afirmativa para ambas.

6.2 Limitações e Trabalhos Futuros

Algumas limitações foram constantemente presentes durante a realização do experimento, representando um obstáculo para otimização dos resultados. Algumas delas foram:

- Limitações técnicas. A autora não dispõe de uma GPU para treinamento de redes neurais, de modo que os treinamentos tiveram de ser realizados por meio de uma máquina virtual. Não obstante, a máquina virtual pôde ser utilizada apenas através de créditos de horas gratuitos fornecidos pela plataforma do Azure; assim, pela limitação de tempo disponível, apenas um treinamento foi realizado para cada modelo de rede neural, sem nenhum reajuste de hiperparâmetros ou de arquitetura. A rede neural recorrente bidirecional teve de ser simplificada para poder ser treinada completamente, sem poder utilizar unidades LSTM (seu treinamento demoraria muito mais).
- Pouca disponibilidade de *datasets* de música. É sabido que a performance de uma rede neural pode ser muito influenciada pela quantidade de dados disponíveis para seu treinamento (GOODFELLOW; BENGIO; COURVILLE, 2016); A quantidade de músicas, principalmente com anotações, disponíveis irrestritamente para uso acadêmico é razoavelmente pequena por uma série de questões tais como *copyright*. Este é um problema recorrente no contexto de *Music Information Retrieval*, e não é tão simples de ser solucionado.
- Disponibilidade de *datasets* de percussão ainda menor. Embora esteja contida na limitação do tópico anterior, essa limitação merece ser abordada a parte. Apenas um *dataset* (*Groove*) é publicamente disponibilizado contendo peças apenas de bateria, contendo apenas 443 músicas. Isso representa um obstáculo para

determinar se a estimativa de tempo de fato é mais efetiva em peças apenas de percussão, assim como para a própria performance da rede neural em si.

Algumas sugestões para trabalhos futuros incluem:

- Aprimorar os modelos de redes neurais, especialmente a B-RNN: pode-se, por exemplo, substituir as camadas "*SimpleRNN*" por LSTM; variar o número de camadas e/ou unidades.
- Recriar o experimento utilizando espectrogramas log-mel.
- Treinar o modelo mais vezes para selecionar o que apresentar melhores resultados.
- Testar outros hiperparâmetros: pode-se testar outros otimizadores (como o Adam para a B-RNN, por exemplo) e outros critérios para *early stopping* (como reproduzir a idéia de (SCHREIBER; MÜLLER, 2018) de considerar a falta de aumento da *validation accuracy*).
- Criação ou busca aprofundada de *datasets* incluindo apenas linhas de percussão.

Referências Bibliográficas

ALONSO, M.; DAVID., B.; RICHARD, G. Tempo and beat estimation of musical signals. In: **5th International Society for Music Information Retrieval (ISMIR) Conference**, 2004, Barcelona.

BERRY, W. **Structural Functions in Music**. New Jersey: Prentice Hall, 1976. 447p.

BÖCK, S. **Onset, Beat, and Tempo Detection with Artificial Neural Networks**. ca. 2010. Disponível em: <http://mir.minimoog.org/sb-diploma-thesis>. Acesso em 23 ago. 2019.

BÖCK, S.; KREBS, F.; WIDMER, G. Accurate tempo estimation based on recurrent neural networks and resonating comb filters. **Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)**, p. 625-631, 2015.

BÖCK, S.; SCHEDL, M. Enhanced beat tracking with context-aware neural networks. **Proceedings of the 14th International Conference on Digital Audio Effects**, p. 135-139, set. 2011.

BRIOT, J.; **Deep Learning Techniques for Music Generation (1)**. 2018. Disponível em: <http://www-poleia.lip6.fr/~briot/cours/unirio/Transparents/dlmg-1-cm-introduction.pdf>. Acesso em: 18 ago. 2019.

BRIOT, J.P.; HADJERES, G.; PACHET, F. Deep Learning Techniques for Music Generation – A Survey. **arXiv:1709.01620**, 2019.

CASEY, M.A. et al. Content-Based Music Information Retrieval: Current Directions and Future Challenges. **Proceedings of the IEEE**, v. 96, n.4, apr. 2008.

CEMGIL, A.T.; BERT, K. Monte Carlo Methods for Tempo Tracking and Rhythm Quantization. **Journal of Artificial Intelligence Research**, v. 18, p. 45-81, 2003.

DECHTER, R. Learning While Searching in Constraint-Satisfaction-Problems. **Proceedings of the 5th National Conference on Artificial Intelligence**, v.1, 1986. p. 178-183.

DONG, H. et al. MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. In: **AAAI Conference on Artificial Intelligence**, 2018, New Orleans.

DONG, M. Convolutional Neural Network Achieves Human-level Accuracy in Music Genre Classification. **arXiv:1802.09697**, 2018.

DOWNIE, J.S. Music information retrieval. **Annual Review of Information Science and Technology**, v. 37, n.1, p. 295-340, 2003.

ECK., D.; SCHMIDHUBER, J. Finding temporal structure in music: blues improvisation with LSTM recurrent networks. **Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing**, 2002, Martigny. p. 747-756.

ELOWSSON, A. Beat tracking with a cepstroid invariant neural network. **Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)**, p. 351-357, 2016.

FLANNERY, M.C. Rhythm & Form in Nature. **The American Biology Teacher**, v. 52, n. 2, p. 118-121, 1990.

FLATTELY, F.W. Rhythm in Nature. **Science Progress in the Twentieth Century (1919-1933)**, v. 14, n. 55, p. 418-426, jan. 1920. Disponível em <www.jstor.org/stable/43431585>. Acesso em: 17 feb. 2020.

GKIOKAS, A.; KATSOUROS, V.; CARAYANNIS, G. Reducing tempo octave errors by periodicity vector coding and SVM learning. In: **13th International Society for Music Information Retrieval Conference**, 2012, Porto.

GÓMEZ, E. et al. Music Information Retrieval: Overview, Recent Developments and Future Challenges. In: **17th International Society for Music Information Retrieval (ISMIR) Conference**, 2016, New York.

GOODFELLOW, I. et al. Generative adversarial nets. **Advances in neural information processing systems**, v. 27, p. 2672-2680, 2014.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. Cambridge: MIT Press, 2016.

GOUYON, F. et al. An experimental comparison of audio tempo induction algorithms. **IEEE Trans. Speech Audio Processing**, v. 14, n. 5, p. 1832-1844, mai. 2006.

GOUYON, F.; DIXON, S. A review of automatic rhythm description systems. **Computer Music Journal**, v. 29, n. 1, p. 34-54, 2005.

GROSCHE, P.; MÜLLER, M.; SERRÀ, J. Audio Content-Based Music Retrieval. **Dagstuhl Follow-Ups**, v. 3, p. 157-154, 2012.

HAINSWORTH, S.; MACLEOD, M.; Particle filtering applied to musical tempo tracking. **EURASIP J. on Applied Signal Processing**, v. 15, p. 2385-2395, 2004.

HAN, W. et al. An efficient mfcc extraction method in speech recognition. 2006 In: **IEEE International Symposium on Circuits and Systems**, Island of Kos, 2006.

HAUSEN, M. et al. Music and speech prosody: a common rhythm. **Frontiers in Psychology**, v. 4, 2013.

HOLZAPFEL, A. et al. Selective sampling for beat tracking evaluation. **IEEE Trans. on Audio, Speech, and Language Processing**, v. 20, n. 9, p. 2539-2548, 2012.

HÖRSCHLAGER, F. et al. Addressing Tempo Estimation Octave Errors in Electronic Music by Incorporating Style Information Extracted from Wikipedia. **Proceedings of the 12th Sound and Music Conference (SMC)**, 2015.

KINGMA, D.P.; BA, J.L. Adam: A method for stochastic optimization. **arXiv:1412.6980**, 2014.

KLAPURI, A.P.; ERONEN, A.J.; ASTOLA, J.T. Analysis of the meter of acoustic musical signals. **IEEE Transactions on Audio, Speech, and Language Processing**, v. 14, p. 342-355, 2006.

KNEES, P. et al. Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections. **Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)**, pp. 364-470, 2015.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. ImageNet Classification with Deep Convolutional Neural Networks. **Advances in neural information processing systems**, v. 25, p. 1097-1105, 2012.

KUBATZKI, J.; Music in Rites. Some Thoughts about the Function of Music in Ancient Greek Cults. **eTopoi Journal for Ancient Studies**, v.5, p. 11-17, 2016.

LI, T.L.; CHAN, A.B.; CHUN, A.H. Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network. **Proceedings of the International MultiConference of Engineers and Computer Scientists**, v. 1, 2010.

LIU, W. et al. SphereFace: Deep Hypersphere Embedding for Face Recognition. In: **Conference on Computer Vision and Pattern Recognition**, 2017, Honolulu.

LOGAN, B.; SALOMON, A. **A Music Similarity Function Based on Signal Analysis**. 2001.

MARCHAND, U.; PEETERS, G. Scale and shift invariant time/frequency representation using auditory statistics: application to rhythm description. In: **IEEE International Workshop on Machine Learning for Signal Processing**, Salerno, set. 2016.

MCAULEY, J.D. Tempo and Rhythm. In: JONES, M.; FAY, R.; POPPER, A. **Music Perception**. New York: Springer, 2010. p. 165-199.

MEDAWAR, P.B.; MEDAWAR, J.S. **Aristotle to zoos**: A Philosophical Dictionary of Biology. Cambridge: Harvard University Press, 1983. 320p.

MONTECCHIO, N.; ROY, P.; PACHET, F. The Skipping Behavior of Users of Music Streaming Services and its Relation to Musical Structure. **arXiv:1903.06008**, 2019.

MÜLLER, M. Fourier Analysis of Signals. In: MÜLLER, M. **Fundamentals of Music Processing**. Switzerland: Springer International Publishing, 2015. p. 39-57.

OLIVER, N.; FLORES-MANGAS, F. MPTrain: a mobile, music and physiology-based personal trainer. **Proceedings of the 8th conference on Humancomputer interaction with mobile devices and services**, 2006.

ORAMAS, S. et al. Multimodal Deep Learning for Music Genre Classification. **Transactions of the International Society for Music Information Retrieval**, v. 1, n. 1, p. 4-21, 2018.

PEETERS, G.; FLOCON-CHLOET, J. Perceptual tempo estimation using GMM-regression. **Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies (MIRUM)**, p. 45-50, 2012.

RAFFEL, C. **Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching**. Dissertação (Doutorado), Columbia University, New York, 2016.

SCHEDL, M. **Automatically extracting, analyzing, and visualizing information on music artists from the World Wide Web**. Dissertação (Doutorado em Ciências). Johannes Kepler University, Linz, 2008.

SCHINDLER, A.; LIDY, T.; BÖCK, S. Deep learning for MIR tutorial. **arXiv:2001.05266**, 2020.

SCHREIBER, H.; MÜLLER, M. A Single-Step Approach to Musical Tempo Estimation Using a Convolutional Neural Network. In: **19th International Society for Music Information Retrieval Conference (ISMIR)**, 2018, Paris.

SCHREIBER, H.; MÜLLER, M. Musical Tempo and Key Estimation using Convolutional Neural Networks with Directional Filters. **Proceedings of the 16th Sound and Music Computing Conference**, 2019.

SCHREIRER, E.D. Tempo and beat analysis of acoustic musical signals. **The Journal of the Acoustical Society of America**, n. 103, v. 1, p. 588-601, 1998.

SCHUSTER, M.; PALIWAL K.K. Bidirectional Recurrent Neural Networks. **IEEE Transactions on Signal Processing**, v. 45, p. 2673-2681, 1997.

SEMAMA, P. **Linguagem e Poder**. Brasília: Editora Universidade de Brasília, 1981.

STEVENS, S.S.; VOLKMANN, J.; NEWMAN, E.B. A Scale for the Measurement of the Psychological Magnitude Pitch. **J.A.S.A.**, v. 8, jan. 1937.

STEVENS, S.S.; VOLKMANN, J.; NEWMAN, E.B. A Scale for the Measurement of the Psychological Magnitude Pitch. **J.A.S.A.**, v. 8, jan. 1937.

SUITS, B.H. **Physics of Musics - Notes**: Frequencies for equal-tempered scale, A_4 = 440 Hz. 1998. Disponível em: <https://pages.mtu.edu/~suits/notefreqs.html>. Acesso em 17 fev. 2020.

TEIE, D. A Comparative Analysis of the Universal Elements of Music and the Fetal Environment. **Frontiers in Psychology**, v. 7, 2016.

TODD, P. M. A Connectionist Approach to Algorithmic Composition. **Computer Music Journal**, v. 13, n. 4, p. 27-43, 1989.

TZANETAKIS, G.; COOK, P. Musical genre classification of audio signals. **IEEE Transactions on Speech and Audio Processing**, v. 10, n. 5, p. 293-302, 2002.

UITDENBOGERD, A.L.; CHATTARAJ, A.; ZOBEL, J. Music IR: Past, Present and Future. In: **1st International Society for Music Information Retrieval (ISMIR) Conference**, 2000, Plymouth.

WU, F.F. Musical tempo octave error reducing based on the statistics of tempogram. In: **2015 23rd Mediterranean Conference on Control and Automation (MED)**, 2015, Torremolinos.

YANG, L.C.; CHOU, S.Y.; YANG, Y.H. MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation. **arXiv:1703.10847**, 2017.

ZHANG, F.; MENG, H.; LI, M. Emotion extraction and recognition from music. In: **2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery**, 2016, Changsha.

ZHANG, M.; BOCKO, M.; BEAUCHAMP, J. Temporal analysis, manipulation, and resynthesis of musical vibrato. **Proceedings of Meetings on Acoustics**, v. 22, 2005.