UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RODNEY SOUZA

RECONHECEDOR E SEPARADOR DE INSTRUMENTOS MUSICAIS

BAURU
Outubro/2019

RODNEY SOUZA

RECONHECEDOR E SEPARADOR DE INSTRUMENTOS MUSICAIS

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.
Orientador: Prof. Associado Aparecido Nilceu Marana

BAURU
Outubro/2019

Rodney Souza Reconhecedor e Separador de Instrumentos Musicais/ Rodney Souza. – Bauru, Outubro/2019- 42 p. : il. (algumas color.) ; 30 cm.
Orientador: Prof. Associado Aparecido Nilceu Marana
Trabalho de Conclusão de Curso – Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Bacharelado em Ciência da Computação, Outubro/2019.
1. Tags 2. Para 3. A 4. Ficha 5. Catalográfica

Rodney Souza

Reconhecedor e Separador de Instrumentos Musicais

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Banca Examinadora

Prof. Associado Aparecido Nilceu Marana
Orientador
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Computação

Profa. Dra. Simone das Graças Domingues Prado
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Computação

Profa. Associada Roberta Spolon
Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Departamento de Computação

Bauru, _____ de _____ de _____.

Dedico esse trabalho a todos que me inspiram.

Agradecimentos

Gostaria de agradecer primeiramente à minha família, que sempre me inspirou a ser uma grande pessoa, e sempre me deu suporte incondicional para que eu persiga meus sonhos em qualquer circunstância.

Gostaria de agradecer também ao circo, pessoas maravilhosas que conheci e que transformaram minha graduação, depois de incontáveis horas juntos, já não consigo imaginar minha vida sem essa palhaçada.

Agradeço ao Bauru Badgers e a todas as pessoas que passaram por esse time, que sempre me inspiram a nunca desistir e sempre melhorar, que compartilharam comigo a emoção de estar em campo e também a emoção de cada treino, nunca houve um treino sequer no qual eu não tenha me divertido.

Agradeço ao meu orientador Prof. Associado Aparecido Nilceu Marana por aceitar ser meu orientador mesmo estando carregado de tarefas e outros orientandos, e por realizar um extenso trabalho no acompanhamento e correção deste trabalho.

Agradeço à universidade proporcionar um ambiente prazeroso e aos professores por todo conhecimento transmitido.

Agradeço ao ex-aluno Gustavo Rosa pela construção do modelo de TCC em Latex.

Agradeço também ao Chiquinho, grande músico, compositor, desenhista, editor, fotógrafo, animador e programador, mas acima de tudo um grande amigo.

Agradeço a Alan Turing, ateu e homossexual, pai da Ciência da Computação(1912-1954) e a todas as pessoas da história que agregaram e a todas que agregarão conhecimentos para a construção de um futuro mais brilhante.

Meaning is a jumper that you have to knit yourself.

- Exurb1a

Resumo

O reconhecimento de sons de instrumentos musicais pode ser uma tarefa difícil até para seres humanos. Essa habilidade está relacionada diretamente com a separação de instrumentos presentes em um áudio, sendo esta uma atividade de alta complexidade, e que demanda expertise e tempo. No âmbito deste trabalho foi proposta uma solução automatizada de reconhecimento e separação de instrumentos com uma abordagem de aprendizado de máquina. Foram utilizadas para a realização deste trabalho redes neurais recorrentes LSTM. Apesar dos resultados obtidos com a solução proposta terem sido inferiores aos obtidos por métodos do estado da arte da área, eles podem ser considerados satisfatórios dados os recursos e o tempo limitados para o desenvolvimento do trabalho. Além disso, os processos de projeto e desenvolvimento da solução apresentada neste trabalho ensejaram ao aluno aplicar conhecimentos obtidos durante o curso de graduação e também a estudar e aplicar conceitos obtidos durante o curso de graduação e também estudar e aplicar conceitos e tecnologias bastante novas e atuais nas áreas de Aprendizado de Máquina e Reconhecimento de Padrões.

Palavras-chave: Aprendizado de máquina, separador de som, classificador de som, processamento de sinais digitais.

Abstract

Musical instruments' sounds recognition may be a hard task even for humans beings. This skill is directly related with sound source separation, which requires expertise and time. In this project scope is proposed an automated solution for both problems with a machine learning aproach. LSTM recorrent neural network were used for the development of this project. The project results did not meet expectations but still were satisfactory.

Keywords: Machine Learning, sound separation, sound classification, signal processing.

Lista de figuras

Lista de quadros

Lista de abreviaturas e siglas

FT	Fourier Transform
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
STFT	Short-Time Fourier Transform
RNN	Recurrent Neural Network
BRNN	Bidirecional Recurrent Neural Network
LSTM	Long Short-Term Memory
BLSTM	Bidirectional Long Short-Term Memory
VP	Verdadeiro Positivo
FP	Falso Positivo
FN	Falso Negativo
VN	Verdadeiro Negativo
ISR	Source Image to Spatial distortion Ratio
SIR	Source to Interference Ratio
SAR	Sources to Artifacts Ratio
SDR	Source to Distortion Ratio

Sumário

1 Introdução

O ouvido humano é um órgão muito avançado e sensível capaz de distinguir, em média, cerca de 1400 frequências discretas de ondas sonoras, traduzi-las para impulsos elétricos e enviá-los ao cérebro, que por sua vez os interpreta, percebendo diversas nuances de uma música como tons, timbres, intensidades, inicio de cada som (MOORE, 2012).

O reconhecedor de instrumentos musicais é um *software* que utiliza redes neurais, capaz de reconhecer os instrumentos de uma música e isolar seus respectivos sons em trilhas musicais diferentes (ZHAO et al., 2018a; JUNIOR; FARIA; YAMANAKA, 2007).

Um ser humano ao ouvir um instrumento reconhece características como gênero musical, ritmo e o tipo do instrumento, como por exemplo instrumentos de sopro ou percussão.

A capacidade de processamento dos computadores vem aumentando junto com seu conjunto de habilidades, cada vez executando mais tarefas subjetivas consideradas antes impossíveis para uma máquina.

1.1 Problema

Classificar é um grande desafio da inteligencia artificial, um ser humano ao escutar uma música composta por dois instrumentos pode facilmente identifica-los se conhece-los, pois o cérebro humano se utiliza de informações precedentes e de processos cognitivos complexos automaticamente, porém, para uma máquina a mesma tarefa não é tão simples, pois para analisa-la dispõe apenas de um conjunto de números que representam a musica.

O desenvolvimento de técnicas que identifiquem as fontes de sons e os separa estão sendo estudadas por diversos pesquisadores, com algoritmos supervisionados e não supervisionados (ZHAO et al., 2018b; JUNIOR; FARIA; YAMANAKA, 2007).

1.2 Justificativa

Desde a revolução industrial, a humanidade tem avançado rapidamente, grande parte deste avanço se deve à automação de tarefas manuais. Atualmente esperasse continuar avançando, mas com automação de tarefas subjetivas, como interpretação e geração de conteúdo.

Atualmente a separação de áudio é uma tarefa manual imprecisa e complicada, logo a exploração de outras formas de faze-la é imprescindível para o meio musical.

A separação de áudio

1.3 Objetivos

1.3.1 Objetivos gerais

Realizar um estudo sobre aprendizado de máquina, conceitos, tipos e aplicações, e aplica-los no reconhecimento de instrumentos a partir de músicas e separa-los em trilhas.

1.3.2 Objetivos específicos

- Realizar uma revisão bibliográfica sobre redes neurais e métodos de separação de áudio;
- Realizar um levantamento de músicas para o treinamento de classificação;
- Desenvolver um programa capaz de reconhecer um instrumento a partir de uma música solo;
- Desenvolver um programa capaz de reconhecer dois instrumentos de um dueto;
- Desenvolver um programa capaz de separar em trilhas diferentes os sons de uma música.

1.4 Desafios

A obtenção de um conjunto de dados grande, diverso e de qualidade é um desafio, uma grande fonte de músicas e performances de solos e duetos é o *Youtube*, porém os vídeos podem possuir vozes, aplausos e ruídos o que dificulta a análise.

A extração de característica de maneira genérica e sem perda das músicas é um desafio pois cada música possui padrões diferentes.

1.5 Organização da monografia

Este trabalho se organiza em Capítulos, onde este é o primeiro e tem o objetivo de introduzir o trabalho como um todo.

O Capítulo 2 tem como objetivo fornecer uma fundamentação teórica de processamento de áudio, transformadas e aprendizado de máquina para um melhor entendimento de um trabalho como um todo.

O Capítulo 3 é uma coletânea de métodos existentes semelhantes ao método desenvolvido neste trabalho, apresentam o estado da arte e diversas abordagens do problema.

O Capítulo 4 trata sobre as ferramentas e tecnologias utilizadas neste trabalho.

O Capítulo 5 trata sobre o desenvolvimento e criação dos modelos.

O Capítulo 6 discorre sobre os resultados dos modelos obtidos ao testa-los em conjuntos de músicas.

O Capítulo 7 apresenta as conclusões do trabalho e sugere trabalhos que podem ser realizados no futuro.

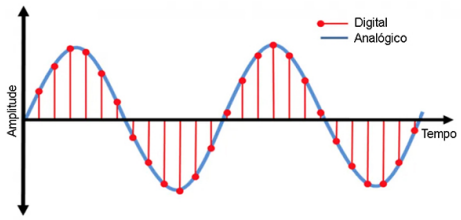
2 Fundamentação Teórica

2.1 Processamento de áudio

Existem diversas formas de representar uma música, como por exemplo partituras, tablaturas e cilindros de piano. São abordada nesse trabalho as formas de representações de ondas, que medem a pressão de ar ao longo do tempo, digitais e tridimensionais (espectrogramas).

As representações digitais, assim como retratado na Figura 1, retratam alguns pontos das ondas em uma determinada taxa de pontos por segundo, a frequência máxima que pode ser armazenada é, teoricamente, metade dessa taxa, mas na realidade um pouco menor. 44100 pontos por segundos (44100Hz) é uma taxa adequada pois o ouvido humano só é capaz de detectar em média cerca de 14000 frequências, pessoas mais jovens podem ouvir até 20000 frequências diferentes.

Figura 1 – Visualização da amostragem de um sinal.



Fonte: <https://www.mobilebeat.com/audio-bit-depth-and-sample-rate/>

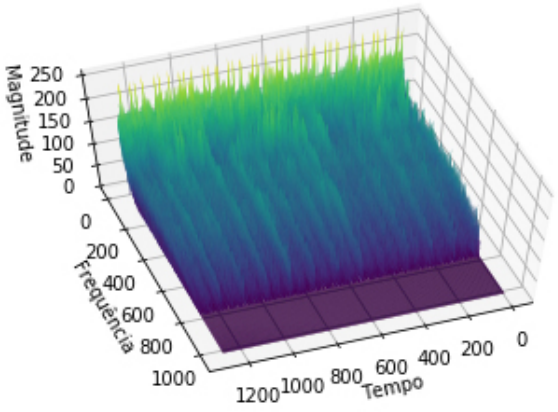
Arquivos MP3 podem ser monofônicos ou estereofônicos, diferenciados apenas pelo numero de canais, onde monofônicos possuem um único canal e estereofônicos possuem dois, normalmente notados com uso de fones de ouvido ou auto falantes especializados de som como o *home teather*, possuindo certos sons mais altos ou levemente adiantados em um canal do que em outro para simular uma noção espacial do som. Arquivos STEM, especializados no armazenamento de músicas, possuem até quatro canais, onde cada um armazena os sons de um instrumento.

2.2 Espectrogramas

Espectrogramas são uma forma de representar sons em um domínio intermediário entre o domínio do tempo e o domínio da frequência. Assim como curvas de nível representam funções, espectrogramas são imagens que representam uma função $\mathbb{R}^2 \to \mathbb{R}, M = s(t, f)$, onde o deslocamento no eixo horizontal representa a variação de tempo, o deslocamento no eixo vertical representa a variação de frequência, e a intensidade de cada pixel representa a

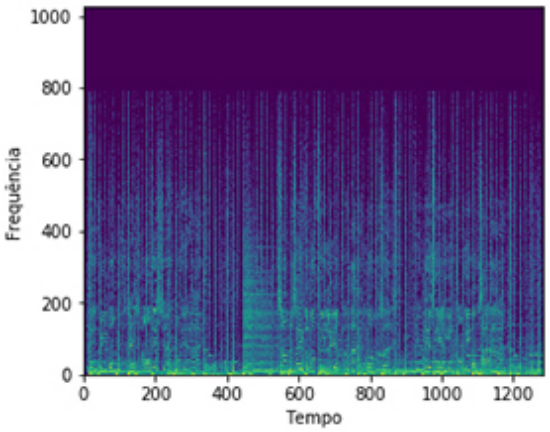
magnitude para um determinado tempo e uma frequência, as figuras 2 e 3 mostram o mesmo sinal no domínio intermediário em diferentes tamanhos de dimensão.

Figura 2 – Superfície do espectrograma.



Fonte: Elaborada pelo autor

Figura 3 – Curvas de nível do espectrograma.



Fonte: Elaborada pelo autor

2.2.1 A transformada de Fourier

A transformada de Fourier parte do princípio de que qualquer função pode ser expressa como um somatório de funções de base sinusoidal (seno e cosseno).(ALLEN, 1977)

A transformada de Fourier possui muitas aplicações, mas será tratada aqui para separação de sons com base na fonte. Em música, todos os sons são ondas sonoras, que primordialmente são somas de frequências puras (ondas senoidais). Para reescrever a função $g(t)$, que representa no domínio do tempo o som em questão, em termos de somas de frequências puras, é necessária uma função intermediaria $\widehat{g}(f)$ que representa o som no domínio de frequências.

A transformada funciona da seguinte forma: representa-se a função $g(t)$ escrita ao redor de uma circunferência de raio um, pra isso é usada a representação no plano complexo, pois a fórmula de Euler $e^{-t2\pi i} = isen(-2\pi t) + cos(-2\pi t)$ representa bem a rotação em sentido horário de t voltas ao redor de um círculo, ou seja, dado um valor de t a fórmula retornará um numero complexo que corresponde à rotação, $e^{-t2\pi i}$ é multiplicado por $g(t)$, representando assim a função ao redor do círculo. Existem inúmeras formas de representar a função $g(t)$ ao redor do círculo, pois para cada volta no círculo é possível representar uma quantidade de ciclos diferentes de $g(t)$ dependendo de uma frequência, encapsulando a frequência na fórmula obtém-se $g(t)e^{-t2\pi if}$.

Agora com a função $g(t)$ descrita ao redor de um círculo, calcula-se a função $\widehat{g}(f)$, que é baricentro (centro de massa) de $g(t)$ para uma determinada frequência f. O baricentro é a somatória de todos os pontos dividido pela quantidade de pontos, mas como o baricentro é, no

contexto da transformação de Fourier, quase sempre próximo a zero, não é necessário dividir pela quantidade de pontos, e o somatório dos infinitos pontos da função pode ser escrito como a integral da função.

$$\widehat{g}(f) = \int_{-\infty}^{\infty} g(t)e^{-2\pi ift}dt$$

Basta agora a análise da função $\widehat{g}(f)$, que é quase sempre muito próxima a zero, mas apresenta alguns picos em seu valor. Os valores de f que refletirem tais picos em $\widehat{g}(f)$ correspondem às frequências das ondas puras que constituem o som, a magnitude (módulo do numero complexo) representa a amplitude, e o ângulo do vetor representa o deslocamento da função senoidal de frequência f.

2.2.2 A transformada discreta de Fourier

Em computação, operações analíticas como somatórios infinitos, derivadas e integrais não são bem vindas, pois para realizar infinitas operações demandaria uma quantidade infinita de tempo. Para que a transformada de Fourier possa ser computada é usada a transformada discreta de Fourier(DFT), que é seu método iterativo e discreto.

A DFT usa para seus cálculos um vetor X, constituído por N elementos de uma amostragem uniformemente distribuída de g (t) em um determinado intervalo, para determinar o vetor $\widehat{X}$ que possui $N/2$ elementos. Determinar o valor de N é crucial para a análise futura devido ao Teorema da amostragem de Nyquist–Shannon, pois as frequências que podem ser analisadas no final da operação depende de N, Para analisar k frequências, deve-se usar $N > 2k$.

n é usado para iterar o vetor X, análogo a t em $g(t)$, e k como f,

$$\widehat{X_k} = \sum_{n=0}^{N-1} X_n e^{-2\pi in\frac{k}{N}}$$

2.2.3 A transformada rápida de Fourier

Embora seja possível computar a transformada de Fourier com a DFT, ela possui complexidade quadrática $o(\frac{N^2}{2})$, pois calcula $\frac{N}{2}$ elementos realizando N operações para cada um. O que pode tornar inviável calcular a DFT com um valor de N grande por demandar muito tempo.

A FFT é um algoritmo eficaz de calcular a DFT com complexidade linearitmica $o(\frac{Nlog(N)}{2})$, tal complexidade é alcançada pois o método explora a natureza periódica das funções sinusoidais com a estratégia de dividir e conquistar, dividindo o somatório em dois somatórios, um com os índices ímpares e outro com os pares.

$$\widehat{X}_k = \sum_{n=0}^{\frac{N}{2}-1} X_{2n} e^{\frac{-2\pi i(2n)k}{N}} + \sum_{n=0}^{\frac{N}{2}-1} X_{2n+1} e^{\frac{-2\pi i(2n+1)k}{N}}$$

Rearranjando:

$$\widehat{X}_k = \sum_{n=0}^{\frac{N}{2}-1} X_{2n} e^{\frac{-2\pi ink}{\frac{N}{2}}} + \sum_{n=0}^{\frac{N}{2}-1} X_{2n+1} e^{\frac{-2\pi ink}{\frac{N}{2}}} e^{\frac{-2\pi ik}{N}}$$

$$\widehat{X}_k = \sum_{n=0}^{\frac{N}{2}-1} X_{2n} e^{\frac{-2\pi ink}{\frac{N}{2}}} + e^{\frac{-2\pi ik}{N}} \sum_{n=0}^{\frac{N}{2}-1} X_{2n+1} e^{\frac{-2\pi ink}{\frac{N}{2}}}$$

Cada somatório é dividido em outros dois somatórios até que todos os somatórios representem a soma de um termo. E depois agrega-se dois somatórios por vez, reutilizando valores calculados por agregações antecessoras já feitas que se repetem em outras agregações.

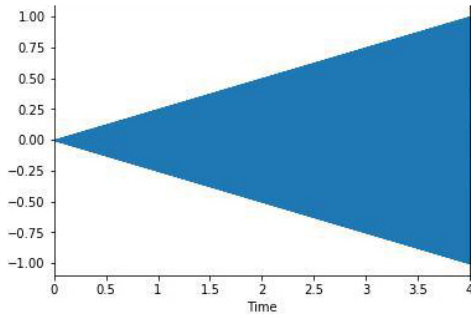
2.2.4 A transformada de Fourier de tempo curto

Em música, as frequências analisadas são variáveis com o decorrer do tempo, trechos de músicas são constituídos por séries de padrões de frequências diferentes. Para amenizar a perda de informação da duração e de fase das ondas, que são muito bem representadas no domínio do tempo, porém se perdem ao transformar para o domínio das frequências, é utilizada a *Short Term Fourier Transform* (STFT), que consiste em aplicar a FFT em segmentos do sinal definidos por janelas curtas de tempo, gerando assim, uma função de três dimensões que dado um tempo e uma frequência obtêm-se um valor. Definir o tamanho das janelas é uma tarefa importante, pois quanto maior o tamanho janela maior a resolução das frequências do sinal, em contra partida, quanto menor o tamanho da janela maior a resolução da duração das ondas (tempo).

2.3 Mudança de escala

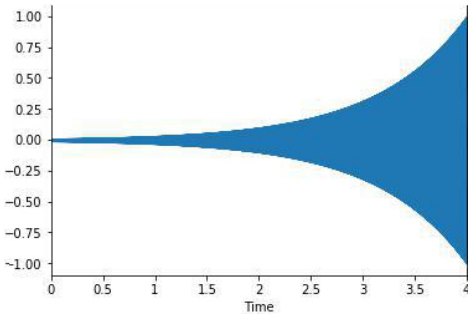
Como dito na seção anterior, na análise um espectrograma, deve-se considerar seus picos de valores. Com intuito de não considerar valores intermediários e enfatizar a diferença das magnitudes, aplica-se a mudança de escala para uma escala logarítmica, é possível notar a diferença observando as figuras 4 e 5.

Figura 4 – Sinal em sua escala original.



Fonte: Elaborada pelo autor

Figura 5 – Sinal em escala logarítmica.



Fonte: Elaborada pelo autor

2.4 Aprendizado de máquina

Em Ciência da Computação, *Machine Learning* é um ramo de pesquisa da Inteligência Artificial que aborda problemas de maneira diferente, treinando um modelo com base em uma série de exemplos, com um algoritmo sendo usado para modificá-lo e otimizá-lo quando preciso (ALPAYDIN, 2009).

Um modelo de aprendizado de máquina é uma função matemática, que tem como objetivo ser generalista o possível para que, dada uma entrada qualquer, seja capaz de prever o resultado correto.

2.4.1 Aprendizado supervisionado

O aprendizado supervisionado é a maneira que o modelo evolui, usando exemplos rotulados com o resultado esperado.

O treinamento segue os seguintes passos: Inicia-se o modelo com pesos aleatórios, calcula-se, com o modelo, uma predição de alguns exemplos e o erro dessas predições em relação ao resultado esperado, calcula-se então o gradiente da função erro afim de minimiza-lo, com os valores do gradiente é possível saber quanto cada peso deve ser alterados, altera-os e repete o processo a partir do segundo passo.

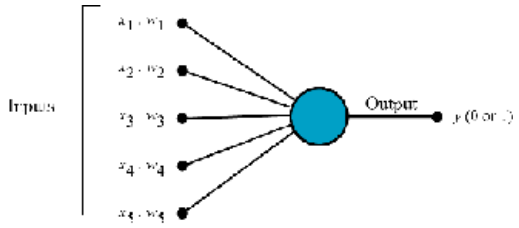
Existem dois tipos de modelos que usam o aprendizado supervisionado, modelos de classificação, que sua saída indica as probabilidades da entrada pertencer às classes pré definidas, e modelos de regressão, que geram como predições números reais que podem ter interpretações variadas de acordo com o problema tratado.

2.4.2 Perceptron

A arquitetura do perceptron é inspirada no funcionamento de neurônios, que recebem estímulos e se ativam caso os estímulos sejam altos o suficiente.

O perceptron possui uma quantidade fixa de pesos e admite entradas de mesmo tamanho, multiplica a entrada pelos pesos, soma os resultados e por fim aplica uma função, usualmente a função sigmoide, que determina sua ativação. A figura 6 representa a estrutura do preceptron. Perceptrons são usualmente chamados de neurônios ou nós.

Figura 6 – Perceptron.

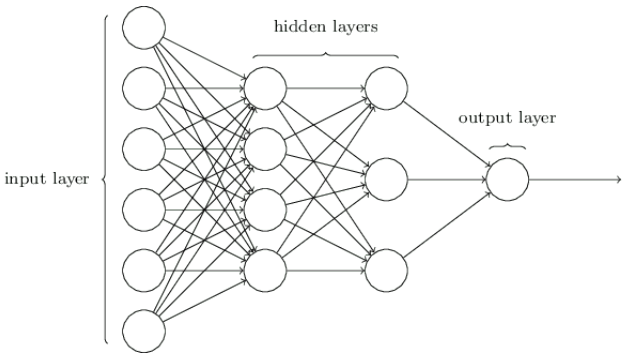


Fonte: <http://deeplearningbook.com.br/>

2.4.3 Redes neurais feed-forward

As redes neurais ampliam a ideia do perceptron com uma arquitetura de camadas. Como é possível observar na figura 7, uma série de perceptrons são organizados em camadas, onde cada camada possui ao menos um neurônio. Uma rede neural deve ter a camada de entrada e a de saída, as camadas escondidas são opcionais. Cada perceptron se conecta com todos os outros perceptrons da camada vizinha seguinte, estimulando-os sempre que ativo e sendo estimulado pelos perceptrons da camada vizinha anterior. Nunca interagindo com outros da mesma camada.

Figura 7 – Redes neurais feed-forward.



Fonte: <http://deeplearningbook.com.br/>

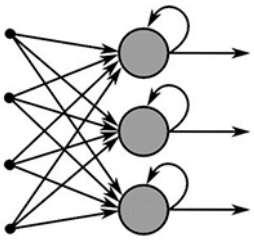
2.4.4 Redes neurais recorrentes

RNN's foram criadas para incluir a ideia de contexto e ordem dos dados. Para tornar uma camada comum em uma camada recorrente é preciso adicionar um peso a cada perceptron da mesma, além de ser estimulado pela camada anterior, ele passa a ser estimulado pela sua

ativação do estado anterior, assim como ilustrado na Figura 8. Na figura 9 é possível observar como a informação se propaga através do tempo. Na primeira iteração da rede, como não há um estado anterior para estimular os neurônios, usa-se valores nulos ou dados gerados artificialmente para simular este estado.

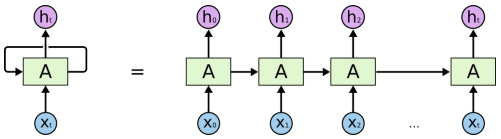
RNN's são muito usadas para problemas em que a entrada e a saída de dados não possuem tamanho padronizado.

Figura 8 – Redes neurais recorrentes.



Fonte: <http://deeplearningbook.com.br/>

Figura 9 – Redes neurais recorrentes através do tempo.



Fonte: <http://deeplearningbook.com.br/>

2.4.5 Redes neurais recorrentes bidirecionais

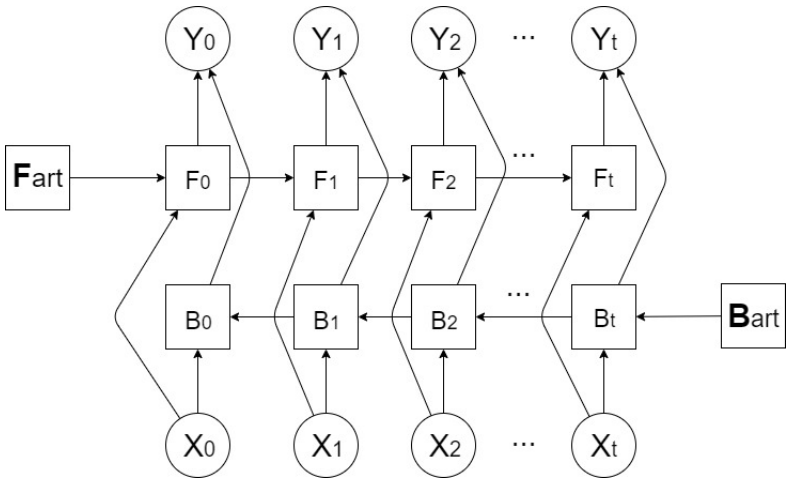
Redes neurais recorrentes bidirecionais (BRNN's) são uma adaptação das RNN's, para que elas além de terem influência da dados anteriores, elas tenham também influência de dados posteriores, para isso é necessário o dobro de nós nesse tipo de camada para gerar a mesma quantidade de saídas, pois na prática existem duas redes recorrentes calculando o resultado.

Como é possível observar na figura 10, primeiro calcula-se todas as previsões na ordem original usando a primeira rede recorrente e guarda-se todos os resultados, em seguida, calcula-se todas os resultados na ordem reversa usando a segunda rede recorrente, por fim soma-se os resultados correspondentes em relação ao tempo para a aplicação de uma função de ativação.

2.4.6 Redes LSTM

As redes LSTM são uma adaptação das RNN comuns, esta adptação muitas vezes é precisa, pois RNN's sofrem de memória de curto prazo, podendo perder a relação entre informações distantes ao analisarem uma sequência muito longa. A arquitetura das redes LSMT possui quatro *gates* que as diferenciam das redes recorrentes comuns. Para melhor

Figura 10 – Redes neurais bidirecionais recorrentes através do tempo.



Fonte: Desenvolvida pelo autor

compreendimento da figura 11 é preciso ter o entendimento de seus símbolos, $\otimes$ representa uma multiplicação entre valores de dois vetores que resulta outro vetor, $\oplus$ representa a soma entre vetores, $\ominus$ representa a junção lógica de dois vetores, linhas pontilhadas representam o fluxo de informações da etapa anterior.

Sempre que uma nova informação entra em um nó, ela é agrupada com a predição anterior e ambas são processadas em quatro redes feed-forward diferentes, gerando os valores usados pelos *gates inputgate*, *forgettinggate*, *outputgate* e a predição. O *inputgate* funciona como uma máscara que filtra o resultado da predição , em seguida a predição é adicionada com informações relevantes da etapa anterior e é salva para etapa posterior, o *forgettinggate* é uma mascara responsável por filtrar memórias de etapas anteriores. E então a máscara do *outputgate* filtra os resultados que passaram pela função de ativação, esse resultado final é usado como predição e para o cálculo dos *gates* e predição da próxima etapa.

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$

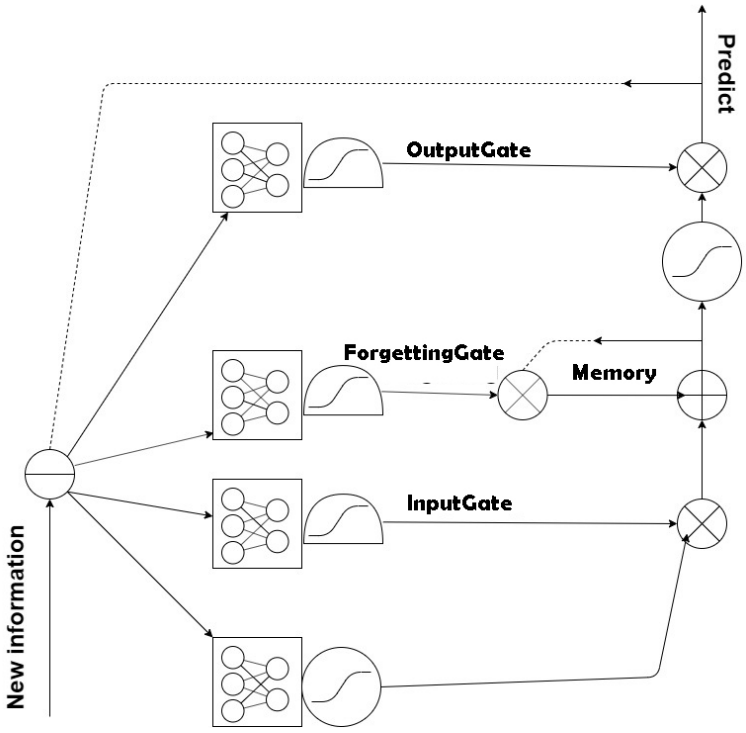
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = (f_t \otimes c_{t-1}) + (i_t \otimes \sigma_c(W_c x_t + U_c h_{t-1} + b_c))$$

$$c_t = o_t \otimes \sigma_h(c_t)$$

Figura 11 – Arquitetura LSTM.



Fonte: Desenvolvida pelo autor

2.5 Métricas

2.5.1 Métricas para métodos de classificação

Um algoritmo ao prever a classe de um dado pode, pela natureza dos problemas de classificação, cometer dois tipos de erros, falsos positivos e falsos negativos, e dois tipos de acertos, verdadeiros positivos e verdadeiros negativos. Suas predições são normalmente visualizadas com uma matriz de confusão em contraste com as classes reais dos dados.

Quadro 1 – Matriz de confusão.

Real/Previsto	**Verdadeiro**	**Falso**
Verdadeiro	VP	FP
Falso	FN	VN

Fonte: Autor.

As métricas mais importantes e mais utilizadas são *accuracy*, que é a a porcentagem dos dados que foram classificados corretamente, *precision* que é razão entre os verdadeiros positivos e de todas as predições dadas como verdadeiros, *recall*, que para todos os dados reais da classe, representa a porcentagem dos que realmente foram classificados corretamente, e por fim, *f1-score*, que não possui uma interpretação intuitiva por ser uma combinação das métricas *precision* e *recall*.

$$Accurary = \frac{VP + VN}{VP + VN + FN + FP}$$

$$Precision = \frac{VP}{VP + FP}$$

$$Recall = \frac{VP}{VP + FN}$$

$$F1 - score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

2.5.2 Métricas para métodos de separação de som

As métricas de separação partem do principio que toda separação de uma música $s(t)$ pode ser escrita como:

$$\widehat{s(t)} = s_{target}(t) + e_{spat} + e_{interf} + e_{artif}$$

Onde $s_{target}(t)$ representa na separação o som puro do instrumento, $e_{spat}, e_{interf}, e_{artif}$ são componentes de erros presentes no resultado. e_{spat} representa a distorção espacial ou de filtragem, , e_{interf} são os sons residuais de outros instrumentos que não foram excluídos,e_{artif} que são ruídos que podem ser gerados por etapas do método, como por exemplo as aproximações realizadas no processo do calculo do espectrograma.

Esses 4 componentes são utilizados para calcular as métricas SDR, SIR, SAR e ISR. SDR é o principal indicador da qualidade da separação, calculando a razão entre o componente alvo e todos os componentes de erro. SIR é um indicador da qualidade da remoção de outros instrumentos. SAR é um indicador da qualidade do método. ISR indica a qualidade espacial. A Figura 12 é uma representação gráfica das métricas.

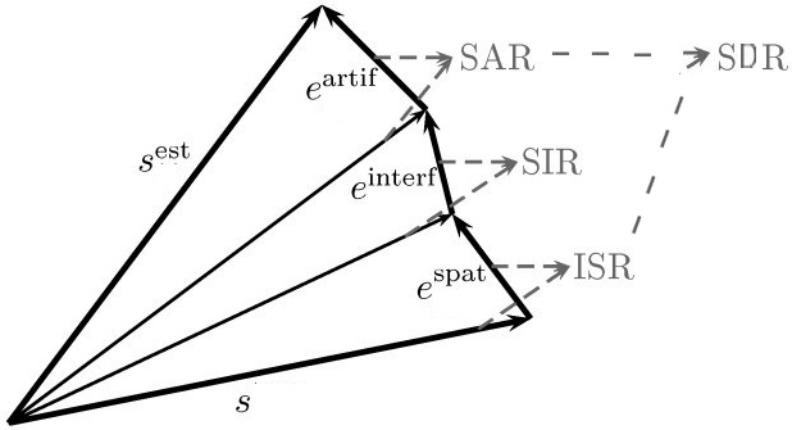
$$SDR = 10\log_{10}\frac{\|s_{target}(t)\|^2}{\|e_{interf} + e_{artif} + e_{spat}\|^2}$$

$$SIR = 10\log_{10}\frac{\|s_{target}(t) + e_{spat}\|^2}{\|e_{interf}\|^2}$$

$$SAR = 10\log_{10}\frac{\|s_{target}(t) + e_{spat} + e_{interf}\|^2}{\|e_{artif}\|^2}$$

$$ISR = 10\log_{10}\frac{\|s_{target}\|^2}{\|e_{spat}\|^2}$$

Figura 12 – Representação visual das métricas de áudio.



Fonte: <https://www.irisa.fr/metiss/SASSEC07/?show=criteria>

3 Aplicações e soluções existentes

Separação de sons é um problema que pode ser encontrado em muitas áreas, inicialmente sendo abordado como *the Cocktail Party Problem*. O problema consiste em um ambiente com pessoas conversando, o desafio é isolar a voz de cada uma delas em áudios, técnicas como *Non-Negative Matrix Factorization*(SCHMIDT; OLSSON, 2006) e *computational auditory scene analysis*(SHAO; WANG, 2008) foram aplicadas.

Um problema especifico da área é a separação de sons de instrumentos musicais. Nesta seção serão abordadas alguns métodos existentes.

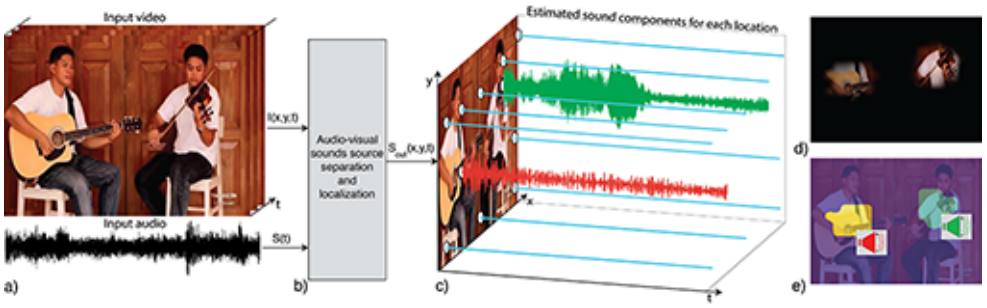
3.1 Pluggins de softwares de audio

A separação de instrumentos musicais é comumente feita por meio de softwares de edição de áudio, como por exemplo audacity[1] e sony vegas[2], com o uso de pluggins que dependem da expertise de quem os usa para aplicar a técnica correta para cada música.

3.2 Sound of pixels

Sound of pixels é um sistema que aprende a localizar regiões de vídeos que produzem sons e separa esse sons em conjuntos de componentes que representam o som gerado por cada pixel. É possível observar seu funcionamento na Figura 13.

Figura 13 – Funcionamento do Sound of Pixels.



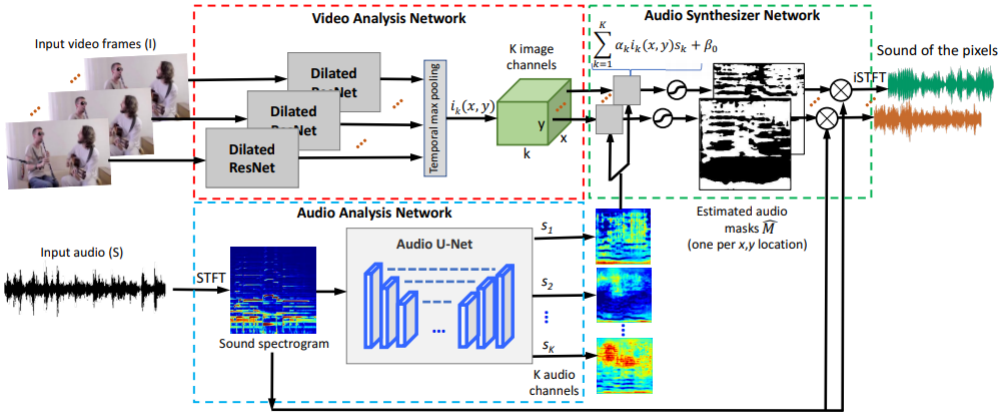
Fonte: <http://sound-of-pixels.csail.mit.edu/>

Com uma abordagem de aprendizado de máquina, o Sound of Pixels usa 3 módulos para a execução da tarefa: o módulo de análise de vídeo, o módulo de análise de áudio e o módulo síntese de vídeo.

[1] https://www.audacityteam.org/
[2] www.vegascreativesoftware.com/

O módulo de análise de vídeo, composto pelo redimensionamento dos *frames* do vídeo, e aplicação do max-pooling no resultado do processamento dos mesmos por redes neurais residuais dilatadas. O módulo de análise de áudio, que primeiramente calcula o espectrograma do áudio, altera a escala do espectrograma para a escala logarítmica, e o processa em uma rede neural convolucional de 14 camadas, sendo 7 de *enconding* e 7 de *deconding*. O último módulo, síntese de áudio, usa os resultados dos módulos anteriores para gerar o resultado. Os módulos estão representados na figura 14.

Figura 14 – Arquitetura do Sound of Pixels.



Fonte: <http://sound-of-pixels.csail.mit.edu/>

3.3 SigSep

A SigSep promove, desde 2008, uma campanha de separação de áudio. Os principais o objetivo da campanha é comparar a performance de métodos e sistemas, padronizando o *dataset* e as métricas utilizadas. Em 2018, em sua sexta edição, houveram 31 submissões de métodos para a campanha.

IBM: IBM é um método usado apenas para referência, como diz o nome *Ideal Binary Mask*, este método calcula uma máscara, que quando multiplicada pelo espectrograma da mistura, resulta no espectrograma de um instrumento isolado. Apesar de alcançar resultados ótimos, para calcular esta máscara, usa-se o próprio resultado, o que impossibilita o uso deste método sem os rótulos.

UHL2: Desenvolvido por uma equipe da *Sony Corporation*, este método utiliza uma rede com camadas Bidirectional Long Short-Term Memory (BLSTM), treinada somente com o dataset da campanha com geração artificial de músicas, aleatoriamente de dessincronizando os instrumentos da mesma música e combinando trechos das performances de instrumentos em músicas diferentes, reduzindo assim o problema de *overfitting*. Esta

rede gera uma espectrograma que representa a separação. Atualmente os métodos dessa equipe atingem o estado da arte quase se equiparando com o método IBM em termos de resultado.

UHL1: Desenvolvido pela mesma equipe que produziu UHL2, UHL1 é um método que utiliza um PCA para pré-processar os dados para alimentar a rede neural ReLU, usando diversos *datasets* musicais e gerando músicas artificialmente. Esta rede, assim como a anterior, gera uma espectrograma que representa a separação.

CHA: Este método faz uso de redes neurais convolucionais para gerar uma máscara, e assim como gera o resultado assim como a IBM, porém calcula a mascara apenas com o espectrograma da mistura.

Open-unmix: Em setembro de 2019 foi lançada a Open-unmix, que é uma aplicação Open-Source com resultados comparáveis ou até superiores às aplicações UHL1 e UHL2, sendo implementada em diferentes *frameworks*. Este método utiliza a versão mais atualizada da base de dados MUSDB2018, a base de dados MUSDB2018HQ, que possui sequências de músicas descomprimidas com maior resolução. O modelo aprende a comprimir os eixos da frequência e canais e durante o processamento utiliza *batch normalization* para que possa convergir mais rápido. Este método faz uso de três camadas LSTM bidirecionais.

4 Materiais ultilizados

4.1 Datasets

4.1.1 MUSDB2018

MUSDB2018 é um *dataset* composto por 150 músicas gravadas profissionalmente por diferentes estúdios, de diferentes gêneros, com duração somada de aproximadamente 10 horas, salvas em formato STEM, com um canal para voz, um para bateria, um para baixo e um para melodias, todos os canais são estereofônicos com taxa de amostragem de 44,1kHz. O MUSDB2018 é dividido em um conjunto de treino, contendo 100 músicas e um conjunto de teste e validação, contendo 50 músicas.

Este *dataset* foi criado para uma campanha de pesquisa em separação de áudio e não deve ser utilizado para fins comerciais sem permissão expressa dos detentores de seus direitos autorais.(RAFII et al., 2017)

4.1.2 MUSMAG

MUSMAG é a versão processada do *dataset* MUSDB, para cada musica foram gerados seis espectrogramas, um para cada canal, um para a soma de todos os canais e um para a soma de todos os canais com exceção da voz.

4.1.3 MUSPIX

Este *dataset* é composto por 766 espectrogramas de músicas, divididos em 19 classes, onde cada classe representa os instrumentos presentes nas música, que podem ser solos ou duetos. Criado a partir do *dataset* usado em *Sound of Pixels* [1], incrementado com outros vídeos do *youtube* e musicas de MUSBD2018.

4.2 Pytorch

Pytorch[2] é uma biblioteca *python* de aprendizado de máquina de código aberto criado pelo *Facebook* em 2016, inspirado pela biblioteca *torch* implementada em LUA.

As principais características da Pytorch são:

[1] https://github.com/roudimit/MUSIC_dataset
[2] https://pytorch.org/resources

- *TorchScript*: disponibiliza flexibilidade e facilidade no uso, criando modelos otimizados e serializados para serem usados em qualquer plataforma sem dependência de *pyhon*;
- Treino paralelizado: possibilita a paralelização do treinamento com o uso de placas de video CUDA;
- Ferramentas e bibliotecas: fornece diversas funções de ativação e erro, otimizadores, *datasets*, ferramentas de processamento de áudio, imagens e texto.

4.3 Bibliotecas

- Numpy: Python implementa nativamente apenas funções básica e não implementa vetores nativamente, numpy é uma biblioteca que implementa vetores e funções matemáticas complexas.
- Pandas: Biblioteca que implementa a leitura de arquivos diversos, implementa também series e dataframes que são tipo de dados que facilitam manipulação e analise de dados.
- Seaborn e matplot: Ultilizadas para a visualização de dados e geração de imagens.
- Musdb: Biblioteca usada para manipular o dataset MUSDB2018.
- Museval: Além de implementar as métricas SRD, SIR, SAR e ISR, agrega os resultados de cada música de um método e agrega também os resultados de vários métodos para facilitar as comparações.
- Sklearn: Biblioteca que implementa algoritmos de aprendizado de máquina e testes estatísticos para a validação de modelos.
- Norbert: Biblioteca de processamento de áudio utilizada para aplicar o filtro *Wiener*, que diminui ruídos da separação com o uso da mistura.
- OpenCV: Utilizada para salvar e abrir espectrogramas como imagens com um único canal.
- YoutubeDl, Librosa, STEMpeg e FFMPEG: Ultilizados para baixar músicas do *Youtube* e manipular arquivos de audio.

5 Desenvolvimento

Foi desenvolvido neste trabalho dois tipos de rede e uma interface. As redes tem estruturas diferentes e apresentam resultados diferentes dependendo do seu tipo.

Todas as redes foram treinadas com uma taxa de aprendizado de 0,01. Tanto a taxa de aprendizado quanto as dimensões das redes foram escolhidas empiricamente.

5.1 Rede de reconhecimento

A rede de reconhecimento tem como objetivo reconhecer a partir do espectrograma de uma música os instrumentos que a compõem. Foram criadas duas instancias dessa rede, onde uma pode reconhecer até 15 instrumentos diferentes, e outra pode reconhecer os mesmos 15 instrumentos e 4 composições de duetos.

Esta rede é formada por três camadas: a camada de entrada que possui 1025 nós, correspondentes a todas as frequências que o espectrograma pode representar, uma camada intermediária LSTM com 512, e uma camada de saída que possui 15 ou 19 nós, dependendo se reconhecerá duetos ou não. A ativação de cada um dos nós da camada de saída representa a probabilidade da entrada pertencer a uma determinada classe, que são definidas de acordo com o instrumento presente.

A rede processa uma coluna de *pixels* do espectrograma por vez, da esquerda para a direita, e cada coluna processada gera uma predição. Somente a última predição é utilizada, pois ao ser gerada ela é influenciada por resultados anteriores.

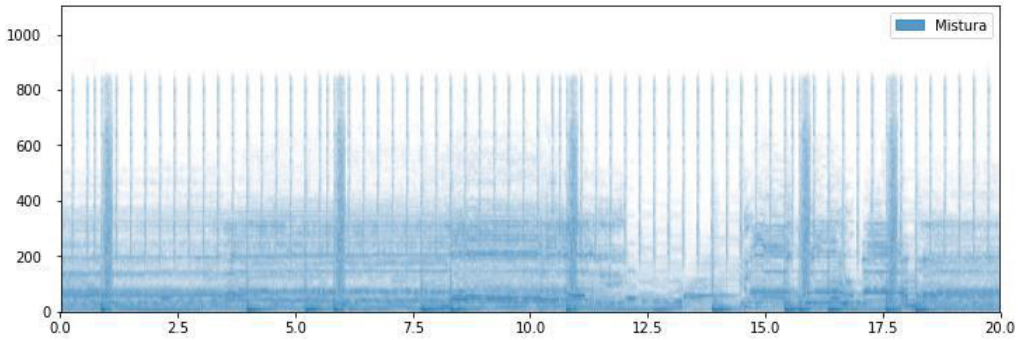
Como é possível que performances direto do *youtube* sejam utilizadas, descarta-se um oitavo do inicio e um oitavo do fim da música para evitar que introduções e aplausos sejam computados a custo de perda de determinadas músicas.

5.2 Rede de separação

Esta rede como objetivo construir a partir do espectrograma de uma música como o apresentado Figura 15, outro espectrograma como indicado pela Figura 16 que representa com cada cor um instrumento separado.

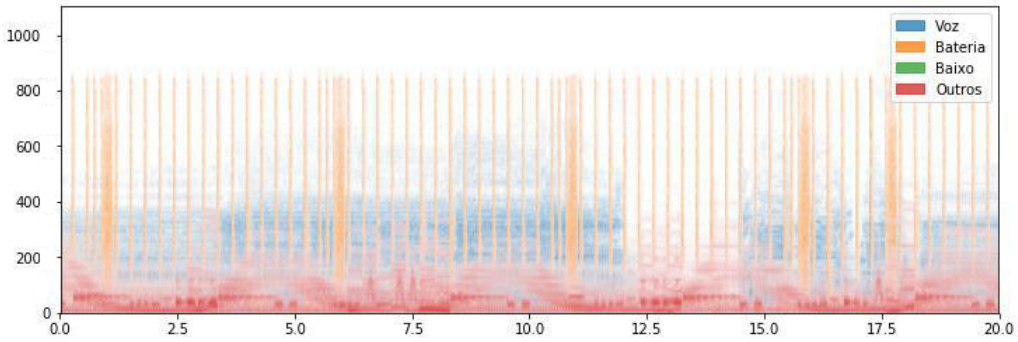
Esta rede é formada por três camadas: a camada de entrada que possui 1025 nós, correspondentes a todas as frequências que o espectrograma pode representar, uma camada intermediária LSTM bidirecional com 512 nós que geram 256 saídas, e uma camada de saída que possui 1025 nós, para gerar um resultado semelhante à entrada.

Figura 15 – Espectrograma de uma mistura de instrumentos



Fonte: Desenvolvido pelo autor

Figura 16 – Espectrograma com indicações de cada instrumento



Fonte: Desenvolvido pelo autor

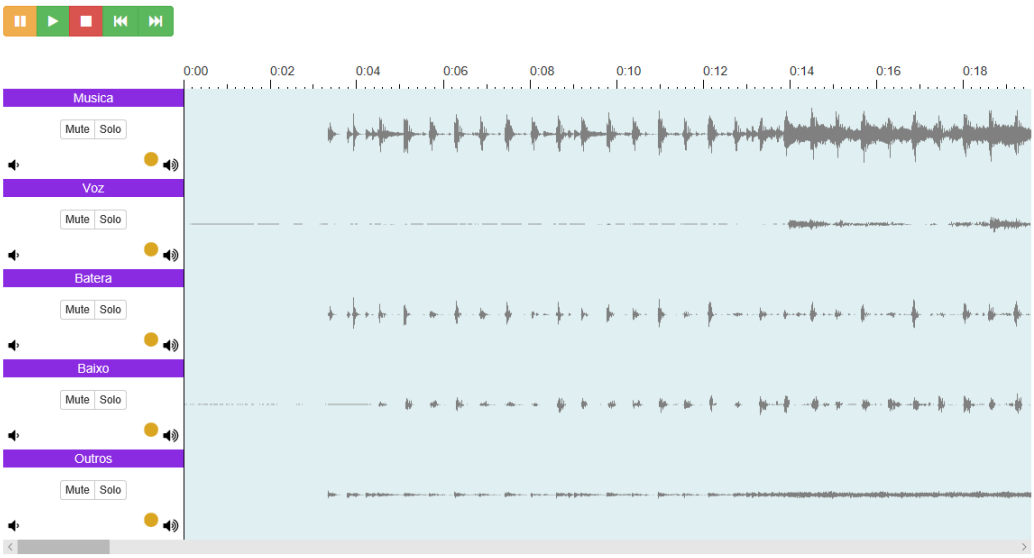
Esta rede processa uma coluna de *pixels* do espectrograma por vez, da esquerda para a direita, e para cada coluna processada gera-se uma coluna de valores, e em seguida processa as colunas de *pixels* novamente, porém da direita para esquerda, gerando uma coluna de valores a cada iteração, depois combina-se as colunas geradas que representam o mesmo tempo na música, gerando assim o espectrograma que representa apenas o som de um determinado instrumento para aquele *frame* (instante).

Foram instanciadas quatro redes de separação, cada uma se especializando na separação de apenas um tipo instrumento.

5.3 Desenvolvimento interface

Para o desenvolvimento da interface foi utilizado o editor de áudio para web desenvolvido e mantido por Naomiaro.

Figura 17 – Espectrograma de uma mistura de instrumentos



Fonte: Desenvolvido pelo autor

6 Validação

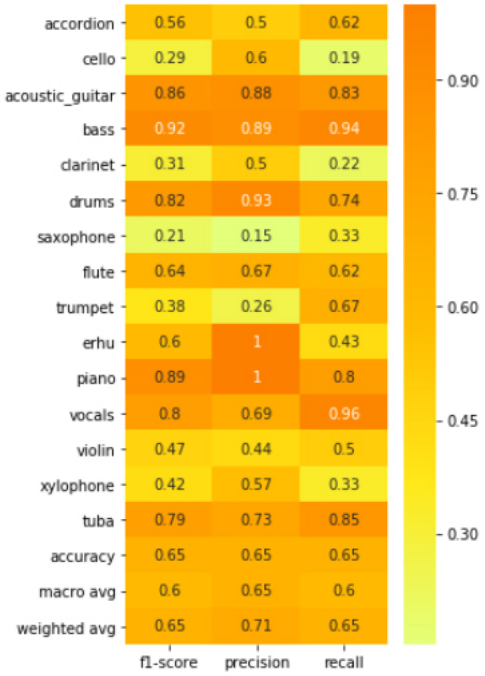
6.1 Rede de reconhecimento

Para o teste foi utilizado um terço do *dataset* MUSPIX que não foi utilizado na etapa de treino.

6.1.1 Modelo reconhecedor de solos

Na fase de testes, o modelo que reconhece solos apresentou uma taxa de acertos de 65%, o que se aproxima do estado da arte se comparado com a taxa de acertos do trabalho *Sound of pixels* que possui 68%. Porém *Sound of pixels* usa de outras informações presentes em video e ambos os trabalhos possuem classes, *datasets* e conjunto de testes diferentes. Portanto essa comparação não é totalmente precisa

Figura 18 – Desempenho de acertos por classe obtidos pelo reconhecedor de solos desenvolvido nesse trabalho.

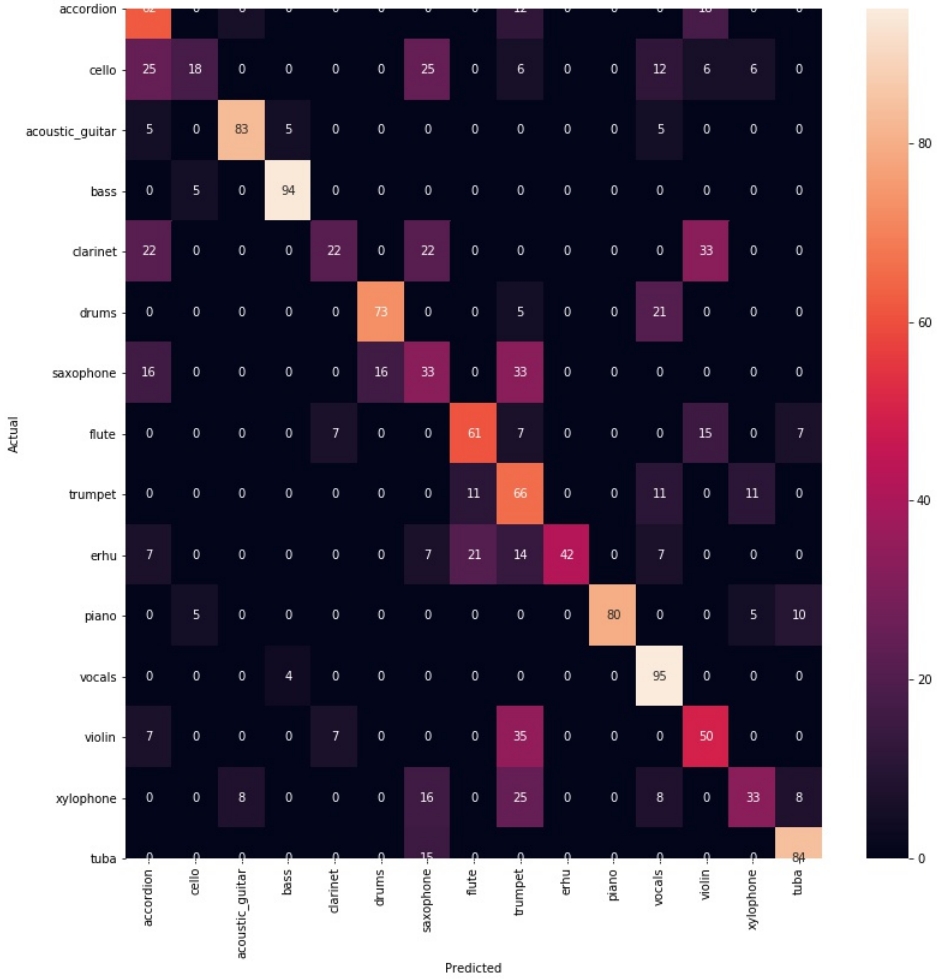


Fonte: Desenvolvida pelo autor

É possível observar na Figura 18 que as classes que apresentam melhor resultado são *bass*, *voice* e *drums*, que são os instrumentos adicionados pelo autor ao *dataset*, pode-se dizer que por serem gravados profissionalmente, possuírem maiores durações e serem os instrumentos mais distintos do *dataset* a rede é capaz de identifica-los com maior facilidade.

Na figura 19 é possível observar quais instrumentos a rede de reconhecimento de solos reconhece com maior facilidade e quais pares de instrumentos ela mais confunde.

Figura 19 – Matriz de confusão referente ao teste da rede de reconhecimento de solos desenvolvidos neste trabalho.



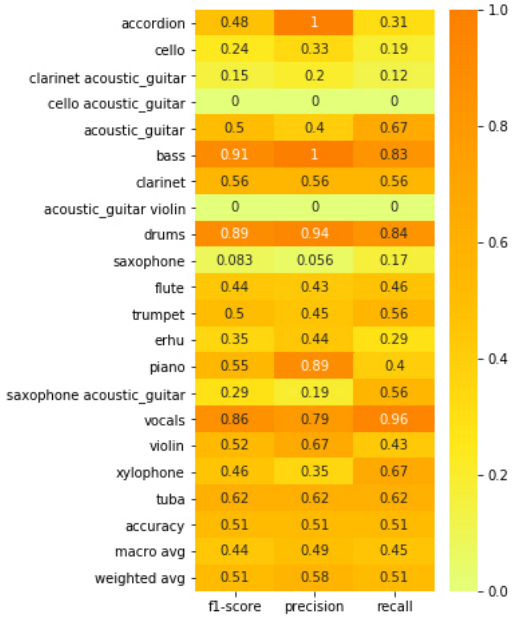
Fonte: Desenvolvida pelo autor

6.1.2 Modelo reconhecedor de solos e duetos

O modelo possui taxa de acerto de 51% com o *dataset* de testes, e como é possível ver na figura 20 as classes que apresentam melhor resultado também são *bass*, *voice* e *drums*.

O modelo possui classes semelhantes, como é possivel ver na figura 21, por exemplo as classes *acoustic_guitar* e *cello acoustic_guitar*, se a classe prevista for *cello acoustic_guitar* e o modelo prever *acoustic_guitar*, sua resposta está errada, porém pode ser considerada satisfatória

Figura 20 – Desempenho de acertos por classe obtidos pelo reconhecedor de solos e duetos desenvolvido neste trabalho.



Fonte: Desenvolvida pelo autor

por ao menos reconhecer o violão presente na música, o que, se levado em consideração, torna mais difícil a análise feita na figura 20.

6.2 Rede de separação UEPA

O metodo de separação deste trabalho foi nomeado UEPA. Na fase de testes, os quatro modelos separaram as 50 musicas de teste do MUSDB2018. A cada intervalo de tempo da separação é calculada seu desempenho para este intervalo, e seu desempenho na música é considerada a mediana dessas medidas.

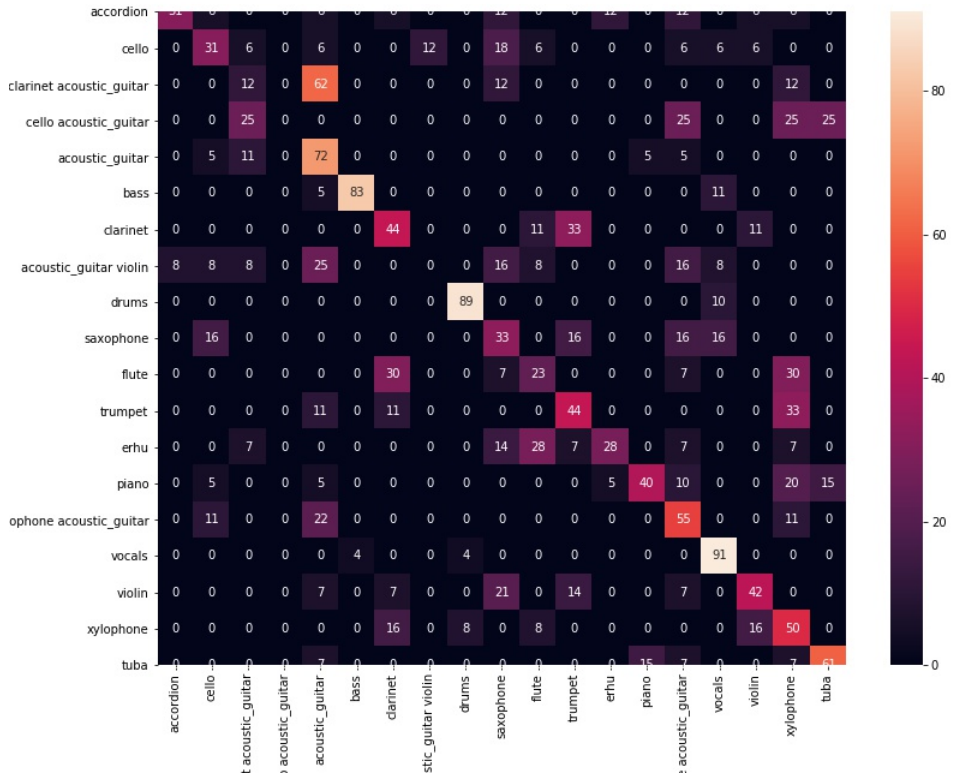
6.2.1 Dados SigSep2018

Os resultados dos métodos submetidos na campanha estão disponíveis em arquivos CSV que facilitam sua manipulação e comparação.

Na figura 22 foi levada em consideração a performance de cada modelo nas 50 músicas de teste, onde é possível identificar em cada modelo seus valores de mínimo e máximo representados pelos inícios e finais das linhas do box plot, uma noção de distribuição com a sinalização da mediana representada por um rico vertical central e medidas de quarto, a representação de um quarto dos dados pode ser uma linha ou meia caixa.

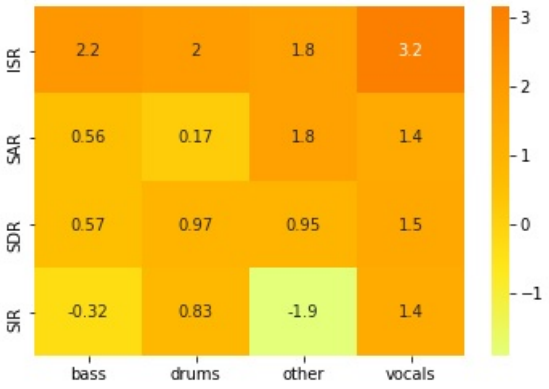
É possível observar na Figura 23 que o método UEPA apesar apresentar baixos resultados ele apresenta maior constância nas separações, e em casos específicos, apresenta resultados

Figura 21 – Matriz de confusão referente ao teste da rede de reconhecimento de solos e duetos desenvolvido neste trabalho.



Fonte: Desenvolvida pelo autor

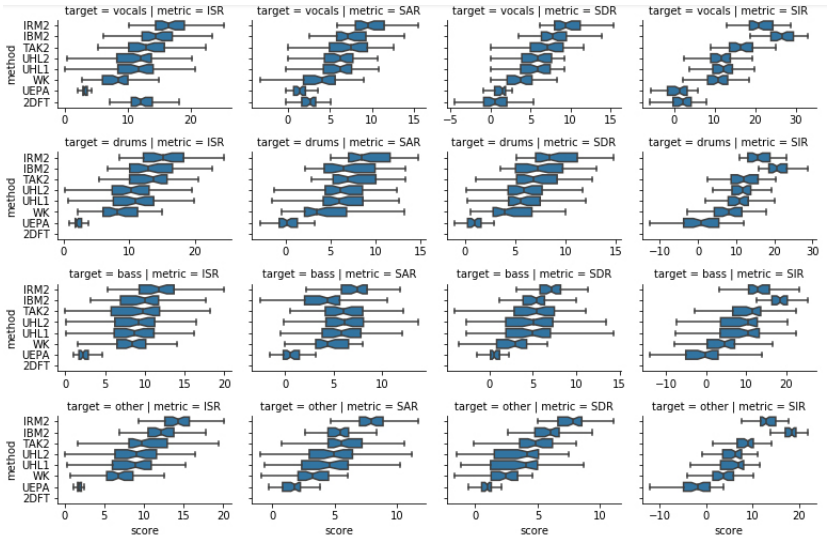
Figura 22 – Desempenho dos modelos em relação às métricas



Fonte: Desenvolvida pelo autor

melhores que UHL1 e UHL2, como por exemplo é possível ver na Figura23, onde a marcação miníma do *boxplot* desses métodos são inferiores à marcação miníma do *boxplot* mínimo do método UEPA.

Figura 23 – Comparação com alguns métodos de separação



Fonte: Desenvolvida pelo autor

7 Conclusão

Neste trabalho foram apresentados conceitos de matemática, aprendizado de máquina e processamento de sinais digitais, apesar de não alcançar o estado da arte, obteve-se resultados considerados satisfatórios. As tecnologias abordadas neste estudo, por sua vez, são atuais, consideradas por muitos revolucionárias e ainda estão longe de atingir seu potencial completo.

7.1 Trabalhos futuros

Para trabalhos futuros, existem diversos caminhos para se explorar, somente a área de aprendizado de máquina abre inúmeras possibilidades. Pode-se sugerir a melhoria dos modelos de separação com a ampliação real do dataset e aumento artificial das músicas, misturando instrumentos de músicas diferentes e misturando até instrumentos do *dataset* MUSPIX para verificar se o modelo se torna mais genérico com o aumento da variedade de instrumentos. Pode-se sugerir também a ampliação dos instrumentos alvos de separação com o *dataset* MUSPIX.

Outra sugestão é a aplicação de redes LSTM em processamento de linguagem natural para gerar letras de músicas a partir de separações de vocais, ou até mesmo traduzi-las com a voz do artista.

Mais uma sugestão é a criação de um modelo capaz de converter os sons de uma música tocada em sons da mesma música performada por outro instrumento.

A ultima sugestão que este trabalho propõe é a criação de um método para identificação do nome de uma música independente do instrumento que performa-la-á.

Referências

ALLEN, J. Short term spectral analysis, synthesis, and modification by discrete fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, IEEE, v. 25, n. 3, p. 235–238, 1977.

ALPAYDIN, E. *Introduction to Machine Learning*. [S.l.: s.n.], 2009. <https://books.google.com.br/books?hl=pt-BR&lr=&id=TtrxCwAAQBAJ&oi=fnd&pg=PR7&dq=machine+learning+introduction&ots=T5elKJ-7rR&sig=PSu1GHskI1jQwvBmce_4mdecQc4#v=onepage&q=introduction&f=false>. Accessed: 2019-03-08.

JUNIOR, C. R. F. de M.; FARIA, E. S. J. de; YAMANAKA, K. *Reconhecendo Instrumentos Musicais Através de Redes Neurais Artificiais*. 2007. <http://revistaseletronicas.pucrs.br/ojs/index.php/hifen/article/viewFile/3847/2921>. Accessed: 2019-03-13.

MOORE, B. C. J. *An Introduction to the Psychology of Hearing*. [S.l.: s.n.], 2012. <https://books.google.com.br/books?hl=pt-BR&lr=&id=LM9U8e28pLMC&oi=fnd&pg=PP1&dq=hearing+introduction&ots=L2Vje0UEA8&sig=_NzoKfhacW1nwX5sPA4ZAvxGz2w#v=onepage&q=introduction&f=false>. Accessed: 2019-03-10.

RAFII, Z.; LIUTKUS, A.; STÖTER, F.-R.; MIMILAKIS, S. I.; BITTNER, R. *The MUSDB18 corpus for music separation*. 2017. Disponível em: <https://doi.org/10.5281/zenodo.1117372>.

SCHMIDT, M. N.; OLSSON, R. K. Single-channel speech separation using sparse non-negative matrix factorization. In: *Ninth International Conference on Spoken Language Processing*. [S.l.: s.n.], 2006.

SHAO, Y.; WANG, D. Robust speaker identification using auditory features and computational auditory scene analysis. In: IEEE. *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. [S.l.], 2008. p. 1589–1592.

ZHAO, H.; GAN, C.; ROUDITCHENKO, A.; VONDRICK, C.; MCDERMOTT, J.; TORRALBA, A. *The Sound of Pixels*. 2018. <https://arxiv.org/pdf/1804.03160.pdf>. Accessed: 2019-02-22.

ZHAO, H.; GAN, C.; ROUDITCHENKO, A.; VONDRICK, C.; MCDERMOTT, J.; TORRALBA, A. The sound of pixels. In: *The European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018.