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

RODNEY RENATO DE SOUZA SILVA

RECONHECEDOR E SEPARADOR DE INSTRUMENTOS MUSICAIS EM ÁUDIO

BAURU
Novembro/2019

RODNEY RENATO DE SOUZA SILVA

RECONHECEDOR E SEPARADOR DE INSTRUMENTOS MUSICAIS EM ÁUDIO

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

BAURU
Novembro/2019

Rodney Renato de Souza Silva Reconhecedor e Separador de Instrumentos Musicais em Áudio/ Rodney Renato de Souza Silva. – Bauru, Novembro/2019-
43 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, Novembro/2019.
1. Aprendizado de máquina 2. Separador de som 3. Classificador de som 4. Processamento de sinais digitais

Rodney Renato de Souza Silva

Reconhecedor e Separador de Instrumentos Musicais em Áudio

Trabalho de Conclusão de Curso do Curso de Bacharelado em 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 artificiais recorrentes LSTM. Apesar dos resultados obtidos com a solução de separação 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 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

Sound recognition of musical instruments can be a difficult task even for humans. This ability is directly related to the separation of instruments present in an audio, which is a highly complex activity that requires expertise and time. As part of this work, an automated instrument recognition and separation solution with a machine learning approach was proposed. Recurrent neural networks LSTM were used for this work. Although the results obtained with the proposed separation solution were below to the state of the art methods, they can be considered satisfactory given the limited resources and time for the development of the work. In addition, the design and development processes of the solution presented in this paper have enabled the student to apply knowledge gained during the undergraduate degree and to study and apply very new and current concepts and technologies in the areas of Machine Learning and Pattern Recognition.

Keywords: Machine Learning, Sound Separation, Sound Classification, Signal Processing.

Lista de figuras

Figura 1 – Visualização da amostragem de um sinal. 17
Figura 2 – Superfície do espectrograma. 18
Figura 3 – Curvas de nível do espectrograma. 18
Figura 4 – Sinal em sua escala original. 21
Figura 5 – Sinal em escala logarítmica. 21
Figura 6 – Perceptron. 22
Figura 7 – NN *feed-forward*. 22
Figura 8 – RNN. 23
Figura 9 – RNN através do tempo. 23
Figura 10 – BRNN através do tempo. 24
Figura 11 – Arquitetura LSTM. 25
Figura 12 – Representação visual das métricas de áudio. 27
Figura 13 – Funcionamento do Sound of Pixels. 28
Figura 14 – Arquitetura do Sound of Pixels. 29
Figura 15 – Espectrograma de uma mistura de instrumentos. 35
Figura 16 – Espectrograma com indicações de cada instrumento. 35
Figura 17 – Interface desenvolvida para facilitar a avaliação subjetiva das redes de separação em conjunto. 36
Figura 18 – Desempenho de acertos por classe obtidos pelo reconhecedor de solos desenvolvido nesse trabalho. 37
Figura 19 – Matriz de confusão referente ao teste da rede de reconhecimento de solos desenvolvidos neste trabalho. 38
Figura 20 – Desempenho de acertos por classe obtidos pelo reconhecedor de solos e duetos desenvolvido neste trabalho. 39
Figura 21 – Matriz de confusão referente ao teste da rede de reconhecimento de solos e duetos desenvolvido neste trabalho. 39
Figura 22 – Mediana do desempenho dos modelos em relação às métricas. 40
Figura 23 – Comparação com alguns métodos de separação. 41

Lista de quadros

Quadro 1 – Matriz de confusão. 26

Lista de abreviaturas e siglas

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

Sumário

1 INTRODUÇÃO . . . 14
1.1 Problema . . . 14
1.2 Justificativa . . . 14
1.3 Objetivos . . . 15
1.3.1 Objetivos gerais . . . 15
1.3.2 Objetivos específicos . . . 15
1.4 Desafios . . . 15
1.5 Organização da monografia . . . 16

2 FUNDAMENTAÇÃO TEÓRICA . . . 17
2.1 Processamento de áudio . . . 17
2.2 Espectrogramas . . . 17
2.2.1 A transformada de Fourier . . . 18
2.2.2 A transformada discreta de Fourier . . . 19
2.2.3 A transformada rápida de Fourier . . . 19
2.2.4 A transformada de Fourier de curto termo . . . 20
2.3 Mudança de escala . . . 20
2.4 Aprendizado de máquina . . . 21
2.4.1 Aprendizado supervisionado . . . 21
2.4.2 Perceptron . . . 22
2.4.3 Redes neurais artificiais *feed-forward* . . . 22
2.4.4 Redes neurais artificiais recorrentes . . . 23
2.4.5 Redes neurais artificiais recorrentes bidirecionais . . . 23
2.4.6 Redes de memória de longo prazo . . . 24
2.5 Métricas . . . 25
2.5.1 Métricas para métodos de classificação . . . 25
2.5.2 Métricas para métodos de separação de som . . . 26

3 APLICAÇÕES E SOLUÇÕES EXISTENTES . . . 28
3.1 Pluggins de softwares de audio . . . 28
3.2 Sound of pixels . . . 28
3.3 SiSEC . . . 29

4 MATERIAIS UTILIZADOS . . . 31
4.1 Bases de dados . . . 31
4.2 Pytorch . . . 31

4.3 **Bibliotecas** **32**

5 **DESENVOLVIMENTO** **34**
5.1 **Rede neural artificial de reconhecimento** **34**
5.2 **Rede neural artificial de separação** **34**
5.3 **Desenvolvimento da interface** **35**

6 **VALIDAÇÃO** **37**
6.1 **Rede de reconhecimento** **37**
6.1.1 Modelo reconhecedor de solos 37
6.1.2 Modelo reconhecedor de solos e duetos 38
6.2 **Rede de separação UEPA** **40**
6.2.1 Dados SigSep2018 40

7 **CONCLUSÃO** **42**
7.1 **Trabalhos futuros** **42**

REFERÊNCIAS **43**

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 artificiais, 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 já os conhecer, 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 analisar a música dispõe apenas de um conjunto de números que a representam.

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 espera-se 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 possui diversas aplicações, como por exemplo:

Karaoke: versão da música sem as vozes;

Acapella: versão da música somente com as vozes;

Mashup: formado pela combinação de trilhas de duas músicas distintas, comumente são usadas as vozes de uma música, e a melodia de outra;

***Sample*:** trechos marcantes de uma música tocados por um certo instrumento que podem ser utilizados na criação de novas músicas;

Edição: aplicação de efeitos somente na voz ou de um instrumento especifico sem alterar os demais canais.

1.3 Objetivos

1.3.1 Objetivos gerais

Realizar um estudo sobre aprendizado de máquina, conceitos, tipos e aplicações, e aplicá-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 artificiais 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 execução 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, o primeiro Capítulo tem o objetivo de introduzir o trabalho, definindo seus problemas, objetivos, justificativas, desafios e sua organização.

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

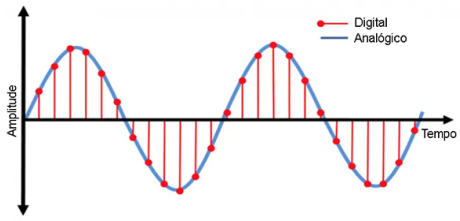
2 Fundamentação Teórica

2.1 Processamento de áudio

Existem diversas formas de representar uma música, como por exemplo partituras, tablaturas e cilindros de piano. São abordadas 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/>

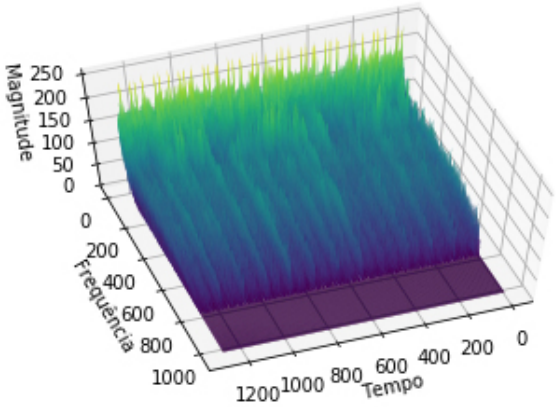
Arquivos MP3 podem ser monofônicos ou estereofônicos, diferenciados apenas pelo numero de canais, onde monofônicos possuem um único canal e estereofônicos possuem dois, normalmente notados com uso de fones de ouvido ou auto falantes especializados, como o *home teather*, possuindo certos sons mais altos 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

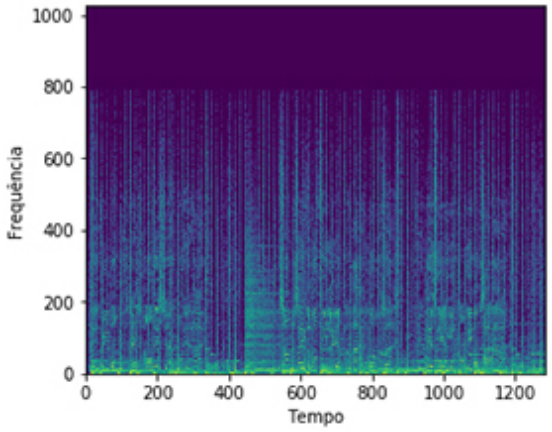
magnitude para um determinado tempo e uma frequência, as Figuras 2 e 3 mostram o mesmo sinal no domínio intermediário em diferentes tamanhos de dimensão.

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

A FT 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

$$e^{-t2\pi i} = isen(-2\pi t) + cos(-2\pi t) \tag{2.1}$$

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 número complexo que corresponde à rotação, o lado esquerdo da Equação 2.1 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 a Equação 2.2

$$g(t)e^{-t2\pi if} \tag{2.2}$$

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 FT, 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 2.2, que é representada pela Função 2.3 (MüLLER, 2015)..

$$\widehat{g}(f) = \int_{-\infty}^{\infty} g(t)e^{-2\pi ift}dt \tag{2.3}$$

Basta agora a análise da Função 2.3, 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 número 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 FT possa ser computada é usada a transformada discreta de Fourier (do inglês *Discrete Fourier Transform*, 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 $N > 2k$.

Como X é uma função discreta e g é uma função continua, usa-se variáveis diferentes para indicar seus parâmetros ou índices, n é usado para indicar os índices do vetor X, análogo a como t indica o parâmetro de $g(t)$, e k indica os índices do vetor $\widehat{X}$, análogo como f indica o parâmetro de $\widehat{g}$, alterando as variáveis continuas pelas variáveis discretas na Fuunção 2.3 obtém-se a Função 2.4 (MüLLER, 2015).

$$\widehat{X}_k = \sum_{n=0}^{N-1} X_n e^{-2\pi in\frac{k}{N}} \tag{2.4}$$

2.2.3 A transformada rápida de Fourier

Embora seja possível computar a FT 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 transformada rápida de Fourier (do inglês *Fast Fourier Transform*, FFT) é um algoritmo eficaz para calcular a DFT com complexidade $o(\frac{Nlog(N)}{2})$, que é inferior a $o(N^2)$, tal complexidade é alcançada pois o método explora a natureza periódica das funções sinusoidais com a estratégia de dividir e conquistar, dividindo o somatório da Função 2.4 em dois somatórios, um com os índices ímpares e outro com os pares como representado na Função 2.5.

$$\widehat{X}_k = \sum_{n=0}^{\frac{N}{2}-1} X_{2n} e^{\frac{-2\pi i(2n)k}{N}} + \sum_{n=0}^{\frac{N}{2}-1} X_{2n+1} e^{\frac{-2\pi i(2n+1)k}{N}} \tag{2.5}$$

Rearranjando a Função 2.5 obtém-sem a Função 2.6 .

$$\widehat{X}_k = \sum_{n=0}^{\frac{N}{2}-1} X_{2n} e^{\frac{-2\pi ink}{\frac{N}{2}}} + e^{\frac{-2\pi ik}{N}} \sum_{n=0}^{\frac{N}{2}-1} X_{2n+1} e^{\frac{-2\pi ink}{\frac{N}{2}}} \tag{2.6}$$

Cada somatório da Função 2.6 é dividido em outros dois somatórios e repete-se o processo 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 que se repetem em outras agregações (MüLLER, 2015).

2.2.4 A transformada de Fourier de curto termo

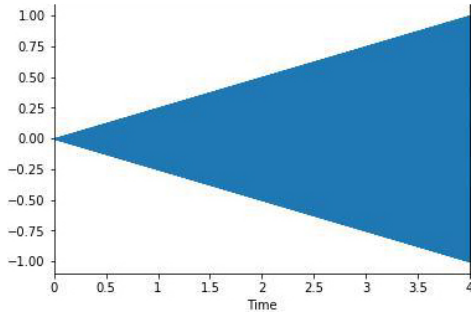
Em música, as frequências analisadas são variáveis com o decorrer do tempo, trechos de músicas são constituídos por séries de padrões de frequências diferentes. Para amenizar a perda de informação da duração e de fase das ondas, que são muito bem representadas no domínio do tempo, mas que se perdem no processo de transformação para o domínio das frequências, é utilizada a transformada de Fourier de curto termo (do inglês *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 contrapartida, quanto menor o tamanho da janela maior a resolução da duração das ondas (tempo) (MüLLER, 2015).

2.3 Mudança de escala

Como dito na seção anterior, na análise de 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

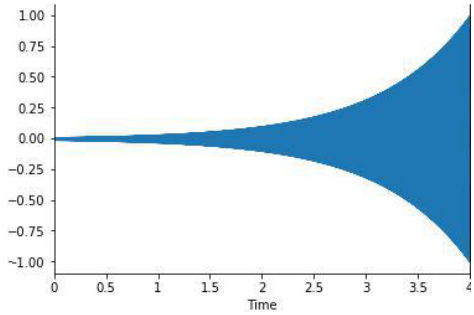
diferença observando a Figura 4, que representa um sinal crescente em função do tempo em sua escala linear original com muitos valores intermediários que dificultam a identificação do inicio do pico de valor, e a Figura 5 que representa o mesmo sinal crescendo em função do tempo em uma escala logarítmica que possui menos valores intermediários, facilitando assim a identificação de picos de valores.

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, Aprendizado de Máquina (do inglês *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 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 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 alterado, altera-os e repete o processo a partir do segundo passo.

Existem dois tipos de modelos que usam o aprendizado supervisionado, modelos de classificação, sua saída indica as probabilidades da entrada pertencer às classes pré definidas,

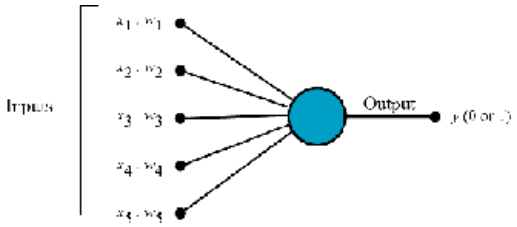
e modelos de regressão, modelos que geram como predições números reais que podem ter interpretações variadas de acordo com o problema tratado.

2.4.2 Perceptron

A arquitetura do Perceptron Artificial (PA) é inspirada no funcionamento de neurônios biológicos, que recebem estímulos e se ativam caso os estímulos sejam altos o suficiente.

O PA possui uma quantidade pré determinada 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 perceptron.

Figura 6 – Perceptron.

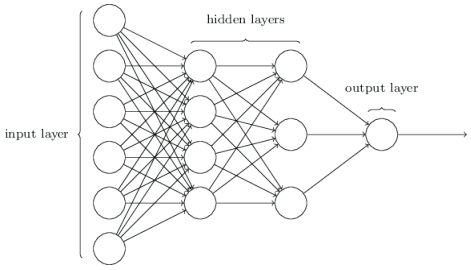


Fonte: <http://deeplearningbook.com.br/>

2.4.3 Redes neurais artificiais *feed-forward*

As redes neurais artificiais (do Inglês *Neural Networks*, NN) ampliam a ideia do PA com uma arquitetura de camadas. Como é possível observar na Figura 7, uma série de PA's 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 PA se conecta com todos os outros PA's da camada vizinha seguinte, estimulando-os sempre que ativo e sendo estimulado pelos PA's da camada vizinha anterior. Nunca interagindo com outros da mesma camada.

Figura 7 – NN *feed-forward*.

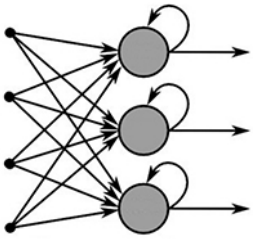


Fonte: <http://deeplearningbook.com.br/>

2.4.4 Redes neurais artificiais recorrentes

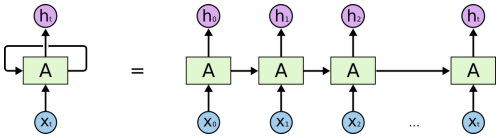
Redes neurais artificiais recorrentes (do inglês *Recurrent Neural Network*, RNN) foram criadas para incluir a ideia de contexto e ordem dos dados. Para tornar uma camada comum em uma camada recorrente é preciso adicionar um peso a cada PA da mesma, além de ser estimulado pela camada anterior, ele passa 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, onde A representa um neurônio, X_i uma informação a ser processada no tempo i e h_i que representa um resultado gerado no tempo i, é possível observar também que o resultado h_i é influenciado pelo resultado h_{i-1} e influencia o resultado h_{i+1}. 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 – RNN.



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

Figura 9 – RNN através do tempo.



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

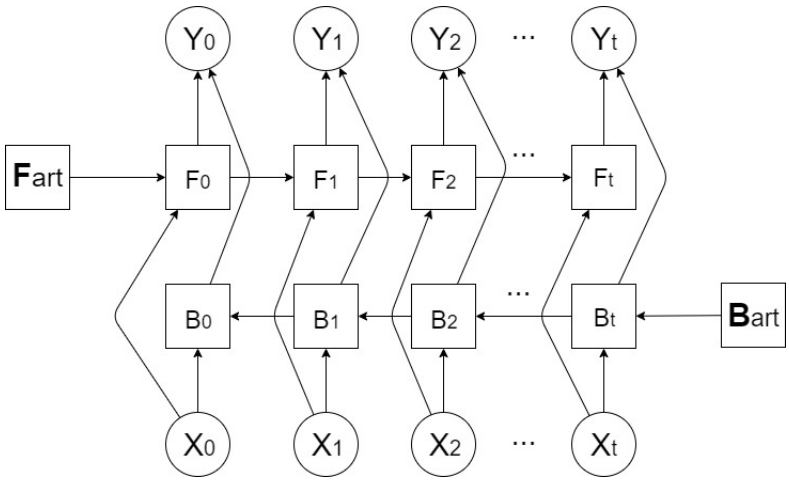
2.4.5 Redes neurais artificiais recorrentes bidirecionais

Redes neurais artificiais recorrentes bidirecionais (do inglês *Bidirecional Recurrent Neural Network*, BRNN) 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.

Figura 10 – BRNN através do tempo.



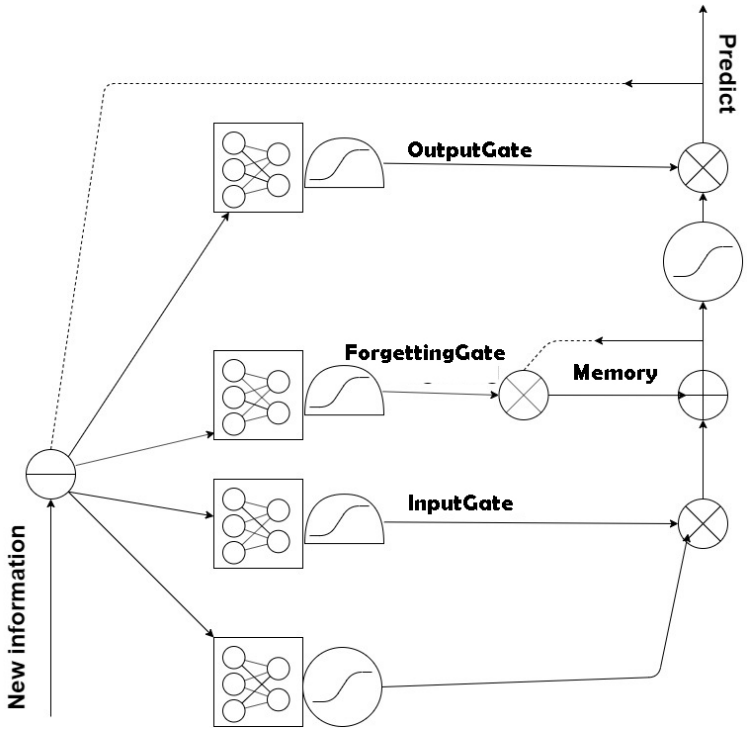
Fonte: Desenvolvida pelo autor

2.4.6 Redes de memória de longo prazo

As redes de memória de longo prazo (do inglês *Long Short-Term Memory*, LSTM) são uma adaptação das RNN comuns, esta adaptação 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 LSTM possui quatro *gates* que as diferenciam das redes recorrentes comuns. Para melhor compreendimento da Figura 11 que representa a arquitetura de um nó de uma rede LSTM,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.

Como é possivel observear na figura 11, sempre que uma nova informação entra em um nó, ela é agrupada com a predição anterior e ambas são processadas em quatro NN feed-forward diferentes, gerando os valores usados pelos *gates inputgate*, *forgettinggate*, *outputgate* e a predição. O *inputgate*, como representa a Equação 2.7, funciona como uma máscara que filtra o resultado da predição, em seguida a predição é adicionada com informações relevantes da etapa anterior e, como representado na Equação 2.10, é salva para etapa posterior, o *forgettinggate*, como representa a Equação 2.8, é uma mascara responsável por filtrar memórias de etapas anteriores. E então a máscara do *outputgate*, como representa a Equação 2.9, filtra os resultados que passarão pela função de ativação, esse resultado final é usado como predição, como visto na Equação 2.11, e para o cálculo dos *gates* e predição da próxima etapa, como já vistos nas Equações 2.7, 2.8, 2.9, 2.10 e 2.11.

Figura 11 – Arquitetura LSTM.



Fonte: Desenvolvida pelo autor

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2.7}$$

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{2.8}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{2.9}$$

$$c_t = (f_t \otimes c_{t-1}) + (i_t \otimes \sigma_c(W_c x_t + U_c h_{t-1} + b_c)) \tag{2.10}$$

$$h_t = o_t \otimes \sigma_h(c_t) \tag{2.11}$$

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

Quadro 1 – Matriz de confusão.

Real/Previsto	**Verdadeiro**	**Falso**
Verdadeiro	Verdadeiro Positivo (VP)	Falso Positivo (FP)
Falso	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Fonte: <http://ai.stanford.edu/ ronnyk/glossary.html>

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.

As métricas mais importantes e mais utilizadas são *accuracy*, representada na Equação 2.12, que é a a porcentagem dos dados que foram classificados corretamente, *precision*, representada pela Equação 2.13, que é razão entre os verdadeiros positivos e de todas as predições dadas como verdadeiros, *recall*, representada pela Equação 2.14, que para todos os dados reais da classe, representa a porcentagem dos que realmente foram classificados corretamente, e por fim, *f1-score*, representada pela Equação 2.15,que não possui uma interpretação intuitiva por ser uma combinação das métricas *precision* e *recall*.

$$Accurary = \frac{VP + VN}{VP + VN + FN + FP} \tag{2.12}$$

$$Precision = \frac{VP}{VP + FP} \tag{2.13}$$

$$Recall = \frac{VP}{VP + FN} \tag{2.14}$$

$$F1 - score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{2.15}$$

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 na Equação 2.16.

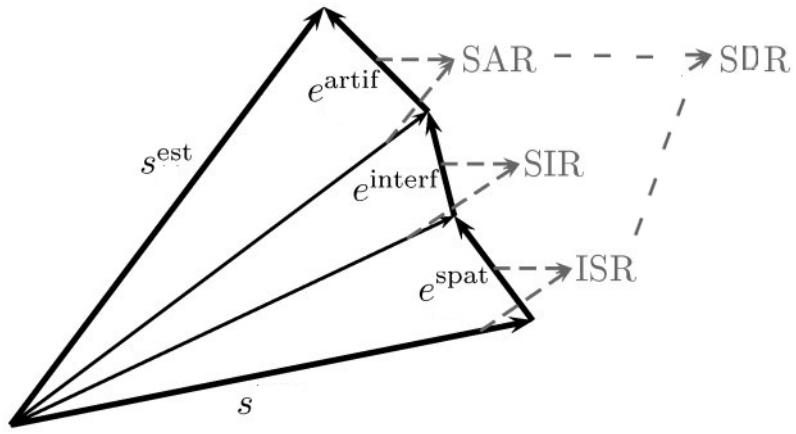
$$\widehat{s(t)} = s_{target}(t) + e_{spat} + e_{interf} + e_{artif} \tag{2.16}$$

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 *Source to Distortion Ratio* (SDR), *Source to Interference Ratio* (SIR), *Sources to Artifacts Ratio* (SAR) e *Source Image to Spatial distortion Ratio* (ISR). SDR é o principal indicador da qualidade da separação, como visto na Equação 2.17, calculando a razão entre o componente alvo e todos os componentes de erro. SIR, como representado na Equação 2.18, é um indicador da qualidade da remoção de outros instrumentos. SAR, como indicado pela Equação 2.19, é um indicador da qualidade do método. ISR, como representado pela Equação 2.20, indica a qualidade espacial. A Figura 12 é uma representação gráfica das métricas, onde a separação esperada e a separação obtida pelo método são vetores e a distância entre eles é representada pelos vetores que retratam os erros.

Figura 12 – Representação visual das métricas de áudio.



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

$$SDR = 10\log_{10}\frac{\|s_{target}(t)\|^2}{\|e_{interf} + e_{artif} + e_{spat}\|^2} \tag{2.17}$$

$$SIR = 10\log_{10}\frac{\|s_{target}(t) + e_{spat}\|^2}{\|e_{interf}\|^2} \tag{2.18}$$

$$SAR = 10\log_{10}\frac{\|s_{target}(t) + e_{spat} + e_{interf}\|^2}{\|e_{artif}\|^2} \tag{2.19}$$

$$ISR = 10\log_{10}\frac{\|s_{target}\|^2}{\|e_{spat}\|^2} \tag{2.20}$$

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.

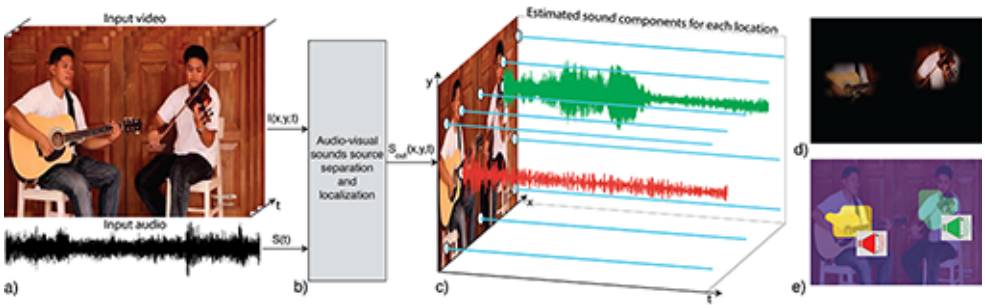
3.1 Pluggins de softwares de audio

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[3] 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.



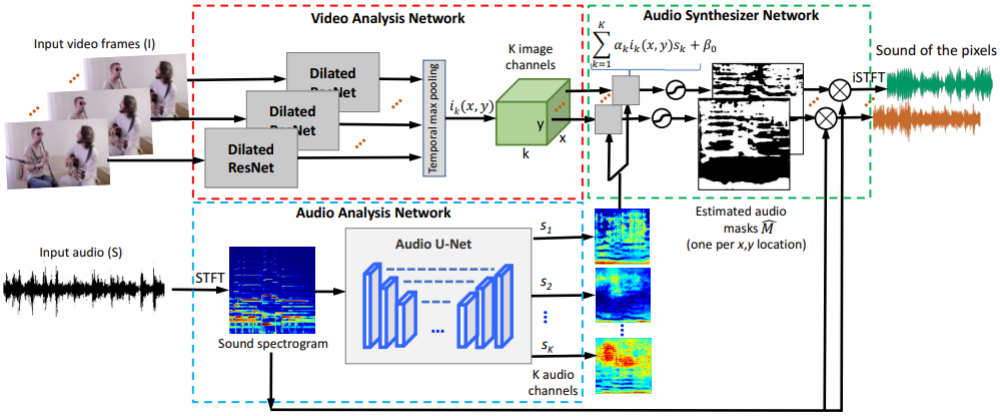
Fonte: <http://sound-of-pixels.csail.mit.edu/>

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/>
[3] <http://sound-of-pixels.csail.mit.edu/>

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 artificial 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 artificial 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 SiSEC

A SiSEC[4](*based Signal Separation Evaluation Campaign*) é uma campanha de separação de áudio iniciada em 2008. O principal 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*, o método UHUL2[5] utiliza uma rede com camadas BLSTM, treinada somente com a base de dados da campanha com geração artificial de músicas, aleatoriamente dessincronizando os instrumentos da mesma música e

[4] <http://sisec.inria.fr/>
[5] <http://sisec17.audiolabs-erlangen.de/#/methods/UHL2>

combinando trechos das performances de instrumentos em músicas diferentes, reduzindo assim o problema de *overfitting*. Esta rede neural artificial 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[6] é um método que utiliza um PCA para pré-processar os dados para alimentar a rede neural artificial ReLU, usando diversos *datasets* musicais e gerando músicas artificialmente. Esta rede, assim como a anterior, gera uma espectrograma que representa a separação.

CHA: O método CHA[7] faz uso de redes neurais artificiais convolucionais para gerar uma máscara, e assim como o método IBM gera o resultado, porém calcula a mascara apenas com o espectrograma da mistura.

Open-unmix: Em setembro de 2019 foi lançada a Open-unmix[8], 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[9], 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.

[6] <http://sisec17.audiolabs-erlangen.de/#/methods/UHL1>
[7] <http://sisec17.audiolabs-erlangen.de/#/methods/CHA>
[8] <https://open.unmix.app/#/>
[9] <https://sigsep.github.io/datasets/musdb.html#musdb18-hq-uncompressed-wav>

4 Materiais utilizados

4.1 Bases de dados

MUSDB2018: MUSDB2018[1] é uma base de dados composta 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 (RAFII et al., 2017).

Esta base de dados foi criada 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.

MUSMAG: MUSMAG[2] é a versão processada da base de dados MUSDB, para cada música foram gerados seis espectrogramas, um para cada um dos quatro canais, um para a soma de todos os canais e um para a soma de todos os canais com exceção da voz.

MUSPIX: Esta base de dados é composta por 766 espectrogramas de músicas, divididos em 19 classes, onde cada classe representa os instrumentos presentes na música, que podem ser solos ou duetos. Criado a partir da base de dados usada em *Sound of Pixels*[3], incrementado com outros vídeos do *youtube* e músicas de MUSBD2018.

4.2 Pytorch

Pytorch[4] é uma biblioteca *python* de aprendizado de máquina de código aberto criado pelo *Facebook* em 2016, inspirado pela biblioteca *Torch*[5] implementada em LUA.

As principais características da Pytorch são:

- *TorchScript*: disponibiliza flexibilidade e facilidade no uso, criando modelos otimizados e serializados para serem usados em qualquer plataforma sem dependência de *pyhon*;

[1] <https://sigsep.github.io/datasets/musdb.html#sisec-2018-evaluation-campaign>
[2] <https://s3.eu-west-3.amazonaws.com/sisec18.unmix.app/dataset/MUSMAG.zip>
[3] <https://github.com/roudimit/MUSIC_dataset>
[4] <https://pytorch.org/resources>
[5] <http://torch.ch/>

- Treino paralelizado: possibilita a paralelização do treinamento com o uso de placas de video CUDA[6];
- Ferramentas e bibliotecas: fornece diversas funções de ativação e erro, otimizadores, base de dados, ferramentas de processamento de áudio, imagens e texto.

4.3 Bibliotecas

Numpy: Python implementa nativamente apenas funções básicas e não implementa vetores nativamente, *Numpy*[7] é uma biblioteca que suporta vetores e matrizes multidimensionais, possuindo uma larga coleção de funções matemáticas para trabalhar com estas estruturas.

Pandas: Biblioteca que implementa a leitura de arquivos diversos, *Pandas*[8] implementa também *series* e *dataframes* que são estruturas de dados que facilitam manipulação e análise com alto desempenho de dados.

Matplot: *Matplot*[9] é uma biblioteca de esboço de gráficos que produz imagens de alta qualidade em diferentes formatos e ambiente em diversas plataformas, com o intuito de manter coisas simples simples de se fazer e tornar coisas complexas possíveis a serem feitas.

Seaborn: *Seaborn*[10] é uma biblioteca de visualização de dados, baseada na biblioteca *Matplot*, que fornece uma interface de auto nível para o esboço de gráficos intuitivos e agradáveis.

MUSDB: MUSDB[11] é uma biblioteca que converte o formato dos arquivos da base de dados MUSDB2018 e fornece uma interface de alto nível para manipulá-la.

Museval: Além de implementar as métricas SRD, SIR, SAR e ISR, *Museval*[12] agrega os resultados de cada música de um método e agrega também os resultados de vários métodos para facilitar as comparações.

scikit-learn: *scikit-learn*[13] é uma biblioteca que fornece ferramentas simples e eficientes para mineração de dados e implementa algoritmos de aprendizado de máquina e testes estatísticos para a validação de modelos.

Norbert: Norbert[14] é uma biblioteca de processamento de áudio que implementa filtros de áudio, como o ffitro *Wiener*, que diminui ruídos da separação com o uso da mistura.

[6] <https://developer.nvidia.com/cuda-zone>
[7] <https://numpy.org/>
[8] <https://pandas.pydata.org/>
[9] <https://matplotlib.org/3.1.0/index.html>
[10] <https://seaborn.pydata.org/>
[11] <https://sigsep.github.io/sigsep-mus-eval/>
[12] <https://sigsep.github.io/sigsep-mus-eval/>
[13] <https://scikit-learn.org/stable/>
[14] <https://sigsep.github.io/norbert/>

OpenCV: *OpenCV*[15] é uma biblioteca de visão computacional e aprendizado de máquina, criada para acelerar aplicações com visão computacional, utilizada neste trabalho para salvar e abrir espectrogramas como imagens com um único canal.

YoutubeDl: YoutubeDL[16] é um programa capaz de fazer *download* de vídeos do *Youtube* e algumas outras plataformas.

Librosa: Librosa[17] é uma biblioteca para análise e manipulação de áudios e músicas.

FFMPEG: FFMPEG[18] é um *framework* capaz de codificar, decodificar, transcodificar, multiplexar, de-multiplexar, transmitir, filtrar e reproduzir áudios em diversos formatos Utilizada converter os formatos nos quais as músicas são salvas.

STEMpeg: Utilizados para, ler, escrever e manipular músicas no formato STEM, a biblioteca STEMpeg[19] foi construída com base no *framework* FFMPEG.

[15] <https://pypi.org/project/opencv-python/>
[16] <https://ytdl-org.github.io/youtube-dl/index.html>
[17] <https://librosa.github.io/librosa/>
[18] <https://www.ffmpeg.org/>
[19] <https://pypi.org/project/stempeg/0.1.2/>

5 Desenvolvimento

Foram desenvolvidos neste trabalho dois tipos de redes neurais artificiais e uma interface. As redes tem estruturas diferentes e apresentam estilos de 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 neural artificial de reconhecimento

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

Esta rede neural artificial 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 neural artificial 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.

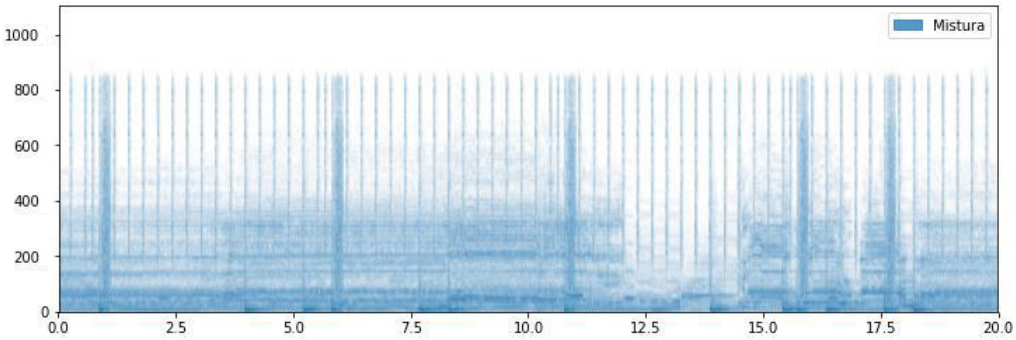
Como é possível que *performances* direto do *Youtube* sejam utilizadas, descarta-se um oitavo do inicio e um oitavo do fim da música para evitar que introduções e aplausos sejam computados a custo de perda de determinadas músicas.

5.2 Rede neural artificial de separação

A rede neural artificial de separação 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.

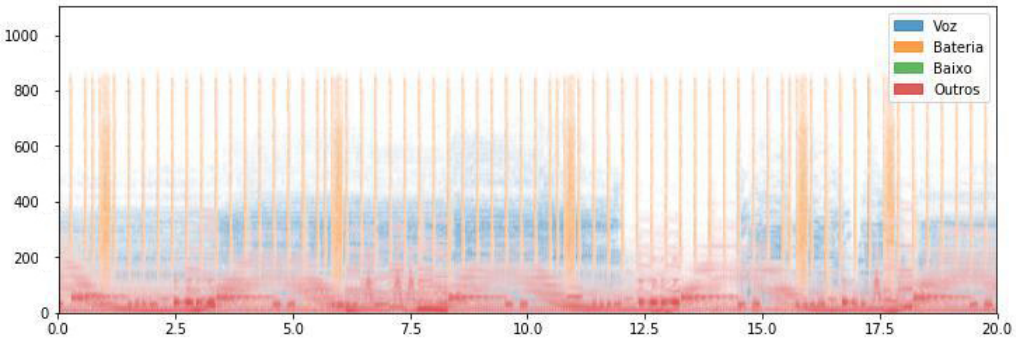
Esta rede neural artificial é formada por três camadas: a camada de entrada que possui 1025 nós, correspondentes a todas as frequências que o espectrograma pode representar, uma camada intermediária BLSTM com 512 nós que geram 256 saídas, e uma camada de saída que possui 1025 nós, para gerar um resultado 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 neural artificial 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 neurais artificiais de separação, cada uma se especializando na separação de apenas um tipo instrumento.

5.3 Desenvolvimento da interface

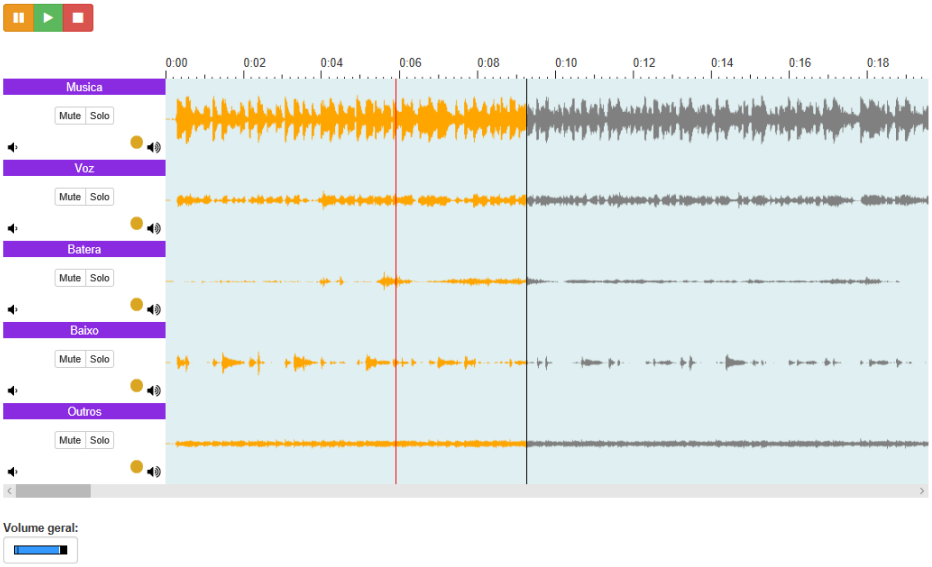
Para o desenvolvimento da interface foi utilizado o editor de áudio para web desenvolvido e mantido por Naomiaro[1].

Como é possível ver na Figura 17 a interface possui um *player* de áudio que permite pausar, retomar e reiniciar a execução de múltiplas trilhas, onde cada trilha representa a

[1] https://github.com/naomiaro/waveform-playlist

música original ou um instrumento especifico, permite a visualização das ondas sonoras de cada instrumento e das ondas da mistura, e permite também controlar o volume geral, o volume de cada trilha e silenciar trilhas ou reproduzir apenas uma trilha.

Figura 17 – Interface desenvolvida para facilitar a avaliação subjetiva das redes de separação em conjunto.



Fonte: Desenvolvido pelo autor

6 Validação

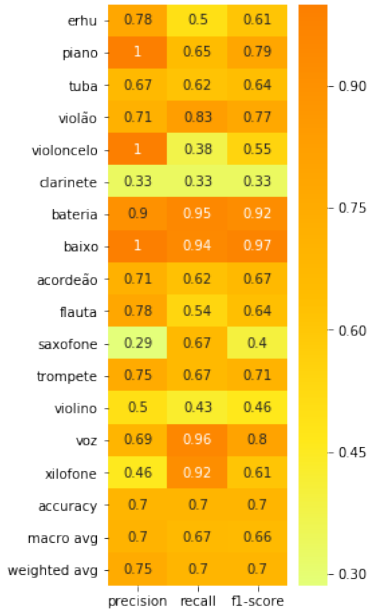
6.1 Rede de reconhecimento

Para o teste foi utilizado um terço da base de dados 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 70%, o que se aproxima do estado da arte se comparado com a taxa de acertos do trabalho *Sound of pixels* que possui 71%. Porém *Sound of pixels* usa de outras informações presentes em video e, apesar de semelhantes, ambos os trabalhos possuem classes, base de dados 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.

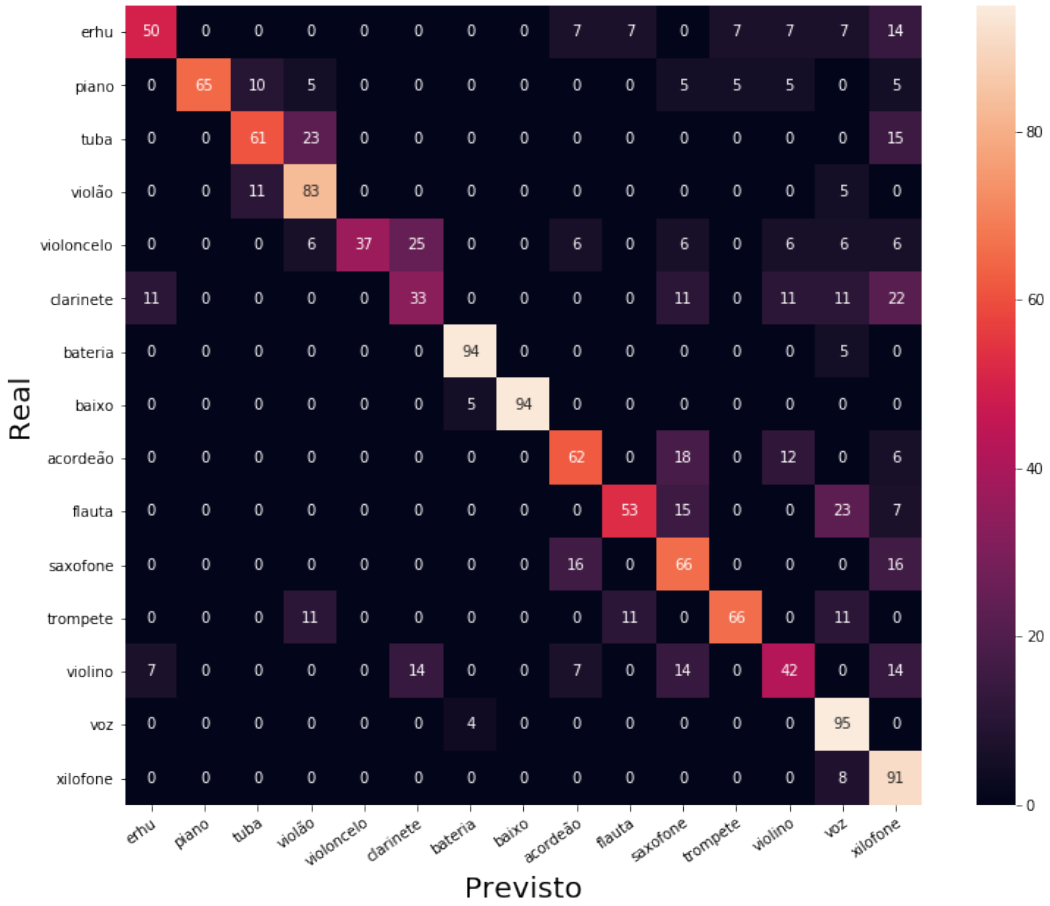


Fonte: Desenvolvida pelo autor

É possível observar na Figura 18 que as classes que apresentam melhor resultado são baixo, voz e bateria, que são os instrumentos adicionados pelo autor à base de dados, pode-se dizer que por serem gravados profissionalmente, possuírem maiores durações e serem os instrumentos mais distintos da base de dados a rede neural artificial 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.



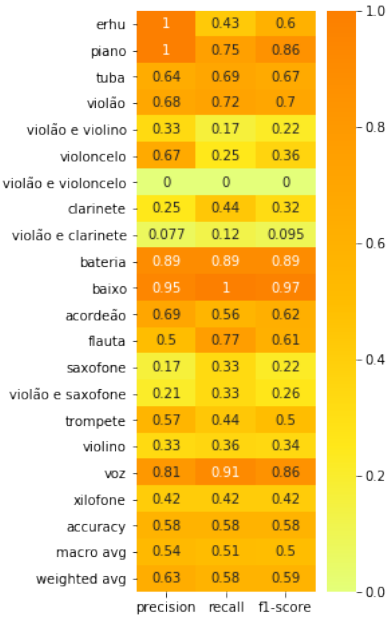
Fonte: Desenvolvida pelo autor

6.1.2 Modelo reconhecedor de solos e duetos

O modelo possui taxa de acerto de 58% com a base de dados usada para testes, e como é possível ver na Figura 20 as classes que apresentam melhor resultado também são baixo, voz e bateria.

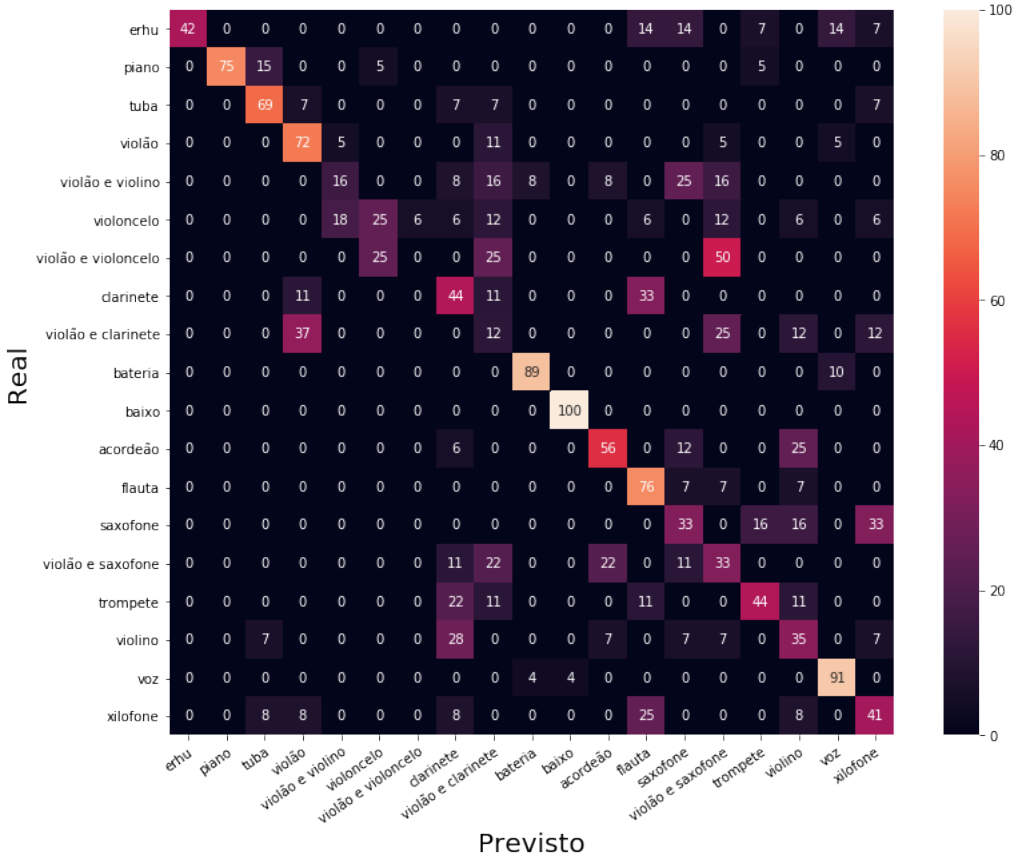
O modelo possui classes semelhantes, como é possível ver na Figura 21, por exemplo a classe violão e a classe violão e violoncelo , se a classe prevista for violão e violoncelo e o modelo prever a classe violão, sua resposta está errada, porém pode ser considerada satisfatória por ao menos reconhecer o violão presente na música, o que, se levado em consideração, torna mais difícil a análise feita na Figura 20.

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



Fonte: Desenvolvida pelo autor

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



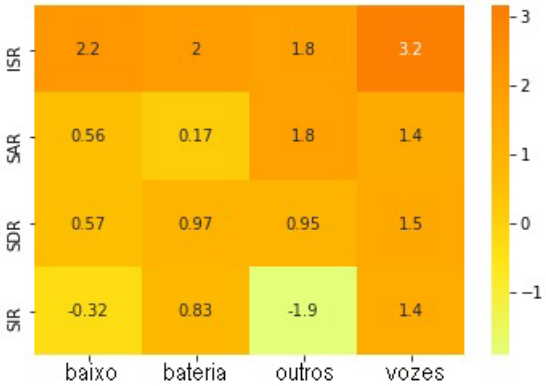
Fonte: Desenvolvida pelo autor

6.2 Rede de separação UEPA

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

Na Figura 22 foi levada em consideração a mediana do desempenho de cada modelo nas 50 músicas de teste, e é possível observar que o modelo separador de vozes desempenha melhor se levada em consideração a mediana dos valores da métricas.

Figura 22 – Mediana do desempenho dos modelos em relação às métricas.



Fonte: Desenvolvida pelo autor

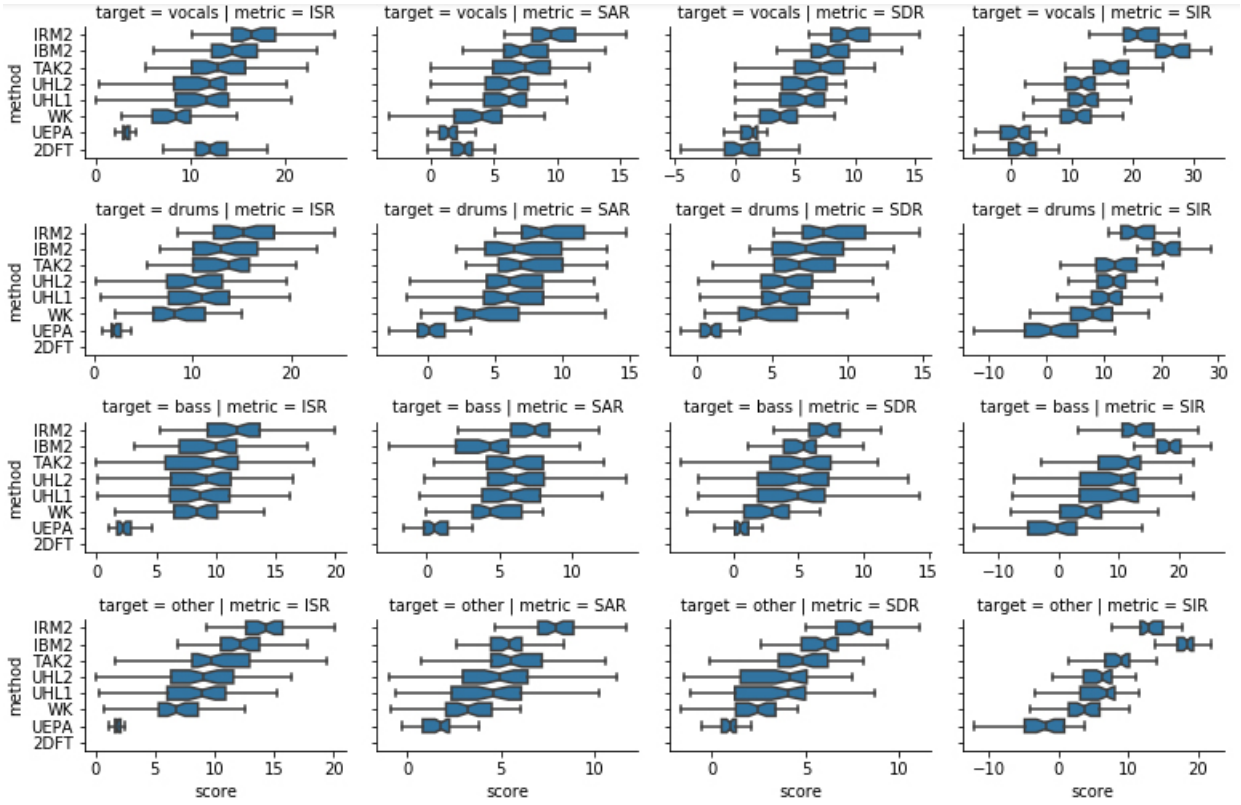
6.2.1 Dados SigSep2018

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

O desempenho do método UEPA desenvolvido neste trabalho é comparado com outros métodos na figura 23, 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 risco vertical próximo ao centro da caixa e medidas de quarto, a representação de um quarto dos dados pode ser uma linha horizontal 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 melhores que UHL1 e UHL2, como por exemplo é possível ver na Figura 23, onde a marcação miníma do *box plot* desses métodos são inferiores à marcação miníma do *box plot* mínimo do método UEPA.

Figura 23 – Comparação com alguns métodos de separação.



Fonte: Desenvolvida pelo autor

7 Conclusão

No decorrer deste trabalho foram desenvolvidos diversos modelos de redes neurais artificiais com diferentes finalidades. Para tal foram realizados estudos sobre conceitos das áreas de Matemática, Aprendizado de Máquina, Reconhecimento de Padrões e Processamento de sinais digitais.

O modelo de reconhecimento de solos obteve resultados bons se comparados com os resultados do estado da arte da área, reconhecendo com maior acurácia instrumentos mais comuns como violão e baixo, porém reconhecendo com menor acurácia instrumentos menos conhecidos como por exemplo o clarinete

O modelo de reconhecimento de duetos obteve acurácia geral inferior ao modelo de reconhecimento de solos devido à sua maior complexidade de suas classes e conjunto de dados adicional desbalanceado. Apesar da baixa acurácia em relação ao outro modelo, o modelo é capaz de classificar satisfatoriamente mesmo em alguns casos em que a classe esperada é diferente da prevista, pois, pela natureza deste problema, identificar um instrumento dentre dois presentes em um dueto é uma resposta satisfatória.

Apesar dos resultados obtidos com a solução de separação 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.

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.

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 última sugestão para a continuidade deste trabalho é a criação de um método para identificação do nome de uma música independente do instrumento.

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>. Acesso em: 08 Mar. 2019.

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>. Acesso em: 13 Mar. 2019.

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>. Acesso em: 10 Mar. 2019.

MüLLER, M. *Fundamentals of Music Processing*. [S.l.: s.n.], 2015. <https://www.springer.com/gp/book/9783319219448>. Acesso em: 21 Ago. 2019.

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>. Acesso em: 22 Fev. 2019.

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.