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

Relatórios Técnicos
do Departamento de Informática Aplicada
da UNIRIO
n° 0006/2020

Modelos de *Deep Learning* para Estimativa de Tempo em Músicas

Mila Soares de Oliveira de Souza
Pedro Nuno de Souza Moura
Jean-Pierre Briot

Departamento de Informática Aplicada

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
Av. Pasteur, 458, Urca - CEP 22290-240
RIO DE JANEIRO – BRASIL

Relatórios Técnicos do DIA/UNIRIO, No. 0006/2020 Julho, 2020

Modelos de *Deep Learning* para Estimativa de Tempo em Músicas

Mila Soares de Oliveira de Souza[1] Pedro Nuno de Souza Moura[1]
Jean-Pierre Briot[1 2]

[1]Escola de Informática Aplicada – Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
[2]Computer Science Laboratory of Paris 6 (LIP6) – Sorbonne Université, CNRS – Paris, France

milasoaresdeoliveira@gmail.com, pedro.moura@uniriotec.br, Jean-Pierre.Briot@lip6.fr

Abstract. This paper proposes the training and evaluation of 2 neural network models (1 CNN and 1 B-RNN, convolution and recurrent-based) which perform the tempo estimation of musical pieces. The first model implementation comes from its original article, while the second model was implemented in this paper based on the first one. We have designed and constructed an extensive dataset (12.550 samples in total) for conducting our comparative quantitative and qualitative evaluation. The 2 trained models' performances are also compared to one state of the art model. The paper reports about results and analyzing them, lessons learned and future prospects.

Keywords: deep learning, music information retrieval, tempo, estimative, bpm

Resumo. Este artigo propõe o treinamento e avaliação de 2 modelos de redes neurais (1 CNN e 1 B-RNN) capazes de estimar o tempo em bpm de uma peça musical. A implementação do primeiro modelo provém de seu artigo original, enquanto o segundo modelo foi implementado neste artigo com base no primeiro. Foi planejado e construído um *dataset* extensivo (12.550 peças no total) para conduzir uma avaliação comparativa quantitativa e qualitativa. As performances dos 2 modelos são comparadas também com a de um modelo estado da arte. Esse artigo apresenta resultados e análises destes, pontos observados, aprendidos e ideias para futuras pesquisas.

Palavras-chave: deep learning, music information retrieval, tempo, estimativa, música

Sumário

1 Introdução

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 e Richard 2004] — que é o próprio tema deste artigo —, identificação de um estilo musical [Oramas et al. 2018] e comparação de similaridade entre duas músicas [Logan e Salomon 2001]. A estimativa do tempo de uma música é considerada uma das tarefas mais fundamentais de MIR [Böck, Krebs e Widmer 2015].

O tempo de uma música corresponde à quantidade de batidas por minuto (bpm) contidas na mesma [Berry 1976]; Em termos genéricos, pode ser entendido como a velocidade em que humanos frequentemente batem os dedos ou os pés conforme ouvem uma música [Böck, Krebs e Widmer 2015].

O estudo da música pela computação e suas aplicações vem recebendo grandes investimentos tanto pelo âmbito acadêmico quanto pela indústria — Como exemplo, o Facebook recentemente divulgou sua pesquisa de separação de fontes musicais no domínio de Waveforms. Sendo o ritmo uma das características importantes da música [Berry 1976], que tem como o tempo uma de suas características mais importantes, mostra-se interessante buscar métodos de detectá-lo com tecnologias modernas e com boa acurácia, e verificar se dentro desta tecnologia há algum modelo que se destaque em performance.

Este artigo tem como objetivo treinar e avaliar dois modelos de redes neurais — uma rede neural convolucional e uma rede neural recorrente bidirecional — que sejam capazes de realizar a estimativa de tempo em bpm de uma música. Busca-se verificar, então, se há algum modelo cuja performance possa ser considerada superior à dos outros dois, apresentando um percentual maior de estimativas corretas para a maioria dos *datasets* de avaliação. Também busca-se verificar se há diferença significativa de precisão quando se analisa uma peça apenas de percussão em relação à análise de peças com mais instrumentos.

2 Base de Dados

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

Os *datasets* selecionados totalizaram 12.550 peças/*samples*. 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 em bpm de cada peça, o que também foi considerado um fator essencial durante a escolha. A Tabela 1 lista seus nomes, número de peças e breve descrição, contendo os gêneros musicais das peças que compõem cada um.

O único *dataset* que contém peças apenas de percussão é o Groove (exclusivamente composto por linhas de bateria). Todos os outros possuem apenas músicas com

outros instrumentos e, possivelmente, linhas vocais. 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 e Courville 2016].

Tabela 1. Datasets selecionados e suas características

Nome	Qtd. de samples	Duração (s)	Extensão	Gêneros
ACM [Peeters e Flocon-Chloet 2012]	1.410	30	.wav	Pop, rock
Extended Ballroom [Marchand e Peeters 2016]	3.826	30	.mp3	Salsa, foxtrot, samba etc.
GiantSteps Tempo [Knees et al. 2015]	664	120	.mp3	EDM
GiantSteps MTG	1.158	120	.mp3	EDM
Groove	443	10-60+	.wav	Reggae, pop, rock, jazz etc.
GTzan [Tzanetakis e Cook 2002]	999	30	.wav	Pop, rock etc.
Hainsworth [Hainsworth e MacLeod 2004]	222	40-60+	.wav	Folk, jazz etc.
LMD [Raffel 2016]	3.611	30	.mp3	Pop, rock, clássica etc.
SMC [Holzapfel et al. 2012]	217	40	.wav	Clássica, romântica, acústica etc.

3 Metodologia

O artigo trabalha sobre o problema da estimativa de tempo em músicas que não apresentem variação de tempo. Para abordar este problema, foi escolhida a utilização de redes neurais. Visto que as redes neurais vêm apresentando resultados interessantes, sendo cada vez mais utilizadas em conferências de MIR [Gómez et al. 2016], elas se mostram opções modernas e com possibilidades de exploração, sendo optadas para este trabalho em vez de métodos tradicionais que contam com algoritmos (como o de Klapuri, Eronen e Astola (2006)).

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.

As etapas executadas foram a seleção de *datasets* (introduzidos na Seção 2), definição da representação do *input* para as redes neurais, definição dos modelos de redes neurais, treinamento da rede e avaliação. As três primeiras etapas são apresentadas e

discutidos nas subseções a seguir, enquanto a avaliação dos modelos e resultados são discutidos na Seção 4.

3.1 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.

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 torna-se 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.

Sendo assim, 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 é replicado de Schreiber e Müller (2019).

3.2 Modelos

Nesta seção, os modelos utilizados para o experimento são introduzidos. Escolheu-se utilizar uma CNN (rede neural convolucional) e uma B-RNN (rede neural recorrente bi-direcional).

3.2.1 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 e 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.

O modelo CNN proposto por Schreiber e Müller (2018) foi selecionado. A escolha se deu por ser um trabalho consideravelmente recente, apresentar resultados excelentes, performando tão bem quanto ou superando outros considerados estado da arte [Schreiber e Müller 2018], e boa reprodutibilidade.

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. 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.

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.

É interessante ressaltar que, apesar do modelo selecionado para a RNN ser reproduzido do trabalho de Schreiber e Müller (2018), os datasets selecionados para o experimento e os parâmetros utilizados para treinamento são diferentes.

3.2.2 B-RNN

Para este artigo, foi proposto um novo modelo simples de redes neurais recorrentes bidirecionais. 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 podem ser utilizados 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 3.2.1; 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, enquanto a camada densa seria mantida.

O objetivo das camadas recorrentes é identificar *onsets*, analisando as freqüências 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.

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

A etapa seguinte é o processamento por camadas densas, conforme explicado na Subseção 3.2.1, cujo propósito é classificar as *features* detectadas pelas camadas re-

correntes. Primeiramente, o *output* das camadas recorrentes passa por um *average pooling* (5 × 1). Logo após, seguem exatamente as mesmas camadas densas da CNN: após uma *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 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.

3.3 Treinamento

Para a etapa de treinamento, é preciso selecionar uma parte do dataset que será destinada ao treino das redes neurais. 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.

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 Seção 2 foi realizada. Os *datasets* selecionados para treinamento dos modelos foram Extended Ballroom, GiantSteps MTG, Hainsworth, LMD e parte do Groove (90%), totalizando 9.215 faixas. Como abordado na Subseção 3.1, não é fornecido o sinal de música inteiro como *input* para a rede neural. O espectrograma 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. 20% de cada dataset foi separado para a validação dos modelos durante o treinamento, e 80% para o treinamento dos modelos propriamente dito. 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.

O treinamento deve ser encerrado (*early stopping*) quando não há mais descréscimo no erro de validação (validation loss) nos últimos 100 Epochs.

4 Avaliação e Resultados

Seguindo os critérios da Subseção 3.3, os *datasets* para a etapa de avaliação dos modelos de redes neurais 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 seguida a metodologia de Schreiber e Müller (2018), na qual múltiplas saídas de ativação são utilizadas com uma janela deslizante com *overlap* de metade da janela (128 frames). As ativações são por classe, e, assim, a classe de tempo com a maior ativação é escolhida como o resultado da estimativa de tempo em bpm.

Ainda inspirado em Schreiber e Müller (2018), assim como outros trabalhos de estimativa de tempo, são definidas 3 acurácias para a estimativa do tempo em bpm: Acurácia0, Acurácia1 e Acurácia2. A Acurácia0 considera diretamente os valores detectados (arredondados para o número inteiro mais próximo) que forem equivalentes ao tempo anotado; Acurácia1 considera valores detectados com desvio de ±4% do valor anotado do tempo; E Acurácia2 considera valores detectados duas ou três vezes maiores do que o tempo anotado, também considerando uma margem de ±4%.

Os resultados das avaliações dos 2 modelos treinados são comparados com os resultados apresentados por Schreiber e Müller (2018). As Tabelas 3, 4 e 5 apresentam cada uma o percentual de acerto para Acurácia0, Acurácia1 e Acurácia2, respectivamente. Os melhores resultados estão em negrito.

Tabela 2. Resultados de Acurácia0 para os modelos

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 3. Resultados de Acurácia1 para os modelos

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 4. Resultados de Acurácia2 para os modelos

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

Dataset	CNN	B-RNN	(SCHREIBER; MÜLLER, 2018)
GTzan	91.9	85.2	**92.6**
SMC	40.5	30.4	**50.2**

Os resultados tendem a evidenciar que, no geral, há uma diferença considerável de performance entre a CNN e a B-RNN. 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, enquanto a CNN foi replicada de um artigo que passou por múltiplos testes.

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.

Era esperado que a performance geral da CNN deste artigo fosse inferior à do modelo de estado da arte, uma vez que este apresenta resultados após testes e ajustes realizados múltiplas vezes. Entretanto, para os *datasets* que não foram abordados pelo artigo de Schreiber e Müller (2018) (GiantSteps Tempo e Groove), o modelo do estado da arte apresentou resultado inferior a pelo menos um dos modelos treinados neste experimento (por vezes, foi 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.

5 Conclusão

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), e comparou os resultados dos dois modelos treinados com um modelo considerado estado da arte [Schreiber e Müller 2018].

Ambos os modelos apresentaram acurácias satisfatórias (ultrapassando 90% para quase todos os *datasets*, conforme a Acurácia2). 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, ainda que este experimento tenha selecionado *datasets*, inputs e hiperparâmetros diferentes.

Não obstante, a performance da B-RNN sobre o dataset Groove (que contém apenas áudios com linhas de bateria) foi superior à de ambas as CNNs, um resultado que não era esperado. Assim, embora a arquitetura CNN apresente resultados mais precisos do que a B-RNN na maior parte dos datasets, isso não ocorreu quando a análise era feita sobre linhas de percussão.

O modelo de estado da arte apresentou um resultado superior aos dos 2 modelos do experimento nos *datasets* que foram abordados em seu artigo original; Entretan-

to, para *datasets* que não cumprem este requisito, os modelos deste experimento apresentaram uma performance superior.

Com base nos resultados, é possível perceber que a B-RNN apresentou melhor performance na estimativa de tempo de faixas de percussão, enquanto as arquiteturas CNN apresentaram melhor performance para estimar o tempo em músicas de outros instrumentos ou com mais de um. Nenhum dos modelos apresentou uma performance superior em todos os *datasets*, o que pode indicar que a performance de uma rede neural na estimativa de tempo ainda depende muito dos *datasets* que são selecionados para treinamento, assim como a quantidade de faixas utilizadas para treinamento, e não necessariamente da categoria de rede neural escolhida. Os modelos apresentaram uma acurácia melhor nos *datasets* de apenas percussão, mas é necessário considerar o viés de treinamento e avaliação.

Para trabalhos futuros, seria interessante explorar novos hiperparâmetros para o treinamento das redes neurais, assim como refinar o modelo de B-RNN. Também é interessante buscar reunir ou mesmo criar novos *datasets* que contenham apenas linhas de percussão, para reduzir a possibilidade de viés da avaliação.

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.; KREBS, F.; WIDMER, G. Accurate Tempo Estimation based on Recurrent Neural Networks and Resonating Comb Filters. In: **16th International Society for Music Information Retrieval Conference**, Madrid, 2015.

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

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

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.

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.

LOGAN, B.; SALOMON, A. A Music Similarity Function Based On Signal Analysis. In: **International Conference on Multimedia and Expo**, 2001, Tokyo.

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.

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 GMMregression. **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). Johannes Kepler University, Linz, 2008.

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.

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.