UNIVERSIDADE FEDERAL DO MARANHÃO

Graduação em Ciência da Computação

Gustavo Roberth Cruz Gomes

Classificando sotaques de Bumba-meu-boi utilizando Deep Learning

São Luís - MA

2022

Gustavo Roberth Cruz Gomes

Classificando sotaques de Bumba-meu-boi utilizando Deep Learning

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Graduação em Ciência da Computação

Universidade Federal do Maranhão

Orientador: Prof. Dr. Geraldo Braz Junior

São Luís - MA

2022

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Gomes, Gustavo Roberth Cruz.
Classificando sotaques de Bumba-meu-boi utilizando Deep Learning / Gustavo Roberth Cruz Gomes. - 2022.
65 p.

Orientador(a): Geraldo Braz Junior.
Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, São Luis, 2022.

1. Bumba-meu-boi. 2. Classificação Musical. 3. Redes Neurais Convolucionais. 4. Redes Neurais Recorrentes. I. Junior, Geraldo Braz. II. Título.

Gustavo Roberth Cruz Gomes

Classificando sotaques de Bumba-meu-boi utilizando Deep Learning

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho Aprovado, São Luís - MA, 07 de fevereiro de 2022:

Prof. Dr. Geraldo Braz Junior
Orientador
Universidade Federal do Maranhão

Prof. Dr. Anselmo Cardoso de Paiva
Universidade Federal do Maranhão

Prof. Dr. João Dallyson Sousa de Almeida
Universidade Federal do Maranhão

São Luís - MA
2022

Agradecimentos

Agradeço a Deus, todo poderoso, aquele que me permite e dá forças todos os dias, para que eu possa seguir em frente, construindo o que deve ser construído, fazer o que deve ser feito e tornar possível aquilo que me é de direito, realizando a minha vida, assim como a este processo acadêmico. Aos meus pais, Carlos Roberte e Odília, por serem minha base, me dando incentivo, apoio, dedicação e amor, contribuindo para aquilo que sou hoje, amo muito vocês. À minha irmã, Amanda Jéssica, por estar pacientemente me incentivando com seu apoio criteriosamente crítico, pelas suas orientações e sugestões valiosas para me tornar alguém melhor. Ao meu orientador, Geraldo Junior, por suas críticas, atenção e paciência depositados em mim, pela disponibilidade de me orientar em minhas primeiras experiências científicas, e confiança para a realização deste trabalho. Ao professor Anselmo de Paiva, por me confiar a oportunidade, em minha primeira experiência científica, de um horizonte de possibilidades para meu desenvolvimento acadêmico e pessoal. A professora Coordenadora da Coordenação de Informática, Simara, pela atenção e disponibilidade durante meu percurso acadêmico, assim como a todos os professores do curso de Ciência da Computação da UFMA, por ceder seus conhecimentos e tempo a outros, e por contribuírem na minha jornada acadêmica. Aos amigos e a todos que me incentivaram de alguma forma a realizar esta conquista.

Obrigado !!!

"Se consegui ver mais longe é porque estava aos ombros de gigantes."

(Isaac Newton)

Resumo

A integral compreensão da importância da garantia de preservação da cultura maranhense nos passa a responsabilidade de dar continuidade ou permitir que as práticas culturais tenham sua continuidade para as gerações que estão por vir, é percebido que práticas culturais regionais, locais, que nos conectam ao nosso passado mais natural e mais humano, estão perdendo força para atrações modernas e globalizadas. Para que estas práticas locais, no estado do Maranhão, possam ter sua identidade preservada, permitindo sua continuidade, devemos criar tais caminhos, em meio a cultura moderna, que possa repercutir estas práticas e as promover no meio cultural. Em um mundo em que a informação se torna o meio de promoção do interesse do indivíduo em se conectar a este meio, por via de ferramentas baratas ao alcance econômico e social, a cultura deve ser promovida de tal maneira a estar tão próxima quanto possível destes para ser comumente inserida neste novo mundo. Sistemas de Inteligência Artificial já se portaram de grande valia para o processamento de imagens, obtendo resultados promissores e amplamente utilizados para o desenvolvimento de modelos de classificação de imagens. Este trabalho, tem como objetivo aproximar a cultura local, de origem nordestina e com variações e enraizamentos pelo Brasil a tecnologia da Inteligência Artificial, mais precisamente as Redes Neurais Artificiais, com o desenvolvimento de um classificador de sotaques de Bumba-meu-boi. Assim, este projeto busca com o auxílio da computação do Aprendizado de Máquina e Aprendizado Profundo, a permanência e repercussão de nossa cultura no mundo das informações sob demanda. Por fim, a base de dados de áudio, a classificação sobre mel-espectrogramas e duas arquiteturas de aprendizado profundo produziram dois modelos que apresentaram resultados de 99% e 100% de acurácia, com poder de classificação de alta precisão.

Palavras-chave: Redes Neurais Convolucionais, Redes Neurais Recorrentes, Classificação Musical, Bumba-meu-boi.

Abstract

The complete understanding of the importance of guaranteeing the preservation of Maranhão's culture gives us the responsibility of giving continuity or allowing cultural practices to have their continuity for the generations that are to come. It is perceived that regional, local cultural practices, which connect us to our more natural and more human past, are losing strength to modern and globalized attractions. In order for these local practices, in the state of Maranhão, to have their identity preserved, allowing their continuity, we must create such paths, in the midst of modern culture, that can echo these practices and promote them in the cultural environment. In a world where information becomes the means of promoting the individual's interest in connecting to this medium, via cheap tools within economic and social reach, culture must be promoted in such a way as to be as close as possible to these people in order to be commonly inserted in this new world. Artificial Intelligence systems have already been of great value for image processing, obtaining promising and widely used results for the development of image classification models. This work aims to approach the local culture, of northeastern origin and with variations and roots throughout Brazil, to the technology of Artificial Intelligence, more precisely the Artificial Neural Networks, with the development of a classifier of Bumba-meu-boi accents. Thus, this project seeks with the aid of Machine Learning and Deep Learning computing, the permanence and repercussion of our culture in the world of on-demand information. Finally, the audio database, classification over honey-spectrograms, and two deep learning architectures produced two models that showed results of 99% and 100% accuracy, with classification power of high accuracy.

Keywords: Convolutional Neural Networks, Recurrent Neural Networks, Music Classification, Bumba-meu-boi.

Lista de ilustrações

Figura 1 – Brincantes tocando pandeirões de onça a altura da cabeça, típico do sotaque de matraca. 21
Figura 2 – Pandeirões afinados a fogo 22
Figura 3 – Cazumbás em apresentação, suas caretas e sinos são a principal marca do sotaque da baixada 23
Figura 4 – Apresentação de grupo folclórico ao som de matracas 23
Figura 5 – Perceptron 25
Figura 6 – Rede Neural Adaline 26
Figura 7 – Desempenho de uma função de desvio mínimo no Hiperplano ótimo . . 27
Figura 8 – Rede Neural Multicamada 27
Figura 9 – Gráfico de tanh(x) 28
Figura 10 – São 3 canais de cores em uma imagem de codificação RGB 29
Figura 11 – Convolução com filtro de extração 5x5 29
Figura 12 – Processo de "lembrança"de parâmetros ao longo do processamento das camadas ocultas 31
Figura 13 – Processo de feedback de informações desenrolado ao longo da rede . . . 31
Figura 14 – Estrutura de uma célula RNN 32
Figura 15 – Estrutura de uma célula LSTM 33
Figura 16 – Conexão da transmissão de estado entre células 33
Figura 17 – Função e gráfico Sigmoid 33
Figura 18 – Gráfico da Função de Ativação Sigmoid e comportamento da sua derivada 35
Figura 19 – Gráfico da Função de Ativação tanh e comportamento da sua derivada 36
Figura 20 – Gráfico da Função de Ativação ReLU e comportamento da sua derivada 37
Figura 21 – Gráfico da Função de Ativação LeakyReLU e comportamento da sua derivada 37
Figura 22 – Processos de Max Pooling e Average Pooling sobre uma imagem 38
Figura 23 – Exemplo de uma imagem mel-espectrograma. 40
Figura 24 – Etapas da metodologia proposta. 44
Figura 25 – O mapa da arquitetura CRNN inspirado no trabalho de (CHOI et al., 2016) 49
Figura 26 – O mapa da arquitetura CRNN modificado para esta aplicação com uso de LSTM como RNN. 50
Figura 27 – O mapa da arquitetura PCRNN inspirado no trabalho de (FENG; LIU; YAO, 2017) com processamento em paralelo 50
Figura 28 – O mapa gerado pela matriz de confusão com a relação entre a predição e o resultado esperado 53

Figura 29 – Gráficos da acurácia para os modelos (a) CRNN e (b) PCRNN 54
Figura 30 – Gráficos da perda para os modelos (a) CRNN e (b) PCRNN 55
Figura 31 – Matriz de confusão resultante dos modelos (a) CRNN e (b) PCRNN . . 56

Lista de tabelas

Tabela 1 – Coletâneas de áudio de Bumba-meu-boi para cada sotaque 45
Tabela 2 – Base de dados de áudio de Bumba-meu-boi após o particionamento . . 46
Tabela 3 – Acurácia na validação e teste para CRNN e PCRNN 55
Tabela 4 – Resultado do teste do modelo CRNN 55
Tabela 5 – Resultado do teste do modelo PCRNN 56
Tabela 6 – Acurácia geral de classificação entre trabalhos estudando espectrogramas particionados e sinais de áudio 58
Tabela 7 – Acurácia geral de classificação entre trabalhos estudando espectrogramas particionados 58
Tabela 8 – Acurácia geral de classificação entre trabalhos estudando sinais de áudio 59

Lista de abreviaturas e siglas

Adaline	*Adaptive Linear Element*
Bi-RNN	*Bi-directional Recurrent Neural Network*
CNN	*Convolutional Neural Network*
CRNN	*Convolutional Recurrent Neural Network*
dB	*Decibéis*
DFT	*Discret Fourier Transformation*
FFT	*Fast Fourier Transformation*
GRU	*Gated Recurrent Unit*
Hz	*Hertz*
LeakyReLU	*Leaky Rectified Linear Unit*
LSTM	*Long-Short Term Memory*
MLP	*Multi Layers Perceptron*
PCRNN	*Paralleling Convolucional Recurrent Neural Network*
Qtd	*Quantidade*
ReLU	*Rectified Linear Unit*
RGB	*Red - Green - Blue*
RNN	*Recurrent Neural Network*
tanh	*Tangent Hiperbolic*
VGG11	*Visual Geometry Group 11*

Sumário

1 INTRODUÇÃO 14
1.1 Objetivos 16
1.2 Trabalhos relacionados 16

2 FUNDAMENTAÇÃO TEÓRICA 19
2.1 Bumba-meu-boi 19
2.2 Sotaques e toadas 21
2.3 Aprendizado de Máquina 24
2.4 Redes Neurais 25
2.4.1 Redes Neurais Convolucionais 28
2.4.2 Redes Neurais Recorrentes 30
2.4.3 LSTM 31
2.5 Funções de ativação 34
2.5.1 Sigmoid 35
2.5.2 TANH 35
2.5.3 ReLU 36
2.5.4 LeakyReLU 36
2.5.5 Softmax 37
2.6 Pooling 38
2.7 Espectrogramas 39
2.7.1 Transformada de Fourier 40
2.7.2 Transformação de Fourier de Tempo Curto 42

3 METODOLOGIA PROPOSTA 44
3.1 Coleta de dados 44
3.2 Pré-processamento 45
3.2.1 Limpeza e particionamento dos dados de áudio 45
3.2.2 Ordenação dos conjuntos de dados de imagens 47
3.2.3 Transformação em espectrogramas 48
3.3 Classificação 49
3.4 Avaliação 53

4 RESULTADOS 54
4.1 Comparação com Trabalhos Relacionados 57

5 CONCLUSÃO 60

REFERÊNCIAS 62

1 Introdução

Para um povo, a cultura é um conjunto de expressões, interpretações e tomadas de decisão que regem os meios de vida, comunicação e manutenção desse povo, uma sociedade em um ambiente em que esta sobrevive.

Outra definição para cultura, seria o conjunto do resultado ou previsão de interações entre os integrantes de um povo, que determinam as interações destes integrantes entre si, como também medem a relação destes com outros povos, estando no mesmo ambiente de sobrevivência destes ou no exterior. A cultura seria produzida "através da interação social dos indivíduos, que elaboram seus modos de pensar e sentir, constroem seus valores, manejam suas identidades e diferenças e estabelecem suas rotinas" (BOTELHO, 2001).

A cultura brasileira integra, ao mesmo momento que não subjuga, em sua prática, a ponto de extinguir as diversas culturas que a constitui, normalizando um ambiente composto de diversas interações de grupos específicos, que se preservam ao longo do tempo, como uma ponte de observação, resistência e restauração das origens deste e daqueles que entraram em contato, como uma miscelânea de construções em modo de reforma, ou preservação da consistência que as torna específicas em suas crenças e tradições. Seria a cultura a própria tradição comum e suas características que a tornam comum dentro de um povo ou território, o contato mútuo de diversos indivíduos que sobrevivem e perecem, resguardando estes traços para a mocidade, as novas gerações.

Dentro de um espaço em que se encontram diversas culturas, a perpetuação se volta para aquelas em que suas traduções se resumem aos anseios desta comunidade, seja por resposta ou manutenção do convívio saudável, pacífico ou até uma sobrevivência, ou também para um súplica por raízes distantes que trazem respostas rápidas para problemas que talvez o indivíduo não consiga compreender, ou não consiga ver e provar aos seus próprios olhos e razão. A grande variação de costumes e crenças, ao longo do território nacional, permite que estas localidades tenham manifestações culturais características, propriamente específicas de cada uma destas regiões, baseadas em todos os costumes iniciais daqueles povos pioneiros ou de crenças e costumes adquiridos ao longo do tempo, como também a união destas duas para a formação das identidades culturais comuns destas regiões, resultando em uma pluralidade, pouco vista em outros lugares do mundo, nos tornando únicos e ao mesmo tempo diferentes entre si.

A variação regional e a incorporação das crenças, anteriormente adquiridas, se deu por construir esta pluralidade, expressa em conteúdos artísticos destas regiões, em produção de afazeres ou expressões artísticas humanas, como artesanato e dança, ou cantigas e teatro, que podem exprimir as histórias de origem destas culturas ou de contos advindos

destes, como também das correntes culturais que se cede origem. “O multiculturalismo pode designar, na atualidade, um complexo de problemáticas que remete à presença de universos culturais diferentes e que necessitam ser enfrentados” (SILVA; JúNIOR, 2017). Um exemplo muito atual para estas manifestações do estilo do forró, com provável origem nordestina, produzida durante o Império Brasileiro através dos sons instrumentais bem rítmicos em bailes, adaptado em outras regiões brasileiras com variadas vertentes subsequentes adquiridas ao longo do tempo, gerando, por fim, uma gama de experiências vinda de uma única vertente, mas se especializando a partir do toque artístico de cada geração artística.

Estas manifestações subsequentes engrossam o dicionário cultural popular confirmando as identidades locais como povos pertencentes àqueles locais, que se beneficiam e reproduzem essas manifestações em forma de entretenimento ou meio de vida. A expressão destas como seu fundamento de origem, suas raízes, as raízes destes povos com a terra em que nasceram e cresceram, após um processo de pertencimento destes indivíduos a estes meios culturais locais, a preservação cultural se torna uma tarefa paralela a vivência dentro deste meio. “A máxima somos todos iguais porque somos diferentes oferece um status de pertencimento (ser membro) à coletividade humana no seu conjunto” (MELO, 2015) , em que cada indivíduo passa a ver estas manifestações como parte integrante do seu meio, e este parte necessária para o acontecimento e perpetuação destas atividades, a preservação é tida como natural.

Em um meio de informação, o indivíduo pode não conhecer sua casa, de onde veio e assim por diante, já que este observa o todo como todo, como uma complexa massa, não observa o que o cerca ou tem conhecimento para identificar e assim discernir, pois seu olhar está no todo, na massa, perdendo sua identidade para a identidade global, sua vida se torna global, como tudo o que o cerca. Estes indivíduos, necessitam assim de conhecer a si mesmos e sua terra natal, em que “o estudo do patrimônio cultural promove a valorização e consagração daquilo que é comum a determinado grupo social no tempo e no espaço” (TOMAZ, 2017), situando o ser a suas raízes e origens, naquilo que o deu origem e o tornou o que é hoje, como também o local onde vive.

A preservação cultural pode se tornar tarefa diária dentro de uma sociedade globalizada e multicultural, em que todas as manifestações culturais podem ser realizadas por aqueles indivíduos que, por estar em contato com a tecnologia, se conectam a distância através dos meios de informação, dos meios de conexão via internet, adquirindo todas estas informações disponíveis na rede, desenvolvendo uma ponte para trazer a proximidade de interessados a novas cultura, por meio de um computador ou de uma aparelho celular.

Para integrar maranhenses e visitantes, interessados a cultura local e outros que desejam nos conhecer nossas origens, este trabalho busca solucionar um problema observado em nosso dia a dia, em nosso meio atual, para nos aproximar de nossas raízes,

aproximar a informação sobre nossos costumes a visitantes e estrangeiros, como para traduzir aos mais jovens aquilo que pertencia a seus entes passados, os conectando através de aparelhos eletrônicos de informações como computadores, celulares entre outros, a adquirir informações sobre nossa cultura.

O Bumba-meu-boi é uma das manifestações mais evidentes no estado do Maranhão, e seu enredo é uma tradução à união de culturas europeia, indígena e africana, caracterizada ao período colonial escravista, observada pelas relações econômicas, sociais, pela monocultura, em um ambiente sertanejo com a presença de uma cultura de criação de gado, com forte presença religiosa (e em algumas interpretações de ocultismo), contendo algumas adaptações dependendo da região que se reconta a história.

Esta manifestação da cultura brasileira, conta especificamente com 5 sotaques mais evidentes presentes nas regiões do Maranhão, muito semelhantes entre si e ricos nos mais diversos aspectos. Com característica sonora muito semelhante entre seus grupos, pode ser uma tarefa difícil a sua identificação de sotaque, sendo assim, este trabalho visa solucionar este problema, utilizando das toadas e sotaques de Bumba-meu-boi conhecidos para construir um modelo de classificação de sotaques de Bumba-meu-boi, utilizando tecnologias de Deep Learning, mesclando cultura e computação.

1.1 Objetivos

Este trabalho tem por objetivo realizar o desenvolvimento de um modelo de Aprendizado de Máquina Profundo capaz de classificar sotaques de Bumba-meu-boi utilizando processamento de imagens de espectrogramas adquiridas através de transformações dos sinais sonoros em gráficos de imagens.

O trabalho desenvolvido teve os seguintes objetivos específicos:

- Desenvolver duas metodologias de classificação de sotaques de Bumba-meu-boi através de imagens;
- Analisar a eficiência no processamento de imagens para classificar dados de áudio;
- Analisar e aplicar a transformação de sinais sonoros em imagens.

1.2 Trabalhos relacionados

Dentre as diversas aplicações de Redes Neurais Convolucionais com arquiteturas obtendo resultados muito satisfatórios desde a primeira aplicação de sucesso por (LECUN et al., 1998), alguns trabalhos se destacam como o de Choi et al. (2016) e o de Feng,

Liu e Yao (2017), por apresentarem um estudo sobre processamento de imagens para classificação de gêneros musicais.

No trabalho de Choi et al. (2016), foram desenvolvidos experimentos de objetivo comparativo de diferentes arquiteturas CNN para processamento de classificação de música. Este trabalho comparativo realizou uma comparação entre 4 (quatro) arquiteturas, dentre elas a introdução da CRNN (*Convolucional Recurrent Neural Network*). Uma arquitetura composta por camadas de CNN, que tem como função realizar a extração de características, e em sua saída uma conexão com uma camada RNN - GRU, que realiza a classificação através de relações recorrentes. Por fim, a CRNN obteve melhores resultados, já que contém como classificador, uma Rede Neural Artificial, e se diferente das demais que apenas contem *Flatten* e *Fully Connected*.

Já Feng, Liu e Yao (2017) desenvolveram um modelo CNN-RNN de processamento paralelo que alcançou resultados satisfatórios em relação a outras arquiteturas, como AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) e VGG11 (SIMONYAN; ZISSERMAN, 2015). Este trabalho desenvolveu esta técnica com o intuito de realizar um paralelismo não apenas espacial, mas um paralelismo computacional que resolve o problema da perda relacional dos quadros temporais ao longo do tempo que as CNNs não conseguem lidar. Realiza isto adicionando as características desta relação ao fim do processamento desta camada, em sua saída, em uma cruzamento de informações relacionais das classes e das características das imagens. Esta arquitetura utilizou uma CNN e uma Bi-RNN de tipo GRU em um modelo paralelo, nomeado pelo autor por PCRNN (Paralleling Convolucional Recurrent Neural Network).

Em (SENAC et al., 2017), o processo de classificação utiliza uma arquitetura de blocos de CNN para extração de características, estes são agrupados em uma sequência de 4 camadas convolucionais. A saída desta leva para duas camadas de Pooling em paralelo, MaxPooling e AveragePooling, seguidas por camadas de classificação entre 10 gêneros. Este modelo contém na saída 4 tipos de técnicas para classificação nomeadas de net_STFT, net_MUSIC, Fusion1 e Fusion2. Na net_STFT o processamento ocorre sobre as características de cada espectrograma. Na net_MUSIC ocorre sobre o Timbre, Detecção de Mudança Harmônica, entre outras características de dados de áudio gerando a probabilidade de cada clipe de 3 segundos. A FUSION1 recebe as medidas da média da probabilidade das duas estratégias anteriores, retornando a medida de maior valor entre elas. Já a FUSION2 recebe as medidas de probabilidades de cada clip e classe, retornando a classe majoritária com maior número de decisões.

Costa et al. (2011) utilizam um SVM para classificação em imagens de espectrogramas gerados sobre uma base de dados do *Latin Music Database*, extraindo sobre cada música 3 trechos, com intuito de não coincidir em momentos de pausa ou pouco conteúdo, cada um destes contém 9 segundos de áudio aproximadamente, para os

momentos inicial, meio e final. Cada espectrograma é dividido horizontalmente em 10 zonas, gerando um vetor de característica e decisão de classe sobre cada um destes. A escolha de classe para a música é medido pelo voto majoritário das classes em cada vetor, e assim por diante para cada música.

Em (JR.; KAESTNER; KOERICH, 2007) é utilizado para processamento múltiplos vetores de características extraídas de algumas partes das músicas da base de dados *Latin Music Dataset*, contando com mais de 3000 músicas de 10 gêneros latino-americanos. As músicas são processadas como dados de áudio em formato digital, particionado em 3 trechos de 30 segundos cada, um inicial, um no meio e outro no final. Após a geração das características, estes passam pelo classificador, para então retornarem a classe na predição. Como classificadores foram utilizados comparativamente o *Naive Bayes*, o *Support Vector Machines* e uma rede neural *MultiLayer Perceptron*. Foram aplicadas 4 estratégias de classificação nas saídas, contando com o valor majoritário, regra do valor máximo com o maior valor de probabilidade, a regra soma retornando a maior soma das probabilidades das classes, a regra do produto com o maior produto das probabilidades das classes e a *emsemble* reunindo os diversos trechos em um só conjunto de dados de classificação, apresentando melhores resultados de classificação.

Este trabalho baseou suas pesquisas para o desenvolvimento dos modelos nos trabalhos de Feng, Liu e Yao (2017) e Choi et al. (2016). Para comparação dos resultados deste trabalho, serão utilizado os trabalhos de (SENAC et al., 2017), Costa et al. (2011) e (JR.; KAESTNER; KOERICH, 2007).

2 Fundamentação Teórica

O presente estudo apresentado neste trabalho consiste em uma aplicação para a identificação de estilos de sotaques de Bumba-meu-boi, reunindo conjuntos de dados sobre detecção de áudios e toadas dos estilos de 5 sotaques diferentes existentes.

A aplicação consiste na tarefa de geração de um modelo que classifica estilos de sotaques de áudios recebidos, retornando os estilos de sotaques em que estes áudios fazem parte. Este capítulo apresenta o referencial teórico necessário para a construção do estudo proposto.

2.1 Bumba-meu-boi

Uma manifestação da cultura popular brasileira, presente em diversas regiões nacionais com diferentes nomes e diferentes interpretações, que em sua história de origem a sua apresentação ao público se difere de região para região. Com a região, se tem a origem da história, seu desenrolar e o nome dado a história, de acordo com MARQUES (1999):

- Maranhão, Rio Grande do Norte, Alagoas e Piauí é chamado de Bumba-meu-boi;
- Pará e Amazonas é boi-bumbá;
- Pernambuco é boi-calemba ou bumbá;
- Ceará é Boi-de-reis, Boi-surubim ou Boi-zumbi;
- Bahia é Boi-janeiro e Boi-estrela-do-mar;
- Paraná e em Santa Catarina é Boi-de-mourão ou Boi-de-mamão;
- Minas Gerais, Rio de Janeiro, Cabo Frio e Macaé é Bumba ou Folguedo-do-boi;
- Espírito Santo é Boi-de-reis;
- Rio Grande do Sul é Bumba, Boizinho ou Boi-mamão;
- São Paulo é Boi-de-jacá e Dança-do-boi.

A história se passa em uma fazenda escravista, em que uma escrava gestante, Mãe Catirina tem o desejo de comer língua de boi, Pai Francisco também escravo e seu marido, satisfaz o desejo de sua esposa, matando um boi da fazenda de seu senhor. Quando o fazendeiro sente a falta do boi, mandou procurá-lo, até descobrir o que aconteceu,

mandando Pai Francisco trazer o boi de volta, sob pena de ser morto como punição. Sem ter para quem recorrer, Pai Francisco busca Pajés e curandeiros para trazer o boi de volta a vida, o reanimando. O boi volta a vida, ressuscitando com um urro bem alto, momento em que a festa pela volta do boi se inicia, com Pai Francisco se livrando da punição.

As personagens dessa história são:

- Mãe Catirina, a escrava gestante, que tem o desejo de comer língua de boi, recebendo o nome de Catarina ou apenas Catirina;
- Pai Francisco, o escravo marido de Mãe Catirina, realiza o desejo de sua esposa colocando sua vida em risco, também nomeado de Pai Chico ou apenas Francisco;
- Fazendeiro, o senhor do casal Mãe Catirina e Pai Francisco;
- o Boi, sacrificado ao desejo humano, retornando a vida após ser ressuscitado por pajés e curandeiros;
- Cazumbá, personagem típico do estilo Sotaque de zabumba de Guimarães, interior do Maranhão, como uma representação animalesca dos bois da fazenda, para assustar após o sacrifício do boi e para animar após a ressurreição do mesmo;
- Padres, não é comum, mas algumas interpretações o incluem como uma personagem para realizar a cerimônia de ressurreição.

"As festividades do boi ocorrem no período junino, onde tem maior representatividade no nordeste, principalmente no Maranhão, em que o bumba-meu-boi é a principal atração dos festejos juninos, estando fortemente ligado ao ciclo de homenagens a São João, Santo Antônio, São Pedro e São Marçal"(FURLANETTO, 2010).

Por ser uma manifestação muito difundida na cultura maranhense, diversas regiões realizam interpretações sobre a ótica da história original, alterando as vestimentas de representação dos personagens, instrumentos do grupo ou a técnica para tocá-los, diferenciando em 5 estilos bem característicos, nomeados de sotaques, de origem e produção no interior e capital maranhense, os conhecidos sotaque de baixada, sotaque de costa de mão, sotaque de matraca, sotaque de orquestra e sotaque de zabumba.

As toadas, as músicas tocadas durante a festa, recontam trechos da história durante a apresentação e interpretação, em que durante o festejo, remonta as passagens e os personagens em forma de atuação narrada a versos, também retratando os personagens da história ou apresentando os ambientes em que essa se passa, não sendo incomum destacar os locais de origem e o grupo folclórico que está apresentando, como uma auto reflexão sobre o grupo e a festa junina, com São João normalmente sendo mencionado.

2.2 Sotaques e toadas

As expressões culturais estão relativamente ligadas a sociedade e como ela percebe a realidade ao seu redor, como a interação com o meio e como o meio influencia sobre o ser humano que a habita e observa. A música, a considerada a primeira arte, também tem ligação como uma das primeiras manifestações artísticas do ser humano para o conjunto social, como "o conteúdo da música é a experiência de um compositor nunca é puramente musical, mas pessoal e social, isto é, condicionada pelo período histórico em que ele vive e que afeta de muitas maneiras" (HEGEL; FISCHER, 1989), modificando e fixando as experiências humanas a seu contexto de interações e interpretações.

As toadas são as expressões musicais dos grupos folclóricos, com caráter de conto, descrição da história ou do lugar onde está o rodeia, parafraseando os personagens do conto, como também a personificação de João Batista, São João, o “anfitrião” da festa em que o mesmo se apresenta, muitas vezes como menino, como uma criança, retratando sua inocência em relação aos acontecimentos da história do boi.

Figura 1 – Brincantes tocando pandeirões de onça a altura da cabeça, típico do sotaque de matraca.

Fonte: (ABREU, 2017)

"Os sotaques do bumba boi são elaborados como categorias classificatórias que se baseiam em características distintivas do folguedo e de afirmação de pertencimento, bem como em significados específicos relativos à tradição"(ALBERNAZ, 2013), como expressão das manifestações culturais que se derivam e diferenciam nas regiões do Maranhão e onde estas expressões são realizadas. Estas expressões têm forte teor religioso, ligado a crenças de renascimento, como também ao ocultismo pelo curandeirismo e a crenças informais de relações sociais comuns, como a satisfação dos prazeres uma mulher em período gestacional buscando evitar a negatividade e problemas de caráter estético a criança que está por vir, comum em sociedades sertanejas pelo Brasil.

Os sotaques expressam estas características, tanto a cerimônia para ressuscitar o boi, quanto para evitar que a criança ou sua mãe gestante venham a sofrer por seus desejos gestacionais, com versos cheios de súplicas, lampejos e bênçãos. Estes versos mudam a cada sotaque, sendo realizados com maior frequência a grupos folclóricos mais tradicionais, estando presentes em toadas sobre o conto, desenvolvendo a história sobre cada uma de suas passagens. Na Figura 2 pandeirões afinados a fogo, uma necessidade para o couro da boi esticar e produzir o som desejado.

Figura 2 – Pandeirões afinados a fogo

Fonte: (SOUSA; BOGéA, 2021)

No Maranhão, as manifestações dos sotaques de Bumba-meu-boi tem relação direta a maneira como são caracterizadas as personagens, os instrumentos a serem utilizados para extrair os sons e a maneira com que são tocados, assim como explica Loureiro (2009) com sobre gêneros musicais, "... considerá-los como parte de um conjunto mais amplo de manifestações culturais. Os gêneros são comumente determinados pela tradição e por suas apresentações e não só pela música de fato", se somando em 5 (cinco) manifestações diferentes de sotaques de Bumba-meu-boi, são elas:

- Sotaque de baixada, utilizando roupas de penas e muito bem bordadas, tem produção sonora mais leve com pandeirões pequenos e tocados a altura do peito, maracás e matracas, é característico por contar com o Cazumbá em suas composições, utilizando uma careta, bata e um sino ao pescoço (como sinos de vaca), como exibido na Figura 3;
- Sotaque de costa de mão, referente a como é tocado nos pandeirões tocados a frente do tocador, utilizando a costa da mão para extrair o som desses instrumentos, com

Figura 3 – Cazumbás em apresentação, suas caretas e sinos são a principal marca do sotaque da baixada

Fonte: (PALHANO, 2013)

uma representação bem mais característica, muito rara em expressões musicais se tornando uma das mais primitivas desta manifestação;

- Sotaque de matraca, tendo como o principal instrumento a matraca, sendo o instrumento utilizado em maior quantidade pelos instrumentistas, contando com zabumbas, pandeirões tocados a altura cabeça e tambores, como exibido na Figura 1 e Figura 4, principalmente o tambor de onça, que rege o ritmo do grupo, mas em menor escala no conjunto final;

Figura 4 – Apresentação de grupo folclórico ao som de matracas

Fonte: (PANDA, 2018)

- Sotaque de orquestra, a expressão mais diferente das demais, por adicionar um caráter mais lúdico a peça, contém instrumentos de sopro e cordas, com pandeiros tocados a altura da cabeça e tambores com pouca participação, sendo a orquestra a regente da melodia e ritmo;

- Sotaque de zabumba, com principal instrumento a zabumba, com som extremamente característico, tem forte ligação ao Bumba-meu-boi tradicional como o sotaque costa de mão, apesar do instrumento principal ter caráter mais moderno, os tocadores a utilizam em pé com uma haste de madeira servindo de apoio ao chão, um padeirinho ou pandeirito afinado ao fogo e apito para reger a melodia, com ritmo leve e animado, semelhante ao samba.

2.3 Aprendizado de Máquina

Com ramo de estudo acadêmico e científico derivado da Inteligência Artificial, o Aprendizado de Máquina, em que literalmente como o seu nome descreve, um campo de estudo e aplicação que ensina as máquinas instruções para solução de problemas complexos, tarefas minimalistas, reconhecimento de padrões e classificação por características (ou que consomem demasiado tempo e são entediantes), tomando do ser humano o mínimo de atividade possível, aliando o aprendizado, uma característica da consciência humana e a capacidade de resolução de tarefas físicas com esforço repetitivo.

Atualmente, o Aprendizado de Máquina tem grande importância para a humanidade, desde o advento da era da informação, em que a troca, armazenamento e a necessidade de se realizar tarefas mais rapidamente com menor custo se tornou cada vez mais vantajosa.

Este meio de informações gerou um grande acúmulo de dados, que seja por necessidade ou em exploração de novos métodos de análise, amplia técnicas e áreas de estudo, seja com Big Data ou outros métodos, a busca de soluções práticas e baratas para problemas e diagnósticos com poucos retorno solucionável (ou em que esta ainda não foi pensada em ser solucionada), mas que pode apresentar uma alta eficiência para computadores, por sua substancial capacidade de precisão.

Com intuito em projetar neste campo da capacidade humana, de raciocínio e geração de conclusões por análise eventos e padrões, gerar conclusões, sistemas são desenvolvidos para adaptar tais funcionalidades, visando consumir uma gama de grande quantidade de recursos a disposição e um interesse crescente em adaptar o comportamento computacional ao comportamento humano, se estabelecendo um campo de estudo sobre o desenvolvimento de sistemas computacionais capazes de imitar o funcionamento de neurônios cerebrais, as Redes Neurais.

2.4 Redes Neurais

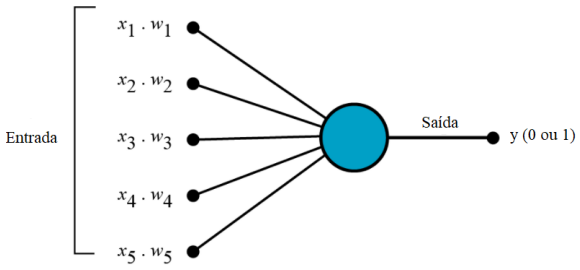
Com a grande percepção que a Inteligência Artificial trouxe a ciência moderna, sua capacidade de integrar máquinas físicas a máquinas computacionais, e simular o funcionamento biológico do cérebro humano em computadores, com intuito imitar o sistema neurológico humano, a maneira como estas estruturas enviam e recebem sinais, em suas pequenas estruturas celulares, os neurônios.

A Redes Neurais Artificiais (NNs) são estruturas compostas de neurônios artificiais interligados, formando camadas, estas camadas têm funções específicas, em que na grande maioria das aplicação se utiliza dos 3 (três) tipos de camadas existentes:

- Camada Entrada de Dados, em que os neurônios recebem as informações da rede neural e realizam as primeiras etapas de processamento;
- Camada Oculta, em que o processamento de dados se dá de fato, pode ser uma ou mais, dependendo do problema a ser resolvido ou da estratégia de resolução;
- Camada de Saída para retorno das informações.

A redes neurais artificiais funcionam como estímulos nos nós desta rede, como um neurônio biológico, que ativa o processamento de uma função que calcula as somatórias dos vetores de dados de entrada, seus pesos e o viés limite, testando o resultado em uma função específica para ativar ou não a saída do neurônio artificial. Estas etapas acontecem em um único neurônio, que por sua vez pode ser considerado uma rede neural de apenas uma camada, chamada de Perceptron. Este sistema tem processamento linear e tem como função classificar a entrada de dados fornecida em relação aos parâmetros dados.

Figura 5 – Perceptron



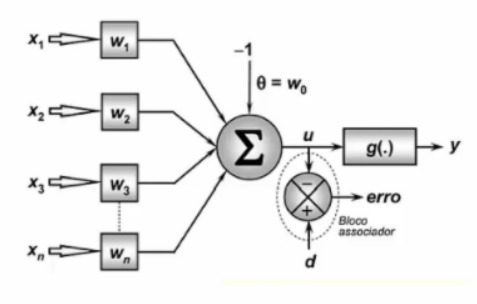
Fonte: (BOOK, 2021b)

Para a grande maioria das necessidades de aplicação desta técnica o resultado obtido com o perceptron pode não ser considerado ideal ou próximo do ideal, visto que, como sua interpretação dos padrões visto na entrada são de natureza linear, ou seja, a característica linear em que o Perceptron é construído impede que o mesmo perpasse o

fator de linearidade, não sendo possível o reconhecimento de características suficientemente aceitável para um mapeamento não linear, por exemplo, como observado por Minsky e Papert (1969). Um exemplo de um Perceptron é dado na Figura 5.

Por se tratar de uma única etapa de processamento, e dependendo dos pesos e bias, para resultados não lineares, o resultado pode ser rejeitado ou pouco aproveitado. Para solucionar esse erro, e aproximar os valores resultantes para próximo do valor desejado, diminuindo o erro resultante, Widrow e Hoff (1960) propuseram a rede Adaline (*Adaptive Linear Element*), uma solução muito semelhante a do Perceptron, diferenciando apenas do treinamento entre elas, um exemplo de um neurônio Adaline é dado na Figura 6.

Figura 6 – Rede Neural Adaline



Fonte: (MOREIRA, 2018)

Para solucionar a diferença de erro, adicionaram após o treinamento, uma função Regra Delta que altera os pesos, diminuindo o valor da função de erro, aplicando o método do gradiente descendente, possibilitando assim a convergência para um mínimo da função de erro.

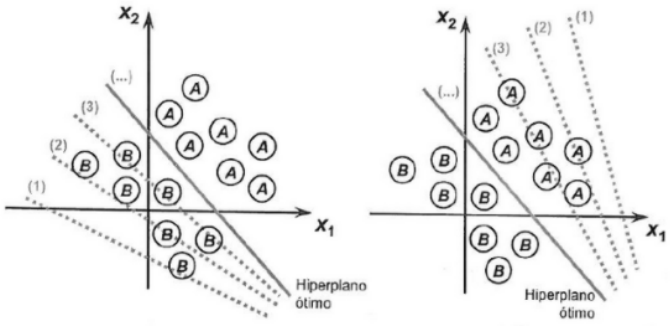
$$erro = d \cdot u \tag{2.1}$$

Essa técnica também conta com a iteração até um valor limite. Caso seja alcançado o valor esperado, respeitando o erro percentual mínimo ou valor de desvio mínimo, o resultado é aceito como a melhor solução. A Figura 7 exibe o desempenho de uma função de erro na Equação 2.1 em busca do melhor resultado possível.

Outra solução, ainda mais sofisticada, seria a inclusão de novos neurônios Perceptrons rede, chamada de Multi-layers Perceptron (Perceptron Multicamadas), Figura 8, desenvolvida por Rumelhart, Hinton e Williams (1986), após perceberem que mais Perceptrons podem ser incluídos, como também agrupar mais camadas a rede.

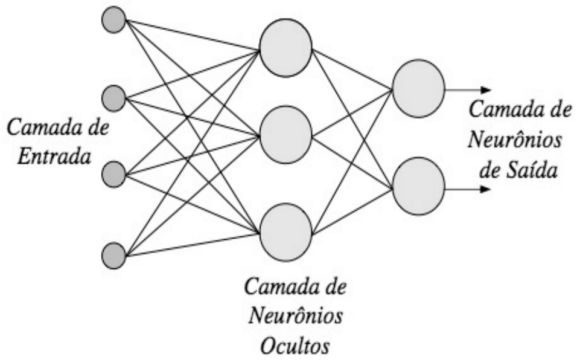
Agora as redes neurais contam com uma camada de entrada de padrões, pesos e viés, a camada de saída de dados com a quantificação do erro obtido durante o processamento e uma ou mais camadas intermediárias, também chamadas de camadas ocultas, que realizam o processamento dos padrões, passando a próxima camada os valores resultantes, até

Figura 7 – Desempenho de uma função de desvio mínimo no Hiperplano ótimo



Fonte: (MOREIRA, 2018)

Figura 8 – Rede Neural Multicamada



Fonte: (CINTRA, 2018)

a camada de saída. Este modelo resolve o problema do Perceptron simples, entretanto adiciona um pouco mais de complexidade para tal solução.

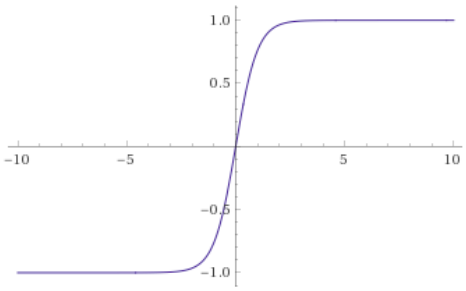
Uma outra característica das redes neurais MLP é a inclusão da retropropagação (backpropagation ou propagação reversa) que consiste em associar valores da camada de saída atual ao valor desejado, se não estiver correta, calcula o erro resultante e na próxima etapa altera os valores dos pesos na próxima camada de entrada. Redes de retropropagação (backpropagation) trabalham com uma variação da regra delta, a delta generalizada, apropriada para redes multicamadas.

A principal característica desta aplicação é a função de ativação na camada de saída, como a rede multicamadas contém camadas ocultas, que realizam processamento de natureza não-linear, como também a sua saída deve ser de natureza não-linear, utilizando funções que geram hipérboles, por exemplo, a função de ativação tangente hiperbólica, Figura 9 e Função 2.2, que é utilizada em aplicações onde a saída gerada tem valores entre

1 e -1.

$$\phi(x) = \tanh x \tag{2.2}$$

Figura 9 – Gráfico de tanh(x)



Fonte: (JACQUES, 2020)

2.4.1 Redes Neurais Convolucionais

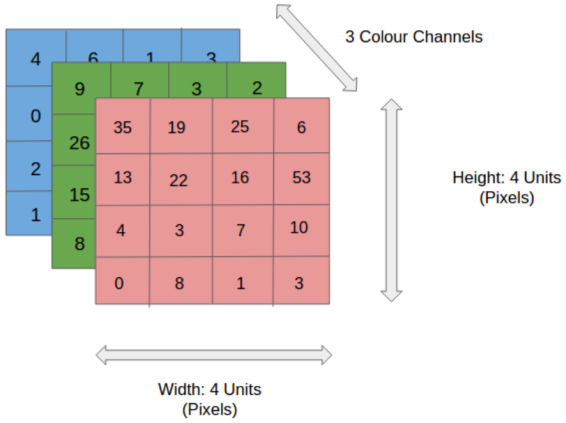
A partir das Redes Neurais Multicamadas Perceptron, outras redes foram desenvolvidas para resolver problemas específicos, modificando e implementando novas técnicas para as conexões entre os neurônios e camadas, apresentando estas em padrões de Redes Neurais ao longo de cada trabalho publicado. As Redes Neurais Convolucionais (ou ConvNet ou Convolutional Neural Network ou CNN) foram desenvolvidas com o intuito de acelerar o processamento de imagens digitais, respeitando a natureza digital e fatores biológicos do funcionamento do córtex humano.

Com o estudo inicial sobre a neurociência humana, em 1962, Hubel e Wiesel (HUBEL; WIESEL, 1962), analisaram as conexões neurais com o córtex humano, e descobriram uma certa característica, alguns neurônios são ativados ao reconhecer linhas, curvas e bordas, ou seja, silhuetas de formas reais, destacando que nosso cérebro reconhece objetos primeiramente por suas formas e bordas, esta declaração o primeiro passo para o desenvolvimento das CNNs.

Para as Redes Neurais Convolucionais, a entrada se resume, em matrizes para altura, largura e profundidade, determinadas nas dimensões da imagem para os dois primeiros e na quantidade de escala de cores para o último, como é exibida na Figura 10.

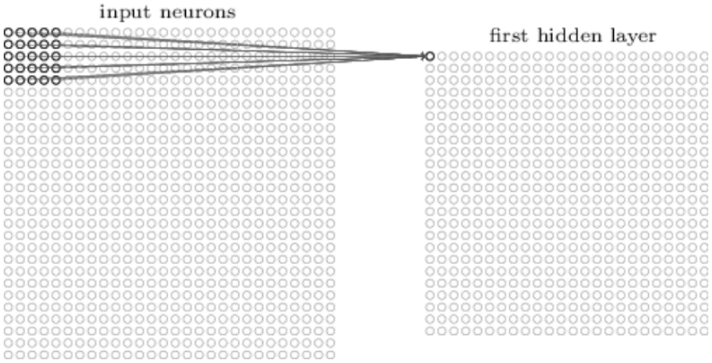
As convoluções consistem em adicionar os pesos na primeira execução aleatoriamente, ajustando-os conforme o sistema avança em seu processamento. Um filtro que captura parcialmente várias porções da imagem original, recebe a característica mais marcante desta porção e a salva em uma matriz resultante referente a porção extraída, realizando estes processos continuadamente até construir a nova imagem de características resultante a partir da imagem original. A Figura 11 exemplifica este processo de convolução.

Figura 10 – São 3 canais de cores em uma imagem de codificação RGB



Fonte: (BOOK, 2021a)

Figura 11 – Convolução com filtro de extração 5x5



Fonte: (ALVES, 2018)

Assim como as Redes Neurais Multicamadas, as funções de ativação servem para garantir a não-linearidade da rede, para que a saída obtida alcance todos os padrões de características possíveis (dentro do conjunto de características possíveis). Para esta tarefa, pode-se utilizar, as funções sigmoid, tangente hiperbólica e softmax, entretanto, para redes neurais convolucionais a mais indicada é a ReLU, por ser mais econômica, já não ativa todos os neurônios ao mesmo tempo, exceto aqueles com valores iguais ou superiores a zero, substituindo por zero os demais valores negativos obtidos após a geração das derivadas da saída da camada anterior, reduzindo o custo de processamento.

O processo Convolucional consiste em 3 (três) camadas principais:

- Camada Convolucional é responsável pela extração de características das imagens de entrada e retornar o vetor de imagens resultantes na forma de mapa de recursos;

- Camada de Pooling é responsável por resumir a informação contida na matriz resultante obtida na camada de convolução, com o objetivo de evitar o overfitting e diminuir a quantidade de pesos da camada. Por exemplo, em uma matriz 24x24, uma unidade de área de tamanho 3x3 é escolhida para transitar por esta, resultando em uma matriz de tamanho 8x8, que também depende do tamanho do stride. Esta unidade é responsável por resumir as informações obtidas na área escolhida em um único valor. Existem alguns tipos de escolha do valor a ser resumido, o método mais comum é o MaxPooling, resumindo a área no maior valor obtido;
- Camada Fully Connected é responsável por organizar a saída da camada anterior para a entrada da próxima camada, está com N neurônios, de tamanho igual para a quantidade de classes para o modelo.

2.4.2 Redes Neurais Recorrentes

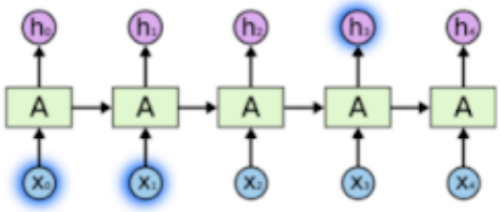
A leitura humana é uma aquisição constante de símbolos, com significados e signos, construídos e disponibilizados em uma sequência lógica que respeita regras de linguagem pré-estabelecidas, mas estas ao serem lidas, não são esquecidas logo em seguida, são guardadas para que a sequência tenha coesão e uma harmonia contextual, como para extrair seus significados ou signos da linguagem.

Assim, durante uma leitura de um texto, o ser humano lê as palavras sequencialmente, não esquecendo as palavras lidas anteriormente para ler as novas, é mantida uma leitura constante guardando em sua memória o que leu para compreender aquilo que é mais relevante que cada palavra lida, o contexto que esta descreve. Seguindo esta análise e observando o padrão de construção das redes neurais, cada estrutura de camada contém sua função, com os parâmetros dos dados processados na etapa de tempo presente refletindo nas etapas de tempo seguintes, nos dando uma ideia de sequencialidade durante o processamento, com um passado, presente e futuro.

Logo foi proposto um sistema de Redes Neurais capaz de passar a novas camadas aquilo que havia sido processado anteriormente (ou parte do processamento) para as novas camadas a seguir, as Redes Neurais Recorrentes (Recurrent Neural Network ou RNN), que focam na filosofia de processamento com "lembranças" do passado, para que aquilo a ser construído tenha as características tanto do todo, e não somente das últimas camadas e pesos, agregando maior probabilidade de resolução do problema com maior precisão. Como é exemplificado na Figura 13.

O propósito central é contar com informações anteriores à tarefa presente, e assim passar as informações obtidas da tarefa atual às tarefas seguintes. A sequência de Funções 2.3 exibe este processo, em que a tarefa na camada oculta anterior realiza seu processamento e passa para a próxima etapa as informações das variáveis independentes $\underline{X}$ e o resultado

Figura 12 – Processo de "lembrança"de parâmetros ao longo do processamento das camadas ocultas



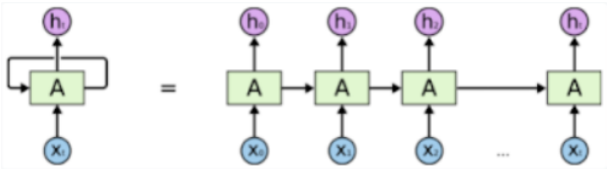
Fonte: (JUNIOR, 2019)

do seu processamento Ht-1, e assim por diante, cada processamento realizado é um período de tempo a ser contabilizado, ou realizado.

$$
\begin{aligned}
H_0 &= \phi\left(b_h + XW_x\right) \\
H_1 &= \phi\left(b_h + XW_x + H_0W_h\right) \\
H_2 &= \phi\left(b_h + XW_x + H_1W_h\right) \\
H_3 &= \phi\left(b_h + XW_x + H_2W_h\right) \\
\hat{y} &= b_0 + H_3W_0 \\
\lambda &= \frac{1}{n}\sum_{i=0}^{n}\left(\hat{y} + y_0\right)
\end{aligned}
\tag{2.3}
$$

O processamento ocorre da seguinte maneira, no estado inicial, os dados são inserido na camada oculta H0 (etapa inicial), processados e passados para a próxima etapa os seus dados de processamento H1, H2 e H3, e as variáveis independentes X, que as processa na etapa atual e passa para a próxima etapa os seus dados de processamento Hx e as variáveis independentes, assim por diante. Resumindo, as RNNs são Redes Neurais muito profundas, que se desenrolam ao longo do tempo.

Figura 13 – Processo de feedback de informações desenrolado ao longo da rede



Fonte: (JUNIOR, 2019)

2.4.3 LSTM

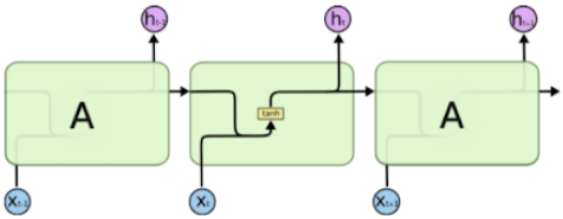
As Redes Neurais Recorrentes conseguirem adicionar uma característica importante, um carregamento de porções de informações relevantes dos períodos de tempos anteriores,

em que as informações são passadas para a próxima etapa, salvas e carregadas nos próximos períodos de tempo seguintes, em que o algoritmo percebe que estas são relevantes. Esta estratégia se propaga a cada período de tempo, até o momento em que o algoritmo necessita substituir alguns dos dados obtidos em períodos de tempo anteriores por dados de períodos de tempo mais atuais, perdendo conexões com etapas anteriores, esta situação é nomeada de vanish gradient problem, não havendo possibilidade de carregar determinada informação necessárias em determinado ponto distante do ponto de interesse e as informações obtidas deste ponto de interesse já foram perdidas na entrada, é a perda de informações quando a distância entre os períodos de tempo são muito grandes, tornando-se incapaz de aprender a conectar estas informações.

Sendo refém deste problema, percebido por Hochreiter (1991) e estudado mais a fundo por Bengio, Simard e Frasconi (1994), dando origem aos LSTMs desenvolvida por Hochreiter e Schmidhuber (1997), com intuito de solucionar este problema principal das RNNs, com algumas mudanças realizadas na estrutura principal dos neurônios do RNN comum.

As cadeias de neurônios do RNN contém módulos repetidos de Redes Neurais em períodos de tempo, que internamente contém uma única camada de uma função tanh, como exibida uma célula RNN comum internamente na Figura 14.

Figura 14 – Estrutura de uma célula RNN



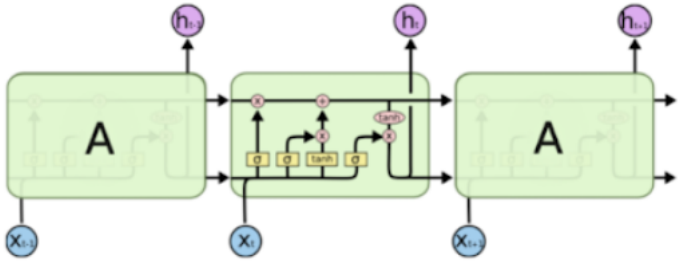
Fonte: (JUNIOR, 2019)

Seguindo uma estrutura parecida, os LSTMs contém algumas outras funções internas, alpha, operações em alpha e tanh em x, contendo quatro camadas de interação interna em cada neurônio.

Internamente cada neurônio LSTM contém uma linha de transmissão de informações para as células seguintes, chamada de estado da célula, Figura 16, com possibilidade de remover ou adicionar informações. Estas informações são controladas por portas, como observado na Figura 15, ligadas as passagens de informações que se conectam diretamente do centro da célula, onde realizados os processamentos.

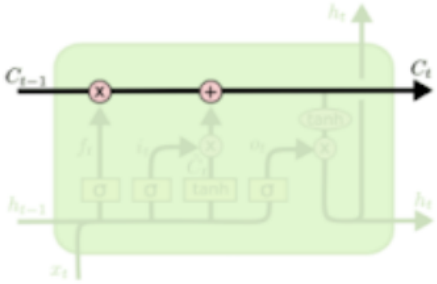
As portas nada mais são que uma medida de probabilidade sobre o grau de importância desta sobre a célula e sobre o estado da célula, que permite que o estado Ct-1 se altere ao longo do processamento da mesma, medida sobre uma função sigmoid,

Figura 15 – Estrutura de uma célula LSTM



Fonte: (JUNIOR, 2019)

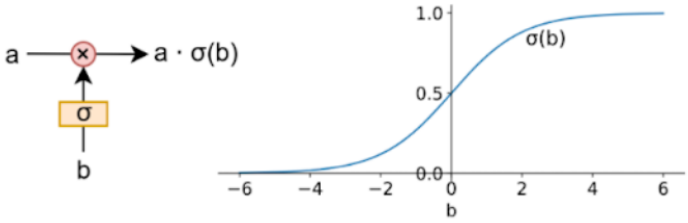
Figura 16 – Conexão da transmissão de estado entre células



Fonte: (JUNIOR, 2019)

que observa os valores de Ht-1 e Xt, produzindo valores em escala de 0 a 1, em que 1 as informações são enviadas e 0 ficam retidas, sendo replicada esta tarefa nas 3 portas existentes.

Figura 17 – Função e gráfico Sigmoid



Fonte: (MATSUMOTO, 2019)

As portas são:

- forget gate, responsável por esquecer determinados dados selecionados;
- input gate, responsável por decidir se determinadas informações podem ser adicionadas ao estado da célula;

- output gate, responsável por decidir quais partes da célula atual são importantes para serem passada adiante.

Na primeira etapa de processamento de portas, as informações são lidas através dos vetores Ht-1 e Xt, sendo calculadas pela função sigmoid, Figura 17, em seguida pelo estado da célula anterior, para gerar o estado da célula atual.

$$\begin{aligned} f_t &= \sigma\left(U_f x_t + V_f h_{t-1} + b_f\right) \\ C_t^{'} &= f_t \cdot C_{t-1} \end{aligned} \tag{2.4}$$

Em seguida, no input gate, é recebido uma lista C+ com todas as informações do estado da célula atual, em que vão ser escolhidas por uma função tanh, que serão mantidas por um grau de importância em um vetor Ct, adquiridas junto ao vetor i.

$$\begin{aligned} i_t &= \sigma\left(U_i x_t + V_i h_{t-1} + b_i\right) \\ C_t^{+} &= \tanh\left(U_c x_t + V_c h_{t-1} + b_c\right) \\ C_t &= C_t^{'} + i_t \cdot C_t^{+} \end{aligned} \tag{2.5}$$

Por fim, a output gate recebe o vetor Ct de estado da célula atual, que passa para uma função de ativação tanh, e processa quais informações são relevantes para se passar adiante e quais devem ser descartadas juntamente a um vetor ot.

Neste sistema, o tratamento de memória da leitura e aprendizado ocorre a cada gate, quando são aplicadas as funções de ativação, as funções tanh e sigmoid, impedindo que o vanish gradient ocorra por falta de função de ativação na saída da célula.

2.5 Funções de ativação

Ao longo do processamento, as Redes Neurais Artificias adicionam em seus neurônios de saída informações adquiridas ao longo das atividades realizadas nas suas camadas, entretanto algumas dessas informações podem não ser tão relevantes para a cadeia de processamento, como para a classificação ou previsão, então estas necessitam ser descartadas, ou desativadas, da saída destes neurônios, sendo necessárias funções específicas para tal necessidade.

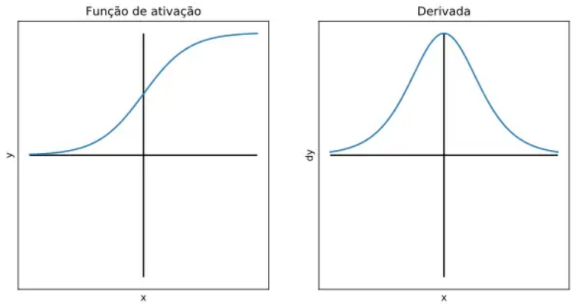
Um problema que pode aparecer ao longo do treinamento é ao fim de camadas de Redes Neurais Artificiais, como os perceptrons, que produzem uma saída de formato linear, para problemas de domínio linear não representam nenhum problema, entretanto para problemas de natureza não-linear, que necessitam solucionar problemas de classificação por exemplo, com uma saída linear gera um resultado inválido, necessitando de um tratamento. As Funções de Ativação acabam por selecionar as informações ao fim das camadas, ativando ou desativando neurônios e seus respectivos conteúdos que não são interessantes.

2.5.1 Sigmoid

É uma das funções mais amplamente utilizadas, pelo fato de que esta não é linear, mas não-linear, formando um S, uma hipérbole, com valores variando entre 0 e 1. Seu objetivo é tentar manipular os pesos para próximo de um valor específico, sendo um dos motivos por ser muito utilizada, perfeita quando se deseja criar um modelo classificador. Sua fórmula, Função 2.5.1 desenvolve uma hipérbole, Figura 18.

$$\sigma(x) = \frac{1}{1 + e^{(-x)}} \qquad \sigma'(x) = \sigma(x)(1 - \sigma(x)) \tag{2.6}$$

Figura 18 – Gráfico da Função de Ativação Sigmoid e comportamento da sua derivada



Fonte: (CECCON, 2020)

Apresenta uma solução simples que resolve o problema da não-linearidade para classificação para mais de 2 classes, mas sua solução é um de seus problemas, como tem a tendência de aproximar muito seus pesos para determinado sentido (para próximo de 0 e 1) o nível de aprendizado sobre as classes se altera, entretanto, quando seus pesos se aproximam demais de zero o modelo não aprende, comprometendo a continuidade do processamento. Outro problema seria a unisinalidade, seus pesos são apenas positivos, algo não desejável em certas ocasiões, em que seria mais interessante que alguns pesos fossem decrementados para haver um equilíbrio entre as interpretações das classes.

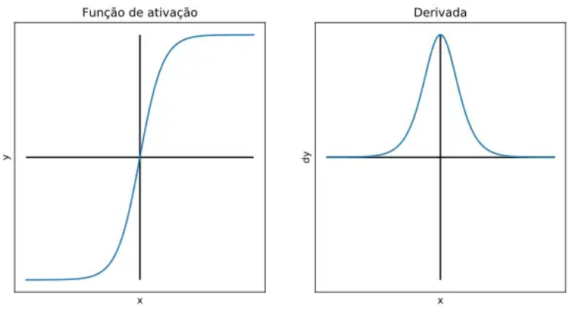
2.5.2 TANH

Com a função Sigmoid havendo alguns problemas, e sendo mais desejável em algumas ocasiões que alguns pesos sejam negativos (valores inferiores aos existentes no conjunto de valores reais 0 a 1), a função tanh alcança este conceito, alcançando valores em -1 e 1, como observado na Figura 19.

$$\tanh(x) = 2\sigma(2x) - 1 \qquad \tanh'(x) = 1 - \tanh^2(x) \tag{2.7}$$

A sua fórmula de função aparenta muito com a função sigmoid, isto por que a função tanh deriva desta, com leves ajustes, como observado na Função 2.7.

Figura 19 – Gráfico da Função de Ativação tanh e comportamento da sua derivada



Fonte: (CECCON, 2020)

2.5.3 ReLU

A função de ativação mais amplamente utilizada para camadas de processamento de Redes Neurais, pois contém a eficiência e simplicidade que nenhuma outra função contém, ao mesmo tempo que resolve com robustez o problema de linearidade. Isto se deve ao fato de que ao receber de neurônios da camada de processamento, verifica-os e desativa aqueles que têm valores abaixo de zero, simplificando o resultado, deixando a rede mais leve e esparsa para a continuidade do processamento. Entretanto, sua estratégia pode apresentar problemas a gradientes que se aproximam de zero, por sua estratégia de retornar zero a valores negativos acaba por "matar" alguns neurônios por receber estes valores, impedindo que progridem ou melhorem seu desempenho, sem haver uma chance de contribuir positivamente com a continuidade do processamento. Outro problema seria a "explosão" de valores positivos, já que apenas retorna o próprio valor positivo, sem haver qualquer tratamento.

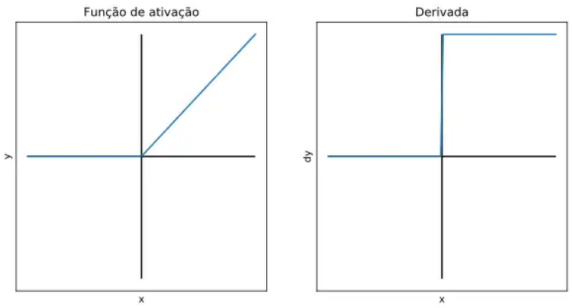
$$ReLU(x) = max\{0, x\} \qquad ReLU'(x) = \begin{cases} 1, & \text{se x} \geq 0; \\ 0, & \text{c.c.} \end{cases} \tag{2.8}$$

Retornando o próprio valor a entradas positivas e apenas retornando zero a valores negativos.

2.5.4 LeakyReLU

Esta função continua com a estratégia anterior, Figura 2.9, apenas adicionando uma mudança a valores negativos. Com o problema da função ReLU, a LeakyReLU aplica, sobre a mesma estratégia da anterior, uma divisão do valor negativo por um valor especificado pelo desenvolvedor, removendo o problema de "morte" a neurônios com valores indesejados,

Figura 20 – Gráfico da Função de Ativação ReLU e comportamento da sua derivada

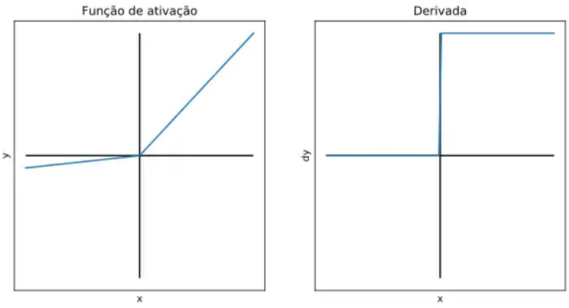


Fonte: (CECCON, 2020)

apenas aproximando levemente estes ao zero.

$$LeakyReLU(x, \alpha) = max(\alpha x, x) \qquad LeakyReLU'(x, \alpha) = \begin{cases} 1, & \text{se x} \geq 0; \\ \alpha, & \text{c.c.} \end{cases} \tag{2.9}$$

Figura 21 – Gráfico da Função de Ativação LeakyReLU e comportamento da sua derivada



Fonte: (CECCON, 2020)

Como exibido no gráfico da Figura 21, o retorno a valores negativos não é zero, mas valores bem próximos de zero. É normalmente utilizada quando a função ReLU tem dificuldades de convergir.

2.5.5 Softmax

Para problemas de classificação, soluções simples com saídas binárias, sim ou não, não resolvem o problema, necessitam de algo mais robusto, que gere uma relação entre as classes em questão. A Softmax apresenta uma função tipo sigmoid, Função 2.10, com uma saída com valores entre 0 a 1, entretanto, adiciona uma relação de probabilidade dos valores obtidos na saída para cada classe, realizando uma correspondência da probabilidade

de cada valor objeto pertencer a uma classe, especificamente.

$$\sigma(z)_j = \frac{e^{zj}}{\sum_{k=1}^{K} e^{z_k}} \qquad , j = 1, 2, 3, ..., K \tag{2.10}$$

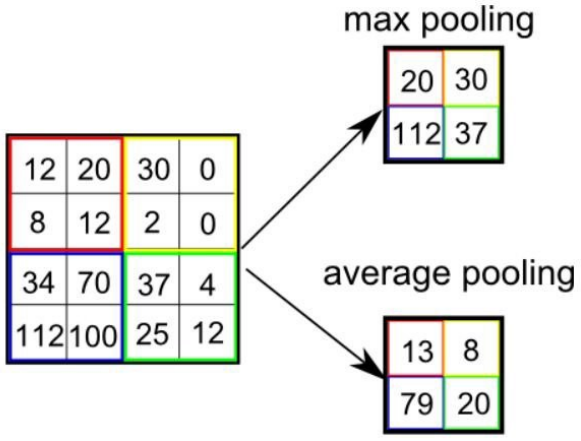
Esta equação resolve o problema de linearidade adicionando um viés de classificação ao resultado, direcionando as probabilidades da saída às classes pertencentes ao modelo.

2.6 Pooling

Uma camada de ajuste de mapa de características adicionada logo após a camada de ajuste de Função de Ativação de não-linearidade, com o objetivo de reunir os recursos obtidos das imagens em uma resolução inferior à recebida em entrada, visto que, uma grande quantidade de recursos pode inviabilizar a percepção das características necessárias a identificação de padrões de classificação, em linhas e objetos ao longo das imagens.

A camada de Pooling realiza uma extração do mapa de características da superfície da imagem a partir de uma mapa menor, 2x2 com saltos de 2 em 2, por exemplo, capturando as informações nesse mapa de extração e realizando operações matemáticas de resumo deste mapa em um pixel, adicionado a uma novo mapa de características resultante que definirá o mapa original, como observado na Figura 22. Há dois tipos de resumo matemático realizado em camada de ajuste de Pooling, o Average Pooling (agrupamento de média) e o Maximum Pooling (agrupamento máximo ou normalmente conhecido em Redes Neurais Artificiais, MaxPooling):

Figura 22 – Processos de Max Pooling e Average Pooling sobre uma imagem



Fonte: (YINGGE; ALI; LEE, 2020)

- Average Pooling realiza a extração básica e aplica um filtro de média sobre os valores obtidos ao mapa extraído, a média obtida é o valor resumo deste mapa de extração.

- Maximum Pooling realiza a extração básica e aplica um filtro de valor máximo entre os valores obtidos no mapa extraído, este valor máximo é o valor resumo dos valores deste mapa de extração. É o método de resumo de agrupamento mais utilizado por ser mais eficiente, não realizando excessivas operações matemáticas em passos de menor escala.

2.7 Espectrogramas

Dentro do plano cartesiano, um objeto matemático composto por duas retas numéricas perpendiculares em um ângulo de 90º, podemos desenhar dados de pontos numéricos, funções lineares e hiperbólicas e figuras geométricas apenas visualizada em 2(duas) dimensões, que representam sinais de energia, mudanças de estado, dados de funções e/ou valores diversos. Se tratando de funções neste plano, seus valores podem ser estudados a ponto de realizar o planejamento de desempenho dos momentos ou estados de transição, em uma apresentação visual e matemática.

Espectros são sinais de energia em propagação, em “movimento”, representadas geralmente na forma de ondas por sua instabilidade em se conter no meio, seja por sua fonte ou resistência durante a transmissão, sua propagação depende da sua natureza, e sua existência de uma fonte geradora ou transformadora para que tal fenômeno aconteça.

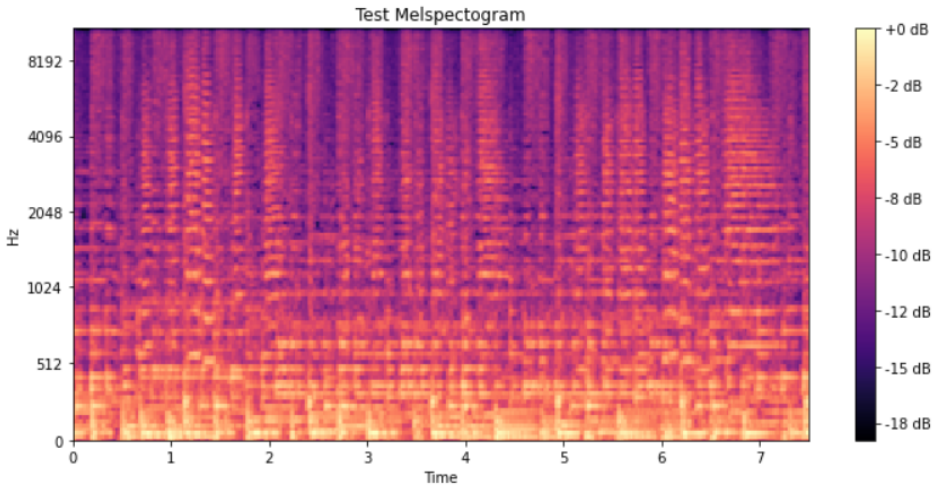
"O espectrograma é um gráfico tridimensional, com o domínio das frequências no eixo vertical, o domínio do tempo no plano horizontal e a amplitude dos componentes da onda sonora representada pelo contraste de cores no traçado" (LOPES; ALVES; MELO, 2017). Assim, a análise espectrográfica depende da análise visual sobre as medições acústicas realizada sobre a produção sonora, em uma plano gráfico multicolorido e numérico direcionando a 3 (três) grandezas, frequência, tempo e amplitude. É exibido na Figura 23, no plano sobre as grandezas da frequência, tempo e intensidade sonora (dB).

Os sinais de áudio na verdade são no nível de som em determinadas frequências expostas ao longo do tempo, ou seja, uma frequência que se dispersa ao longo do tempo.

A medida de escala de som, em níveis é medida em decibéis (dB), uma escala relativa para diferença de níveis de som no ambiente, uma relação entre picos e depressões das ondas sonoras, recebendo também o nome de volume. Também é utilizada como uma grandeza para medir pressão e potências sonoras.

A frequência de um som é definida pela quantidade de vibração das ondas mecânicas do som, ao número de ciclos de uma onda sonora ao longo do tempo, quantificando sua altura determinando sons graves e agudos, quanto maior a vibração sonora mais agudo será o som, quanto menor a vibração sonora mais grave será o som. A unidade de medida é Hertz (Hz), quantifica o número de ciclos/segundos (frequência).

Figura 23 – Exemplo de uma imagem mel-espectrograma.



Fonte: acervo do autor.

A taxa de amostragem é a quantidade de vezes que a amplitude é medida a cada segundo, usada para conversão de sinais analógicos para sinais digitais. Também determina a dimensão de amostras por segundo, sendo as amostras, os valores de um sinal analógico medidos em um período de tempo. Assim, em relação ao número de amostras no arquivo de áudio pela taxa de amostragem, se obtém a duração total desta produção de áudio.

2.7.1 Transformada de Fourier

Para geração de gráfico de sinais, necessitamos de um estudo sobre propriedades matemáticas que calculam a geração ou transformação de sinais, como nosso enfoque são sinais de áudio, devemos partir para o estudo das transformações de frequência de sinal.

A Transformada de Fourier permite analisar funções não periódicas, de modo a analisar como uma soma de ondas senoidais, ondas conhecidas como infinitas. A estratégia é o particionamento do tempo de todas as ondas espaços de tempo menores e processar cada um desses individualmente, retornando a soma de todos estes como o resultado total.

Para o matemático Joseph Fourier (1768-1830), sinais periódicos podem ser decompostos em uma única soma de ondas senoidais puras e, em cada uma dessas ondas, a frequência de cada senóide é um múltiplo, com valor inteiro, da frequência fundamental da onda original, denominados “harmônicos”. Cada um desses sinais é nomeado como série trigonométrica de Fourier, série por conter o conjunto dessas frequência ao longo do tempo. A série pode ser descrita por estas Equações 2.11, 2.12, 2.13 e 2.14:

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left[a_k \cdot \cos\left(\frac{2\pi}{T} \cdot kt\right) + b_k \cdot \text{sen}\left(\frac{2\pi}{T} \cdot kt\right) \right] \tag{2.11}$$

sendo:

$$a_k(t) = \frac{2}{T}\int_T x(t) \cdot \cos\left(\frac{2\pi}{T} \cdot kt\right) dt \qquad , k = 1, 2, 3, ..., K \tag{2.12}$$

$$b_k(t) = \frac{2}{T}\int_T x(t) \cdot \text{sen}\left(\frac{2\pi}{T} \cdot kt\right) dt \qquad , k = 1, 2, 3, ..., K \tag{2.13}$$

onde:

$$T = \frac{2\pi}{\omega_0} \tag{2.14}$$

obseem que T é o período da fundamental do sinal x(t), retornando o período de todas as amostras do conteúdo,

A função x(t) é conhecida como equação de síntese e ak e bk como equações de análise. A série de Fourier é utilizada para realizar a representação de uma função periódica em uma soma de senos e cossenos. Já a Transformada de Fourier realiza uma representação de uma função geral, que engloba não somente funções periódicas, mas qualquer solução que necessite gerar agrupamento de exponenciais complexos.

Inicialmente o sinal é gerado através do espectro percebido, por exemplo na Função 2.15:

$$x(t) = A_1 \cdot \text{sen}\,(2 \cdot \pi) \cdot f_1 t \tag{2.15}$$

Resumidamente, processo da FFT é decrementar a DFT em funções menores, gerando uma sequência de amostras do sinal de entrada convertendo o resultado em espectro de amplitude e fases harmônicas na saída.

Assim como na Função 2.16, são N amostras espaçadas ao longo de intervalos T de um espectro de sinal, em que:

$$T = \frac{T_1}{N} = \frac{1}{Nf_1} \qquad \rightarrow \qquad f_1 = \frac{1}{NT} \tag{2.16}$$

com frequência f1, N para o número de amostras e T1 para o período.

E as amostra de x(t), podem ser expressas levando em consideração T na função discreta, como na Função 2.17:

$$x(nT) = A_1 \cdot \text{sen}\,(2\pi) \cdot f_1\,(nT) = A_1 \cdot \text{sen}\left(\frac{2\pi}{N}n\right) \triangleq x(n) \qquad , n = 1, 2, 3, ..., N \tag{2.17}$$

Acrescentando a harmônica h, exibida na Função 2.18, a série de amostras resulta em:

$$y(n) = A_1 \cdot \text{sen}\left(\frac{2\pi}{N}n\right) + A_h \cdot \text{sen}\left(\frac{2\pi h}{N}n\right) \qquad , n = 1, 2, 3, ..., N \tag{2.18}$$

Por fim, a solução para a saída é substituir por valores em N, h, A1 e Ah com o objetivo de se alcançar o período de sinal N vezes. A saída z(n) da FFT corresponde as entradas y(n) em que n=1,2,3,...N, com valores complexos no domínio da frequência. Observando que para se obter uma FFT basta dividir em N parte uma DFT, então cada magnitude de saída deve ser dividida por N, assim como pode ser observado na sequência de Funções 2.19, 2.20 e 2.21:

$$A_0 = \frac{Abs\,(Z_1)}{N} \qquad \text{, para } I = 1 \text{ resulta o nível CC.} \tag{2.19}$$

$$A_{i-1} = \frac{Abs\,(Z_1)}{N} \qquad \text{, para } I > 1 \text{ resulta metade das amplitudes das harmônicas.} \tag{2.20}$$

$$\theta_{i-1} = tg\left[\frac{lm\,(Z_1)}{Re\,(Z_1)}\right] \qquad \text{, para } I > 1 \text{ dependendo do quadrante e de Re(.)} \neq 0. \tag{2.21}$$

Para testar todo o processo da FFT precisa-se inicialmente gerar a sequência de amostras do sinal de entrada e no final converter o vetor de saída em espectro de amplitude e fase das harmônicas.

2.7.2 Transformação de Fourier de Tempo Curto

Dentre todas as aplicações anteriores das interpretações da Transformada de Fourier, esta se baseia focando em trabalhar sobre o espectro da frequência quantificável e tempo infinito, com objetivo de simplificar os cálculos, mas subjugando uma destas grandezas.

Com objetivo de também simplificar os cálculos e utilizando o fator do tempo, utiliza espaços de tempos menores e realiza o enjanelamento, para obter as amplitudes destes momentos, como é realizado em FFT, ou seja, aplicar a transformada de Fourier em espaços de tempo curtos, o menor possível para que a redução da complexidade das amostras não interfiram no cálculo, aumentando o número de blocos (imagens) a ser produzidas.

Da mesma maneira que em FFT, utilizando de pequenas partes da amostra, como pode ser observado na Função 2.22 em que é quantificada através da função de enjanelamento w[n, t]:

$$w\,[n,t] \qquad \rightarrow \qquad w\,[(n-\tau)] \tag{2.22}$$

As funções de enjanelamento se deslocam em função de t para a sub-amostragem do sinal. Após uma série de interpretações, a função discreta resultante para uma

subamostragem da Transformada de Fourier de Tempo Curto é definida na Função 2.23:

$$x\left(m,k\right)=\frac{1}{N}\sum_{n=0}^{N-1}x\left[n\right]\cdot w\left[n-m\right]e^{-i\omega_0 kn} \tag{2.23}$$

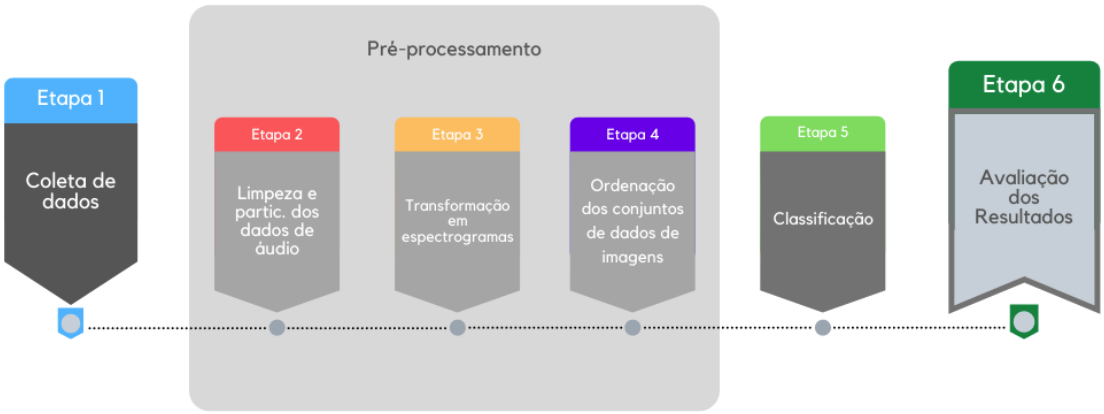
Em que uma sub-amostra do sinal x[n] é o produto definido na Função 2.24:

$$x\left[n\right]\cdot w\left[n-m\right] \tag{2.24}$$

3 Metodologia Proposta

Este capítulo apresenta a metodologia proposta, assim como os materiais utilizados para sua construção. A metodologia proposta é descrita resumidamente na Figura 24 e suas etapas descritas nas seções a seguir.

Figura 24 – Etapas da metodologia proposta.



Fonte: acervo do autor.

3.1 Coleta de dados

A base de dados consiste em um conjunto de dados de arquivos de áudio, nomeadas aqui pra frente de coletâneas, que variam de tamanho, obtidos através de um site na internet de compartilhamento de vídeos, o Youtube, em vídeos compilados de músicas de apresentações e gravadoras em formato de áudio .wav, reunidas e disponibilizadas neste site de compartilhamento de vídeo por estes grupos e por espectadores, sendo variada a qualidade de áudio deste em relação uns aos outros.

Estas coletâneas constituem o conjunto de dados de áudio inicial de toadas de Bumba-meu-boi, com durações que variam de 45 minutos a 1 hora e 30 minutos, organizadas em 5 sotaques diferentes, totalizando coletâneas de áudio bem pequenas em quantidade de arquivos, mas grande em volume de dados armazenados.

Cada sotaque tem uma quantidade diferente de coletâneas, organizada em 5 pastas específicas para cada estilo de sotaque de Bumba-meu-boi manifestado pelas brincadeiras culturais, sendo 6 coletâneas para sotaque de Baixada, 5 coletâneas para sotaque de costa de mão, 6 coletâneas para sotaque de matraca, 6 coletâneas para sotaque de orquestra e 6 coletâneas para sotaque de zabumba, organizadas em pastas diferentes para cada sotaque.

Estes foram adquiridos através de conteúdo de vídeo disponível no Youtube, em seus canais oficiais vídeo de cada grupo folclórico, como canais de fãs, admiradores e colecionadores de toadas de Bumba-meu-boi. Sem esta fonte, a aquisição seria um pouco mais complicada, já que alguns grupos ou sotaques não têm coletâneas realizadas em gravadoras, em formato digital ou em suas publicações na mídia por distribuidoras, que por muitas vezes são gravadas pelo público que compartilha por conta própria para outros poderem ter acesso. Sem esse conteúdo, alguns sotaques teriam uma quantidade de coletâneas defasada por indisponibilidade de acesso a esses conteúdos.

3.2 Pré-processamento

Para o processamento de classificação de imagens é preferível que se tenha uma grande quantidade de dados a serem processados. Redes Neurais Convolucionais tem a necessidade de grandes conjuntos de dados rotulados, assim, a tarefa de organizar os dados obtidos se reserva previamente ao momento de processamento.

3.2.1 Limpeza e particionamento dos dados de áudio

Como os dados obtidos, apesar de estarem bem organizados em seus respectivos estilos para classificação, são muito grandes e contém um grande número de dados de áudio individuais, toadas que podem ser processadas individualmente.

Tabela 1 – Coletâneas de áudio de Bumba-meu-boi para cada sotaque

Coletâneas de áudio de Bumba-meu-boi						
Sotaque	Baixada	Costa-de-mão	Matraca	Orquestra	Zabumba	**Total de arquivos**
Qtd. de arquivos	6	5	6	6	6	29

Visualizando a Tabela 1, observa-se a pequena quantidade de arquivos de áudio, não grande o suficiente para um bom modelo de processamento de imagens. Temos toadas completas, entretanto com tamanhos diferentes, que ao serem transformadas em arquivos de imagens, compreenderão tamanhos de espaços de tempo desiguais, além de conter grande quantidade de informações importantes que serão descartadas ou atrapalharão os sistemas no momento do processamento. Muitos desses arquivos de áudio contém momentos não interessantes para o objetivo, como pausas, ruídos e conversações descartáveis, entre outros, sendo necessária uma nova estratégia de reorganização de nossos dados.

Em produções musicais, uma sequência de notas musicais respeita um tempo, para que a combinação sonora desejada possa se dispersar pelo ar e transmitir as informações sonoras desejadas. Estas sequências compreendem como vibrações harmoniosamente arranjadas, que para o ouvido humano são traduzidas como uma melodia. Uma melodia geralmente respeita tempos específicos, 7 (sete) segundos e 50 (cinquenta) milésimos ou 15

(quinze) segundos. Com as toadas não é diferente. As melodias se organizam em conjuntos de 7 (sete) segundos e 50 (cinquenta) milésimos. Melodias completas em 15 (quinze) segundos em média, podendo ser divididas em um número muito maior de partes de áudio, que geram um número maior de arquivos de imagens para a entrada da arquitetura.

A escolha do momento de início e fim de cada toada também é muito importante na tarefa de limpeza, pois algo muito comum em todos os sotaques é a existência de uma conversação ou momentos não importantes para identificação dos sotaques já que geralmente estão presentes ao longo da toada. Logo no início pode existir uma chamada a um integrante do grupo (ou ao boi do grupo), brincadeira ou uma chacota dada ao boi do grupo folclórico (ou a outros grupos folclóricos), muito comum em toadas clássicas de grupos folclóricos de mesmo sotaque ou região, identificando uma disputa de espaços de apresentação, sendo assim removidos. Pode existir também um soar de apito ou zabumba iniciais e finais do tocar de instrumental, como uma chamada para os instrumentistas iniciarem ou finalizarem a melodia, sendo escolhido este último como critério para os momentos limites de cada toada.

Após esta limpeza, as toadas foram desagregadas das coletâneas e reorganizadas em novos conjuntos de dados, disponibilizando um número maior de arquivos de áudio em cada conjunto. Os novos arquivos correspondem a trechos de 45 segundos de áudio extraídos de cada toada. Este processo realizado manualmente, por existir a possibilidade de novos arquivos de áudio serem removidos, tanto para 15 segundos ou 7 (sete) segundo e 50 (cinquenta) centésimos de segundo cada. Com esta técnica obtivemos 474 novos arquivos de áudio de 28 coletâneas iniciais.

Por motivos de aproveitamento máximo das toadas obtidas por estas estarem organizadas em sequências de melodias completas de 15 (quinze) segundos e a possibilidade de disponibilizar a quantidade maior possível de arquivos de imagens, estas foram divididas em vários arquivos de áudio de duração máxima de 7 (sete) segundos e 50 (cinquenta) centésimos.

Assim, os arquivos de áudio finais comportam 7 (segundos) e 50 (cinquenta) centésimos de segundo de instrumental com vocais sobrepondo ou apenas o instrumental do grupo folclórico.

No total a base de dados contém arquivos de áudio, dividida em 5 sotaques (classes) de Bumba-meu-boi, como descrito na Tabela 2.

Tabela 2 – Base de dados de áudio de Bumba-meu-boi após o particionamento

Base de dados de áudio de Bumba-meu-boi - FINAL						
Sotaque	Baixada	Costa-de-mão	Matraca	Orquestra	Zabumba	**Total de arquivos**
Qtd. de arquivos	2.160	1.392	2.586	2.682	1.644	10.464

Estes arquivos resumem, por fim, em quantidade não igual de dados disponíveis entre todos os conjuntos de arquivos de áudio de sotaques de Bumba-meu-boi para o processamento.

3.2.2 Ordenação dos conjuntos de dados de imagens

Logo após realizar a separação dos arquivos de áudio, estes devem ser organizados de modo a facilitar a leitura e o processamento, então neste momento estes arquivos são separados em 3 (três) conjuntos de dados, treinamento, validação e teste, sendo organizados da seguinte maneira:

- São lidos todos os arquivos de um determinado sotaque e armazenados seus nomes em uma lista;
- Os nomes contidos nesta lista são embaralhados aleatoriamente, a fim de que os arquivos de imagens não possam ser lidos de maneira sequencial, garantindo que todos os trechos pertencentes de cada música sejam reunidos em apenas 1 único conjunto;
- É realizada a primeira leitura e escrita de movimentação para os dois diretórios correspondentes, treinamento (com nome "training") e validação (com nome "validation"), com um particionamento de 70% para o conjunto de treinamento e 30% para o conjunto de validação;
- Então é realizada a segunda leitura e escrita de movimentação, agora para o último diretório, o de teste (com nome "test"), com um particionamento de 33,333% para o conjunto de teste, restando 66,666% para o conjunto de validação.

Assim, ao finalizar o particionamento, é verificado que os 3 conjuntos têm tamanhos totais de 70% de dados para treinamento, 20% de dados para validação e 10% de dados para teste, suficiente para o processamento e teste posteriores.

Desta maneira, os arquivos de áudio são organizados de acordo com sua pasta (com o mesmo nome do sotaque correspondente), alimentando uma grande tabela de informações sobre cada arquivo, com seu nome (nome de arquivo sem a sua extensão), classe (sotaque) e a que conjunto de dados de processamento esta pertence, treino (training), validação (validation) ou teste (test).

Neste momento os arquivos de áudio são transformados em arquivos de imagens de espectrogramas digitais na forma de matrizes de dados, normalizadas as suas dimensões no padrão de 323 pixels de comprimento e 128 pixels de altura, já que há possibilidade de algumas imagens terem sido criadas em tamanhos diferentes (este momento é melhor discutido na subseção 3.2.3). Este processo é realizado para cada uma das imagens, que

são salvas em um objeto de dados tipo lista para cada um dos conjuntos de dados, que por sua vez são salvos em uma lista indexada por palavra contendo como índice o nome do conjunto de dados (training, validation ou test). Para o processo de treino, como existe uma grande quantidade de dados, este é dividido em 3 (três) listas de dados. Concomitante a esse processo, é alimentada uma outra lista para cada conjunto de dados, sendo recebido o nome de cada arquivo (sem extensão) e o seu nome de sotaque (classe) correspondente, esta é salva em outra lista indexada por palavra, semelhante a lista de imagens, contendo como índice o nome do conjunto de dados (training, validation ou test).

As listas de validação e teste são salvas em dois arquivos de dados diferentes no sistema de arquivos, "test_arr.npz" para os dados de teste e "valid_arr.npz" para os dados de validação. As listas de treinamento são salvas em "train1_arr.npz", "train2_arr.npz" e "train3_arr.npz".

As 3 (três) listas de treinamento geradas anteriormente são concatenadas, realizando uma conversão destes dados em graus de decibéis para potência, em escala numérica com referência para valores em 10, repetindo esta conversão para o conjunto de validação, que será utilizado durante o treinamento. Então é realizada uma permutação de modo aleatório entre estes dentro de cada conjunto de dados (somente treinamento e validação), com intuito de contribuir com o treino e impedir uma leitura de dados sequenciais, caso sejam de mesma classe (sotaque). Estes são reunidos em um segundo arquivo, dados de imagens e informações de imagens e classes (labels), sendo realizado para cada conjunto de dados (treinamento e validação), com nomes "shuffled_train.npz" para treinamento e "shuffled_valid.npz" para validação.

3.2.3 Transformação em espectrogramas

Os espectrogramas são representações gráficas dos sinais, convertidos de dados de frequência, tempo e amplitude adquiridos de sinais digitais obtidos de arquivos de áudio com formatação de bits de frequência de dados. Neste trabalho utilizaremos a conversão de arquivos de áudio em arquivos de imagens mel-espectrogramas, desenvolvidos a partir da conversão através da função de Transformação Rápida de Fourier.

Em dado momento de aquisição de dados para a construção dos conjuntos de dados, os arquivos de áudio são lidos para uma série temporal de áudio e convertidos em mel-espectrogramas, através da chamada das funções "load" e "melspectrogram" das bibliotecas auxiliares "librosa"[1] recebendo a série temporal de áudio e a taxa de amostragem da primeira, e o mel-espectrograma da segunda função.

Inicialmente, o arquivo de áudio necessita ser carregado em uma estrutura de dados para então ser computado. Esta tarefa é realizada pela função "load", recebendo como

[1] Librosa (raiz) e "feature" da biblioteca librosa, na sua versão 0.8.1, em Python, em sua versão 3.8

parâmetros, o nome do arquivo de áudio a ser lido e a sua duração total, e retornando em "y" a série temporal de áudio em dados numéricos em formato de matriz de dados e em "sr" o dado numérico da taxa de amostragem obtida na série temporal gerada a partir do áudio recebido.

Com os dados de áudio e a taxa de amostragem geral, é possível ser gerada um gráfico de espectrograma utilizando a Transformada de Fourier, entretanto, por motivos de desempenho e alocação de recursos, a Transformada de Fourier de Curto Tempo é mais eficiente, retornando resultados semelhantes. Assim, a função "melspectrogram" recebe a série temporal do áudio gerado, a taxa de amostragem da série temporal, o comprimento da janela a ser gerada e o número de amostras entre quadros sucessivos a ser gerada, a qualidade da janela de gráfico a ser gerada, quanto maior, mais detalhes estarão contidos na imagem. É retornada a imagem espectrograma em escala de potência em uma matriz de dados.

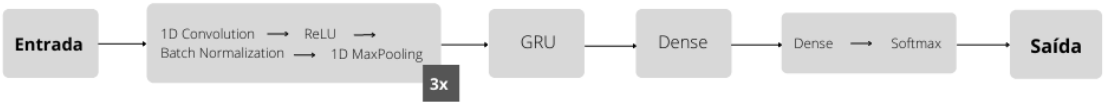
Após esta etapa, os dados recebidos do espectrograma de potência são convertidos em unidades de decibéis, através da função "power_to_db", Equação 3.1, da biblioteca librosa. Esta recebe a matriz de dados do espectrograma em escala de potência e a referência para o limite máximo de dados a ser transformados, neste caso o maior valor recebido, por fim, retornando o espectrograma em unidade de decibéis após realizar uma transformação numérica logarítmica estável, permitindo que os dados do espectrograma estejam visíveis a interpretação numérica e gráfica:

$$p(Spect) = 10 \cdot \log 10 \left(\frac{Spect}{valor\_ref} \right) \tag{3.1}$$

3.3 Classificação

Foi tomada inspiração para a construção de duas arquiteturas, uma CRNN (Convolucional Recurrent Neural Network ou Rede Neural Convolucional Recorrente) e uma PCRNN (Parallel Convolucional Recurrent Neural Network ou Rede Neural Convolucional Recorrente Paralela).

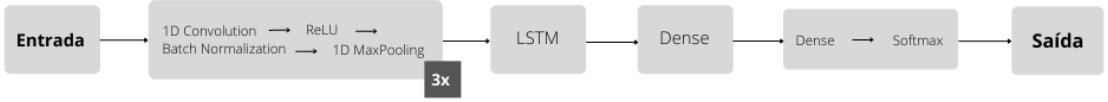
Figura 25 – O mapa da arquitetura CRNN inspirado no trabalho de (CHOI et al., 2016)



Fonte: acervo do autor.

O primeiro, baseado nos trabalhos de Choi et al. (2016), chamado CRNN, aplicando uma arquitetura que utiliza uma CNN na entrada, com sua saída uma RNN - GRU para classificação gêneros musicais (como introduzido na seção 1.2), e como apresentado na Figura 25.

Figura 26 – O mapa da arquitetura CRNN modificado para esta aplicação com uso de LSTM como RNN.

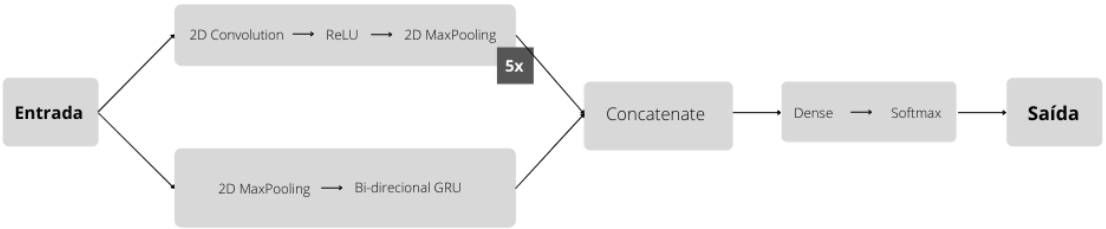


Fonte: acervo do autor.

A implementação neste trabalho, se baseando na arquitetura para desenvolvimento do modelo CRNN (CHOI et al., 2016), foi modificada a camada RNN, originalmente uma GRU para utilizar uma LSTM, como observado nas Figura 25 e Figura 26, para tratar o problema das RNNs de vanish gradient implementando uma estratégia de curto-longo prazo com maior controle, bem interessante para a grande quantidade de conteúdo esparso que está a ser trabalhado ao receber os dados das camadas de convolução.

O segundo, baseado no trabalho de Feng, Liu e Yao (2017), chamado PCRNN, aplicando uma arquitetura de sistemas CNN e RNN, mas aqui estas duas em paralelo uma com a outra, utilizando uma camada adicional ao fim destas para concatenação, como observada na Figura 27. Esta diferença é pertinente, pois as funções de tratamento de dados ao longo do processamento, realizando as relações de características a longos espaços de tempo dos sistemas RNN separadas das etapas de extração de características da CNN, resultando em informações não totalmente conectadas.

Figura 27 – O mapa da arquitetura PCRNN inspirado no trabalho de (FENG; LIU; YAO, 2017) com processamento em paralelo



Fonte: acervo do autor.

As duas arquiteturas do modelo gerado por este trabalho utilizam estruturas de redes neurais CNN para extração de características das imagens obtidas, ligada a estruturas de redes neurais RNN tipo GRU ou LSTM para conectar as estruturas de curto-longo prazo reconhecendo os padrões das sequências obtidas da camada anterior.

Na CRNN (CHOI et al., 2016), se faz como uma sequência de processamento ao longo das camadas, a entrada da arquitetura recebe as imagens e labels (rótulos), passando os mesmos a uma CNN 1D de 1(uma) única dimensão, realizando o processamento em função do tempo, que recebe as imagens e os features e extrai os seus recursos, construindo os vetores de features da convolução, seguida de uma camada de Função de Ativação ReLu para evitar a saturação do gradiente. Logo em seguida, é adicionada um Batch Normalization para auxiliar no decrescimento de pesos para a convergência na saída da ReLu. Em seguida, uma camada de Pooling para resumir o mapa de características a uma versão em resolução mais baixa após o ajuste de não-linearidade, para acelerar o processamento de recursos realmente necessários. Essas séries de sequências de camada de convolução e tratamento se repetem em 3 (três) vezes antes da sua saída, construída uma pilha de convolução que servem para que a extração de recursos obtenha o máximo possível de características das formas, já que, uma série de pilhas de convolução na entrada melhora a performance do modelo para uma melhor interpretação de linhas, e sendo os gráficos das imagens de espectrograma da base de dados que utilizamos aqui serem compostas essencialmente por linhas.

A camada seguinte é de uma RNN que recebe em sua entrada um input de tamanho de 96 unidades, contendo o vetor de features das camadas anteriores, realizando o reconhecimento das sequências de imagens através de persistência a longo prazo com baixa perda das dependências reconhecidas anteriormente, permitindo que padrões e diferenças sejam identificadas e devidamente separadas em classes determinadas. A saída da LSTM leva a uma camada Dense que recebe 64 unidades. A saída do modelo é uma Dense com a Função de Ativação Softmax, para realizar a ativação de uma função não-linear para gerar dados satisfatórios da probabilidade das 5 classes, e 5 unidades ocultas, para as 5 classes que se deseja prever. Ao fim desta camada, é inserido um Dropout.

O Dropout e L2 Regularization é utilizado na saída de toda camada de Redes Neurais com o objetivo de reduzir o ajuste excessivo, diminuindo a probabilidade de gerar um modelo excessivamente ajustado.

Como otimizador foi utilizado o Adam, com taxa de aprendizado em 0,001, com modelo compilado para realizar uma redução de aprendizagem por taxa de perda utilizando Entropia Cruzada Categórica (Categorical Cross Entropy) e utilizando como métrica para visualização a acurácia, não há uma necessidade real de adicionar mais métricas, a acurácia já basta durante o treinamento.

Na PRCNN (FENG; LIU; YAO, 2017), os blocos convolucionais e recorrentes estão em paralelo com apenas 1 bloco processando as informações temporais, e o outro processando as definições de características mais pertinentes existentes, sendo estas duas tarefas integradas ao fim destes dois através de uma concatenação. O bloco da CNN consiste em uma camada de convolução 2D realizando o processamento tanto em função do tempo quanto em função da frequência. recebendo as imagens e os features e extraindo os recursos e construindo vetores de features da convolução. A camada segue para uma Função de Ativação RELU e um MaxPooling de 2 (dimensões), para suportar a saída da convolução, resumindo os mapas de características em resoluções mais baixas, ainda suportando os detalhes necessários para a continuidade e acelerando o processamento. Estas sequências de camadas são repetidas 5 vezes antes de realizar a saída do bloco convolucional, obtendo o máximo possível de informações das imagens recebidas, com saída de tamanho de forma None, 256.

Em paralelo a este processamento está o bloco recorrente, recebendo a entrada com uma redução do frame por um MaxPooling de 2 (duas) dimensões, para que sua entrada tenha as mesmas características de dimensões do bloco anterior na primeira etapa convolucional com o objetivo de acelerar o processamento. Logo após este, segue os dados para uma camada convolucional de GRU Bi-direcional, com 64 unidades na entrada, retornando na saída um tamanho de forma Nona, 128. A concatenação da saída dos blocos de camadas CNN e RNN ocorre com tamanho de forma de None, 384, unindo as informações destas, seguindo para uma saída com uma camada Dense com Função de Ativação Softmax, para realizar a ativação com saída não-linear com geração de probabilidades de cada imagem para cada uma das 5 classes

É utilizado o otimizador Adam, com taxa de aprendizado em 0,001, também utilizando uma redução de aprendizagem por taxa de acurácia utilizando Entropia Cruzada Categórica (Categorical Cross Entropy) e utilizando como métrica para visualização a acurácia.

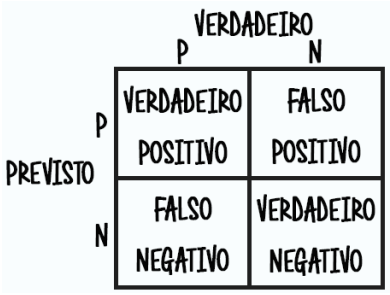
O modelo CRNN foi treinado em 70 épocas necessitando aproximadamente 20 minutos para processamento, já o modelo PCRNN utilizou apenas 50 épocas e aproximadamente 40 minutos para completar o processamento do treino, utilizando uma máquina dedicada e o sistema do Jupyter Notebook, para os dois processamento dos dois modelos. Ocasionalmente, a taxa de aprendizagem é decrementada caso a acurácia não obtiver um melhor valor em 10 épocas consecutivas, com delta mínimo de 0,01 e fator 0,5. Em adição, os pesos de treinamento são salvos a cada melhor resultado de acurácia durante o treinamento, para uso posterior.

3.4 Avaliação

O treinamento por si só pode não apresentar consistência sobre a veracidade dos dados obtidos, pois apesar de se apresentarem com valores de validação, estes veem repetidamente os mesmos dados, não entrando em contato com valores não vistos, fora de seus conjuntos de treinamento, sendo assim necessário realizar tais testes para verificar tais consistências nos resultados. Algumas funções são aplicações práticas de teste de modelos de treinamento de Redes Neurais Artificiais, são alguns destes:

- evaluate - realiza operações de avaliação no modo teste sobre os parâmetros obtidos no modelo durante o treinamento, retornando a perda e acurácia em uma lista de valores.
- confusion_matrix - função da biblioteca auxiliar sklearn.metrics, realiza uma avaliação no modo teste sobre os resultados obtidos por uma previsão do modelo, obtendo os erros e acertos sobre estas previsões em relação aos resultados esperados, realizando operações de probabilidade para cada classe prevista e esperada.

Figura 28 – O mapa gerado pela matriz de confusão com a relação entre a predição e o resultado esperado



Fonte: (DIAS, 2020)

- accuracy_score - função da biblioteca auxiliar sklearn.metrics, Função 3.2, realiza uma avaliação no modo teste sobre os resultados obtidos por uma previsão do modelo, retornando um valor real de acurácia do modelo.

$$\text{accuracy\_score}(y, y') = \frac{1}{n_{amostras}} \cdot \sum_{i=0}^{n_{amostras}-1} 1 \cdot (y' = y_i) \tag{3.2}$$

Estas são as funções utilizadas neste trabalho a fim de verificar os resultados obtidos.

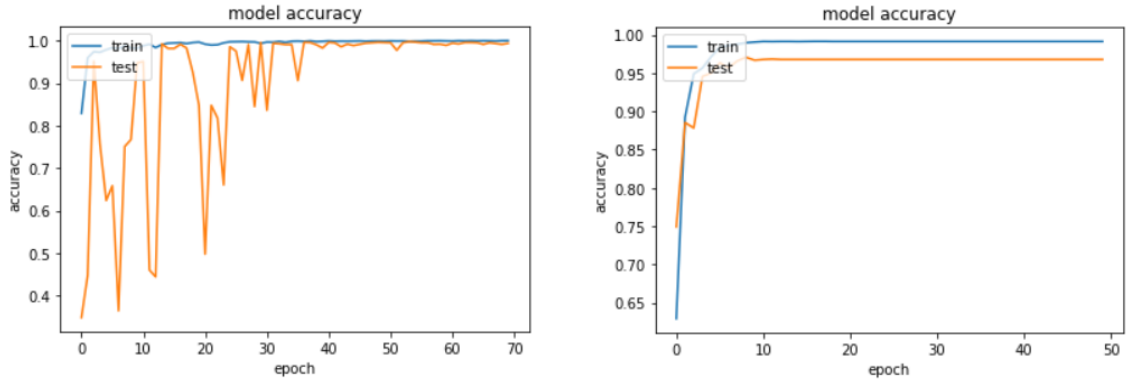
4 Resultados

Após os treinamentos, os modelos obtidos foram avaliados utilizando o conjunto de teste da base de dados, que corresponde a 10% (dez por cento) do conjunto de dados total. Através de métricas de obtenção de acurácia, evaluate e matriz de confusão podemos analisar valores de acerto e erro durante o teste e alguns gráficos de desempenho de treino final e teste final, para realizar uma comparação visual do desempenho ao longo do treinamento e desempenho pós treino.

Os gráficos de desempenho podem auxiliar na busca por um modelo mais robusto para a resolução do problema, até mesmo aqueles que apenas medem acurácia e perda, o que utilizamos aqui, já que pode ser facilmente observado se este contém algum ajuste excessivo. Taxas altas de acurácia com baixas perdas durante treino seguida de taxas baixas de acurácia e altas perdas ao longo de várias épocas indica uma clara possibilidade do modelo está entrando em ajuste excessivo, um overfitting. O ideal é a validação durante o treino apresentar uma acurácia próxima a acurácia de dados de treino.

Como observado nos resultados obtidos, Figuras 29, 30 e ainda na Tabela 3, o modelo CRNN obteve 0.0305 de valor para perda e 1.0 de valor para a acurácia, semelhante aos valores obtidos na acurácia durante o teste de treinamento (validação). Já o modelo PCRNN obteve 0.0281 de valor para perda e 0.99 de valor para a acurácia, recebendo no teste valor de acurácia em 0.97, aproximado aos val ores obtidos na acurácia durante o teste de treinamento (validação).

Figura 29 – Gráficos da acurácia para os modelos (a) CRNN e (b) PCRNN

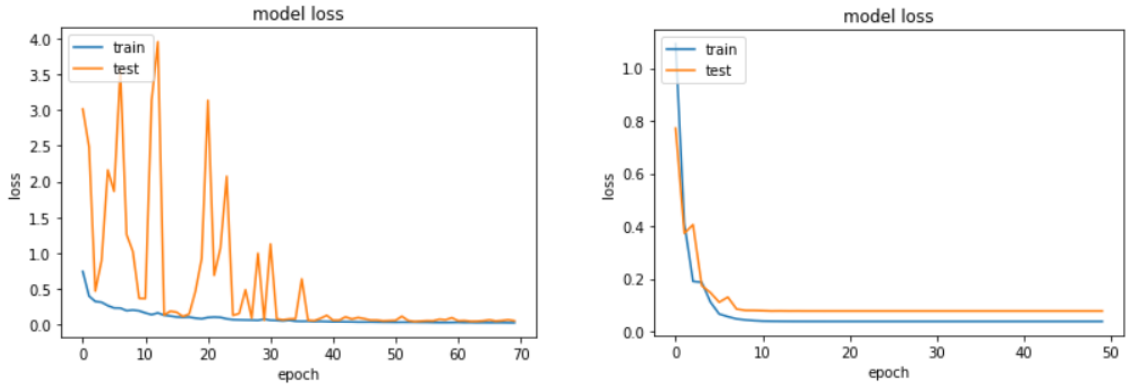


(a) (b)

Fonte: acervo do autor.

Os modelos desenvolvidos neste trabalho alcançaram resultados, com taxa de acerto acima de 95% (noventa e cinco por cento), e baixa perda, comprovada através dos dados obtidos nas Tabelas 4 e 5. Novamente os modelos, CRNN e PCRNN, apresentam resultados

Figura 30 – Gráficos da perda para os modelos (a) CRNN e (b) PCRNN



(*a*) (*b*)

Fonte: acervo do autor.

Tabela 3 – Acurácia na validação e teste para CRNN e PCRNN

	CRNN		PCRNN	
	Validação	Teste	Validação	Teste
Acurácia	99%	100%	97%	99%
Perda	<0,001%	3,05%	<0,001%	2,81%

consistentes em relação observado anteriormente, acertando as previsões sobre as classes, assim como uma acurácia alta o suficiente que condizem com tais resultados.

Tabela 4 – Resultado do teste do modelo CRNN

Sotaques	Precisão	Recall	F1-score	Qtd. base
baixada	1.00	1.00	1.00	192
costa de mao	1.00	1.00	1.00	114
matraca	1.00	1.00	1.00	216
orquestra	1.00	1.00	1.00	210
zabumba	1.00	1.00	1.00	210
Acurácia			1.00	942
Média Macro	1.00	1.00	1.00	942
Média Ponderada	1.00	1.00	1.00	942

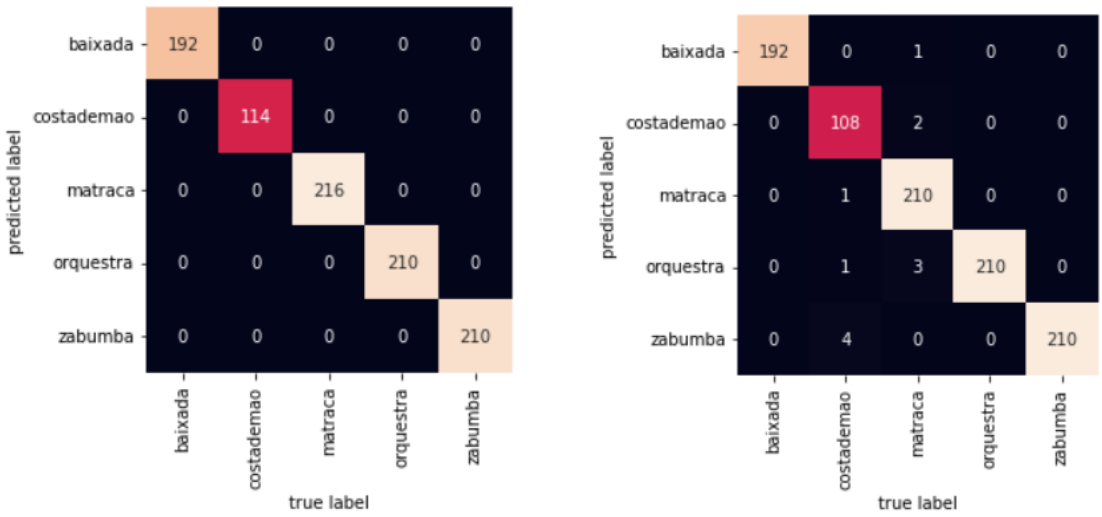
O resultado apresentado na Figura 31, apresenta os acertos sobre cada classe em uma matriz de confusão, notando-se que os dados mais ao centro sobre a linha diagonal decrescente mostra a quantidade de resultados de previsões corretamente declaradas para cada classe.

Umas das características presentes aqui que pode ser destacada para o alcance de tais resultados seriam as variadas escolhas de técnicas para a construção desta arquitetura, como para a base de dados, seu pré-processamento e classificação.

Tabela 5 – Resultado do teste do modelo PCRNN

Sotaques	Precisão	Recall	F1-score	Qtd. base
baixada	0.99	1.00	1.00	192
costa de mao	0.98	0.95	0.96	114
matraca	1.00	0.97	0.98	216
orquestra	0.98	1.00	0.99	210
zabumba	0.98	1.00	0.99	210
Acurácia			0.99	942
Média Macro	0.99	0.98	0.99	942
Média Ponderada	0.99	0.99	0.99	942

Figura 31 – Matriz de confusão resultante dos modelos (a) CRNN e (b) PCRNN



(*a*) (*b*)

Fonte: acervo do autor.

Outro fator importante, é a utilização de CNNs para o processamento da extração de características das imagens da base de dados, esta se demonstra em grande valia, já que as imagens, por conter uma grande quantidade de informações em regiões não sequenciais discrepante ao uso de técnicas tradicionais que processam os dados sequencialmente. O processamento com uso deste sistema auxilia na identificação de linhas, formas e texturas, se tornando um fator importante para o alcance de resultados favoráveis.

A aplicação de RNNs no processamento de imagens auxilia a classificação, guardando estados do processamento ao longo do tempo, permitindo que determinadas informações de camadas anteriores sejam acessadas em camadas futuras, assim gerando relações mais confiáveis entre os dados, resultando em um poder de classificação mais robusto, indicado quando se está trabalhando em grandes conjuntos de dados que contém características semelhantes em modo sequencial, como em determinado ponto neste trabalho.

Um outro fator que pode ser destacado aqui seria a construção de uma grande base de dados a partir da base de dados inicialmente coletada, por motivos de que a última continha poucos dados individualmente mas com uma grande quantidade de volume, o que consequentemente acarretaria problemas em extratores de características e em classificadores. Assim, estes foram particionado em diversos trechos menores, aumentando a quantidade de dados da base, permitindo que as estruturas CNN pudessem gerar relações mais consistentes a partir de suas extrações, requisito inicial para uso desta.

Apesar dos bons resultados obtidos, houveram diferenças não tão significativas para com os resultados dos modelos, apresentada no resultado da arquitetura PCRNN, por motivo de suas duas estruturas CNN e RNN em paralelo, com a estrutura CNN perdendo a capacidade de realizar conexões de informações de suas estruturas a longo prazo, característica obtida através do processamento em sequência a camadas RNNs. Os dados foram apenas concatenados ao fim das duas estruturas, não havendo possibilidade das características entre os dados pudessem ser "conectadas" corretamente, justificando a perda de acertos em sotaques de caráter bastante ruidoso, como o sotaque de matraca e orquestra, por utilizar em grande quantidade instrumentos de percussão, seguida de ritmo e tons melódicos mais complexos, assim, em dados momentos muito semelhantes.

4.1 Comparação com Trabalhos Relacionados

Algo claro a se destacar são as características possíveis a serem obtidas através do processamento, sejam em espectrogramas ou diretamente de sinais de áudio, compreendendo que sua complexidade em conteúdo o torna uma fonte valiosa para extração de características, além das utilizadas aqui (frequência , tempo, amplitude e decibéis, por exemplo).

Entretanto, quando estamos a comparar modelos de processamento de áudio para classificação de gêneros entraves podem aparecer, não conter na literatura algum trabalho de conteúdo aproximado, assim, estes que foram separados, por apresentar gêneros latino-americanos, e se aproximar ao máximo dos gêneros aqui trabalhados.

Assim, este tópico tem por objetivo a comparação dos trabalhos, modelos apresentados em cada um destes e apresentar suas diferenças, que por sinal, neste temos um conjunto produzido a mão e não disponível como os outros. Os modelos desenvolvidos neste trabalho recebem o nome de BumbaNet - CRNN e BumbaNet - PCRNN, para os modelos de CRNN e PCRNN, respectivamente.

Como observado nas Tabelas 6, 7 e 8, os resultados obtidos neste trabalho são superiores, em comparação aos outros trabalhos relacionados aqui, se destacando por sua ótima performance na classificação dos sotaques de Bumba-meu-boi.

Este fato se dá por motivos de que os outros trabalhos, nesta comparação, realizam classificação sobre músicas utilizando SVM, Naive Bayes e MLP, exceto Senac et al. (2017), diferentemente neste trabalho, em que são utilizadas 2 (duas) arquiteturas a partir de CNN e RNN, que juntas, apresentam um bom histórico de resultados em classificação de imagens.

Tabela 6 – Acurácia geral de classificação entre trabalhos estudando espectrogramas particionados e sinais de áudio

Acurácia geral de classificação entre os trabalhos (%)						
Modelo	BumbaNet - CRNN	BumbaNet - PCRNN	n_STFT	n_MÚSICA	FUSION1	FUSION2
Acurácia	100	99	87.8 ± 1.8	89.6 ± 2.4	90.5 ± 0.7	91 ± 1.2

No trabalho de Senac et al. (2017) é utilizado CNN para processamento de imagens para extração de características, com camadas adicionais de ReLU e Softmax para ativar a saída e gerar as probabilidades sobre as classes. Esta técnica demonstra grande eficiência em comparação com os outros modelos, por conter uma maior quantidade de dados extraídos das imagens no momento de classificação. Além disso, aplica 4 tipos de classificação, utilizando imagens obtidas a partir da Transformada de Fourier de Curto Tempo, dados extraídos e processados diretamente dos sinais de áudio, e 2 modelos de classificação desenvolvidos pelos autores, com resultados de acurácia apresentados na Tabela 6

Em contrapartida, o trabalho de Costa et al. (2011), que apresenta o uso de SVM, garantem a geração de classificações com particionamento dos dados de áudio em 3 segmentos, um inicial, no meio, e no final, em cada áudio da base de dados, assim, o processamento é acelerado, por não gerar vetores exorbitantemente grandes, ao mesmo tempo que aumenta o volume de dados para utilização na classificação, expandindo a base de dados ao mesmo tempo que não sobrecarrega o modelo utilizando vetores de tamanho reduzido. Resultados apresentados na Tabela 7.

Tabela 7 – Acurácia geral de classificação entre trabalhos estudando espectrogramas particionados

Acurácia geral de classificação entre os trabalhos (%)					
Modelo	BumbaNet - CRNN	BumbaNet - PCRNN	AvgRR	AvgRR - SUM	AvgRR - MAX
Acurácia	100	99	60.1	67.2	65.7

Em adição, no trabalho de Costa et al. (2011), aplica a transformação de dados de áudio em espectrogramas, dividindo horizontalmente cada janela de imagem em 10 partes. A estratégia está na classificação de cada uma destas zonas, realizando a taxa média de recognição entre estas, retornando a probabilidade sobre cada classe. Também aplica a técnica de soma de probabilidades de cada classe em cada zona, a classe de maior valor é definida como a classe da imagem, repetindo este mesmo processo e aplicando a máxima das probabilidades, como discutido em (KITTLER et al., 1998). Para o índice de Taxa

Média de Recognição será utilizada AvgRR, para soma AvgRR - SUM e para a máxima AvgRR - MAX, apresentada na Tabela 8.

Tabela 8 – Acurácia geral de classificação entre trabalhos estudando sinais de áudio

Acurácia geral de classificação entre os trabalhos (%)					
Modelo	BumbaNet - CRNN	BumbaNet - PCRNN	SVM	NaiveBayes	MLP
Acurácia	100	99	65.06	46.03	59.43

No trabalho de Jr., Kaestner e Koerich (2007) aplica como classificadores uma rede MLP, um SVM e um Naive Bayes, para aplicando sobre sinais de áudio extraídos através de diversos momentos das músicas da base de dado. Aqui será indexado por SVM para aplicação com classificador SVM, Naive Bayes para classificação com Naive Bayes e MLP para classificação com MLP.

5 Conclusão

Este trabalho teve como objetivo produzir um modelo de classificação para sotaques de Bumba-meu-boi, produzido a partir da metodologia proposta em uma arquitetura de Redes Neurais Artificiais. Este modelo foi produzido em uma base de dados de áudio desenvolvida neste trabalho a partir de toadas adquiridas em fontes disponíveis na internet pelos grupos folclóricos e por espectadores.

A metodologia empregada neste trabalho foi executada por meio de uma base de dados de áudio formada por toadas de diversos sotaques de Bumba-meu-boi. Foi desenvolvida uma base de dados de áudio para realizar este trabalho, adquiridos na internet e formatados manualmente. Logo no início da coleta, foi percebido o desafio de tratar os dados de áudio, sendo necessária uma limpeza para aquisição apenas dos momentos de conteúdo mais importantes. Um pré-processamento foi realizado com o intuito de adquirir apenas os trechos do instrumental, uma das características principais que definem os sotaques de Bumba-meu-boi. A proposta deste trabalho, como visto anteriormente na seção 3, de realizar uma classificação de sotaques de Bumba-meu-boi utilizando imagens, foi possível através do uso de bibliotecas auxiliares que implementam a Transformada de Fourier sobre sinais sonoros salvos em arquivos áudio digitais, gerando gráficos em função da frequência pelo tempo e amplitude sonora que podem ser processados como quaisquer outros arquivos de imagens. Outra necessidade observada é que, para obter um melhor resultado utilizando Redes Neurais Convolucionais, deve-se atentar para o uso de uma base de dados robusta e com seus dados rotulados, sendo necessário reestruturar a base já coletada, particionando as toadas em trechos menores de áudio de 7 (sete) segundo e 50 (cinquenta) centésimos de segundo.

Contudo, mesmo com os desafios encontrados, os resultados obtidos ao longo do treinamento demonstravam que o modelo estava lidando muito bem com os dados reestruturados.

Este trabalho alcançou as expectativas iniciais, obtendo resultados acima do esperado nos dois modelos desenvolvidos, CRNN e PCRNN, com resultados acima de 95% (noventa e cinco por cento) de acurácia, se entende que em situações de aplicação real conseguiria classificar os áudios recebidos com baixa chance de erro.

Para trabalhos futuros, pretende-se realizar o desenvolvimento de uma aplicação, para aparelhos celulares ou computadores, para que possa reconhecer o sotaque de Bumba-meu-boi, recebendo um trecho de áudio através de um microfone conectado ou de um arquivo da máquina, retornando em tela o sotaque do arquivo de áudio recebido. Esta pretensão vai de encontro a acessibilidade devida a se dar a cultura local, de comunidades

quilombolas e indígenas, como gerar na sociedade um sentimento de pertencimento e presença de uma cultura que não se faz apenas de apresentações, mas é parte de nossa comunidade. Esta proposta idealiza a aproximação da cultura local, de festividades e cerimônias, com a tecnologia em constante evolução, levando a sociedade estes frutos de acessibilidade à informação.

Referências

ABREU, G. *Arraial Pertinho tem vasta programação com forró e bumba-meu-boi*. 2017. Disponível em: <http://www.genivaldoabreu.com.br/2017/06/arraial-pertinho-tem-vasta-programacao.html>. Citado na página 21.

ALBERNAZ, L. S. F. Dinâmicas do bumba meu boi maranhense: classificação em "sotaques" e participação do público. Revista Olhares Sociais, v. 2, n. 2, p. 3–24, 2013. Citado na página 21.

ALVES, G. *Entendendo Redes Convolucionais*. Medium, 2018. Disponível em: <https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184>. Citado na página 29.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, v. 5, n. 2, p. 157–166, 1994. Citado na página 32.

BOOK, D. L. *Capítulo 40 – Introdução as Redes Neurais Convolucionais*. Data Science Academy, 2021. Disponível em: <https://www.deeplearningbook.com.br/introducao-as-redes-neurais-convolucionais/>. Citado na página 29.

BOOK, D. L. *Capítulo 6 – O Perceptron – Parte 1*. Data Science Academy, 2021. Disponível em: <https://www.deeplearningbook.com.br/o-perceptron-parte-1/>. Citado na página 25.

BOTELHO, I. Dimensões da cultura e políticas públicas. *São Paulo em Perspectiva*, São Paulo, Brasil, v. 15, n. 2, p. 2, 2001. Citado na página 14.

CECCON, D. iaexpert.academy.com.br, 2020. Disponível em: <https://iaexpert.academy/2020/05/25/funcoes-de-ativacao-definicao-caracteristicas-e-quando-usar-cada-uma/>. Citado 3 vezes nas páginas 35, 36 e 37.

CHOI, K.; FAZEKAS, G.; SANDLER, M.; CHO, K. Convolutional recurrent neural networks for music classification. 2016. ISSN 1609.04243. Citado 7 vezes nas páginas 8, 16, 17, 18, 49, 50 e 51.

CINTRA, R. *MC3 - Introdução à Neurociência*. INPE, 2018. Disponível em: <http://www.inpe.br/elac2018/arquivos/ELAC2018_MC3_apostila.pdf>. Citado na página 27.

COSTA, Y. M. G.; OLIVEIRA, L. S.; KOERICB, A. L.; GOUYON, F. Music genre recognition using spectrograms. In: *2011 18th International Conference on Systems, Signals and Image Processing*. [S.l.: s.n.], 2011. p. 1–4. Citado 3 vezes nas páginas 17, 18 e 58.

DIAS, T. *Classificação com scikit-learn*. 2020. Disponível em: <https://dadosaocubo.com/classificacao-com-scikit-learn/>. Citado na página 53.

FENG, L.; LIU, S.; YAO, J. Music genre classification with paralleling recurrent convolutional neural network. 2017. ISSN 1712.08370. Citado 5 vezes nas páginas 8, 17, 18, 50 e 52.

FURLANETTO, B. H. O bumba-meu-boi do maranhÃo: TerritÓrio de encontros e representaÇÕes sociais. *Raega - O Espaço Geográfico em Análise*, v. 20, n. 0, 2010. ISSN 2177.2738. Disponível em: <https://revistas.ufpr.br/raega/article/view/20615>. Citado na página 20.

HEGEL; FISCHER, E. A necessidade da arte: Uma interpretação marxista. Zahar Editores, Rio de Janeiro, 1989. Citado na página 21.

HOCHREITER, S. *Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München*. 1991. Citado na página 32.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 32.

HUBEL, D. H.; WIESEL, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, v. 160, 1962. Citado na página 28.

JACQUES, M. *Introdução a Redes Neurais*. EDGE AI Guru, 2020. Disponível em: <https://edgeaiguru.com/Introdu%C3%A7%C3%A3o-a-Redes-Neurais>. Citado na página 28.

JR., C. N. S.; KAESTNER, C. A. A.; KOERICH, A. L. Automatic music genre classification using ensemble of classifiers. In: *2007 IEEE International Conference on Systems, Man and Cybernetics*. [S.l.: s.n.], 2007. p. 1687–1692. Citado 2 vezes nas páginas 18 e 59.

JUNIOR, J. R. F. *Redes Neurais Recorrentes — LSTM*. Medium, 2019. Disponível em: <https://medium.com/@web2ajax/redes-neurais-recorrentes-lstm-b90b720dc3f6>. Citado 3 vezes nas páginas 31, 32 e 33.

KITTLER, J.; HATEF, M.; DUIN, R. P.; MATAS, J. On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 20, n. 3, p. 226–239, 1998. Citado na página 58.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. Curran Associates Inc., Red Hook, NY, USA, p. 1097–1105, 2012. Citado na página 17.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, p. 2278 – 2324, 1998. ISSN 0018.9219. Citado na página 16.

LOPES, L.; ALVES, G.; MELO, M. Content evidence of a spectrographic analysis protocol. *Revista CEFAC*, v. 19, p. 510–528, 08 2017. Citado na página 39.

LOUREIRO, V. M. R. Música enquanto manifestação cultural e sua relação com o sagrado. In: *Música para ouvidos, fé para a alma, transformação para a vida: música de fé e construção de novas identidade na prisão - Dissertação (Mestrado em Serviço Social)*. [S.l.]: Faculdade de Serviço Social, Pontifícia Universidade Católica do Rio de Janeiro - PUC-RJ, 2009. p. 79–98. Citado na página 22.

MARQUES, F. E. de S. Mídia e experiência estética na cultura popular: o caso do bumba-meu-boi. Imprensa Universitária, São Luís, 1999. Citado na página 19.

MATSUMOTO, F. *Redes Neurais*. Medium, 2019. Disponível em: <https://medium.com/turing-talks/turing-talks-27-modelos-de-predi%C3%A7%C3%A3o-lstm-df85d87ad210>. Citado na página 33.

MELO, J. W. R. de. Multiculturalismo, diversidade e direitos humanos. *UFT Grupo de Trabalho – Educação e Direitos Humanos*, EDUCERE - XII Congresso Nacional de Educação, p. 1495 – 1510, 2015. ISSN 2176.1396. Citado na página 15.

MINSKY, M.; PAPERT, S. A. Perceptrons. *M.I.T. Press*, M.I.T., 1969. Citado na página 26.

MOREIRA, S. *Rede Neural Perceptron Adaline*. Medium, 2018. Disponível em: <https://medium.com/ensina-ai/rede-neural-perceptron-adaline-8f69dc419d4e>. Citado 2 vezes nas páginas 26 e 27.

PALHANO, M. L. L. *Cazumbás - Feiticeiros da Floresta - Bumba Boi Unidos de santa Fé*. 2013. Citado na página 23.

PANDA, J. *Lenda do Bumba meu boi e as festas juninas em São Luís/MA*. 2018. Disponível em: <https://www.indavoula.com.br/bumba-meu-boi-festa-junina-em-sao-luis/>. Citado na página 23.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, p. 533 – 536, Outubro 1986. ISSN 1476.4687. Disponível em: <https://doi.org/10.1038/323533a0>. Citado na página 26.

SENAC, C.; PELLEGRINI, T.; MOURET, F.; PINQUIER, J. Music feature maps with convolutional neural networks for music genre classification. In: *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*. New York, NY, USA: Association for Computing Machinery, 2017. (CBMI '17). ISBN 9781450353335. Disponível em: <https://doi.org/10.1145/3095713.3095733>. Citado 3 vezes nas páginas 17, 18 e 58.

SILVA, R. H. da; JúNIOR, N. K. Multiculturalismo, sociedades complexas e povos tradicionais: uma perspectiva interdisciplinar. *Rev. Humanidades*, Fortaleza, Brasil, v. 32, n. 2, p. 295–304, Julho - Dezembro 2017. Citado na página 15.

SIMONYAN, K.; ZISSERMAN, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. Citado na página 17.

SOUSA, P. M.; BOGéA, G. *Festejos juninos no Maranhão – a maior diversidade rítmica e visual do Brasil*. 2021. Disponível em: <https://jpturismo.com.br/festejos-juninos-no-maranhao-a-maior-diversidade-ritmica-e-visual-do-brasil/>. Citado na página 22.

TOMAZ, P. C. A preservação do patrimônio cultura e sua trajetória no brasil. *Fênix – Revista de História e Estudos Culturais*, v. 7, n. 2, 2017. ISSN 1807.6971. Citado na página 15.

WIDROW, B.; HOFF, M. E. Adaptive switching circuits. In: *1960 IRE WESCON Convention Record, Part 4*. New York: IRE, 1960. p. 96–104. Citado na página 26.

YINGGE, H.; ALI, I.; LEE, K.-Y. *Deep Neural Networks on Chip - A Survey*. 2020. 589-592 p. Disponível em: <https://www.researchgate.net/figure/Pooling-layer-operation-oproaches-1-Pooling-layers-For-the-function-of-decreasing-the_fig4_340812216>. Citado na página 38.