CENTRO UNIVERSITÁRIO DE BRASÍLIA- UniCEUB

PROGRAMA DE INICIAÇÃO CIENTÍFICA

PEDRO HENRIQUE RODIRGUES MENDES

GERAÇÃO DE MÚSICAS POLIFÔNICAS UTILIZANDO REDES NEURAIS ARTIFICAIS

BRASÍLIA

2020

PEDRO HENRIQUE RODIRGUES MENDES

GERAÇÃO DE MÚSICAS POLIFÔNICAS UTILIZANDO REDES NEURAIS ARTIFICAIS

Relatório final de pesquisa de Iniciação Científica apresentado à Assessoria de Pós-Graduação e Pesquisa.

Orientação: Msc. William R. Malvezzi

BRASÍLIA

2020

DEDICATÓRIA

Primeiramente minha família, que presenciaram todo esse turbulento processo de desenvolver uma pesquisa cientifica e que possuem o mérito de criar uma base forte e sólida para mim.

Para os grandes os artistas do gênero Bossa Nova, criaram um estilo musical unindo a harmonia sublime e sofisticada do jazz com a poesia e gingado do nosso querido Brasil, que por sua vez encantou e ainda encanta o mundo. O Destaque fica para o músico João Gilberto que morreu durante a preparação da proposta do PIC e assim pude conhecer e apreciar sua arte que irei de carregar por toda minha vida.

Por fim dedico aos meus companheiros da turma 2019 de formandos do Bacharelado de Engenharia da Computação, que muito me motivou e auxiliou na fase elaboração da proposta para o Programa de Iniciação Cientifica (PIC) do UniCeub.

AGRADECIMENTOS

Primeiramente agradeço a instituição UniCEUB por oferecer aos seus alunos esse programa de iniciação nos estudos científicos durante a graduação.

Segundamente agradeço a tutoria e orientação do MSC. William Roberto Malvezzi, ele acreditou na mesma intensidade quanto eu neste projeto e também pude ter a oportunidade de absorver parte de seu vasto conhecimento no campo que planejo seguir por toda a minha vida.

Agradeço a amizade e conselhos do MSC. Ivandro Ribeiro, me orientou na primeira tentativa de PIC e infelizmente não foi aprovada, mas que influenciou a aprovação e realização dessa, portanto, meus mais sinceros agradecimentos.

Por fim agradeço minha namorada Isadora Garcia Emilio, esteve no meu lado desde o início até o fim do projeto e nunca soltou a minha mão.

"Nossa responsabilidade é fazer o que podemos, aprender o que podemos, melhorar o que podemos e por fim passá-los adiante"

Richard P. Feynman (2015). "The Quotable Feynman", p.359, Princeton University Press

RESUMO

Música é um elemento importante e discreto na vida em sociedade e que possui alcance e relevância em âmbitos social, cultural e econômico na raça humana, está presente desde a pré-história até os dias atuais em uma relação quase simbiótica com a tecnologia, atualmente é possível consumir qualquer estilo ou artista de música em qualquer lugar com suporte das tecnologias de *streaming* e *smartphones*. A produção de materiais audiovisuais é considerada uma indústria bilionária, o mercado competitivo da música impulsiona a evolução de processos e ferramentas. Com os recentes avanços no campo da Inteligência Artificial, portanto, os sistemas inteligentes estão cada vez mais dominando com êxito atividades consideradas complexas, dentre eles estão os de visão computacional (VC) e também de processamento de linguagem natural (PLN). O estudo visa mesclar essas novas técnicas de inteligência computacional, como as Redes Neurais Artificiais e Aprendizado Profundo, com o campo da música com afim de inferir novas músicas do gênero brasileiro Bossa Nova a partir de um modelo preditivo. Para possibilitar essa síntese de músicas, será resgatado e aplicado conhecimento de estudos pregressos na geração de músicas afim selecionar o ideal no gênero proposto com base no método quali-quantitativo, avaliando métricas de treinamento dos modelos e a qualidade das amostras geradas em atributos referentes a música. É estimado que ao fim do projeto realize um enriquecimento cultural brasileiro ao envolver a bossa nova como foco, e também criar precedentes para existência de ferramentas de geração de músicas com o objetivo de auxiliar músicos em âmbito de inspiração em processos de composições autorais ou na criação ligeira de *samples* para materiais audiovisuais.

Palavras-Chave: Inteligência Artificial. Redes Neurais Artificiais (RNA). Aprendizado Profundo. Música. Bossa Nova.

LISTAS DE FIGURAS

1. **RESULTADOS E DISCUSSÕES**

2. **RESULTADOS E DISCUSSÕES**

LISTAS DE FLUXOGRAMAS

1. **METODÓLOGIA**

LISTAS DE TABELAS

2. RESULTADOS E DISCUSSÕES

LISTAS DE SÍMBOLOS E ABREVIAÇÕES

ANN - *Artificial Neural Networks*, no português Redes Neurais Artificiais.

CNN - *Convolutional Neural Network*, no português Redes Neurais Convolucionais.

RNN - *Recurrent Neural Net*, no português Redes Neurais Artificiais.

GANS - *Generative Adversial Neural Nets*, no português Redes Neurais Generativas Adversais.

LSTM – Long and Short Term Memory, no português Memória de Longo e Curto Prazo

ML - *Machine Learning*, no português Aprendizado de Máquina.

DP - *Deep learning*, no português Aprendizado Profundo.

W2V - *Word2Vac*.

PLN - Processamento de Linguagem Natural.

VC - Visão Computacional.

RMH - Redução Musical Harmônica.

SUMÁRIO

INTRODUÇÃO

Música é um elemento importante e discreto na vida em sociedade e que possui alcance e relevância em âmbitos social, cultural e econômico, pois é consumido inúmeros produtos audiovisuais durante o nosso dia a dia e de certa forma não é percebido o esforço laboral e de investimentos por de trás de uma simples música que está tocando em uma rádio, segundo o Relatório Final da Estratégia para Exportação de Música da União Europeia (2019) dentro do Setor de Cultura e Criatividade (CCS) da União Europeia, o setor de música é o terceiro maior empregador com 1.168,000 empregados diretos e gera uma receita anual de 25 Bilhões de Euros. As projeções atuais sobre o mercado da música indicam uma tendência de crescimento no futuro, portanto, com a capacidade de poder absolver novos agentes nesse mercado, conforme a Federação Internacional da Indústria Fonográfica IFPI (2018) após 15 anos de declínio no número de músicas gravadas no mundo, no ano de 2017 foi alcançado a terceira alta consecutiva e somado com o crescimento de 8,1% da Receita Global proveniente da música.

Na contramão desse movimento macro e multibilionário que envolve setores públicos e privados para a dominância do mercado musical, no momento de publicação desse artigo há o surgimento de novos e peculiares agentes de criação nesse mercado, são os artistas independentes que estão cada vez estão mais relevantes e também ditando novas tendências no meio da música. Segundo WALZER (2016), foi observado um aumento na autonomia e oportunidades para que artistas independentes possam produzir o próprio som com alta qualidades de forma autônoma, conseguindo gerar lucros modestos sem ter a presença de uma gravadora na produção e distribuição do conteúdo. Esse movimento de produção independente escora na democratização da internet, um novo canal livre e descentralizado para realização de propaganda e a disseminação de produtos musicais sem a necessidade imposição de vieses ou crivo por uma parte detentora do meio de comunicação tradicional, como as rádios. Segundo FIGUEREDO e ARAÚJO (2019), a produção autoral independente dá liberdade para que os músicos construam a suas próprias identidades musicais e abracem as influências que são realmente quistas pelo

artista, dentro de uma zona de conforto para decisões e com contato direto e em tempo real com consumidores via redes sociais.

O assunto inteligência artificial está em voga na atualidade, criando e revolucionando novos mercados, segundo a empresa PwC (2018), a maior empresa do mundo no fornecimento de serviços profissionais, o PIB mundial será 14% maior em 2030 com os resultados provenientes da utilização de inteligência artificial em variados processos, e esse tal interesse nessa também transpassa para o mundo acadêmico, segundo SHOHAM et al (2018), o crescimento de artigos publicados anualmente continua acima do número normalmente publicado em ciência de computação, sugerindo que esse crescimento da Inteligência Artificial está em uma direção mais ampla que apenas ser um interesse especifico da área do conhecimento ciência de computação. Destarte, soluções com aprendizado profundo podem simplificar o ato de fazer melodia, reduzindo os altos investimentos e eliminando o pré-requisito de possuir experiência ou treinamento para realizar composições, gerando, assim, uma democratização da elaboração de trilhas sonoras, músicas e efeitos sonoros para produções emergentes de conteúdos audiovisuais como filmes, vídeos e jogos digitais.

A síntese de música de forma automática mostra-se ao primeiro momento desafiador, principalmente ao modo que será modelado um agente inteligente que terá como objetivo dominar aspectos como ritmo, harmonia e melodia para enfim concluir em uma música, Segundo FÉRNANDEZ e VICO (2013), é possível delinear variados métodos para realização de composição de músicas ou geração de outros elementos musicais por meio da inteligência computacional, dentre esses existe o uso de Redes Neurais Artificiais (ANN) que busca reconhecer e gerar padrões por inferência conexionista e aprendizado de máquina na resolução de problemas em composição de melodias, improvisação, contraponto ou harmonização. Aproveitando essa recente expansão do campo de inteligência computacional com as RNA já impactam amplamente os setores do entretenimento principalmente com a utilização de algoritmos de recomendação e de previsão do comportamento dos seus clientes, segundo KULESZ (2018), a experimentação do uso de algoritmos de aprendizado de máquina, no inglês *Machine Learning*, em produtos ligados ao entretenimento está em crescimento e que já mostra-se um diferencial estratégico de mercado dentro de grandes empresas digitais do ramo, como *Spotify* e *Netflix*.

O estudo visa permitir a democratização do uso de métodos de inteligência computacional na sociedade como o catalizador na resolução de problemas culturais e socioeconômicos. Segundo KULESZ (2018), com a inteligência artificial pode ajudar a empoderar os inúmeros criadores independentes, fazer a indústria cultural mais eficiente, aumentar o número de obras de artes, que é do interesse do público em geral. Provocar o fator interdisciplinar, abordar o tema Musica, que destoa do que é comumente aplicado em soluções de *deep learning*, portanto, a proposta é desenvolver um sistema inteligente de composição de músicas com atributos harmônicos e rítmicos que permita a síntese de novos arranjos musicais baseados em estilos já existentes e categorizados pela expressão musical de determinados artistas de renome da música brasileira, mais especificamente o gênero Bossa Nova, assim instigando a criatividade, como fonte de inspiração e também gerando insumos para músicos de toda indústria fonográfica, especialmente os produtores independentes.

FUNDAMENTAÇÃO TEÓRICA

Será apresentada a seguir a fundamentação teórica em tópicos para execução do projeto com base em referências bibliográficas reconhecidas.

Redes Neurais Artificiais (ANN)

Redes neurais artificiais consistem em uma forma de computação não-algorítmica baseada no funcionamento da estrutura biológica do neurônio, possuindo uma abordagem conexionista que não precisa a utilização de regras ou lógica previamente estabelecidas. A reprodução artificial de um neurônio é chamada de perceptron, idealizado pelo psicólogo Frank Rosenblatt em 1957 presente no artigo The *Perceptron: A Perceiving and Recognizing Automaton*, o tipo mais simples de um perceptron possui apenas uma camada de pesos conectando as entradas e saídas do sistema (SOMPOLINSKY, 2013), o formalismo matemático desse elemento está definido na função a.

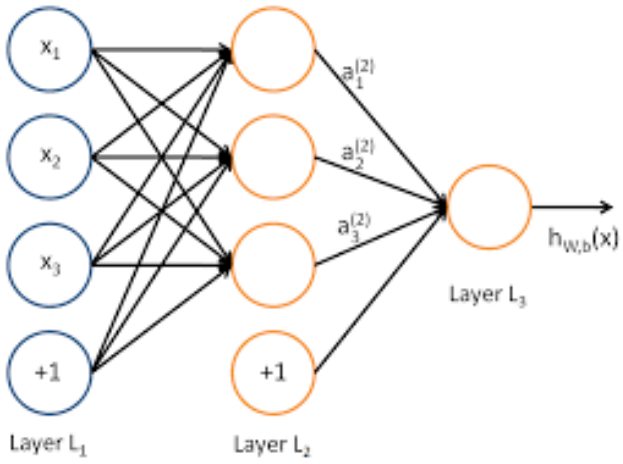
$$y = entrada(\sum_t^N w_t x_t - \theta) \ (a)$$

$$\Delta y = y_o - y \ (b)$$

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial b_{nj}} \frac{\partial b_{nj}}{\partial w_{ji}} = \delta_j \, x_i \, (c)$$

Em que o vetor peso (w_t) possuem valores dentro do conjunto dos Reais (R) que multiplica o valor de entrada, em seguida subtrai-se com valor θ de ativação caso o produto dessa subtração dê maior que o valor zero o perceptron permite a passagem do sinal, se o inverso ocorrer, cessa a emissão do sinal. Uma ANN utiliza da interação de várias unidades desses perceptrons em uma camada conforme a figura 01, a primeira camada de entrada é responsável por captar os dados exteriores da rede e direciona-los para cada um *perceptrons* na camada escondida L2 para depois realizar a soma desses valores na função 1.1 de ativação. Os sinais que resultam da camada L2 escondida na figura 01 convergem para a camada L3 de saída que realiza o processamento de todos esses sinais e resulta em uma única ativação final da rede com a decisão a ou classificação por uma função de ativação, neste caso de estudo função degrau, todo esse processo é descrito como *feedfoward*. Depois de realizar o cálculo dos resultados obtidos da rede neural, é necessário validar esses resultados, mas o comumente utilizado é o erro absoluto presente na função b, com a obtenção do índice de acerto do sistema, é possível verificar eficácia do modelo servindo de parâmetro para a evolução da rede por meio do algoritmo *Backpropagation*, e o ajuste é feito quando mudam os pesos das interconexões entre os neurônios artificiais (YÜKSEL et al. ,2011), esse ajuste é realizado da função c visando a um processo iterativo de ajustes de pesos rumo a minimização do erro do sistema por vetor gradiente descendente do erro (∂E_n), buscando o valor mínimo global do erro absoluto do sistema por meio dessa função integrada no algoritmo de *Backpropagation*.

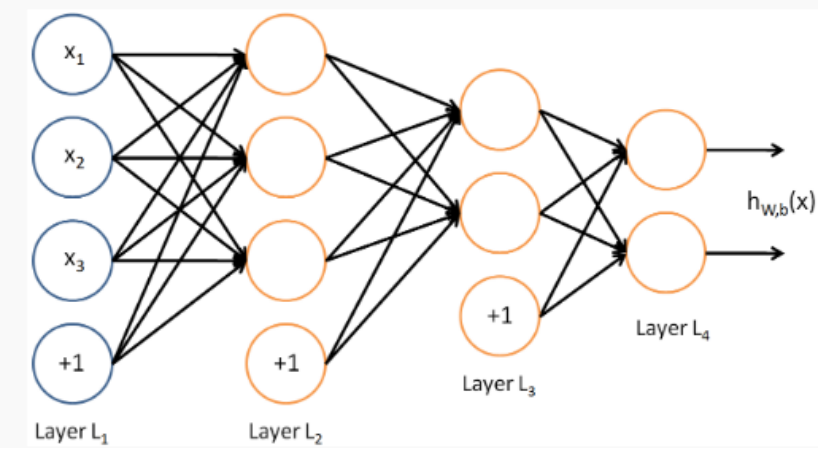
Figura 1.1: Ilustração de uma rede neural de camada escondida única.



Fonte: deeplearning.stanford.edu

Redes neurais de simples camada única possuem limitações, conforme SOMPOLINSKY (2013), “há muitas dicotomias que não podem ser feitas por *perceptrons*, um desses problemas são os não-linearmente separáveis”. Para a solução de problemas linearmente separáveis, foi apresentada a técnica de múltiplas camadas escondidas, utilizando-se de repetidas multiplicações de matrizes e entrelaçando-as com a função de ativação conforme a figura 1.2.

Figura 1.2: Ilustração de uma rede neural de múltiplas camadas escondidas.



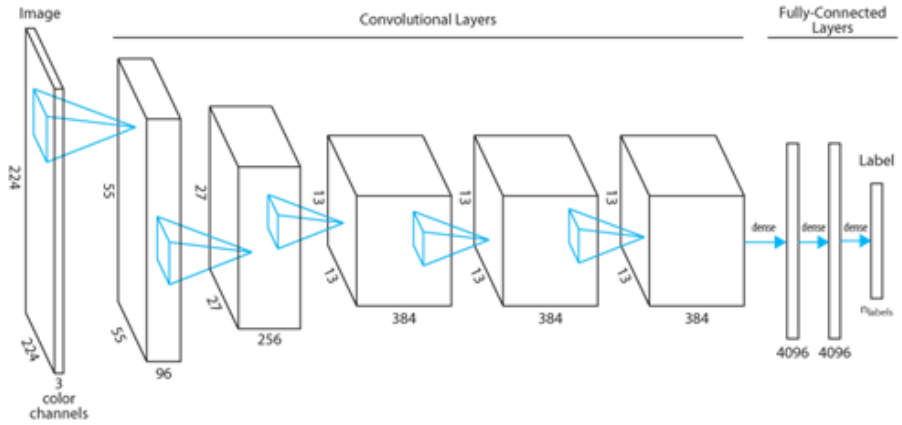
Fonte: deeplearning.stanford.edu

Redes Neurais Convolucionais (CNN)

Em Problemas referentes a VC, como reconhecimento de objetos em uma imagem, as soluções tinham caído em estagnação, pois era ineficiente utilizar a arquitetura ANN para esse tipo de problema pelo fator que a camada de entrada tem que ser dimensionada a dimensões da imagem analisada. Segundo R. MENDES (2019), para processar uma imagem com resolução de 250x250x3, portanto, possui 250 pixels de altura, 250 de largura e 3 canais de cores que pode ser o RGB, a camada de entrada da ANN normal teria que ser modelada para possuir o equivalente a 187.500, fora os pesos presentes nas camadas ocultas e por fim tornando um modelo muito ineficiente no âmbito de processamento computacional. Mesmo implementando com essa camada de entrada dimensionada a imagem, o modelo pode cair em problemas de desempenho, segundo KARPATHY e JOHNSON (2018) a ampla

conectividade presente nas primeiras duas camadas de uma ANN é dispendiosa, o grande número de parâmetros certamente converge para que apareça o fenômeno do *overfitting*, ou seja, modelo ficará super ajustado aos dados de treino e não irá conseguir generalizar para dados desconhecidos com boas métricas.

Figura 1.3: A ilustração da arquitetura CNN AlexNET.



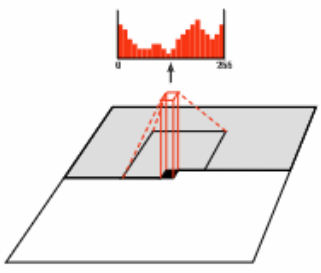
Fonte: Microsoft Machine Learning Blog (2017)

Visando resolver esse problema, foi apresentado durante a competição anual de reconhecimento de objetos chamada *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) de 2012, o problema consistia em uma base de treino de 1.2 milhões de imagens com 1000 categorias e a de teste com 150,000 fotografias e o modelo que melhor performar em classificar essa base de dados é dado como vencedor do ano. O modelo AlexNet com a arquitetura Redes Neurais Convolucionais (CNN) ganhou a edição de 2012 com uma taxa de erro de 15.3, 10.8% menor em comparação ao segundo lugar, um salto nunca visto na competição e que abriu precedentes para que no ano de 2015 o modelo vencedor superasse a habilidade humana na competição.

O uso de CNN consiste em desenvolver uma camada de entrada da rede disposta não linearmente, sim tridimensionalmente, onde lidamos com camadas com largura, altura e profundidade, vemos em ilustração a configuração da CNN na figura 1.3, precisamente nas primeiras 5 camadas. Segundo ALBAWI e MOHAMMED (2017), um dos aspectos mais benéficos de CNNs é a redução do número de parâmetros de entrada em uma ANN e também a melhoria da eficiência da velocidade do processamento de dados na inferência e

treinamento. A camada de entrada com convolução funcionará semelhante a uma “lupa” sobre uma imagem, ela não irá analisar por inteiro uma imagem de uma só vez, sim áreas correspondentes a estrutura em formato de triangulo na figura 1.3, Conforme PONTI e COSTA (2017), a camada convolucional processa a imagem e transforma por meio de cálculo de combinação linear da vizinhança de 27 pixels da imagem em um único pixel de saída que subsequente é integrado na construção do mapa de características do modelo. A camada de convolução é responsável por compilar um mapa de características, Segundo R. MENDES (2019) o mapa armazena informações de padrões detectados no cálculo de convolução em formato de vetores, esses padrões podem ser armazenados por exemplo podem achar em uma imagem formatos referentes a orelhas de gatos e olhos. Outra função que é aplicada no filtro de convolução é a técnica de *pooling*, possui o objetivo de reduzir a escala dos mapas de características, podemos ver isso ocorrendo na função 1.3 que entre a primeira, segunda e terceira camada possuem volumas e formas diferentes, o pooling é responsável por interconectar as camadas vizinhas. A camada convolução resulta na criação do mapa de características e no redimensionamento dos *inputs* para auxiliar a próxima camada da figura 1.3, a antepenúltima e penúltima que são MLP da arquitetura e são responsáveis por realmente fazer as classificações e inferências por meio do aprendizado profundo e ao fim em uma função output para decisão da classe.

Figura 1.4: A visualização da arquitetura *PixelCNN* realizando o mapeamento da vizinhança de pixel para predizer o próximo pixel, para gerar esse pixel o modelo pode apenas usar como condição o pixel anteriormente gerado.



Fonte: OORD et al (2016).

A pesquisa lida com geração de material musical, não classificação como as CNN são normalmente aplicadas, existe uma derivação da mesma que se chama *PixelCNN* e indicada

para geração de imagens. Segundo OORD et al (2016), a ideia básica da arquitetura *PixelCNN* é usar conexões auto regressivas para modelar novas imagens pixel por pixel, decompondo a imagem original como um produto de condicionais. Para processar música em uma CCN, seria preciso transformar a música que normalmente ouvimos em som para uma representação visual, existe o arquivo *mid*i que representa a música digitalmente que carrega informações das notas acionadas por tempo, portanto, transformar essa informação em imagem em que a altura representa a escala do piano, largura progressão do tempo e no fim a profundidade lida com a ativação da nota. Com a arquitetura *PixelCNN* é possível estabelecer coesão ao gerar um pixel baseado no pixel anterior, vemos na figura 1.4 a decisão da arquitetura da geração do pixel na distribuição de probabilidades construída conforme os pixels vizinhos.

Redes Neurais Recorrentes (RNN)

Uma ANN normalmente processa o dado de entrada em suas camadas e emite sua decisão, assim realizando um ciclo, porém, com uso de RNN, um modelo preditivo permite armazenar informações da entrada anterior e usa-las como para parâmetros para novas decisões nas entradas subsequentes. Isso permite que as redes neurais realizem processamento temporal e também aprenda entender as sequências, como por exemplo: realizar reconhecimento, reprodução de sequência ou associação e predição temporal (BULLINARIA, 2015).

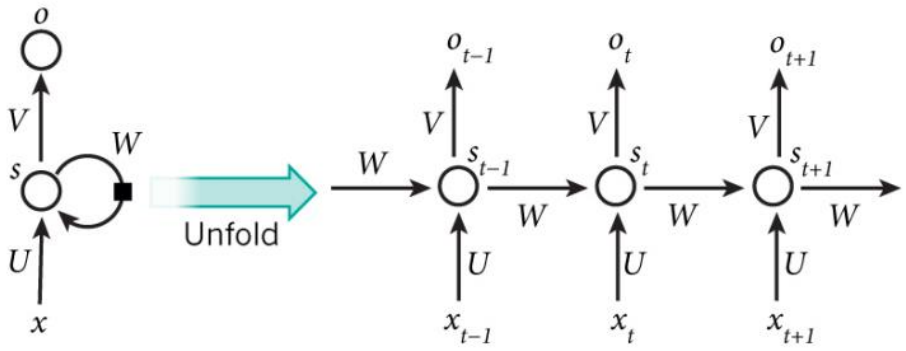
$$h_t = \varphi W(\, Uh_{t-1}\, , x_t\,)\ (d)$$

$$y = entrada(\textstyle\sum_t^N Wh_t\ -\ \theta)\ (e)$$

Na equação e descreve como perpetuar uma informação passada, o sinal x_t passa pela função de ajuste de do estado oculto na figura 1.5, tem o somatório do sinal atual x_t com o estado anterior Uh_{t-1}, o produto da soma por fim é processado em uma função de ativação não linear φ com os parâmetros W do perceptron, resulta em novo sinal de entrada h_t do perceptron da rede com informações do sinal de entrada h_{t-1}, portanto a função a básica do *perceptron* RNN é modificada conforme a função e.

Figura 1.5: Ilustração do perceptron da Rede Neural Recorrente, o valor peso W dos perceptrons mudam conforme é destrinchado (*Unfold*) o estado oculto s_{t+n} em períodos

de tempo: $t + 1$ e $t - 1$, portanto a ilustração implica que o peso W dos perceptrons da rede neural sempre está em alteração contínua por fatores temporais ocorridos previamente.



Fonte. BRITZ (2015).

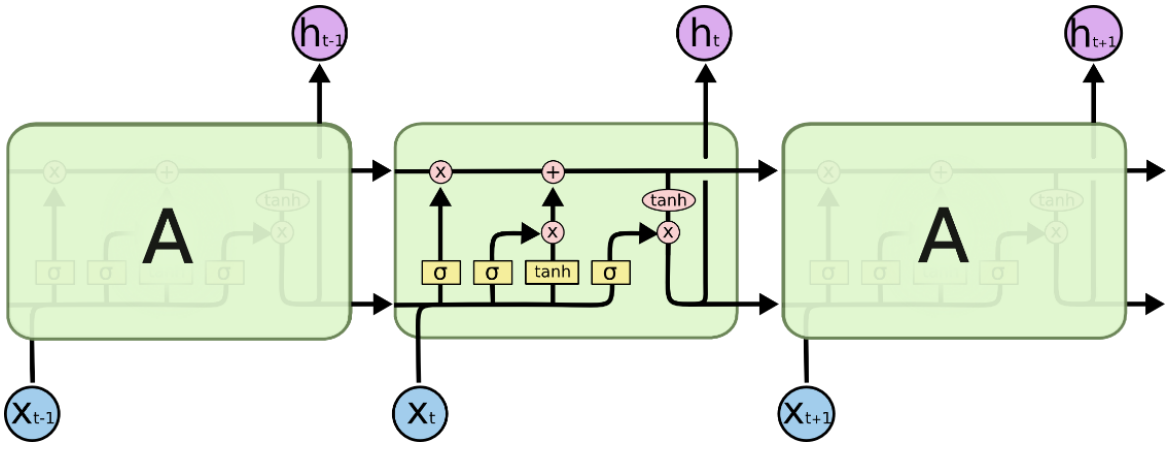
A RNN pode aprender uma distribuição de probabilidade para uma sequência de valores, sendo treinado para prever o próximo símbolo na sequência (CHO et al, 2014). Portanto o fato que a arquitetura RNN armazena informações pregressas de inferências passadas e depois utiliza-se no presente, é ideal como um recurso na síntese de músicas, pois é latente a necessidade de uma "Memoria" quando o modelo irá inferir uma nova nota dependendo da última tocada e assim permitindo uma coesão para ter uma música, segundo HOCHREITER e SCHMIDHUBER (1997) redes recorrentes possuem potencial significante para variados tipos de aplicações, dentre elas incluem o processamento de fala, controle não-markoviano e composição de música.

Redes Neurais de Memória de Longo e Curto prazo (LSTM)

Demostrado anteriormente, as RNN possui uma arquitetura que simula uma "memória de curto prazo" quando armazena estados e *feedbacks* de interações $t - 1$ passadas para poder influenciar em futuras entradas. Uma memória de curto prazo possui desvantagens, segundo P.H. MENDES (2019), em experimentações com RNN simples para geração de músicas, foi observado que o uso de memória de curto prazo por si só não consegue convergir para mínimo local ou global dentro do problema proposto, resultou em

amostra com a presença de repetições de uma única nota, portanto, caindo em um loop de símbolos. Uma maneira de simular uma mémoria de longo prazo computacional foi apresentado no Artigo Long Short-Term Memory de Sepp Hochreiter e Jurgen Schmidhuber de 1997 , segundo HOCHREITER e SCHMIDHUBER (1997), a rede neural LSTM apresenta não necessitar de um ajuste fino hiper parâmetros em treinamentos principalmente em lidar com o Learning rate, tende a generalizar bem as modelagens aplicadas, consegue lidar com ruídos em representações de distribuição e variáveis continuas com espaço de tempo amplo.

Figura 1.6: Ilustração do funcionamento de uma rede LSTM de perpetuamento dos estados ocultos (h_{t-1}) ao longo do tempo, sinal de entrada (x_t) alimenta os retângulos amarelos que indicam uma camada da rede neural com dois diferentes tipos de funções de ativação: sigmoide (σ) e de tangente hiperbólica ($tanh$). Os vetores em preto indicam o fluxo percorrido pelo os sinais. o círculos na coloração rosa indica operações aritméticas com resultados das camadas da rede neural, aplicando operações de soma ($+$), multiplicação ($\times$) e tangente hiperbólica ($tanh$).



Fonte: colah.github.io

A arquitetura LSTM propõe realizar o acumulo, alteração e adição os valores dos estados ocultos $t-1$ em uma célula de memória e com operações com unidades de portão, a figura 04 ilustra o fluxo de administração e destino dos valores pertencentes ao estado oculto, o vetor horizontal que transpassa ao topo do digrama, permite que uma informação flua com

as possíveis operações de alteração desse estado oculto para auxiliar no processamentos de novos sinais de entrada na rede neural.

$$f_t = \sigma(w_f \times [h_{t-1}, x_t] + b_f)\ (f)$$

$$i_t = \sigma(w_i \times [h_{t-1}, x_t] + b_i)\ (g)$$

$$c_t = tanh(w_c \times [h_{t-1}, x_t] + b_c)\ (h)$$

Alteração das informações $t-1$ - na arquitetura LSTM ocorre por meio fluxos e operações chamado *Gates*, segundo OLAH (2015) *Gates* compõe uma camada superior de controle da rede neural com funções de ativação do tipo sigmoide (σ) ou tangente hiperbólica ($tanh$) para controlar os valores armazenados nas células de memória. O Primeiro *Gate*, com a sigmoide (σ) e a operação de multiplicação na figura 1.6, é conhecida como "Porta de Esquecimento" ou "Porta multiplicativa", quando a saída da função f é igual a 0, apaga os valores armazenados no estado escondido (h_{t-1}) multiplicando pelo valor nulo obtido, caso o valor seja 1 o valores (h_{t-1}) na unidade de memória se mantem inalteráveis. A Segunda *Gate* é responsável por lidar coma a inserção de novos valores no estado escondido (h_{t-1}), possui duas camadas da rede neural conforme na figura 1.6 e tem sua função $(i_t \times c_t) + h_{t-1}$, a primeira função g de ativação sigmoide (σ) irá determinar se o novos valores da função ativação da tangente hiperbólica ($tanh$) da equação h serão adicionados no (h_{t-1}) em uma operação de multiplicação, assim adicionando novos valores para novo estado escondido (h_{t-1}) em resultado igual a 1 na sigmoide. e caso dê 0 no resultado da ativação a nova informação não será alocada no h_t.

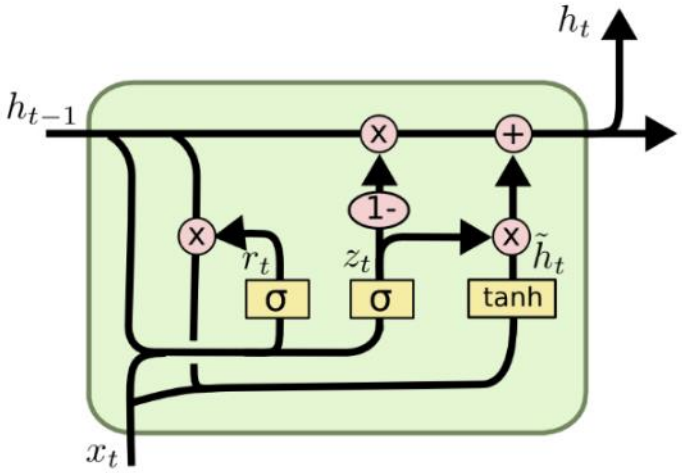
$$o_t = \sigma(w_0 \times [h_{t-1}, x_t] + b_0)\ (j)$$

$$h_t = o_t \times \tanh(ct)\ (i)$$

No fim do processo haverá a última decisão da rede, após a obter todos dos resultados provenientes da *Gate* de esquecimento e da *Gate* de inserção, o sistema possui um h_{t-1} consolidado para enfim ser agregado na inferência. Função da rede neural responsável da inferência é constituída pela função j, com a ativação sigmoide (σ) , portanto, o sinal 0 ou 1 resultante dessa camada é multiplica o produto resultante da função tangente hiperbólica do valor h_{t-1} presente na função i, no final gerando um valor de saída do sistema h_t

influenciado por saídas anteriores para que na sequência também possa influenciar a saída h_{t+1}.

Figura 1.7: Infográfico do funcionamento da arquitetura GRU.



Fonte: DRAKOS (2019).

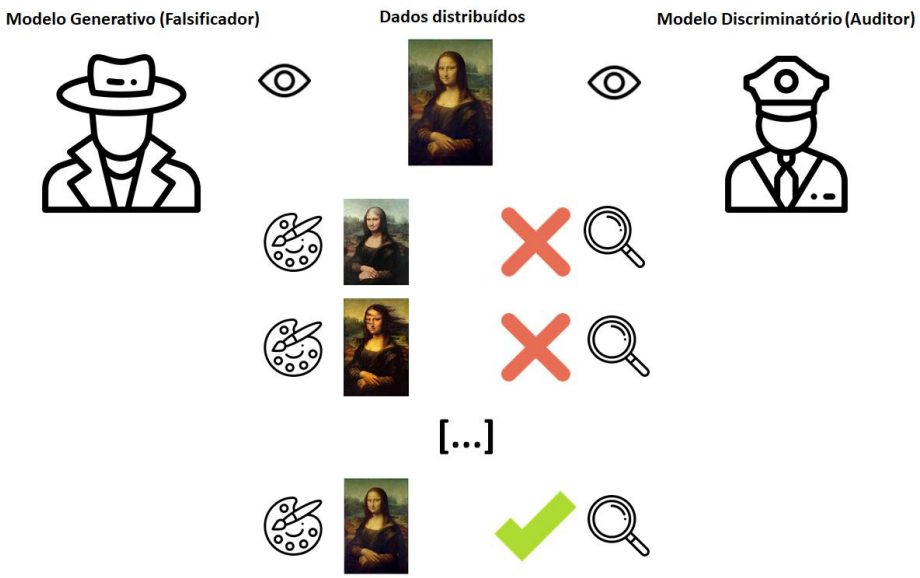
Derivada da arquitetura LSTM, existe a GRU, traduzida para o português Unidade como Portas Recorrentes, é uma arquitetura recorrente com grandes similaridades com a LSTM, a semelhança mais latente é o uso de portões ou *Gates* na manipulação do estado oculto anterior, conforme podemos verificar comparando as figuras 1.6 e 17, podemos destacar que ambas possuem os portões do esquecimento (*Forget Gate* ou *Reset Gate*) e o da inserção (*Update Gate*). A GRU segundo NAYEBI e VITELLI (2015), foi concebida visando resolução de tarefas de PLN, preenchendo uma lacuna existente entre o modelo RNN e LSTM, sendo menos onerosa em processamento computacional que a LSTM, porém, mais eficiente que uma RNN simples. A principal diferença entre a LSTM e GRU segundo MENDES (2019), ocorre primeiramente que a informação $T-1$ na GRU muda mais lentamente que uma RNN simples e mais ágil que uma LSTM.

Redes neurais Generativas Adversais (GANS)

GANS ou Redes neurais Generativas Adversais, pode ser considerada mais uma forma de treinamento de que uma arquitetura em si, consiste em utilizar duas ANN de arquitetura livre para execução de tarefas de geração de dados. A interação entre os dois

modelos é o diferencial encontrado nessa arquitetura, Segundo GOODFELLOW et al. (2014), foi concebido um método de redes adversárias, ou seja, ambas acessam a mesma base de dados e possuem o objetivo de competir entre si em métricas alcançadas durante o treino, o modelo generativo aprende criar novos dados e o outro modelo discriminatório aprende classificar se a amostra inferida é um output do modelo generativo ou da distribuição de dados utilizadas.

Figura 1.8: O modelo GANS pode ser explicado, em um formalismo simples por meio dessa ilustração, ao realizar algo semelhante a uma relação entre um falsificador e um auditor, o falsificador tem o objetivo de criar uma obra de arte a fim de enganar o auditor com vistas a realizar um lucro fácil, que por sua vez o auditor irá verificar essa obra e julgar se a obra apresentada é realmente falsa ou supostamente verdadeira. Assim, cada um desses dois agentes irá testar a habilidade do outro, após interações da recusa do auditor, esse falsificador atinge certo grau de excelência após o repetitivo aperfeiçoamento das técnicas de síntese da obra por meio da tentativa e erro, em um determinado espaço de tempo, o auditor não saberá distinguir se a obra verificada é verdadeira ou falsa por conta desse aprendizado por reforço.



Fonte: MENDES (2019).

O funcionamento é bem com similar o aprendizado em reforço, os dois modelos utilizam de erros e acertos do outro para melhorar nas suas próprias métricas, podemos ver um exemplo metafórico do funcionamento do treinamento de GANS na figura 1.8 na relação que existe entre o falsificador e o auditor de artes. O formalismo matemático, chamamos o modelo generativo de $G(x)$ e discriminatória $D(x)$, um aspecto importante dessa arquitetura é sua a função k de erro, é capaz de gerenciar e utilizar o erro das duas ANN para o benefício de todo o sistema GANS. Ao treinar o gerador, queremos maximizar esse erro enquanto tentamos minimizá-lo para o discriminador (ROCCA, 2019).

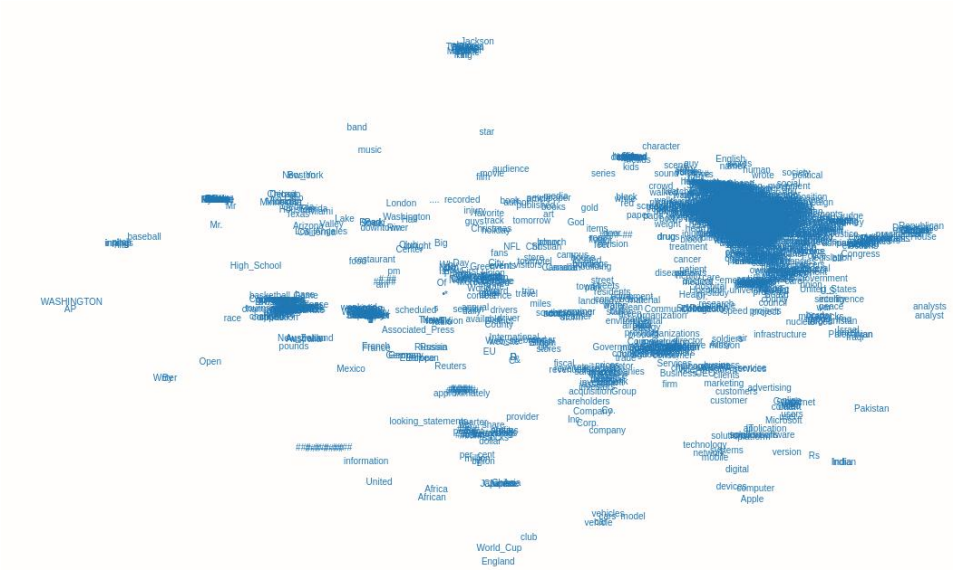
$$E(G, D) = \frac{1}{2}\,\epsilon_{x\sim pt}[1 - D(x)] + \frac{1}{2}\,\epsilon_{z\sim py}[D(G(x))]\,(k)$$

O maior problema identificado é o alto custo de processamento computacional para realizar um treinamento de GANS, os motivos é são duas ANN para realizar um objetivo, portanto, precisam de o dobro de capacidade de processamento e alocação de RAM que uma arquitetural tradicional DP necessitaria para realizar o mesmo objetivo.

Modelo *Word2Vac*

No campo de PLN, um dos grandes desafios enfrentados em pesquisas é criar uma nova forma modelar um corpus e ser capaz de sintetizar as informações imbuídas nesssa base em formato de texto legível ao humano para uma nova forma legível à nível computacional. Podemos listar técnicas tradicionais como o *Bag of Words*, onde transformamos textos, sentenças ou frases em tabelas de frequências de ocorrências de palavras ou símbolos, porém, eliminamos aspectos fundamentais da linguagem humana ao realizar tal técnica, como por exemplo a gramática e a o valor semântico do texto.

Figura 1.9: Visualização do modelo *word2vec* referente a notícias do portal de informações *google news*.



Fonte: SCHMIDT (2017).

Modelos Word2Vac (Word2Vac) inovam na abordagem de modelagem de PLN, podendo capturar informações referentes a similaridades semânticas entre símbolos ou palavras em uma abordagem utilizando técnicas de álgebra linear na criação de um plano de multidimensional conforme a Figura 1.9, onde vemos uma distribuição de palavras ao longo do plano e elas aglomeram-se conforme a similaridade encontrada no corpus utilizado. Segundo MIKOLOV et Al. (2013a), o funcionamento do modelo W2V ou *Skip-gram* ocorrem em um espaço multidimensional contendo n vetores com dados, posição, sentido e direção provenientes de cálculos de similaridade probabilística e de distância euclidiana encontrada a partir dos símbolos presentes no corpus processado. Na figura 1.9, vemos agrupamentos por similaridade propostos pelo modelo W2V, dentre eles vemos o *Investment*, *Banks*, *Assests*, *Financial* e *Investors*, em português Investimento, Bancos, ativos, financeiro e investidores, claramente vemos uma latente similaridade entre essas palavras e que corresponde a um possível tema central sobre investimentos ou mercado de ações. Representar palavras em um espaço de vetores beneficia modelos de ML em performance em problemas de PLN MIKOLOV et Al. (2013b).

Podemos utilizar técnicas de PLN para gerar músicas e modelar a base de dados para aplicações em ANN, semelhante a ao que foi reproduzido com notícias conforme na figura

1.9, Segundo HERREMANS e CHUAN (2017), o modelo é passível de capturar a notação tonal de músicas ao considerar a similaridade entre uma amostra original e as amostras modificadas durante o experimento.

Música teórica

Música é um fenômeno lógico, físico e abstrato que integra a evolução da humanidade, presente em questões sociais, culturais, políticos, históricos e econômicos ao longo das eras. A capacidade humana de entender e processar música é formada por dois domínios aparentemente distintos um do outro: o domínio da subjetividade abstrata que abrange a composição musical e imaginação artística; e o domínio da objetividade abstrata, que abrange operações lógicas e raciocínio matemático (CORRÊA, 2018). O computador pode lidar com eficiência os domínios das operações lógicas e matemáticos ligados a música, para utilizar música em um modelo inteligente, temos que converter os sons analógicos em um protocolo digital no campo discreto para que possa ser interpretado pelo computador, portanto para essa tarefa é amplamente utilizado o tipo de arquivo chamado MIDI, que é um Protocolo de comunicação entre instrumentos musicais e computadores que segundo SILVA (1999) é uma sintaxe para a codificação de informações que sintetiza atributos de ritmo, tons e harmonia em uma sequência de mensagens e dados de comunicação em formato binário. Um grande desafio no projeto é que uma rede neural possa conseguir simular a subjetividade na inferência de melodias, por meio do aprendizado dos padrões presentes nas composições utilizadas no treino.

Melodia, Ritmo, Timbre e Harmonia

O ato de realizar música é muito intuitivo, consiste em manipular sons distintos e aplicar as variáveis de ritmo, timbre e harmonia nessa composição, tanto que uma criança pode utilizar as duas primeiras técnicas ao descobrir ao simples bater uma colher de pau em uma panela de alumínio e logo depois em uma bacia de plástico. Logo ela irá perceber que o som comportará de forma diferente em cada um dos materiais, esse atributo

descoberto chama-se timbre, muito conhecido também como a “cor” do som, o som da panela possui uma frequência mais aguda e também ressoará maior por conta do material, no enquanto a bacia reproduz um som mais seco e grave que não ressoa como o objeto anterior. Segundo LOUREIRO e PAULA (2006), Diferentemente de outros atributos do som musical, tais como altura, volume e duração, o timbre não pode ser associado a apenas uma dimensão física, não podendo ser especificado quantitativamente pelo sistema tradicional de notação musical como são o volume e a altura, descritos a partir de escalonamentos entre fraco-forte e de gamas de alturas. Podemos identificar essa gama de alturas por meio da equação l fundamental da ondulatória para identificar os perfis da amplitude λ da onda para os graves e agudos, também a frequência da onda f.

$$v = \lambda \times f\ (l)$$

Após reconhecer que sons podem ter características distintas, então essa criança experimenta bater a colher de pau em diferentes frequências tanto na panela e quanto na bacia, ele percebe que bater em uma frequência maior na panela, produz uma sonoridade mais interessante por conta da ressonância do metal e ao bater na bacia de plástico de forma marcada como o bumbo da bateria, ele percebeu que ajuda no compasso da batida na panela, portanto essa propriedade musical experimentada é conhecida com ritmo, segundo KIEFER (1973), o primeiro movimento articulado foi o tempo na música, organizando em subdivisões ou seções perceptíveis, configurando a característica rítmica da melodia. Portanto, na figura 1.10, temos a representação gráfica de compasso composto que irá expressar o ritmo de toda a melodia ou partes dela. Outra forma de aplicar ritmo, ocorre quando usamos a notação BPM, batidas por minuto ($\frac{b}{60}$), o instrumento que mede essas batidas é chamado de Metrônomo de Maezel, possuindo o importante atributo da precisão e maiores possibilidades de configurações de ritmo comparado aos modos do compasso composto.

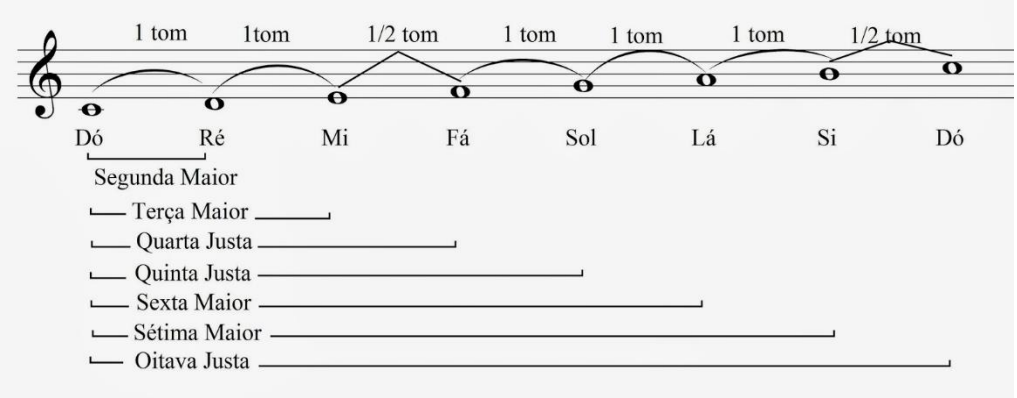
Figura 1.10: Compasso composto na teoria musical, possui modos binário, ternários e quartanários de ritmos aplicada graficamente em partituras.

Unidade de Tempo	Binário	Ternário	Quaternário
U.T. = 𝅗𝅥 = ♩ + ♩	2/2 𝅗𝅥 𝅗𝅥	3/2 𝅗𝅥 𝅗𝅥 𝅗𝅥	4/2 𝅗𝅥 𝅗𝅥 𝅗𝅥 𝅗𝅥
U.T. = ♩ = ♪ + ♪	2/4 ♩ ♩	3/4 ♩ ♩ ♩	4/4 ♩ ♩ ♩ ♩
U.T. = ♪ = 𝅘𝅥𝅯 + 𝅘𝅥𝅯	2/8 ♪ ♪	3/8 ♪ ♪ ♪	4/8 ♪ ♪ ♪ ♪

Fonte: musica.culturamix.com

A próxima propriedade da música talvez uma criança não consiga entendê-la ou praticá-la, pois ela normalmente pode reproduzir os sons da panela e da bacia de forma destoante e que não pode ser muito prazeroso para os ouvintes ao seu redor. A propriedade da harmonia consiste na formação de acordes com a música tonal, utilizando os tons característicos como C = Dó, D = Ré, E = Mi, F = Fá, S = Sol, L= La e B = Si, entanto esses acordes são agrupados em escalas, em sequencias de tons (tons C, D, F, S e L) e semitons (tons E e B), com o uso tom na forma maior ou menor na respectiva escala utilizando-se a sobreposição terças, ou seja, o terceiro grau da escala escolhida, podemos observar esses intervalos de tons na figura 1.11.

Figura 1.11: Intervalos específicos da escala de Dó maior.



Fonte. segredosdofole.blogspot.com

Bossa-Nova

A bossa nova foi um movimento bairrista que ocorreu em bares e casas de shows pertencentes a zona sul da cidade de Rio de Janeiro, precisamente nos bairros notoriamente de classe média de Copacabana, Ipanema e Leblon, ao longo da década de 50 e o início da seguinte, os anos 60. Esse período foi marcado por largos avanços culturais, econômicos e sociais provenientes do mandato do Presidente Juscelino Kubitschek (1951-1961) no Brasil, com o plano de governo conhecido como *cinquenta anos em cinco*, segundo LUIZ e NASCIMENTO (2011), naquele especial momento histórico da Guerra Fria, o Brasil possuía um estreitamento maior de relações com o Estados Unidos da América, essa interferência americana possibilitou um diálogo cultural com jazz e o surgimento de aparelhos de televisão, influenciando os pioneiros do movimento Bossa Nova com um registro musical intimista proveniente do Jazz e subsequente a nova mídia TV ampliava o alcance desse recente gênero para patamares tanto nacionalmente e internacionalmente.

Figura 1.12: Da direita para esquerda em sequência na fotografia estão dispostos: Toquinho, Tom Jobim e Vinicius de Moraes em momento de ensaio.

Fonte. FONSECA (2016).

O gênero bossa nova mirava em uma ruptura da postura da música popular naquele período, ao importar tendências e salientar aspectos já existentes da música popular brasileira, no caso o samba e suas ramificações, assim criando um novo produto musical

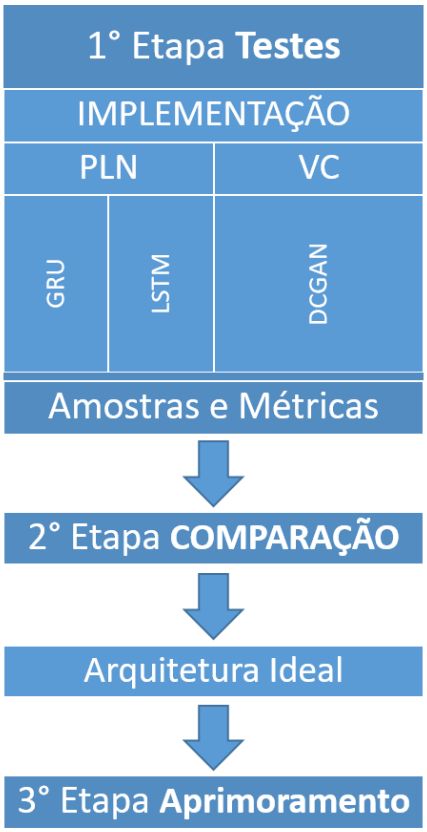
final afável até para o consumo internacional, segundo LUIZ e NASCIMENTO (2011), o movimento rompeu com o padrão da música popular anterior na rejeição dos sambas-canções e também no bolero, para alterar o modo comum de interpretação, composição e principalmente nos temas abordados no gênero bossa nova que foca no amor, a cidade do Rio de Janeiro e a melancolia.

Podemos caracterizar o gênero bossa nova em características práticas segundo LUIZ e NASCIMENTO (2011), com a presença de acordes dissonantes provenientes da "antropofagia" do Jazz, compassos sincopados do Samba, a harmonia e a peculiar batida de violão criada pelo interprete João Gilberto é possível delinear o estilo bossa nova. Podemos ver na figura 1.12 alguns dos artistas mais relevantes do movimento que foram Vinícius de Moraes, Tom Jobim, Nara Leão, João Gilberto e Toquinho, alguns dos artistas citados estão presentes na figura 1.12. No exterior, a Bossa Nova é um marco na modernidade do Brasil durante o século XX, foi o gênero de composição brasileiro mais reconhecido internacionalmente. Os bossa-novistas tiveram a oportunidade de se apresentar e gravar, principalmente nos Estados Unidos, Europa e Japão (MARQUES, 2011).

MÉTODO

O estudo desdobra-se na metodologia quali-quantitativa com a finalidade de comparar arquiteturas de ANN's na tarefa de composição de músicas do movimento bossa nova, utilizando os atributos referente a implementação e qualidade das amostras de músicas geradas para extrair o modelo que melhor performa dentro do ambiente e regras estabelecidas nesta metodologia.

Fluxograma 1.1: Fluxo da metodologia da pesquisa.



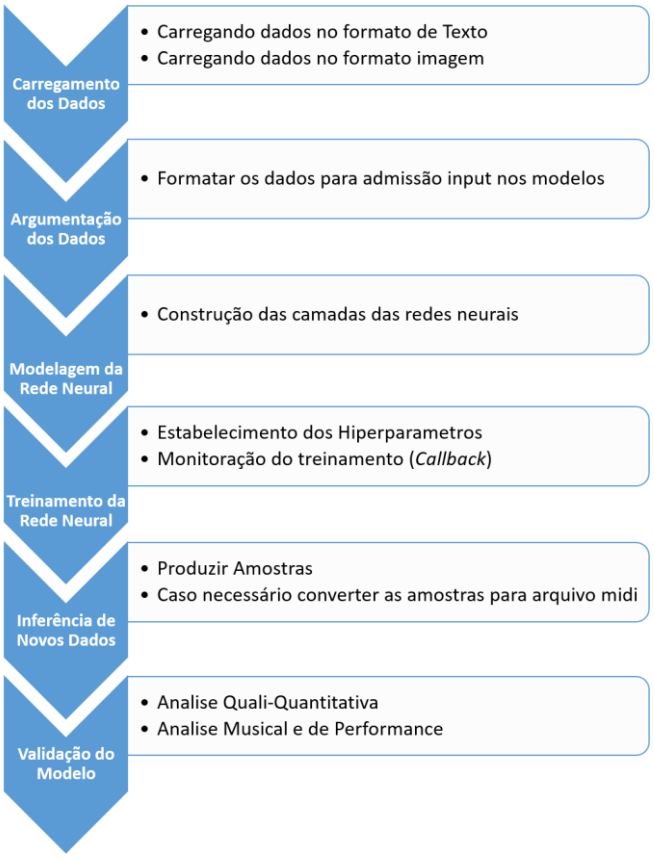
Fonte: Própria.

Conseguimos visualizar as fases e andamento do processo por meio do fluxograma 1.1, na primeira etapa apresentada é a Testes, consiste na implementação das arquiteturas apresentadas e estudadas no referencial teórico e tem como saída as amostras e indicadores gerados da implementação. Dentro dessa fase implementação foi proposta

condições mais igualitárias possíveis entre as arquiteturas aplicadas para que haja condições semelhantes para poder inferir uma comparação, a primeira dentre elas é o uso de uma base de dados padronizado em âmbito de informação e quantidade que ela contém. Outro ponto é o valor base de no mínimo 100 épocas no parâmetro de ciclos de *feedfoward* e *backpropagation* configurado na rede neural implementada no momento do treinamento, portanto, que nos leva reconhecer um forte indicativo de eficiência na obtenção de um valor mínimo da arquitetura abordada com a base de dados compartilhada.

Embora haja condições semelhantes para realização da implementação, temos técnicas em diferentes campos de atuação sendo abordadas na pesquisa, a primeira é o PLN e em seguida a VC, cada uma dessas abordagens requer detalhes de tratamento de dados ou treinamento do modelo preditivo específicos, mesmo com essas diferenças podemos usar um fluxo de desenvolvimento único conforme o fluxograma 1.2 para ambas abordagens.

Fluxograma 1.2: Fluxo do desenvolvimento dos modelos preditivos utilizados na pesquisa.



Fonte: MENDES (2019).

Estabelecendo condições iguais para desenvolvimento dos modelos estudados, geramos resultados e amostras que são entradas para a segunda etapa do fluxograma 1.1, chamada de Comparação, onde é avaliado no âmbito qualitativo da pesquisa o coeficiente lógico *Loss*, traduzido direto do inglês significa perda ou erro e é o índice que norteia o processo *backpropagation*, registra o erro quadrático médio (MSE) alcançado pelo modelo e que pode ser calculado por meio da equação m.

$$Loss(y, \hat{y}) = \sum_{n=1}^{n} (y - \hat{y})^2 \ (m)$$

y é o valor real e o $\hat{y}$ é o valor inferido, portanto, é calculado a diferença do quadrado desses valores que denota um índice de assertividade para essas predições.

Podemos inferir com o *Loss* tanto na eficiência e efetividade alcançada pelo modelo treinado, portanto, valores perto de 0 são considerados o objetivo de um modelo inteligente com essa métrica em foco, com isso com podemos entender se o modelo convergiu para um mínimo local efetivo. Relacionar o indicador de *Loss* com a ocorrência das épocas conseguimos constatar a eficiência do modelo no momento do treinamento para encontrar a convergência, quanto menor o número da época o modelo convergir a um *Loss* considerado baixo, esse modelo pode ser considerado eficiente.

No âmbito qualitativo temos as amostras geradas, primeiramente temos que constatar se essa amostra possui características pertencentes ao gênero “Bossa Nova”, dentre elas temos os atributos da harmonia e ritmo presente nas amostras, segundo SIMONETTA et Al. (2018), o uso da técnica de Redução Musical Harmônica (RMH) é uma representação visual que representa teoricamente uma música pela perspectiva horizontal no contraponto e a vertical na harmônica em um modo simplificação sem perda de contexto e harmônicas. Com a representação RMH, é possível traçar e calcular grau de similaridade harmônica entre músicas na mesma representação proposta, portanto, ao utilizar o W2V com o objetivo de encontrar relação entre acordes em composições “Bossa Nova”, conforme o vetor X extraído da pesquisa de MENDES (2019), informa os principais acordos utilizados na música Águas de Março, informando a estrutura harmônica na notação

numérica romana.

['III64', 'III64', '#IV6', 'II43','IV', 'I', 'IV7'] (12)

Com os atributos quali-quantitativo definidos e analisados, podemos decernir qual das arquiteturas estudadas e desenvolvidas é a que melhor gerou resultados e será encaminhada para a próxima fase, de aprimoramento, sendo o último passo do fluxograma 1.1 que é priorizado a melhoria dos aspectos negativos encontrados no modelo na etapa anterior com o objetivo de otimizar a geração de amostras dentro da base de dados coletada na pesquisa.

RESULTADOS E DISCUSSÃO

A metodologia sugerida para produção de resultados abrange as etapas de Testes, Comparação e Aprimoramento presente no fluxograma 1, de forma sequencial em que as entradas do processo são a fundamentação teórica e base de dados utilizada para a pesquisa. Muitas informações e dados que impactam esse projeto são oriundos de um estudo passado realizado por MENDES (2019), o mesmo autor que desenvolve essa pesquisa, portanto, este relatório é uma progressão direta do estudo "*Analise Quali-Quantitativa de Arquiteturas de Redes Neurais Artificiais para Geração de Música Polifônica*" que abrange até a etapa de comparação da metodologia. O ponto crucial e mais importante do estudo é a base de dados, comparável como o "combustível" da pesquisa, para isso fora adquirido arquivos do formato *midi* de diferentes bases e repositórios na internet, pois não fora encontrado uma fonte única ou acadêmica que contenha arquivos *midi* de músicas do gênero "Bossa Nova", em contrapartida é facilmente encontrado esse tipo de arquivo em repositórios dedicados à música clássica e do Jazz na rede mundial de computadores. O levantamento dos dados se revelou desafiador, segundo MENDES (2019), a procura resultou em amostras de músicas *midi* em bases de terceiros e não oficiais que não é garantido se as composições são originais dos autores ou se é regravações de terceiros, portanto, a integridade do material não pode ser comprovada. Embora existam os problemas de integridade dos dados, é possível por meio de uma análise manual e filtragem dos materiais se as amostras conferem com o estilo proposto comparando com o

material original. Realizando esse batimento podemos levantar uma base útil para a pesquisa e dentro do que foi proposto. Todos os códigos, modelos e dados da pesquisa estão disponíveis no repositório *https://github.com/pedro-h-mendes/Music-Generation*.

Implementação

A primeira etapa conforme a metodologia se baseia na implementação de redes neurais na resolução de problemas de composição musical, dividimos o escopo de abordagem PLN e VC com o objetivo de gerar dados e insumos para a próxima etapa de *Comparação*.

Implementação – Processamento de Linguagem Natural (PLN)

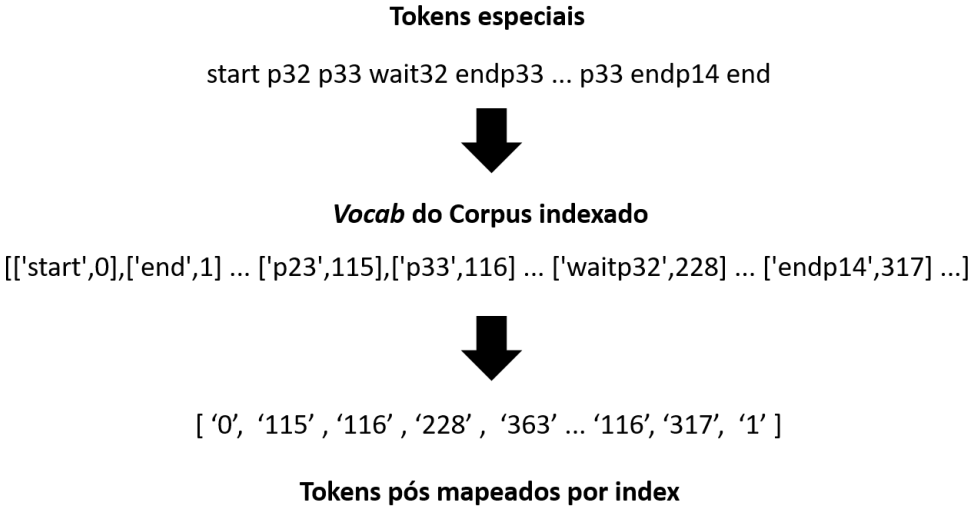
A primeira técnica, a PLN, segundo MENDES (2019), foi o método com maior número de referências de estudos na síntese de música por inteligência artificial, citandos os trabalhos de TODD (1989), CHEN e MIIKKULAINEN (2001), YÜKSEL (2011), COCA et al. (2011), NAYEBI e VITELLI (2015) e PAYNE (2018). Uma possível resposta para esse número de referências é utilizar uma heurística de lidar a estrutura de uma música como uma gramática de símbolos com tons e semitons para construir uma estrutura semelhante em âmbito de regras e intergerações com os idiomas da linguagem humana. A abordagem desse tipo ocorreu primeiramente por TODD (1989) deduziu que cada nota musical gerada possui uma dependência de memorização de notas geradas no passado para ao fim criar uma coesão musical na totalidade da sequência formada.

Dentre inúmeras pesquisas citadas anteriormente, a implementação PLN que foi utilizada como base maioritária dessa pesquisa foi a pesquisa da PAYNE (2018), aplicou abordagens de PLN tradicionalmente estabelecidas em problemáticas em idiomas em síntese de músicas, codificou notas e acordes presentes em arquivos MIDI em textos e com esse corpus treinou um modelo de linguagem na arquitetura AWD-LSTM empenhado em gerar a próxima nota ou acorde tocado com base o que foi tocado anteriormente. A autora dividiu a pesquisa em duas abordagens, utilizando apenas notas ou acordes, dentre os resultados segundo PAYNE (2018), a abordagem com uso de acordes musicais resultou em amostras

com pouca generalização e também não conseguiu lidar com diferentes durações de notas dentro do acorde, portanto, sendo mais efetivo para criar uma amostra com a característica de ritmo constante. A representação com uso de notas musicais criou amostras mais autorais, criou músicas com coerência rítmica e lidou bem com notas com longa durações PAYNE (2018). Conforme a proposta desse estudo, utilizando o gênero Bossa Nova e com base nos resultados da pesquisa da PAYNE (2018) fora decidido aplicar ambas representações, notas e acordes, em todos os modelos de PLN nesta presente etapa.

O desenvolvimento utilizou de ferramentas de programação *python 3.6* com as bibliografias especificas music21 para manipular e processar as músicas no arquivo *midi* em texto, framework *Keras* para modelagem e treino de redes neurais com o *backend tensorflow.* Um ponto importante para processar texto em modelos inteligentes é a técnica de Tokenização, segundo MENDES (2019), a tokenização consiste em quebrar uma estrutura padrão de textos em documentos, estrofes e frases em símbolos como palavras, pontuação e letras na nova disposição organizado em um vetor para fácil analise computacional. Podemos mapear o método da tokenização na figura 2.1, vemos a conversão de corpus de texto na estrutura vetorial, primeiramente é mapeado um vocabulário indexado de palavras distintas que ocorrem no corpus e resulta no tamanho final do vetor, e logo depois nos documentos ou registros serão lastreados nesse index do vocabulário gerado.

Figura 2.1: Infográfico do mapeamento de símbolos de uma sentença.



Fonte: MENDES (2019).

Com base na modelagem dos dados sugerida, Fora tentado aplicar em uma arquitetura de RNN simples semelhante à figura 1.5, em GRU na figura 1.7 e por fim LSTM na figura 1.6 como foi proposto na metodologia. Segundo MENDES (2019), o modelo RNN não convergiu para um mínimo local do indicie de *Loss* mesmo com alterações na estrutura do modelo, ajuste dos hiperparâmetros ou verificação da base de dados, pois as amostras geradas não estavam nada próximas do que é convencionado como música Deste modo percebemos que a arquitetura RNN não é ideal para a problemática proposta na pesquisa a de criar música, segundo MENDES (2019), equação a da RNN guarda sempre as informações de parâmetros de entradas de inferências passadas (t-1) para influenciar na atual inferência (t) e isso ocasiona a tendência criar repetições de outputs de notas musicais quando apenas duas notas se relacionam no aprendizado adquirido no treinamento. Logo após esse resultado, a arquitetura RNN foi retirada do escopo, pois foi percebido uma necessidade de modelos utilizar o estado oculto de longo prazo para a complexa tarefa de composição de músicas.

Para implementação dos modelos GRU e RNN foram usados a mesma abordagem de PAYNE (2018) com o aproveitamento do código fonte do SKÚLI (2017), conforme no anexo 1 e 2, podemos ver que a estruturas dos modelos são semelhantes e só diferem nas camadas especialistas. Ambas essencialmente conseguiram convergir para mínimos locais quistos, não havendo problemas relacionado ao treinamento, a única observação importante é a redução da base de dados treinamento, de primeiro momento tentou usar a totalidade da massa de dados com 93 músicas no treino e que não foi bem efetivo, pois o modelo não conseguia convergir para um mínimo e além do grande tempo para realização de uma época considerando a utilização de um vocabulário grande no modelo. Com a redução para apenas 15 músicas aleatórias para a base de treinamento, houve a obtenção dos mínimos em ambos os modelos, e uma média de tempo de treinamento de 9 horas com 100 épocas.

Implementação – Visão Computacional (VC)

Visão computacional pode resolver variados problemas e tarefas complexos, e na data de publicação dessa pesquisa esse tipo de abordagem está em seu ápice, segundo KRIEGESKORTE (2015), ao longo dos últimos 8 anos as arquiteturas CNN's reduzidas drasticamente as taxas de erros e chegaram ao ponto de disputar com a performance humana em classificação e detecção de objetos. Embora parecera anormal utilizar técnicas que comumente suportam atividades reconhecimento de objetos e lidam com imagem para a proposta dessa pesquisa, composição de músicas, porém, há trabalhos realizados no mesmo escopo, podemos citar os estudos de HUANG et al. (2017); DONG et al. (2017) e do YANG, CHOU e YI-HSUAN (2017). Um ponto em comum observado entre esses estudos é uso de técnicas de VC, dentre elas uso de GANS e CNN, o trabalho considerado mais relevante para esta pesquisa dentre os citados anteriormente é o produzido pelo DONG et al. (2017) com o modelo nomeado por ele de *Musegan* que realiza síntese de músicas com na arquitetura GANS processando imagens chamadas de *pianorolls*, uma representação visual de um intervalo de uma música.

Figura 2.2: Exemplo de uma visualização 128x128 *Pianoroll* de um trecho da música Águas de Março da composição de Tom Jobim 1972.

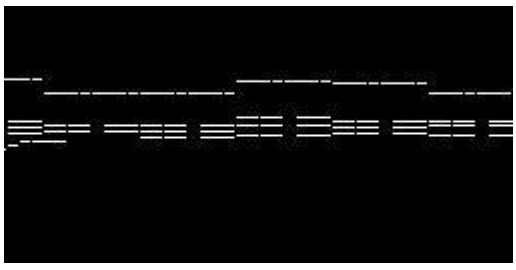
Fonte: MENDES (2019).

Pianoroll representa uma música visualmente em uma matriz esparsa $P^{s \times e}$, o eixo P^{s} corresponde a variavel *step* que é o tempo ou compasso e o eixo P^{e} indica a escala musical

C maior, realizando a coordenada entre esses dois eixos conforme a figura 2.2, consideramos que o valor 1 ou a cor RGB (255,255,255) encontrado representa o acionamento da nota dado ao instante P^s e escala P^e e o valor 0 ou no RGB (0,0,0) denota ausência de som (MENDES, 2019). Podemos indicar que um *pianoroll* é uma fotografia dado um instante de uma música, com essa informação podemos inferir novos dados com obtenção de várias desses recortes de músicas, o *pianoroll* apresentado na figura XX pode apresentar mais uma dimensão caso possua multicanais de instrumentos e terá uma notação $P^{s \times e \times t}$, o P^t pode ter n faixas t que são diferentes ou mesmos instrumentos com interpretação diferentes tocados ao mesmo tempo. Outra pesquisa relevante dentro de VC na problemática dessa pesquisa é o modelo COCONET de HUANG et al. (2017), também se utiliza de *pianorolls* como estrutura de dados aplicado em um modelo convolutivo para geração de harmonias para qualquer música no padrão do compositor Bach, segundo COCONET de HUANG et al. (2017), consiste reconstruir uma nova harmonia em uma *pianoroll* já existente por meio de contraponto de notas com a arquitetura CNN. Um fato importante para o modelo COCONET é sua ampla disponibilização na *homepage* no famoso indexador de conteúdos *Google.com* de todo o mundo na data de aniversário de Bach em 21 de março de 2019.

Outra abordagem realizada para geração de músicas teve na pesquisa de YANG, CHOU e YI-HSUAN (2017) com o modelo MIDINET, consiste em uma arquitetura CNN e GANS com processamento de *pianorolls*, segundo YANG, CHOU e YI-HSUAN (2017), o modelo MIDINET em arquitetura CNN alcançou resultados semelhantes a um modelo RNN com recursividade de informações passadas, comprovando que redes CNN podem lidar com fatores temporais em funções de composição de músicas.

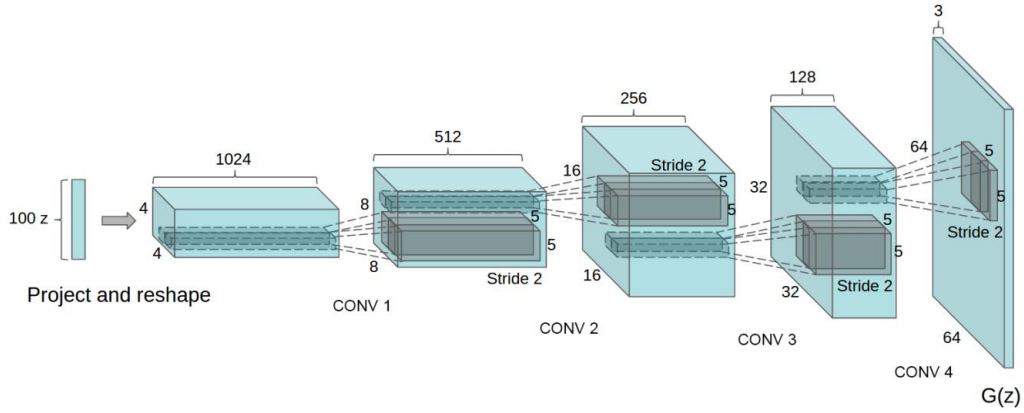
Figura 2.3: Exemplo de uma visualização 128x254 *Pianoroll* de um trecho da música Águas de Março da composição de Tom Jobim 1972.



Fonte: MENDES (2019).

A implementação por parte dessa pesquisa segue a efetividade de geração de músicas por VC observadas e citadas nos três estudos anteriores, será utilizado a mesma estrutura de dados de *pianorolls* e a dimensão dessa matriz foi convencionada a utilização na escala dada pela da figura 2.2 de $P^{128\times128}$ por motivos operacionais, quanto maior a dimensão por parte do eixo P^{S} ocasiona uma menor a quantidade de dados para processar na base e também uma maior o número de parâmetros treinados no modelo em questão, sempre se atentando a limitação computacional do projeto, pois estamos utilizando uma placa gráfica RTX 2070 em um computador pessoal para realizar esses treinamentos. Temos a representação em $P^{254\times128}$ na figura 2.3, uma matriz com eixo *step* maior, assim mais oneroso em atividade de treinamento, no geral foram produzidas 5.154 e 2.525 imagens respectivamente nas escalas $P^{128\times128}$ e $P^{254\times128}$. Como facilitador na geração dessa base de dados foi utilizada e adaptada a biblioteca *pypianoroll* da linguagem *python* elaborada por DONG et al. (2017) para conversão de arquivos *midi* para *pianoroll*.

Figura 2.4: Ilustração de um gerador $G(X)$ DCGAN que o input Z com distribuição dimensional 100 é processado por 4 filtros convulsionais de diferentes dimensões até projetar uma representação em uma imagem gerada 64x64.



Fonte: RADFORD et al. (2015).

Outro fator preponderante para utilização da matriz $P^{128\times128}$ é a arquiteturas GAN, conhecido pela alta demanda computacional na implementação de um modelo com essas características, com a finalidade para otimizar o oneroso processamento de imagens foi indicado no referencial teórico o uso de CNN's, portanto, usaremos um uma derivação do entre os dois modelos, a arquitetura DCGAN, Redes Adversarias com Convolução Profunda,

por RADFORD et al. (2015). Conforme a figura 2.4, temos um representação visual da estrutura da DCGAN na parte do Gerador $G(X)$, segundo RADFORD et al. (2015) a utilização de CNN abriga benefícios em uma rede neural mais estável no processamento e convergência ao aplicar cálculos convulocionais na camada de entrada nos modelos $G(X)$ e $D(X)$.

A execução de um modelo DCGAN precisou de variar os parâmetros comuns estipulados na metodologia, dentre eles foi o aumento do coeficiente de épocas no momento do treinamento, o valor base era 100 e com o decorrer dos resultados e amostras geradas tivemos que dobrar esse valor de épocas para 200 com a finalidade que o modelo conseguisse convergir em um algum mínimo local. Outro problema enfrentado é como entender e avaliar as métricas de treinamento de um modelo GANS, segundo BORJI (2018), explicita que modelos do tipo GANS visam maximizar uma aproximação a distribuição de dados ineridas, utilizando um modelo parametrizado da distribuição real. O ocorrido durante o treinamento que as métricas de treinamento tendiam a minimização do *Loss* para o Discriminador $D(X)$ ou para o Gerador $G(X)$ em certos momentos, segundo MENDES (2019), as métricas do modelo GAN implementado agia semelhante a um cabo de guerra. No campo de estudo de GANS ainda não há um guia ou bibliografia que consiga elucidar a leitura das métricas tanto de treinamento e validação do modelo inteligente, segundo BORJI (2018), há falta de métodos estatísticos ou matemáticos para avaliar arquiteturas GANS. No meio de profissionais ligados na área tem um conhecimento comum que a melhor forma de avaliar um modelo GAN é olhar intrinsecamente os dados gerados, verificar em uma forma subjetiva ou objetiva como as amostras geradas se comportam e comparar com a base original ou o objetivo proposto para o modelo.

Comparação

As saídas da última etapa, Implementação, levantou dados e informações pertinentes. Seguindo a metodologia quali-quantitativa proposta, os resultados irão ser divididos em duas analises, a *Análise de Performance* dos modelos e a *Análise Musical* das amostras, subsequente formando a parte de quantitativa e qualidade.

Análise de Performance

Tabela 1.1. Resultados da etapa *Comparação*, coeficiente *Loss*. *O DCGAN é uma arquitetura que envolve o uso de duas redes distintas, portanto, será informado o *Loss* referente ao $D(x)$ Discriminador e também do $G(x)$ Gerador.

Amostra	*Loss*
LSTM	0.1067
GRU	0.1808
DCGAN*	$D(x)$ 0.1799 / $G(x)$ 4.1790

Fonte: MENDES (2019).

A minimização do coeficiente *Loss* expõem se o sistema inteligente está alcançando os objetivos declarados da problemática, na tabela 1.1 observamos o desempenho das arquiteturas e abordagens propostas na pesquisa e o que conseguiu minimizar melhor o *Loss* foi a arquitetura LSTM, logo em seguida vem o GRU se lidarmos com a abordagem PLN, notando uma diferença de 0,0741 entre os dois modelos, a característica de redes LSTM mudarem mais lentamente o seu estado oculto influenciou em uma melhor convergência do modelo em comparação GRU com uma alteração mais rápida.

Como já discutido na etapa anterior, avaliar eficientemente os coeficientes de *Loss* da arquitetura GANS é um problema que ainda permeia a comunidade acadêmica, em uma avaliação breve como o modelo se comportou podemos notar que o $D(x)$ puxou a

minimização do coeficiente *Loss* na tabela 1.1, que por sua vez, o outro modelo $G(x)$ aumentou o mesmo coeficiente. Uma possibilidade é que a arquitetura por parte do $G(x)$ não conseguiu "enganar" o seu adversário $D(x)$ que tinha uma ampla vantagem seu papel, pois analisava matrizes esparsas como na figura 2.1, sabemos que quase toda a totalidade da imagem *pianoroll* processada é constituída por pixels pertos, nisso o modelo tende em inferir mais ocorrências de pixels pretos RGB (0,0,0) com a certeza de ter uma maior probabilidade de acerto em comparação a inferência no pixel branco RGB (255,255,255) de acionamento da nota musical.

Outro ponto a destacar é o tempo de inferência de cada modelo, as amostras geradas na abordagem PLN em ambas as arquiteturas estudadas precisaram de 127 segundos para gerar 500 notas de output, ou seja, aproximadamente 4 notas geradas por segundo ou 254 milésimos de segundo por inferência, com essa taxa de inferência por tempo podemos facilmente poderia realizar uma reprodução em tempo real de uma música. A síntese de amostras por parte do DCGAN teve por meio de *batch*, configurado no código que a cada época seria gerado 6 amostras de 10 segundos referente nos parâmetros de treinamento da atual época de treinamento. Um problema identificado que o modelo não guarda coesão entre as 6 amostras geradas, sim cada uma sendo independente entre si, assim não podendo criar uma música com vários fragmentos com facilidade.

Avaliação Musical

Foi gerado 4 amostras de cara arquitetura estudada, portanto nessa etapa será verificado um total de 12 amostras e estão compiladas e armazenadas no link https://soundcloud.com/pedro-mendes-116/sets/geracao-de-musicas-polifonicas-por-rede-neurais.

Realizar uma avaliação musical nas amostras geradas levanta certos desafios, um deles que não pode apenas realizar uma análise puramente objetiva e sim que é a que a música possui elementos subjetivos e sujeito a interpretações pessoais. As métricas de performance indicam um desempenho objetivo geral do modelo inteligente, porém, mesmo com boas

métricas esse modelo pode gerar amostras longe do que é definido como música. Outro desafio encontrado é como descobrir se a amostra gerada é similar ou compatível com gênero “Bossa Nova”, para essa tarefa foi empregado a técnica de RMH ou Redução Musical Harmônica, consiste em processar e transformar as amostras na representação RMH e para gerar a comparação utilizamos como parâmetro base a música Águas de Março.

Tabela 1.2. Resultados da análise de Similaridade Harmônica com Redução Harmônica.

Amostras	Similaridade Harmônica
LSTM	0,99998427
GRU	0.99998725
DCGAN	0,99991589

Fonte: MENDES (2019).

Observando a tabela 1.2 vemos o resultado da técnica de RMH e Similaridade Harmônica, os três modelos estudados alcançaram bons índices de similaridade com valores perto de 1 e o modelo GRU alcançou dentre os outros modelos estudados a melhor métrica, porém, é uma diferença bem pequena presente apenas na quinta casa decimal. Podemos analisar ouvindo as amostras, tentando identificar possíveis falhas, ruídos ou repetições da base de treino, segundo MENDES (2019), incidência de repetições e copias dentro de uma amostra gerada pode indicar o fenômeno de *overfitting*, ou um modelo super ajustado a base de treino, ocorrendo no modelo. No modelo LSTM detectamos a ocorrência de copias de composições da base de treino, conseguindo praticamente mimetizar a base com uma boa qualidade, isso é provável de ocorrer quando utilizamos uma base de dados reduzida em 15 músicas, um ponto interessante seria tentar trabalhar o *sampling output*, ou seja, modificar método de escolha do output do modelo para atingir generalização das inferências ou outra opção seria tentar treinar com mais dados. A arquitetura GRU apresentou amostras bem diferentes do LSTM, focado mais em bases harmônicas, onde que na LSTM apresentava a criação de melodia e improvisações. A DCGAN segundo MENDES (2019), possui uma desvantagem em amostras em consideração com as abordagens PLN, sendo uma amostra GANS possui apenas 10 segundos e cada amostra é única. O modelo de VC gerou amostras ainda perto de primitivas, possuindo dissonância em

algumas partes, porém, na amostra 4 conseguiu criar um trecho com harmonia, assim possuindo ainda um potencial caso tivesse mais tempo de treino ou aumento das dimensões do *pianoroll* utilizado.

Em meio os modelos analisados, o que mostrou um potencial que poderia ser explorado foi o LSTM, ao lidar com o problema de *overfitting*, seria possível gerar amostras com melhor qualidade e seria no fim o modelo ideal para seguir na metodologia proposta e aprimorá-lo.

Aprimoramento

A última etapa da metodologia visar ajustar problemas encontrados no modelo LSTM, como discutido houve a presença de overffiting no modelo ao gerar as amostras com copias da base de dados. A primeira possível explicação para a ocorrência desse problema é o método escolhido para o *sampling output* ou *Search Decoder*, nas abordagens de PLN comumente é aplicado nos modelos um *Decoder*, segundo PHY (2020), ao fim do modelo inteligente é aplicado uma função n de *softmax* para converter a inferência da rede em um vetor de probabilidades para a provável ocorrência de cada símbolo ou palavra presente no vocábulo do modelo.

$$P(x_i|x_{1:i-1}) = \frac{\exp(u_i)}{\sum_j \exp(u_j)} (n)$$

Na fase de implementação de PLN foi utilizado em ambos os modelos o método *Greedy*, traduzindo para o português Ganancioso, que é o mais comum de implementar. O *Greedy Search* realiza a seleção da saída do modelo conforme na maior probabilidade encontrada no vetor output da inferência, conforme a função *o*.

$$\max(\vec{P}) \ (o)$$

A Função *Greedy Search* acima retorna o maior valor presente no vetor de probabilidade $\vec{P}$.

Já que o modelo sempre vai tender ao máximo de um vetor de probabilidade no método *Greedy Search*, a amostra pode cair em repetições e tende a generalizar menos. Existem variados métodos de *sampling output* que podem ser implantados, podemos citar o *Beam Search*, *Nucleus Sampling, Random Sampling*, *Temperatura* e o *Top-K Sampling* e abordar e aplicar cada um desses métodos seria uma nova pesquisa aparte, portanto fora do escopo escolhido na metodologia e das motivações do estudo. Podemos selecionar uma das abordagens mencionadas anteriormente, dentre elas temos a Calculo de Temperatura, primeira consiste em criar uma tendência de predição no *softmax* do modelo por meio do coeficiente T conforme na equação p, esse coeficiente idealmente pode ser declarado de 0 á 1.

$$P(x_i|x_{1:i-1}) = \frac{\exp(u_i/T)}{\sum_j \exp(u_j/T)}(p)$$

Aplicando Temperatura em modelo de linguagem que cria textos em inglês a partir de obras literárias, uma temperatura baixa no coeficiente T resulta em termos extremamente repetidos e um texto previsível, porém, com uma estrutura bem condizente, com a temperatura alta T o texto torna-se mais interessante, surpreendente e até mesmo criativo (ALLAIRE, 2018). Para introdução do coeficiente de temperatura no modelo, foi necessário apenas acrescentar uma nova linha de código no modelo LSTM conforme o APENDICE X e precisou ser treinado novamente para cada coeficiente de temperatura. A implementação desse *Decoder* ofereceu melhoria no âmbito de performance, conseguindo reduzir o *Loss* obtido no LSTM da etapa anterior de Implementação conforme na tabela 1.3 e na figura 2.5, todas as implementações de temperatura conseguiram superar o valor de *Loss* base LSTM, a maior diferença encontrada foi de 0,0785 com T igual a 0.9.

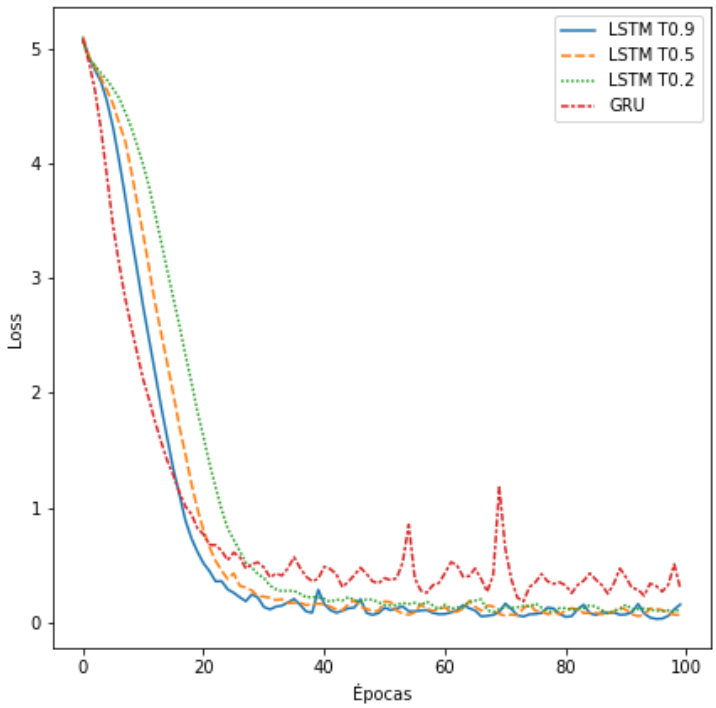
Tabela 1.3: Resultados dos treinos com a adição do coeficiente de Temperatura.

Amostra	*Loss*
LSTM (Base)	0.1067
LSTM T0.9	0.0282
LSTM T0.5	0.0526
LSTM T0.2	0.0749
GRU (Base)	0.1808

Fonte: Própria (2020).

Não houve alteração no tempo de uma época de treinamento no modelo com a implementação do coeficiente de Temperatura nos modelos e também podemos observar na figura 2.5 uma melhor eficiência em comparação com o algoritmo GRU.

Figura **2.5**: Gráfico do *Loss* ao decorrer do treinamento de 100 Épocas e com modelos LSTM com o coeficiente de Temperatura.



Fonte: Própria (2020).

Na parte de avaliação musical, as amostras geradas com o coeficiente de Temperatura trouxeram resultados satisfatórios e que estão também armazenadas no link https://soundcloud.com/pedro-mendes-116/sets/geracao-de-musicas-polifonicas-por-rede-neurais. O modelo T0.9 apresenta uma maior ocorrência de *overfitting*, apresentando cópia da base de dados em todas as amostras geradas por este modelo, e o modelo T0.5 foi que mostrou uma melhor performance com apenas uma das amostras demostrava a ocorrência de *overfitting* em um trecho e as outras conseguiram generalizar e lidar com harmonia e melodia. O último modelo T0.2 teve ocorrência de overfitting em pequenas passagens em todas as amostras e também apresentou trechos desconexos ou dissonantes, foi o modelo

que mais saiu mal no quesito de analise musical. Percebemos ainda a presença de *overfitting* mesmo após o aprimoramento com um novo método de Decoder, provavelmente o modelo LSTM T0.5 mantenha resquícios de *overfitting* por causa da base de dados pequena e para superá-lo seria necessário adquirir maior poder de processamento por meio de serviços escaláveis *cloud* ou a compra de hardware de processamento gráfico de ultima geração.

CONSIDERAÇÕES FINAIS

O principal objetivo da pesquisa é alcançar por parâmetros quali-quantitativos a definição de um modelo inteligente por DP que consiga melhor compor músicas do gênero Bossa Nova sem a intervenção direta do ser humano e essa escolha convergiu para o modelo LSTM, pois sua capacidade de memória curta e longa conseguiu alcançar uma modelagem efetiva no problema de composição músicas com boas métricas de treinamento e também na qualidade sonora das amostras produzidas. Esse modelo passou por uma etapa de aprimoramento para suavizar os problemas identificados, dentre eles o *overfitting*, com o uso de técnica de *Output Decoder* de coeficiente de temperatura foi alcançado amenização desse feito de super ajustamento da base de treinamento e permitir o modelo generalizar e ter mais composições autorais. Um dos problemas encontrados durante a pesquisa foi a capacidade computacional disponível, que era um computador pessoal com uma placa de processamento gráfico (GPU) RTX 2070 com 8 Gb de memória RAM. O uso de DP requisita um grande esforço computacional para treinamento dos pesos em uma rede neural e esse fato pode ter impactado os resultados obtidos desse estudo, dentre eles podemos citar a base de treino reduzida de PLN e as dimensões das *pianorolls* processadas no modelo DCGAN, sendo assim, seria necessário para em pesquisas futuras realizar a escalabilidade do poder de processamento computacional das implementações realizadas neste projeto.

Proposta de Trabalhos Futuros

A pesquisa conversou com a área de inteligência artificial, que atualmente avança passos largos com o surgimento de tecnologias e técnicas revolucionarias dentro da academia e indústria, portanto, é passível de ocorrer que alguma delas fique de fora do escopo da pesquisa e dentre elas podemos citar o atual estado da arte em PLN no momento de finalização dessa pesquisa que é os modelos com arquitetura *Encoder-Decoder*. Essa arquitetura é reconhecida mundialmente pelo modelo GPT-3 de geração de textos da empresa *OpenAI* e que poderia ser uma grande adição para ampliar o escopo de novas arquiteturas abordadas para uma pesquisa futura.

Uma lacuna observada durante o desenvolvimento do são as possiblidades estudo dos algoritmos de *Output Decoder* no desafio de composição de músicas, uma pesquisa futura focada nesse aspecto permitiria uma analise quali-quantitativa dos seguintes métodos: *Beam Search*, *Nucleus Sampling, Random Sampling*, *Temperatura, Greedy Sampling* e o *Top-K Sampling*.

Uma técnica que pode auxiliar nos futuros desenvolvimentos de modelos de composição de música é o uso de *transfering learning*, consiste em reaproveitar pesos de um modelo treinado já existente para auxiliar no treinamento de um novo modelo com menor esforço computacional e período de tempo, semelhante aos modelos RESNET de VC. Uma boa proposta seria criar um modelo genérico ou especifico de algum gênero musical com a finalidade de importação de pesos que seria vantajoso para qualquer pesquisa futura.

Por fim, esbarrou-se durante execução dessa pesquisa no desafio de como poder avaliar um modelo de DP em um problema de composição de músicas, não existe uma métrica ou método relativo a isso. Podemos citar um o exemplo próximo que é a métrica *BLEU score* utilizada para avaliar sentenças traduzidas entre idiomas por um modelo de inteligência computacional, portanto, esse conceito pode ser aproveitado para uma nova métrica focada em avaliar composições de músicas por modelos inteligentes.

REFERÊNCIAS

ALLAIRE, Joseph J. **Text generation with LSTM.** Sitio jjallaire.github.io, 2018. Disponível em:<https://jjallaire.github.io/deep-learning-with-r-notebooks/notebooks/8.1-text-generation-with-lstm.nb.html>. Acesso em outubro de 2020.

ALBAWI, Saad e MOHAMMED, Tareq Abed. **Understanding of a Convolutional Neural Network**. Turquia, Antalya, 2017.

ARANO, Gildo. **Os quatro passos para dominar os intervalos.** Blog Segredos do Fole, 2014. Disponível em: <http://segredosdofole.blogspot.com/2014/03/os-4-passos-para-dominar-os-intervalos.html>. Acesso em outubro de 2020.

BORJI, Ali. **Pros and Cons of GAN Evaluation Measures**. Revista Computer Vision and Image Understanding, Volume 179, Páginas 41-65, 2018.

BRITZ, Denny**. Recurrent Neural Network Tutorial, Part 4 – Implementing a GRU/LSTM RNN with Python and Theano.** Sitio WildML, 2015. Disponível em: <http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/>. Acesso em outubro de 2020.

BULLINARIA, John A. **Recurrent Neural Networks Neural Computation Lecture 12**. Escola de Ciência da computação, Universidade de Birmingham, Reino Unido, 2015.

DONG, Hao-Wen; HSIAO, Wen-Yi; YANG, Li-Chia e YI-HSUAN, Yang. **MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment**. Centro de pesquisa da tecnologia da informação e inovação, Academia Sinica, Taipei, Taiwan, novembro, 2017.

DRAKOS, Georgios. **What is a Recurrent Neural Networks (RNNS) and Gated Recurrent Unit (GRUS)**. Sitio towards data Science, 2019. Disponível em: < https://towardsdatascience.com/what-is-a-recurrent-nns-and-gated-recurrent-unit-grus-ea71d2a05a69>. Acesso em novembro de 2019.

COCA, Andrés E.; Romero, Roseli A. F. e Zhao, Liang. **Generation of composed musical structures through recurrent neural networks based on chaotic inspiration**. International Joint Conference on Neural Networks (IJCNN), San Jose, Califórnia, EUA, de 31 julho à 5 de agosto, 2011.

CORRÊA, Débora Cristina. **Sistema baseado em redes neurais para composição musical assistida por computador**. Universidade Federal de São Carlos, 2008.

CHEN, Chun-Chi J. e MIIKKULAINEN, Risto. **Creating Melodies with Evolving Recurrent Neural Networks** 2001 International Joint Conference on Neural Networks, Washington, DC, 2001.

CHO, Kyunghyun; VAN MERRIËNBOER, Bart; GULCEHRE, Caglar; BOUGARES, Fethi; SCHWENK, Holger; BAHDANAU, Dzmitry e BENGIO, Yoshua. **Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation**. 2014.

CULTURA MIX (2018). **O Que é Compasso Composto na Teoria Musical?** Sitio Cultura Mix, 2018. Disponível em: <https://musica.culturamix.com/curiosidades/o-que-e-compasso-composto-na-teoria-musical>. Acesso em outubro de 2020.

FERNÁNDEZ, Jose David e VICO, Francisco. **AI Methods in Algorithmic Composition: A Comprehensive Survey.** Fundação AI Access, Journal of Artificial Intelligence Research número 48, 2013.

FIGUEIREDO, João Luiz de e ARAÚJO, Lara Muniz. **A Rede Carioca de Rodas de Samba e a produção independente de música.** Revista Conhecimento & Diversidade, Niterói, volume 11, número 24, p. 73 – 90, maio/agosto, 2019.

FONSECA, Eder. **"A Bossa Nova não tem idade" Toquinho – Cantor, compositor e violonista.** Sitio Panorama Mercantil, 2016. Disponível em: <http://www.panoramamercantil.com.br/a-bossa-nova-nao-tem-idade-toquinho-cantor-compositor-e-violonista/>. Acesso em outubro de 2020.

GOODFELLOW, Ian J.; POUGET-ABADIE, Jean; MIRZA, Mehdi; XU, Bing; WARDE-FARLEY, David; OZAIR, Sherjil; COURVILLE, Aaron e BENGIO, Yoshua. **Generative Adversarial Nets**. Département d'informatique et de recherche opérationnelle, Universidade de Montreal, Montreal, Canadá, 2014.

HOCHREITER, Sepp e SCHMIDHUBER, Jurgen. **Long Short-Term Memory**. Neural Computation, dezembro de 1997.

HERREMANS, D. e CHUAN, C. **Modeling Musical Context Using Word2vec**. Primeiro Workshop Internacional de Deep Learning para Música, Anchorage EUA, maio 2017

HUANG, Cheng-Zhi Anna; COOIJMANS, Tim; ROBERTS, Adam; COURVILLE, Aaron e ECK, Douglas. M. **Counterpoint by Convolution.** 18th International Society for Music Information Retrieval Conference, Suzhou, China, 2017.

International Federation of the Phonographic Industry IFPI. **IFPI Global Music Report 2018.** Londres, Reino Unido, 2018.

KIEFER, Bruno. **Elementos da linguagem musical.** Porto Alegre: Movimento/INL/MEC, 1973.

KARPATHY, Andrej e JOHNSON, Justin. **Convolutional Neural Networks (CNNs / ConvNets)** . Universidade de Stanford, Ciência da Computação turma CS231n, 2018. Disponível em: < http://cs231n.github.io/convolutional-networks/>. Acesso em outubro de 2020.

KRIEGESKORTE, Nikolaus. **Deep neural networks: a new framework for modelling biological vision and brain information processing**. Medical Research Council Cognition and Brain Sciences Unit, Cambridge, Reino Unido, 2015.

KULESZ, Octavio. **Culture, platforms and machines: the impact of artificial intelligence on the diversity of cultural expressions**. Intergovernmental Committee for the Protection and Promotion of the Diversity of Cultural Expressions, UNESCO Headquarters, Paris, França, dezembro de 2018.

LOUREIRO, Maurício A.; PAULA, Hugo B. de. **Timbre de um instrumento musical**. Per Musi, Belo Horizonte, n.14, 2006, p.57-81

LUIZ, Douglas Marques e NASCIMENTO, Luciana Marino. **Minha terra tem palmeiras, imagens do Brasil na bossa nova.** DARANDINA revista eletrônica, Programa de Pós-Graduação em Letras UFJF,volume 4, número 1, 2011.

MARQUES, Douglas L. e MARINO, Luciana N. **Minha terra tem palmeiras Imagens do Brasil na bossa nova.** Programa de Pós-Graduação em Letras Universidade Federal de Juiz de Fora, Juiz de Fora, Rio de Janeiro, Brasil, 2011.

MENDES, P. H. R. **Analise Quali-Quantitativa de Arquiteturas de Redes Neurais Artificiais para Geração de Música Polifônica**. Universidade Centro de Ensino Unificado de Brasília (UniCEUB), 2019.

MICROSOFT. **Embarrassingly Parallel Image Classification, Using Cognitive Toolkit and TensorFlow on Azure HDInsight Spark**. Sitio Microsoft Machine Learning Blog, 2017. Disponível em: <https://blogs.technet.microsoft.com/machinelearning/2017/04/12/embarrassingly-parallel-image-classification-using-cognitive-toolkit-tensorflow-on-azure-hdinsight-spark/>. Acesso em outubro de 2020.

MIKOLOV A, Tomas; CHEN, Kai; CORRADO, Greg e DEAN, Jeffrey. **Efficient Estimation of Word Representations in Vector Space**. Universidade de Cornell, EUA, setembro de 2013.

MIKOLOV B, Tomas; SUTSKEVER, Ilya; CHEN, Kai; CORRADO, Greg e DEAN, Jeffrey. **Distributed Representations of Words and Phrases and their Compositionality**. Universidade de Cornell, EUA, dezembro de 2013.

NAYEBI, Aran e VITELLI, Matt. GRUV. **Algorithmic Music Generation using Recurrent Neural Networks**. Departamento de Ciência da computação, Universidade de Stanford, Califórnia, EUA, 2015

NG, Andrew; NGIAM, Jiquan; FOO, Chuan Yu; MAI, Yifan; SUEN, Caroline; COATES, Adam; MAAS, Andrew; HANNUN, Awni; HUVAL, Brody; WANG, Tao e TANDON, Sameep. **Multi-Layer Neural Network.** Sitio Unsupervised Feature Learning and Deep Learning. Disponível em: <http://deeplearning.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/>. Acesso em outubro de 2020.

OORD, Aaron van den; KALCHBRENNER, Nal; VINYALS, Oriol; ESPEHOLT, Lasse; GRAVES, Alex e KAVUKCUOGLU, Koray. **Conditional Image Generation with PixelCNN Decoders**. Universidade de Cornell, EUA, junho de 2016.

OLAH, Christopher. **Understanding LSTMs – Blog colah's blog, 2015**. Disponível em: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Acesso em outubro de 2020.

PwC PricewaterhouseCoopers (2018). **The macroeconomic impact of artificial intelligence**. Londres, Reino Unido, fevereiro de 2018.

PAYNE, Christine. **Clara: Generating Polyphonic and Multi-Instrument Music Using an AWD-LSTM Architecture**. OpenAI Scholars Program, 2018.

PONTI, Moacir A. e COSTA, Gabriel B. Paranhos da. **Como funciona o Deep Learning**. Simpósio Brasileiro de Computação (SBC), Tópicos em Gerenciamento de Dados e Informações, 2017.

RADFORD, Alec; METZ, Luke e CHINTALA, Soumith. **Unsupervised representation learning with deep convolutional generative adversarial networks**. Universidade de Cornell, EUA, novembro, 2015

ROCCA, Baptiste. **Understanding Generative Adversarial Networks (Gans)**. Sitio towards data Science, 2019. Disponível em: <https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29>. Acesso em outubro de 2020.

SCHMIDT, Ben. **A zoomable map of language from news articles** – Sitio ben schmidt, 2017. Disponível em: <http://benschmidt.org/word2vec_map/>. Acesso em outubro de 2020.

SILVA, Paulo Roberto Pereira da. **Batucada: Um Lego Rítmico.** Centro de Ciências Exatas e da Natureza (CCEN), Departamento de Informática (DI), Universidade federal de Pernambuco UFPE, 1999.

SKÚLI, Sigurður. How to Generate Music using a LSTM Neural Network in Keras. Sitio *towards data Science*, 2017. Disponível em:<https://towardsdatascience.com/how-to-generate-music-using-a-lstm-neural-network-in-keras-68786834d4c5> Acesso em outubro de 2020.

SHOHAM, Yoav; PERRAULT, Raymond; BRYNJOLFSSON, Erik; CLARK, Jack; MANYIKA, James; NIEBLES, Juan Carlos; LYONS, Terah; ETCHEMENDY, John; GROSZ, Barbara e BAUER, Zoe. **The AI Index 2018 Annual Report.** AI Index Steering Committee, Human-Centered AI Initiative, Universidade de Stanford, Stanford, Califórnia, EUA, dezembro de 2018.

SOMPOLINSKY, Haim. **Introduction: The Perceptron.** Instituto de Tecnologia e Massachusetts, Cambridge, Massachusetts, EUA, outubro de 2013.

TODD, Peter. M. **A connectionist approach to algorithmic composition.** Computer Music Journal, volume 13 N°4, páginas 27–43, 1989.

UNIÃO EUROPEIA. **Music Moves Europe – A European Music Export Strategy, Final Report.** Escritório de Publicações da União Europeia, Luxemburgo, 2019.

WALZER, Daniel A. **Independent Music Production: How Individuality, Technology, and Creative Entrepreneurship Influence Contemporary Music Industry Practices.** Creative Industries Journal, novembro 2016.

YANG, Li-Chia; CHOU, Szu-Yu e YI-HSUAN, Yang. **MIDINET: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation.** 18° International Society for Music Information Retrieval Conference, Suzhou, China, 2017.

YÜKSEL, Ali Çağatay; KARCI, Mehmet Melih e UYAR, A. Şima. **Automatic Music Generation Using Evolutionary Algorithms and Neural Networks.** Faculty of Computers and Informatics, Istanbul Technical University, Istambul, Turquia, 2011.