

**UNIVERSIDADE FEDERAL FLUMINENSE  
ESCOLA DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO *STRICTO SENSU* EM ENGENHARIA DE  
PRODUÇÃO  
CURSO DE MESTRADO EM ENGENHARIA DE PRODUÇÃO**

**CARLOS HENRIQUE TARJANO SANTOS**

**REDES NEURAIS APLICADAS À MODELAGEM DE INSTRUMENTOS  
ACÚSTICOS PARA SÍNTESE SONORA EM TEMPO REAL**

Dissertação apresentada ao Programa de Pós-Graduação  
*Stricto Sensu* em Engenharia de Produção da  
Universidade Federal Fluminense como requisito parcial  
para obtenção do Grau de Mestre em Engenharia de  
Produção.

**Professor Orientador:**

**Valdecy Pereira, D.Sc.**

Niterói  
2018

Ficha catalográfica automática - SDC/BEE

S237r	Santos, Carlos Henrique Tarjano Redes Neurais Aplicadas à Modelagem de Instrumentos Acústicos para Síntese Sonora em Tempo Real / Carlos Henrique Tarjano Santos ; Valdecy Pereira Pereira, orientador. Niterói, 2018. 88 f. : il.
	Dissertação (mestrado)-Universidade Federal Fluminense, Niterói, 2018.
	DOI: <a href="http://dx.doi.org/10.22409/TPP.2018.m.11839251743">http://dx.doi.org/10.22409/TPP.2018.m.11839251743</a>
	1. Rede neural artificial. 2. Acústica. 3. Inteligência artificial. 4. Produção intelectual. I. Título II. Pereira,Valdecy Pereira, orientador. III. Universidade Federal Fluminense. Escola de Engenharia.
	CDD -

Bibliotecária responsável: Fabiana Menezes Santos da Silva - CRB7/5274

CARLOS HENRIQUE TARJANO SANTOS

**REDES NEURAIS APLICADAS À MODELAGEM DE INSTRUMENTOS ACÚSTICOS  
PARA SÍNTESE SONORA EM TEMPO REAL**

Dissertação apresentada ao Curso de Mestrado em Engenharia de Produção da Universidade Federal Fluminense como requisito parcial para obtenção do Grau de Mestre em Engenharia de Produção.

Aprovada em 31 de Tinhas de 2018.

BANCA EXAMINADORA

Professor Orientador: Valdecy Pereira, D.Sc.  
Universidade Federal Fluminense

Aline Paes, D.Sc.  
Universidade Federal Fluminense

Leonardo Antonio Monteiro Pessôa, D.Sc.  
Marinha do Brasil

*Sertão é onde manda quem é forte, com as astúcias.  
Deus, mesmo, quando vier, que venha armado.*

Guimarães Rosa

## RESUMO

O presente trabalho desenvolve estratégias de aplicação de redes neurais artificiais ao campo de modelagem acústica, tanto no domínio do tempo quanto no domínio da frequência, com foco na síntese sonora em tempo real de instrumentos musicais. Para essa finalidade, o estado da arte da pesquisa relacionada a redes neurais e suas aplicações é investigado, através de uma revisão bibliográfica, que levanta também as principais técnicas e algoritmos utilizados na emulação de instrumentos musicais em tempo real. Duas propostas de modelos completos para síntese sonora são apresentadas e avaliadas, a partir do uso de redes neurais aplicadas à modelagem espectral. A comparação de um dos modelos com implementações apresentadas para os dois algoritmos mais utilizados na modelagem física indica que a aplicação das redes neurais na área de áudio tem o potencial de aumentar a verossimilhança das simulações, reduzindo a carga computacional necessária. Do ponto de vista do desenvolvimento de produtos, o trabalho apresenta, a partir dos métodos espetrais desenvolvidos, uma inovação tecnológica capaz de basear uma linha de instrumentos virtuais comerciais.

**Palavras-chave:** Redes Neurais, Modelagem Acústica, Síntese Sonora em Tempo Real

## **ABSTRACT**

The present work develops strategies for the application of artificial neural networks to the field of acoustic modeling, both in the time and frequency domains, focusing on the synthesis of musical instruments in real time. To this end, the state of the art of the research related to neural networks and their applications is investigated through a bibliographic review, which also identifies the main techniques and algorithms used in the real time emulation of musical instruments. Two complete models are introduced and evaluated, based on the theory of neural networks applied to spectral modeling. The comparison of one of the models with implementations presented for the two most used algorithms in physical modeling indicates that the use of the neural networks in the audio area has the potential to increase the likelihood of the simulations, while reducing the computational load required. From the point of view of product development, the work presents, based on the spectral methods developed, a technological innovation that could be the basis for a line of commercial virtual instruments.

**Key-words:** Neural Networks, Acoustic Modeling, Real-time Sound Synthesis

## SUMÁRIO

1	Introdução .....	9
1.1	Objetivo .....	10
1.2	Escopo.....	11
1.3	Estrutura do Trabalho e Síntese das Etapas de Pesquisa .....	11
1.4	Relevância socio-econômica.....	12
2	Revisão Bibliográfica .....	14
2.1	Redes Neurais Artificiais .....	14
2.2	Redes Neurais - Principais Arquiteturas .....	16
2.2.1	Redes Densas .....	17
2.2.2	Redes Neurais Profundas.....	17
2.2.3	Redes Recorrentes .....	18
2.2.4	Redes Convolucionais .....	18
2.3	Redes Neurais Aplicadas à Musica.....	20
2.4	Redes Neurais Aplicadas ao Discurso Falado .....	22
2.5	Redes Neurais Aplicadas à Imagem .....	23
2.6	Redes Neurais - Ferramentas e Frameworks .....	24
2.7	Síntese Sonora em Tempo Real .....	25
2.7.1	Método das Diferenças Finitas .....	26
2.7.2	Digital Waveguides .....	30
2.7.3	Digital Waveguides X Diferenças Finitas .....	33
3	Referencial Teórico .....	35
3.1	Transformada Discreta de Fourier .....	35
3.2	A Equação da Onda .....	37
3.3	Formalização de uma Rede Neural - Forward Pass .....	40
3.4	Formalização de uma Rede Neural - Backpropagation .....	41
4	Metodologia.....	44
4.1	Amostras Sonoras Utilizadas .....	44
4.2	Redes Densas no Domínio do tempo .....	45
4.2.1	Algoritmos de Otimização.....	47
4.2.2	Busca em grade para topologia e parâmetros .....	48
4.2.2.1	Adagrad .....	48
4.2.2.2	Nadam .....	50
4.2.2.3	Ativações e Otimizadores.....	51
4.2.3	Análise dos resultados .....	52

4.3	Redes Recorrentes.....	54
4.4	Domínio da Frequência.....	58
4.5	Modelo Misto.....	62
4.5.1	Frequências.....	64
4.5.2	Decaimentos e Amplitudes.....	65
4.5.3	Formulação Final.....	66
5	Resultados.....	71
6	Conclusão .....	73
6.1	Desenvolvimentos Futuros .....	74
7	Referências .....	76
	GLOSSÁRIO.....	86

## 1 INTRODUÇÃO

A área denominada inteligência artificial é um dos campos da ciência mais explorados na atualidade, estando presente em uma gama crescente de aplicações que vão desde o entretenimento (Risi e Togelius, 2017) à saúde (Esteva *et al.*, 2017), passando por campos como segurança digital (Aljumah e Ahamad, 2016; Omar, Johari e Smith, 2017), entre inúmeros outros.

Um relatório publicado recentemente pelo *Center for a New American Security* (CNAS) sugere que a inteligência artificial está no âmago de uma nova revolução industrial, com o potencial de introduzir mudanças tão disruptivas quanto as causadas pelas revoluções anteriores (Horowitz *et al.*, 2018).

Isso ajuda a entender a relevância da área no modelo de negócios de algumas das maiores empresas do mundo, como Google, Amazon, Apple e Facebook. Na contramão desse movimento, no entanto, observa-se uma escassa penetração dessas tecnologias na indústria de instrumentos virtuais, acompanhado de um baixo volume de pesquisas na área: Um levantamento bibliográfico na base Scopus revela que trabalhos relacionados à utilização de redes neurais para a simulação de instrumentos musicais ainda são bastante escassos, a despeito do sucesso desta ferramenta em iniciativas afins.

O potencial de inovação nesse campo é sugerido pelo papel central (muitas vezes representando o estado da arte) que as redes neurais artificiais vêm desempenhando em áreas diretamente correlatas, com o campo de síntese de voz, ou ainda em campos menos obviamente relacionados, à exemplo dos grandes avanços na área de visão computacional (*computer vision*), como a geração de imagens e vídeos.

Esses resultados estimulam a transposição de algumas dessas inovações para o contexto da síntese sonora, além do desenvolvimento de novas abordagens. Nesse contexto, o presente trabalho investiga, a partir de diferentes perspectivas, o potencial de aplicação dos recentes desenvolvimentos na área de redes neurais à modelagem de instrumentos acústicos, com vistas à síntese sonora em tempo real.

Ênfase é dada a instrumentos de caráter percussivo, no sentido de sistemas em que o som é gerado por uma excitação inicial, aproximadamente impulsiva do ponto de vista físico, e a subsequente vibração livre do(s) componentes pertinentes do instrumento. Esse é o caso, por exemplo, do conjunto de peças de uma bateria ao serem golpeados, das cordas ou conjuntos de cordas de um piano acionadas pelo martelo ao ser alavancado pelo pressionar

das teclas e, em uma aproximação mais livre, a excitação causada pelos dedos ou palheta em instrumentos de corda.

Do ponto de vista da indústria, embora o interesse por instrumentos musicais digitais tenha crescido bastante na última década (Staudt, 2016), os instrumentos virtuais de ponta, utilizados em estúdios para gravações profissionais ainda baseiam-se prioritariamente em coleções de amostras sonoras pré-gravadas (Smith, 2008), demandando uma alta quantidade de memória em disco e razoável poder de processamento do *hardware* (em geral computadores) utilizado.

Em plataformas onde o poder de processamento e armazenamento são limitados, como é o caso de teclados digitais e baterias eletrônicas, por exemplo, é comum que esses instrumentos sejam garnecidos com bibliotecas de menor tamanho e qualidade, suficientes apenas para a prática do instrumento; a qualidade de produção é alcançada apenas conectando-se o instrumento à um computador, e usando-o como controlador de uma implementação em software, geralmente no formato VST (de forma a acessar técnicas de manipulação e coleções de amostras sonoras de melhor qualidade).

O foco em algoritmos neurais voltados à síntese em tempo real tem origem tanto na escassez de pesquisas sobre o tema quanto nas possibilidades mercadológicas: aumentos de eficiência e redução do tamanho em memória ocupado pelo modelo podem, por exemplo, basear implementações mais verossímeis e baratas nessas plataformas dedicadas.

## 1.1 OBJETIVO

O objetivo geral deste trabalho é desenvolver um modelo de simulação em tempo real de instrumentos acústicos que tire proveito do estado da arte relacionado a redes neurais artificiais para exibir maior eficiência, quando comparado aos algoritmos tradicionalmente utilizados para síntese sonora em produtos comerciais, dando origem a simulações mais realistas, principalmente do ponto de vista da percepção humana, e menos intensivas computacionalmente.

Esse objetivo pode ser desmembrado em objetivos específicos:

- Identificar formas compactas de representação das ondas sonoras mais adequadas à predição/generalização via redes neurais;
- Identificar arquiteturas neurais que possam ser utilizadas na modelagem acústica;
- Identificar na literatura sobre modelos acústicos de instrumentos musicais ferramentas que possam ser utilizadas para simplificação e aumento de eficiência, tanto do modelo quanto das representações;

- Tendo em vista o foco na síntese em tempo real, identificar as arquiteturas e hiperparâmetros neurais e técnicas em geral mais eficientes;
- Delinear a capacidade de generalização do modelo criado.

## 1.2 ESCOPO

O trabalho limita-se à emulação de instrumentos musicais acústicos convencionais e não aborda, senão tangencialmente, a área mais subjetiva de desenho de novos instrumentos, através de um exemplo em que um instrumento híbrido é apresentado; um tratamento razoavelmente completo do tema, pelo grande número de possibilidades que oferece (Dalgleish, Spencer e Foster, 2014), está além do objetivo deste trabalho.

Adicionalmente, limitar o escopo do trabalho à investigação de instrumentos que possam ser aproximados por um modelo de excitação impulsivo, como aqui é feito, possibilita excluir da observação uma subárea significativa do processamento de sinais que lida com a evolução das frequências de uma onda sonora no domínio do tempo, trazendo o volume de pesquisa a níveis práticos, sem uma perda substancial em termos conceituais.

Uma outra vantagem desse recorte é de cunho técnico, já que essa escolha reduz a duração média das ondas investigadas; investigações do comportamento de arquiteturas neurais densas e profundas, por exemplo, tornam-se computacionalmente custosas quando uma dimensionalidade muito grande é envolvida. Esse também é o comportamento de alguns dos algoritmos utilizados durante o trabalho, como a Transformada Discreta de Fourier.

Em relação ao tratamento das redes neurais, prioridade foi dada às arquiteturas já consolidadas na literatura, o que excluiu, por exemplo, uma investigação da promissora arquitetura Capsnet (Hinton, Sabour e Frosst, 2018; Sabour, Frosst e Hinton, 2017). A escolha é necessária devido a limitações de tempo frente ao grande volume de pesquisas envolvendo novas arquiteturas.

## 1.3 ESTRUTURA DO TRABALHO E SÍNTSE DA ETAPAS DE PESQUISA

Durante a revisão bibliográfica, após um breve resumo da evolução das redes neurais do ponto de vista histórico, o trabalho apresenta as arquiteturas mais relevantes encontradas na literatura. O objetivo é introduzir o tema, inicialmente de maneira ampla, lançando as bases para as seções seguintes, que exploram o estado da arte das aplicações dessas arquiteturas em 3 áreas de interesse: música, discurso e imagens. Enquanto o interesse nas duas primeiras áreas é mais evidente, uma investigação dos desenvolvimentos em visão computacional tem o potencial de oferecer inspiração em relação a abordagens; além de ser uma das áreas do

aprendizado de máquina mais ativamente pesquisadas, alguns esforços de transposição dos desenvolvimentos nessa área para o campo da modelagem acústica têm alcançado resultados interessantes.

A seguir, algumas plataformas para a implementação de redes neurais são discutidas e comparadas; ênfase foi dada aos *frameworks* com maior penetração tanto na indústria quanto na academia e a escolha da plataforma utilizada nesse trabalho obedeceu mais a critérios de documentação, adoção na comunidade e flexibilidade, do que a critérios relacionados diretamente à performance.

A seção seguinte investiga a modelagem acústica convencional, com ênfase no estado da arte da síntese sonora em tempo real. Por tratar-se de uma área mais hermética, sobretudo quando comparada ao campo de redes neurais e sua filosofia código aberto, alguns temas relevantes são apresentados com uma maior profundidade conceitual. Por esse mesmo motivo, implementações didáticas dos dois algoritmos mais utilizados são apresentadas.

Em seguida são formalizadas algumas das bases conceituais do trabalho, como a equação da onda e o funcionamento, do ponto de vista matemático, de uma rede neural; ademais, uma interpretação geométrica para a simetria da transformada discreta de Fourier, quando aplicada a sinais no domínio dos números reais, é introduzida.

A seguir, no capítulo dedicado à metodologia, são apresentadas as etapas executadas, desde o processo de obtenção das amostras de som utilizadas na etapa de treinamento das redes, passando pelas investigações no domínio do tempo e da frequência, até a apresentação em mais detalhes do modelo final proposto, que tem seus resultados comentados na seção seguinte. Na conclusão algumas limitações e possibilidades de encaminhamentos futuros são apresentados.

## 1.4 RELEVÂNCIA SOCIO-ECONÔMICA

Os computadores introduziram a produção musical em uma nova era, permitindo, por um lado, experimentações com novos timbres e formas de interação homem máquina para a criação musical e, por outro, a emulação de instrumentos tradicionais, através de técnicas promissoras, como a modelagem acústica (Bovermann *et al.*, 2016).

Avançar essas pesquisas significa, na medida em que, por exemplo, diminui custos de *software* e *hardware* dedicados, oferecer a um maior número de pessoas acesso à educação musical e, por conseguinte, uma formação intelectual mais rica.

Algumas vantagens do acesso precoce à educação musical são delineadas no estudo de Forgeard *et al.* (2008), que sugere uma maior habilidade verbal e uma maior capacidade de

raciocínio não verbal em crianças que praticaram instrumentos em sua infância. Na mesma linha, o trabalho de Vaughn (2000) aponta para uma relação entre o estudo voluntário de música e uma melhoria do desempenho matemático, a partir de uma meta análise de 20 estudos.

Uma outra motivação é estimular a descentralização do capital intelectual relacionado ao campo de instrumentos virtuais, desde a década de 70 fortemente concentrado em algumas empresas japonesas, situadas em sua maioria na cidade de Hamamatsu, como Yamaha e Roland (Reiffenstein, 2006). Essas empresas detêm, exclusivamente ou parcialmente, as patentes para alguns dos mais importantes algoritmos de síntese sonora, como os *digital waveguides*.

## 2 REVISÃO BIBLIOGRÁFICA

Após uma breve exposição sobre o nascimento e evolução das redes neurais, com o objetivo principal de conferir familiaridade à área onde o trabalho é desenvolvido, a revisão bibliográfica aqui apresentada assume um caráter pragmático, investigando na literatura, em especial no que tange as pesquisas relacionadas à área de redes neurais, técnicas pertinentes ao desenvolvimento do modelo proposto.

A última seção elenca de maneira bastante concisa o conteúdo relacionado à modelagem acústica “tradicional”, limitando-se aos modelos fisicamente informados mais importantes para o trabalho em tela. A área de síntese digital, que tem aproximadamente a mesma idade da área de redes neurais é, talvez, mais extensa e errática do que a última em sua evolução, e o leitor interessado em uma revisão histórica mais abrangente é referido ao trabalho de Bilbao (2009), que delinea o desenvolvimento dos esforços de síntese sonora desde as abordagens iniciais baseadas em algoritmos abstratos até os trabalhos mais sofisticados, fisicamente informados, desenvolvidos recentemente.

### 2.1 REDES NEURAIS ARTIFICIAIS

Multidisciplinar desde o nascimento, o desenvolvimento das redes neurais artificiais pode ser remontado aos primeiros esforços para sistematização teórica da forma como o cérebro humano funciona, a partir dos trabalhos de Hermann von Helmholtz, Ernst Mach e Ivan Pavlov, na virada do século 19 (Hagan *et al.*, 2014).

Em 1943, o neurofisiologista Warren McCulloch e o matemático Walter Pitts foram responsáveis por formular o primeiro modelo matemático conhecido do cérebro humano (McCulloch e Pitts, 1943), mostrando que topologias simples podem, em princípio, encarregar-se de operações aritméticas e lógicas (Yadav, Yadav e Kumar, 2015) relativamente complexas. O trabalho desses autores é considerado o início da pesquisa na área de redes neurais artificiais (Russell e Norvig, 2016).

O termo inteligência artificial foi cunhado em 1956 por John McCarthy em uma conferência no Dartmouth College (Coppin, 2004), da qual participaram muitos dos primeiros pesquisadores influentes na área (Brunette, Flemmer e Flemmer, 2009).

Esse termo, assim como o próprio termo inteligência, não se presta a uma definição fácil; para o propósito deste trabalho, no entanto, pode ser entendido como um termo guarda-chuva sob o qual algoritmos que buscam emular aspectos da inteligência humana estão abrigados.

O termo aprendizado de máquina, por sua vez, é utilizado para denominar um subgrupo da área de inteligência artificial, do qual fazem parte as redes neurais, que ocupa-se de técnicas que permitam que máquinas resolvam problemas sem que sejam explicitamente programadas para tal (Raschka, 2015).

Na área mais ampla denominada de inteligência artificial, redes neurais artificiais possuem um papel primordial pois tem sido um dos principais representantes do aprendizado de máquina em diversas aplicações comerciais (Goodfellow *et al.*, 2016).

O primeiro passo para a utilização prática das redes neurais foi dado no final da década de 1950 por Frank Rosenblatt (Hagan *et al.*, 2014), em sua proposta ao laboratório aeronáutico de Cornell de um autômato baseado em seu modelo simplificado de neurônio, o Perceptron (Rosenblatt, 1957).

Tomando como base também as teorias de McCulloch, Widrow desenvolveu a ADALINE (Adaptive Linear Neuron) (Widrow e Hoff, 1960) bastante similar em estrutura ao modelo de Rosenblatt, e partilhando de suas limitações. O modelo de treinamento proposto, no entanto, era consideravelmente mais robusto.

A motivação final de Rosenblatt, inspirada nas teorias sobre o funcionamento do cérebro humano apresentadas por autores como Culbertson, Von Neumann e Ashby (Rosenblatt, 1958) era a construção de uma máquina capaz de aprender a responder diretamente à estímulos físicos externos, como sinais luminosos.

Para tanto, tal máquina utilizaria como unidade fundamental o Perceptron, que funciona matematicamente como uma função aplicada à soma ponderada das entradas e dos vieses dando origem a um classificador linear, capaz de atualizar seus pesos para aprender, através de exemplos de entradas e as correspondentes saídas desejáveis (alvos), a separar linearmente classes diferentes.

A despeito das limitações práticas do modelo proposto, como expostas de maneira um tanto pessimista por Minsky (Minsky e Papert, 1969), é interessante notar que muitas das características contemporâneas já estavam presentes nos trabalho seminais de Rosenblatt e Widrow, notadamente o caráter estocástico das redes neurais, a necessidade de treiná-las com base em grandes conjuntos de dados e a característica de caixa preta do modelo treinado.

É interessante observar que Widrow introduziu, como forma de treinamento, um caso restrito do algoritmo de descida em gradiente, que veio a ser amplamente responsável pelo ressurgimento do interesse em redes neurais décadas mais tarde, em conjunto com uma técnica mais eficiente de atualização dos pesos e vieses de uma rede.

O modelo de Rosenblatt, tendo tido sua primeira implementação na forma de uma simulação em um computador IBM 704 (Bishop, 2006), ao contrário da intenção inicial de utilizar hardware específico, inaugura também a prática da simulação neural.

Enquanto uma arquitetura baseada em uma única camada de Perceptrons apresenta severas limitações, sendo limitada à classificação de variáveis linearmente independentes, o agrupamento sucessivo dessas camadas dá origem a uma topologia, conhecida como Multilayer Perceptron, capaz de atuar como um aproximador universal (Hornik, 1991). Desde que utilize uma função de ativação não-linear (Leshno *et al.*, 1993), e dado um número suficiente de neurons na camada oculta, essa topologia é capaz de mapear qualquer conjunto de números finitos a qualquer outro com precisão arbitrária (Hornik, Stinchcombe e White, 1989).

Um outro impedimento prático do trabalho de Rosenblatt e Widrow foi a ausência de uma metodologia eficiente para a atualização dos pesos da rede, sobretudo envolvendo múltiplas camadas de neurons - caso não coberto pelos algoritmos iniciais tanto de Rosenblatt quanto de Widrow; tal metodologia veio a ser proposta originalmente por Werbos em 1974 (Werbos, 1974). Contudo, essa técnica permaneceu pouco conhecida na comunidade até ser redescoberta por Parker (Parker, 1985) e, pouco tempo depois, Rumelhart (Rumelhart, Hinton e Williams, 1986) na segunda metade da década de 1980 (Mizutani, Dreyfus e Nishio, 2000; Widrow e Lehr, 1990).

O algoritmo, que é conhecido como *backpropagation*, foi um dos responsáveis por reaquecer o interesse no campo (Hagan *et al.*, 2014) na época, inaugurando sua fase atual, com o surgimento dos principais congressos sobre o assunto, como o *IEEE International Conference on Neural Networks* e periódicos, a exemplo do *INNS Neural Networks*, ao fim da década de 1980 (Yadav, Yadav e Kumar, 2015).

Nos anos seguintes, observou-se uma profusão de novas arquiteturas, que foram aprofundando-se na medida em que a utilização de um maior número de camadas ocultas foi possibilitada pelos avanços no *hardware* computacional. Outras formas de organizar camadas sucessivas foram também introduzidas, além de vários avanços incrementais nos algoritmos de treinamento.

## 2.2 REDES NEURAIS - PRINCIPAIS ARQUITETURAS

Esta seção apresenta as principais arquiteturas consolidadas na literatura, dando ênfase às arquiteturas relevantes ao trabalho aqui desenvolvido. Para uma apresentação mais abrangente, ainda que não exaustiva, o leitor é referido à página preparada pelo *The Asimov*

*Institute* (“The Neural Network Zoo”, 2017), onde uma gama maior de arquiteturas é elencada, junto com referência aos artigos onde foram introduzidas.

### 2.2.1 Redes Densas

A arquitetura básica no campo das redes neurais artificiais é a chamada densa, consistindo em várias camadas sucessivas totalmente ligadas entre si por meio de pesos. Os impulsos recebidos pelas camadas anteriores fluem sucessivamente para as camadas posteriores.

São uma generalização do *Multilayer Perceptron* na medida em que utilizam, em geral uma gama mais vasta de funções de ativação, muitas delas de forma sigmoidal, como a função logística  $y = \frac{1}{1+e^{-x}}$  e a tangente hiperbólica  $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$  (Goldberg, 2015).

### 2.2.2 Redes Neurais Profundas

Com a melhoria do hardware, a topologia densa foi recebendo um acréscimo de camadas, gerando as redes profundas (*Deep Neural Networks*) (Patterson e Gibson, 2017). O aspecto mais importante dessa evolução foi que as redes ganharam a possibilidade de gerar representações sucessivas, abstraindo diferentes aspectos dos dados em cada uma de suas camadas (Nguyen et al., 2015).

A grande vantagem desta topologia foi sua capacidade automática de extração dos componentes descritivos dos dados, um trabalho que ficava a cargo dos pesquisadores anteriormente (Socher, 2014). Em contrapartida, a forma como os dados são interpretados pela rede pode não ser facilmente inferida pelo pesquisador, e as redes podem assumir uma forte característica de caixa-preta.

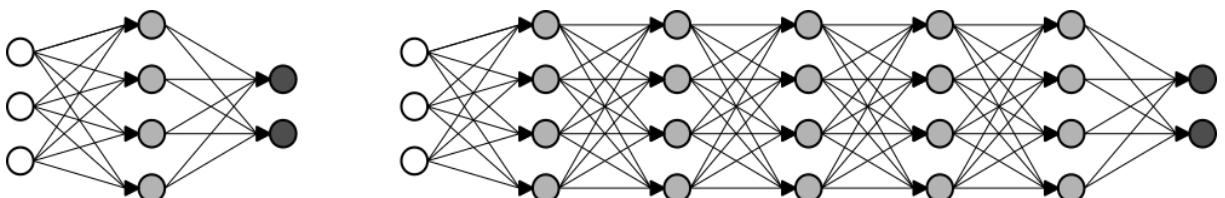


Figura 1: Representação de uma Rede Densa e uma Rede Densa Profunda  
Fonte: Elaboração própria

Na figura 1 observa-se uma representação esquemática de duas arquiteturas densas: uma rede “rasa”, à esquerda, e uma arquitetura profunda, à direita. Os círculos brancos representam as entradas da rede, enquanto os pretos representam suas saídas. Em cinza, temos a representação dos neurons que compõem as camadas ocultas da rede.

### 2.2.3 Redes Recorrentes

As redes recorrentes são representações compactas, do ponto de vista técnico, das arquiteturas profundas (Schmidhuber, 2015), devido ao seu caráter recursivo. Conceitualmente, contudo, a abordagem é distinta: nestas redes os neurons são parcialmente alimentados com seus próprios estados anteriores, emulando um efeito similar à utilização de ligações entre neurons de uma camada anterior com neurons de uma camada posterior não adjacente (Veit, Wilber e Belongie, 2016).

Essa arquitetura foi proposta por Elman (Elman, 1990) com o propósito de capturar informações codificadas no encadeamento temporal de séries de dados, e é bastante poderosa em várias aplicações, como modelos de previsão e classificação de informações (Xu, Auli e Clark, 2015), por exemplo.

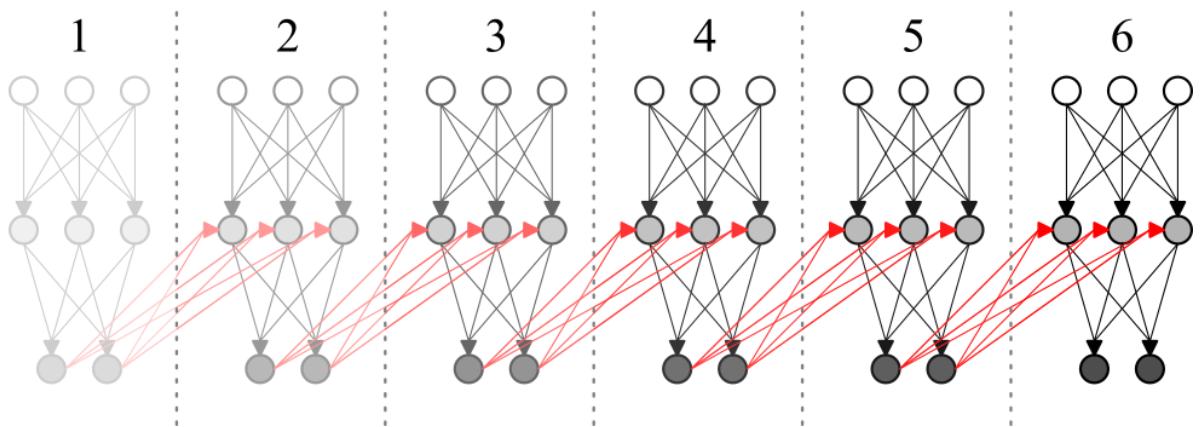


Figura 2: Representação da evolução temporal de uma Rede Recorrente

Fonte: Elaboração própria

A figura 2 representa 6 passos de uma rede recorrente. Observa-se que em cada passo, a partir do primeiro, a rede recebe, além do vetor de entrada externo, as próprias saídas que gerou no passo anterior. É oportuno notar que, em algumas aplicações, a rede recebe apenas um vetor de entradas, no primeiro passo, e opera em todos os passos seguintes processando suas entradas anteriores, em uma configuração que recebe o nome de um para muitos (*one-to-many*) na literatura.

### 2.2.4 Redes Convolucionais

Trata-se de um tipo de arquitetura, geralmente profunda, que é amplamente utilizado em problemas relacionados a imagens, atingindo resultados de ponta em várias áreas relacionadas à visão computacional, como reconhecimento de objetos e rostos em imagens (Pang *et al.*, 2018).

O problema da alta dimensionalidade dos vetores de entrada, como é o caso da representação de uma imagem por exemplo, é amenizado através da substituição de camadas totalmente conectadas por camadas convolucionais, que varrem a imagem, movimentando-se em cada uma de suas dimensões um passo por vez (LeCun *et al.*, 1998), e atualizando todos os pesos da camada convolucional de acordo.

Esse procedimento permite a geração, nessas camadas convolucionais da rede, de uma só representação para padrões que aparecem em diferentes pontos do vetor de entrada; tais representações são geralmente interpretadas em camadas posteriores, totalmente conectadas, de forma a gerar o resultado final.

É comum neste tipo de arquitetura que camadas especiais sejam utilizadas logo após cada uma das camadas convolucionais: essas camadas, denominadas de camadas de *pooling*, ajudam a diminuir ainda mais a dimensionalidade na rede, agregando os outputs de vários neurons em um só valor. Temos, por exemplo, o chamado *max pooling*, onde a saída da camada consiste no valor máximo de saída de um conjunto de neurons.

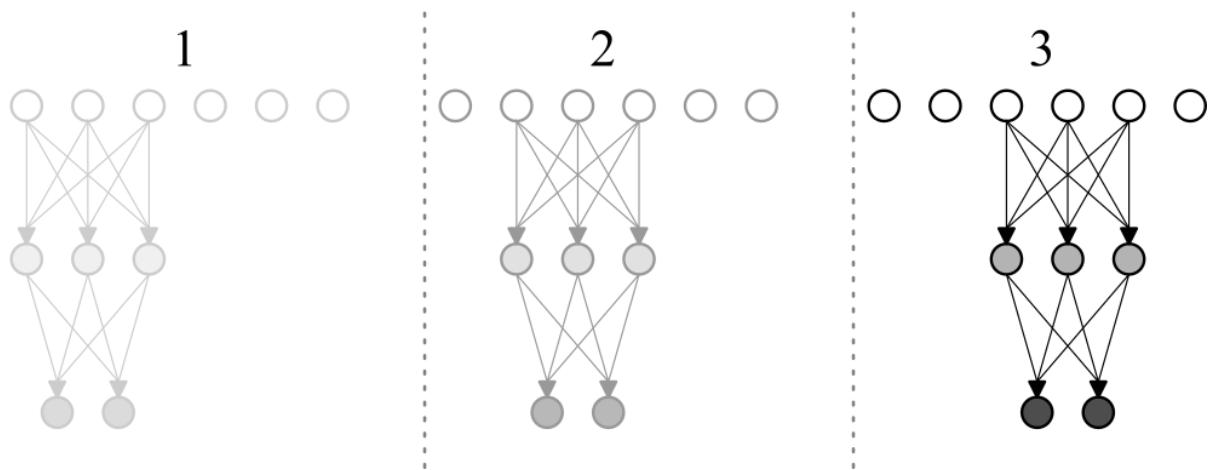


Figura 3: Representação da evolução temporal de uma Rede Convolutacional  
Fonte: Elaboração própria

Na figura 3 é possível acompanhar as 3 primeiras entradas de uma rede convolucional unidimensional: o vetor de entrada é “varrido” pela rede, geralmente andando um passo à cada leitura, em um processo análogo à definição matemática de convolução. O trabalho de Dumoulin e Visin (2016) apresenta uma exposição de vários tipos de convolução, dentre as quais a convolução básica em duas dimensões, apresentada na figura 4

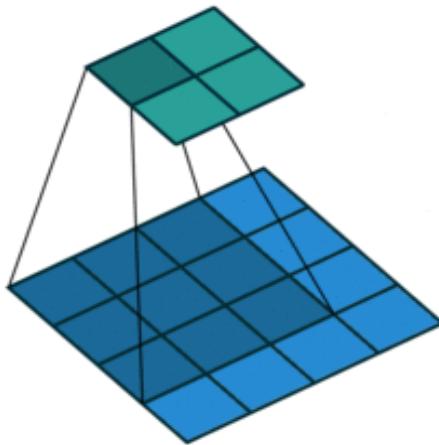


Figura 4: Convolução Bidimensional

Fonte: Dumoulin e Visin (2016)

### 2.3 REDES NEURAIS APLICADAS À MUSICA

A maioria dos trabalhos que investigam a aplicação de redes neurais à música, sobretudo envolvendo sua síntese, ocorrem em um nível de abstração mais alto do que a geração direta dos sons: Geralmente tomam como base a manipulação de representações musicais como partituras, por exemplo, ou representações sonoras compactas, como espectrogramas. Os motivos passam pela alta dimensionalidade dos outputs: no caso de um *framerate* de 44100 amostras por segundo, a qualidade encontrada em CDs, por exemplo, a síntese de 10 segundos de áudio envolve a criação de mais de 4 milhões de samples.

O trabalho desenvolvido em conjunto pelas equipes do Google Brain team e do DeepMind é um exemplo de esforço nesse sentido (Engel *et al.*, 2017). Nele, uma arquitetura desenvolvida a partir da Wavenet é utilizada para a geração de ondas sonoras a partir do treinamento direto com amostras de áudio de vários instrumentos musicais. Os resultados mostram que a arquitetura baseada em várias camadas convolucionais utilizada é capaz de aprender representações no domínio do tempo para vários tipos de instrumentos diferentes.

Uma extensão experimental desse trabalho, denominada de Magenta (“Magenta”, 2018), investiga representações latentes para sequências musicais no domínio do tempo, a partir de uma abordagem probabilística (Roberts, Engel e Oore *et al.*, 2018; Roberts, Engel e Raffel *et al.*, 2018).

Uma plataforma física para a manipulação e reprodução dos sons gerados pela rede, na forma de um *hardware* código aberto denominado *Nsynth Super* (“NSynthSuper”, 2018) é introduzida pela equipe do projeto. Embora não fique claro se a etapa de síntese ocorre

diretamente na plataforma, os esforços corroboram o interesse na aplicação de redes neurais para a síntese sonora em tempo real.

Enquanto as camadas das arquiteturas anteriores são predominantemente convolucionais, Pfalz (2018) apresenta um trabalho bastante abrangente sobre o uso de redes neurais recorrentes puras para essa finalidade. Um dos primeiros trabalhos na área é, talvez, o desenvolvido por Stanley (2007), consistindo na investigação da capacidade de uma rede em prever sons, em analogia direta com séries temporais. Os resultados são bastante tímidos, e uma rede capaz de prever apenas uma onda sonora, oriunda de uma nota de saxofone, é apresentada. De maneira semelhante, Sarroff e Casey (2014) investiga a aplicação direta de redes neurais, de arquitetura densa, na geração direta de múltiplas amostras de áudio

A geração de música em níveis mais altos de abstração é, para o trabalho aqui desenvolvido, apenas marginalmente interessante, na medida em que pode oferecer inspiração sobre representações compactas dos sons. O trabalho de (Hutchings, 2017), por exemplo, utiliza tablaturas para a previsão de partes completas de bateria tomando como base o ritmo de uma de suas peças.

Tomando emprestados os desenvolvimentos obtidos na área da visão computacional, os trabalhos de Grinstein *et al.* (2017) e Mital (2017) buscam transpor os avanços alcançados no campo da transferência de estilo entre imagens para o domínio do som. O primeiro, de cunho exploratório, investiga as arquiteturas mais promissoras para a tarefa, chamando a atenção para o desempenho de arquiteturas rasas, enquanto o segundo investiga, a partir da Wavenet, representações apropriadas. Ainda inspirado em técnicas relacionadas à geração de imagens, Donahue, McAuley e Puckette (2018) fazem uso de redes adversariais (Generative Adversarial Networks - GANs) para a síntese direta de áudio, no contexto da emulação do discurso falado, gerando resultados inteligíveis.

Os esforços de classificação costumam introduzir inovações interessantes nas representações utilizadas: Costa, Oliveira e Silla (2017) e Choi, Fazekas e Sandler ( 2016) abordam a tarefa a partir de redes convolucionais; no primeiro trabalho as redes são alimentadas com música representada por um tipo especial de espectrograma enquanto no segundo arquiteturas convolucionais profundas são investigadas, assim como em Choi *et al.*, (2017) que acrescenta recorrência às arquiteturas convolucionais, no intuito de explorar as inter-relações temporais das amostras sonoras utilizadas.

Representações não canônicas também possuem importante papel nas tarefas relacionadas à transcrição musical a partir do uso de redes neurais, onde busca-se traduzir amostras sonoras em representações de nível mais alto, simbólicas, como tablaturas e

partituras, por exemplo. Uma das primeiras contribuições nesse sentido pode ser vista em Tuohy, Potter e Center (2006) por meio de um modelo que combina redes neurais e uma heurística *hill-climbing*.

Com o uso de uma rede recorrente, Boulanger-Lewandowski, Bengio e Vincent (2013) transcrevem, a partir de espectrogramas, diferentes partes musicais em comandos MIDI. Na mesma linha Böck e Schedl (2012) e Sigtia, Benetos e Dixon (2016) introduzem trabalhos focados na transcrição de sons polifônicos produzidos por pianos. Em relação à transcrição de partes de um *kit* de bateria, Southall, Stables e Hockman (2016) abordam a tarefa a partir de uma arquitetura recorrente alimentada com representações espetrais.

## 2.4 REDES NEURAIS APLICADAS AO DISCURSO FALADO

Um dos trabalhos mais notáveis nesse sentido, desenvolvido pela equipe de inteligência artificial do Google, é a chamada Wavenet (van den Oord *et al.*, 2016; “WaveNet: A Generative Model for Raw Audio”, 2018), uma arquitetura baseada em camadas convolucionais, para a geração ponta a ponta de voz. Uma amostra é gerada por vez, levando em conta um grande número de entradas passadas; os resultados introduzem um novo estado da arte para a tarefa. Os desdobramentos dessa arquitetura na área musical foram apresentados na seção anterior.

Hinton *et al.* (2012) apresenta uma revisão da literatura sobre o uso de modelagem acústica em abordagens baseadas em redes neurais na área de reconhecimento de voz. No mesmo ano Graves, Mohamed e Hinton (2013) alcançam resultados no *TIMIT phoneme recognition benchmark*, com a aplicação de redes recorrentes profundas, que estabelecem o novo estado da arte, enquanto Maas, Hannun e Ng (2013) apontam a superioridade da função de ativação *relu* (*rectifier nonlinearities*) sobre ativações de caráter sigmoidal em tarefas relacionadas ao reconhecimento contínuo da fala.

A partir de uma arquitetura híbrida, combinando um modelo fonético recorrente com um classificador acústico baseado em uma rede neural profunda, Boulanger-Lewandowski *et al.* (2014) aplicam estratégias de sequenciamento telefônico para estabelecer um novo *benchmark* no *TIMIT dataset*, uma técnica que foi comprovada por Sak *et al.* (2015) como superior ao uso de arquiteturas como *deep long short-Term memory recurrent neural networks* e as abordagens mais utilizadas à época baseadas no modelo oculto de Markov.

Com relação à utilização de redes convolucionais, Sainath *et al.* (2015) investiga a otimização de seus hiperparâmetros, além de estratégias de *pooling* e treinamento para aplicações em reconhecimento de fala, enquanto Zweig *et al.* (2016) e Zhang *et al.* (2017)

exploram o desempenho de sistemas baseados somente em redes neurais (*end-to-end*), com o último combinando redes convolucionais hierárquicas com classificação temporal conecionista. Nessa mesma linha, o trabalho de Zhang, Chan e Jaitly (2017) também investiga sistemas *end-to-end* a partir de uma arquitetura combinando redes neurais convolucionais profundas com recorrência e princípios baseados no aninhamento de redes neurais (NIN, *network in a network*). Alguns desses princípios, notadamente a justaposição de convolução e recorrência, inspiraram a implementação apresentada na seção sobre convolução e recorrência.

## 2.5 REDES NEURAIS APLICADAS À IMAGEM

Recentemente tem-se visto esforços para adaptar métodos utilizados com sucesso na área da visão computacional ao campo da modelagem acústica. O trabalho de (Engel *et al.*, 2017), batizado de Nsynth, busca inspiração na área de geração de imagens para elaborar uma arquitetura apropriada a síntese sonora. Ainda em paralelo com a área de relacionada a imagens, o autor propõe uma base de dados sonora à luz de bases de dados clássicas no campo de visão computacional, como o MNIST, em um esforço para alavancar pesquisas nessa direção.

Da mesma forma a Wavenet, antecessora do Nsynth, é uma adaptação das arquiteturas PixelRNN (van den Oord, Kalchbrenner e Kavukcuoglu, 2016) e PixelCNN (van den Oord *et al.*, 2016), desenvolvidas para a geração de imagens.

Os trabalhos que tem como foco a síntese de imagens geralmente envolvem GANs: trata-se de uma abordagem onde uma rede é treinada para gerar imagens, enquanto outra as classifica em artificiais ou não. O objetivo da primeira rede é enganar a segunda, dando origem a imagens mais realistas, do ponto de vista da percepção humana.

Isola *et al.* (2016), por exemplo, utiliza essa técnica para gerar imagens a partir de representações de seus contornos, enquanto Hwang e Zhou (2016), Zhang, Isola e Efros (2016), Larsson, Maire e Shakhnarovich (2016) e Iizuka, Simo-Serra e Ishikawa (2016) abordam a tarefa de colorização automática de imagens.

Ainda usando redes adversariais Frans (2017) introduz controle ao processo de colorização, com a utilização de mapas de cores, que são usados como entradas para as redes, um conceito que é explorado também por Sangkloy *et al.* (2017), onde a interação do usuário e uma rede densa permitem a colorização de imagens em tempo real, através de linhas desenhadas em áreas da imagem que possuam as cores pretendidas.

Gatys, Ecker e Bethge (2016) propõe uma técnica de transferência de estilo, utilizando uma rede convolucional capaz de aprender características do estilo de uma imagem e transferir para outra, sem alterar o seu conteúdo semântico. Nessa linha Kulkarni *et al.* (2015) utiliza uma arquitetura de convolução-deconvolução para separar características de posicionamento de uma imagem, permitindo sua reconstrução posterior em outras posições e situações de iluminação, a partir de mudanças manuais nas variáveis de entrada da rede. Ainda investigando a manipulação de imagens a partir de redes adversariais o trabalho de Zhu *et al.* (2016) tem como objetivo aprender características das imagens de maneira direta.

O trabalho de van den Oord, Kalchbrenner e Kavukcuoglu (2016), que veio a inspirar a Wavenet, propõe uma topologia recorrente profunda com conexões residuais melhoradas capaz de reconstruir imagens parcialmente obstruídas, enquanto Theis e Bethge, (2015) investigam arquiteturas LSTM multidimensionais no contexto da modelagem da distribuição de imagens.

Na área de vídeo, que pode ser entendida como um caso geral da manipulação de imagens, com complexidades adicionais relacionadas à alta dimensionalidade dos dados e ao encadeamento temporal. Karpathy *et al.* (2014) investigam abordagens capazes de estender as redes convolucionais de forma a permitir que tirem proveito das características temporais dos dados de entrada. De forma semelhante, portando desenvolvimentos na área de reconhecimento de expressões faciais obtidos em imagens, Khorrami *et al.* (2016) combinam redes recorrentes e convolucionais, medindo a importância relativa de cada uma nos resultados finais.

Buscando combater a alta dimensionalidade, Yang, Krompass e Tresp (2017) apresentam o conceito de *tensor-train*, permitindo a transferência de conhecimento de outras arquiteturas para a utilização em dados sequenciais com alta dimensionalidade. O trabalho de He *et al.* (2015) introduz novos benchmarks à tarefa de reconhecimento de emoções em vídeos, utilizando uma arquitetura profunda baseada em camadas LSTM bidirecionais no processamento de vídeos, incluindo seu áudio.

## 2.6 REDES NEURAIS - FERRAMENTAS E FRAMEWORKS

A área de aprendizado de máquina encontra-se muito ativa atualmente, tanto no âmbito da academia quanto no da indústria, com várias grandes empresas de tecnologia incorporando essa tecnologia em suas competências essenciais. Dessa forma, assistimos à uma proliferação de ferramentas e plataformas focadas em diferentes aspectos da área, muitas delas desenvolvidos ou endossados por essas empresas, como Microsoft, Google e Amazon.

Somando-se a isso o fato de que, em maior ou menor grau, todos eles apresentam uma curva de aprendizado significativa, fica evidente que uma comparação direta de todos, ou mesmo da maioria das ferramentas disponíveis, encontra-se além do escopo deste trabalho.

Algum direcionamento, contudo, pode ser obtido a partir da literatura disponível. Bahrampour *et al.* (2015), por exemplo, compara quatro ferramentas: Caffe (“Caffe Deep Learning Framework”, 2018), desenvolvida pelo grupo de pesquisas em inteligência artificial de Berkeley (“The Berkeley Artificial Intelligence Research Lab”, 2014); Neon (“Neon”, 2018), desenvolvido pelo setor de inteligência artificial da Intel; Torch (“Torch, a Scientific computing framework for LuaJIT.”, 2018) e Theano (“Theano 1.0.0 documentation”, 2018) os dois últimos sem afiliações diretas com grandes empresas. O autor conclui pela superioridade da plataforma Theano, em termos de velocidade em geral e flexibilidade, chamando atenção para sua capacidade de auto diferenciação.

Embora o time principal tenha abandonado o desenvolvimento do Theano no final de 2017 a plataforma Tensorflow (“TensorFlow”, 2018), introduzida por pesquisadores do *Google Brain Team* em 2015 é, em muitos aspectos, sucessora do Theano, conservando a maioria de suas características.

Se esse estudo, realizado em 2015, indicava alguma ineficiência da plataforma TensorFlow, à época ainda em sua infância, o trabalho de Shi *et al.* (2016), ao comparar 5 plataformas, conclui que a situação 1 ano depois é diferente, e que não há uma plataforma sensivelmente superior às outras em termos gerais.

O trabalho de Erickson *et al.* (2017) é ainda mais abrangente, comparando 12 frameworks, e chamando atenção para a importância das habilidades prévias e características da pesquisa na escolha da ferramenta mais apropriada. Na mesma linha, Parvat *et al.* (2017) compara 5 plataformas, chamando atenção para as potenciais vantagens, em termos do tamanho do código e rapidez na criação de protótipos, da utilização de bibliotecas de nível de abstração mais alto, como a biblioteca Keras (“Keras: The Python Deep Learning library”, 2018), no topo das plataformas convencionais.

## 2.7 SÍNTESE SONORA EM TEMPO REAL

Excluídas as técnicas baseadas em manipulação de amostras pré-gravadas de sons, talvez a abordagem mais utilizada por músicos atualmente (Bilbao, 2009), podemos dividir área de síntese sonora propriamente em duas escolas, na medida em que ocupa-se em modelar os processos físicos que dão origem ao som (Modelagem Física - *Physical Modelling*) ou

diretamente as características das ondas sonoras (Modelagem Espectral - *Spectral Modelling*) (Serra, 2007).

Dessas escolas, a modelagem física é a mais proeminente tanto do ponto de vista da pesquisa quanto de aplicações; a área de modelagem espectral parece ter despertado pouco interesse dos pesquisadores nas últimas duas décadas, encontrando a maioria de suas aplicações recentes em áreas fora da síntese musical. A modelagem física apresenta como principais vantagens um menor número de parâmetros, que costumam ser mais intuitivos, na medida em que são diretamente relacionados a grandezas existentes no mundo real.

Abordagens baseadas na modelagem física ocupam-se majoritariamente em formular modelos discretos baseados em descrições matemáticas dos instrumentos, como por exemplo a solução de D'alembert para a equação da onda (Karjalainen e Erkut, 2004), normalmente incorporando ao modelo termos relacionados por exemplo à rigidez exibida por cordas não ideais, que modelem características relevantes ao timbre do instrumento representado (Bensa *et al.*, 2003). Destacam-se, nessa escola, duas abordagens: o método das diferenças finitas, acurado e computacionalmente intensivo, e os *Digital Waveguides*, um modelo bastante eficiente (Bensa *et al.*, 2005) e elegante que é, talvez, o mais utilizado para a emulação em tempo real de instrumentos acústicos.

### 2.7.1 Método das Diferenças Finitas

A formulação mais básica desse método consiste na resolução da equação da onda propagando-se em uma corda ideal, introduzida abaixo e derivada com mais detalhes na seção seguinte.

$$\frac{\partial^2 y(x, t)}{\partial t^2} = \frac{T}{\rho} \frac{\partial^2 y(x, t)}{\partial x^2} \quad (1)$$

A solução é buscada para pontos em uma malha obtida discretizando-se tanto o tempo  $t$  quanto a posição horizontal  $x$ . Isso permite a aproximação das derivadas parciais de segunda ordem pelo método das diferenças centrais. Assim procedendo, podemos reescrever a equação da onda como abaixo.

$$\frac{y[i, j + 1] - 2y[i, j] + y[i, j - 1]}{\Delta t^2} = c^2 \frac{y[i + 1, j] - 2y[i, j] + y[i - 1, j]}{\Delta x^2}$$

onde

$$c = \sqrt{\frac{T}{\rho}}$$

$$x_i = i\Delta x, i \in \{0, 1, \dots, M\}$$

$$t_j = j\Delta t, j \in \{0, 1, \dots, N\}$$

e, colocando  $y[i, j + 1]$  em evidência:

$$y[i, j + 1] = 2y[i, j] - y[i, j - 1] + C^2(y[i + 1, j] - 2y[i, j] + y[i - 1, j])$$

Em  $C = c \frac{\Delta t}{\Delta x}$ , denominado número de Courant, foram aglutinados todos os parâmetros que governam a qualidade da simulação. Para garantir sua estabilidade, os parâmetros devem ser fixados de modo que  $C$  seja menor que 1.

Repare que  $y[i, j + 1]$  é calculado com base nos último e penúltimo momentos discretos, o que demanda, para o primeiro passo da simulação em  $j = 1$ , a definição do estado inicial do sistema em  $j = 0$ , e informações sobre o sistema no momento discreto indefinido  $j = -1$ . Essa última dificuldade será atacada a partir de considerações adicionais sobre o sistema.

Como condições de contorno para o caso contínuo temos que, no momento inicial, a corda encontra-se deslocada do seu equilíbrio, em repouso. Portanto, a velocidade em  $t = j = 0$ , para qualquer ponto  $x$  da corda, tem valor zero. Ademais, define-se a corda como fixa em suas extremidades nos pontos  $x = 0$  e  $x = L$ . Para primeira condição podemos escrever:

$$\frac{\partial y(x, t)}{\partial t} \approx \frac{y[i, 1] - y[i, -1]}{2\Delta t} = 0 \quad \forall i \in \{0, 1, \dots, M\}$$

o que implica:

$$y[i, 1] = y[i, -1] \quad \forall i \in \{0, 1, \dots, M\}$$

Assim, para o primeiro momento da simulação em  $j = 1$  podemos substituir  $y[i, -1]$  por  $y[i, 1]$ , obtendo a equação para quando  $j = 0$ :

$$y[i, 1] = y[i, 0] + \frac{1}{2}C^2(y[i + 1, 0] - 2y[i, 0] + y[i - 1, 0])$$

Em relação à segunda condição, devemos reforçar o fato de que  $y[0, j] = y[L, j] = 0 \quad \forall j \in \{0, 1, \dots, N\}$  em cada passo da simulação.

É conveniente colocar a formulação em termos físicos: se considerarmos, por exemplo, um *framerate* típico de 44100 amostras por segundo podemos evitar desperdício computacional fazendo com que os passos da simulação coincidam com o intervalo de 1/44100 segundo entre os pontos amostrados. De modo geral, levando em conta que  $C \leq 1$ , podemos parametrizar a simulação em termos dos componentes físicos e o *framerate* desejado da seguinte forma:

$$dt = \frac{1}{FPS} = \frac{D}{N}$$

$$N = D \cdot FPS$$

$$dx = \frac{L}{M}$$

$$M \leq \frac{FPS}{2f}$$

$$c = 2fL$$

onde  $D$  é a duração desejada, em segundos;  $L$  é o comprimento da corda em metros e  $FPS$  é o *framerate* pretendido.

O pseudocódigo apresentado no quadro 1 ajuda a ilustrar o funcionamento do algoritmo. Uma implementação na linguagem Python, versão 3.6, pode ser encontrada no repositório na plataforma Github elaborado como acompanhamento a este trabalho (“Repositório Dissertação”, 2018), no caminho “resources/Finite\_Difference.py”.

```

amplitude <- 0.005                      # metros
L <- 0.6                                # metros
pluck_position <- .5                      # fração do comprimento L
pickup_position <- .5                      # fração do comprimento L
fps <- 44100                             # amostras / segundo
frequency <- 440                          # Hz
duration <- 1                            # segundos
sustain <- .9998
N <- duration * fps                      # número de pontos do grid temporal
dt <- 1 / fps                           # espaço entre pontos no grid temporal
M <- fps / (2 * frequency)               # número de pontos do grid espacial
dx <- L / M                            # espaço entre pontos no grid espacial
c <- frequency * 2 * L                  # metros / segundo
C <- c * dt / dx                        # número de Courant
x <- [0, dx, 2 dx,..., M dx]
t <- [0, dt, 2 dt,..., N dt]
pickup <- M * pickup_position
y_0 <- vetor com "M" zeros
y_1 <- vetor representando o formato inicial do deslocamento vertical
y_2 <- vetor com "M" zeros

ctr <-0
W <- []
FOR i IN [1, 2,..., M]:
    y_0[i] <- y_1[i] + 0.5 * C^2 * (y_1[i+1] - 2 * y_1[i] + y_1[i-1])
y[0] <- 0
y[M] <- 0
APPEND y_0[pickup] TO W
y_2 <- y_1
y_1 <- y_0
ctr <- ctr + 1
FOR j IN [1, 2,..., N]:

```

```

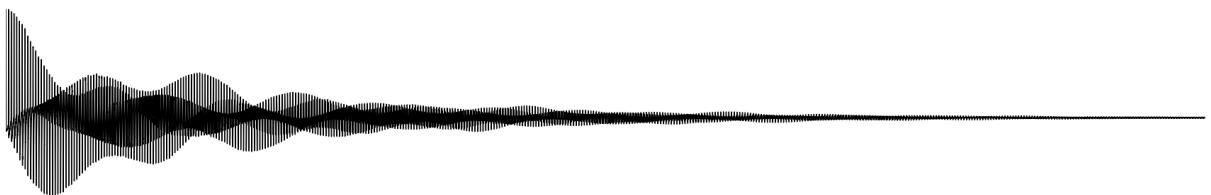
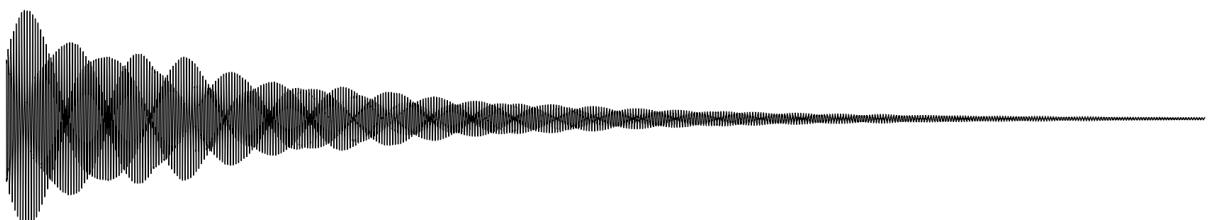
FOR i IN [1, 2, ..., M]:
    y_0[i] <- 2 * y_1[i] - y_2[i] + C^2 * (y_1[i+1] - 2 * y_1[i] + y_1[i-1])
    y_0[0] <- y[0] * 0.5
    y_0[M] <- y[M] * 0.5
    y_0 <- y_0 * sustain
    APPEND y_0[pickup] TO W
    y_2 <- y_1
    y_1 <- y_0
    ctr <- ctr + 1
OUTPUT W

```

Quadro 1: Pseudocódigo do Método das Diferenças Finitas

Fonte: Elaboração própria

Uma animação relativa a uma onda de frequência real igual a 440 Hz, criada a partir do código, também está disponível no repositório da dissertação (“Repositório Dissertação”, 2018), no caminho “resources/Demo Finite Difference/”, junto com áudios de diversas ondas, com locais de excitação e captação variados. É interessante observar o efeito da variação desses parâmetros no timbre da onda sonora gerada. Cabe notar que a velocidade e a escala da animação estão bastante distorcidas, para facilitar a visualização.

Figura 5: Excitação: 0.1 - Captação: 0.1  
Fonte: Elaboração própriaFigura 6: Excitação: 0.1 - Captação: 0.5  
Fonte: Elaboração própria

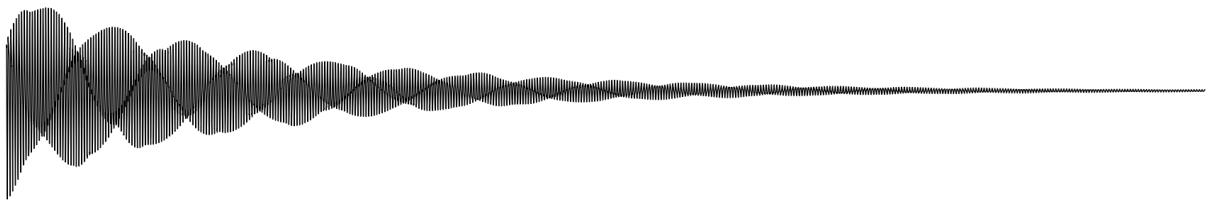


Figura 7: Excitação: 0.3 - Captação: 0.7  
Fonte: Elaboração própria

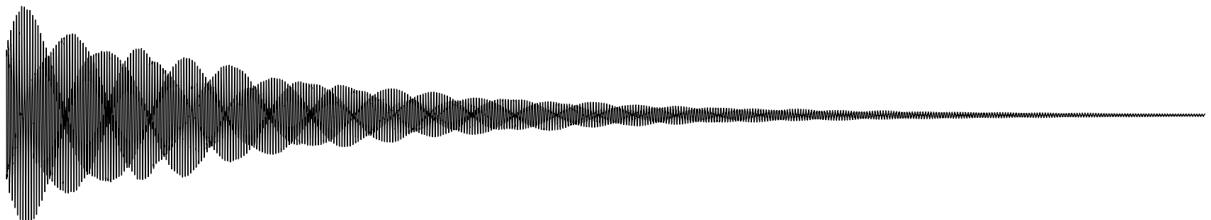


Figura 8: Excitação: 0.5 - Captação: 0.1  
Fonte: Elaboração própria

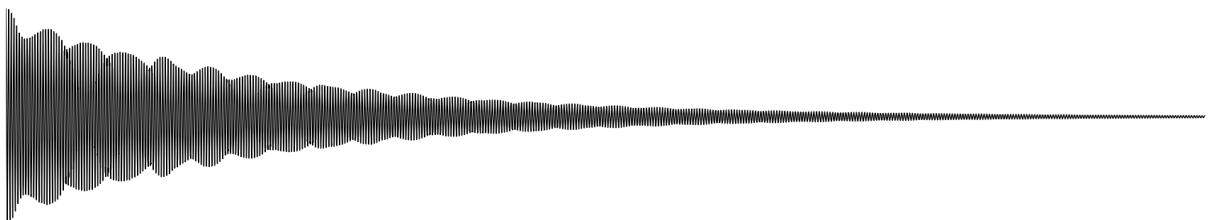


Figura 9: Excitação: 0.5 - Captação: 0.5  
Fonte: Elaboração própria

### 2.7.2 Digital Waveguides

Esse algoritmo consiste em uma abordagem simplificada da modelagem física, na medida em que concentra em alguns pontos discretos os cálculos necessários à simulação, aumentando bastante a eficiência computacional do modelo (Van Duyne e Smith, 1995).

Conceptualmente, podem ser vistos como um caso especial do método das diferenças finitas (Van Duyne e Smith, 1993), e a maioria de suas implementações consistem na utilização de *delay lines* e filtros digitais para a modelagem da propagação da onda (Smith, 2006).

O modelo também parte da equação 1, a equação da onda, discretizada da forma apresentada no contexto do método das diferenças finitas. A formulação analítica, no entanto, é abandonada, em favor de uma formulação baseada na interpretação das duas ondas viajando

em sentidos opostos na corda. A imagem abaixo apresenta a intuição do método, que representa ainda hoje o estado da arte da modelagem física (Bilbao, 2009).

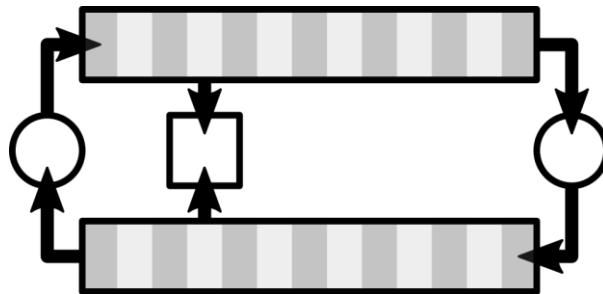


Figura 10: Digital Waveguides  
Fonte: Elaboração própria

As duas linhas quadriculadas apresentadas na figura 10 são as chamadas *delay lines*, por onde a onda circula inalterada, em uma direção: no exemplo, na linha superior a perturbação viaja da esquerda para a direita, e da direita para a esquerda na linha de baixo. Essa movimentação geralmente é implementada a partir estrutura de dados denominada *circular buffer*. O movimento de reflexão da onda ao alcançar as extremidades fixas da corda é calculado nos pontos representados por círculos na figura 10, que são responsáveis por enviar os *samples* de uma *delay line* para a outra, invertendo seu sinal; nesses pontos, geralmente a partir da utilização de filtros digitais, outras manipulações são introduzidas, no sentido, por exemplo, de emular perdas e fatores relacionados à rigidez de uma corda não ideal. O quadrado na figura 10 representa a posição onde a onda é amostrada, a partir da soma do conteúdo das duas *delay lines*. Esse sistema, como um todo, recebe o nome de *digital waveguide*.

Do ponto de vista físico os *digital waveguides* representam uma corda fixa em suas terminações, por onde um pulso viaja, sendo refletida nas extremidades. Ao passar de uma linha pra outra, a onda é espelhada e invertida e todos os cálculos são concatenados nesses pontos ou, dependendo da simulação, em apenas um desses pontos, como é o caso da implementação apresentada no quadro 2, aumentando consideravelmente a eficiência da simulação.

No modelo mais básico, um termo de perda é calculado durante a transição de uma linha para outra, enquanto implementações mais sofisticadas fazem uso dos filtros para emular além da rigidez das cordas, alguns outros fenômenos mais sofisticados, como dispersão, por exemplo. O modelo pode ser estendido também para emulações bidimensionais

(Mullen, 2006) e tridimensionais (Fontana, Rocchesso e Apollonio, 2000; Laird, 2001; Speed, 2012), e presta-se a implementações bastante eficientes, principalmente em uma dimensão.

O pseudocódigo apresentado no quadro 2 ilustra o funcionamento do algoritmo. Observa-se que a implementação é mais simples do que a do método das diferenças finitas, e mais eficiente, como a tabela 1 demonstra. Como antes, a versão funcional do código, em Python, pode ser acessada no repositório da dissertação (“Repositório Dissertação”, 2018), no caminho “resources/Digital\_waveguide.py”, e uma animação ilustrando uma onda gerada pelo método, além dos com diferentes valores para o ponto de excitação e captação áudios, podem ser acessados no mesmo repositório, em “resources/Demo Digital Waveguide/”

```

N <- 44100                                # número final de amostras geradas
fps <- 44100                                 # amostras / segundo
frequency <- 440                            # Hz
L <- fps / (2 * frequency)
pluck_position <- .5                         # fração do comprimento L
pickup_position <- .5                         # fração do comprimento L
sustain <- .99
smoothing <- 3
pickup <- L * pickup_position
delay_r <- vetor representando o formato inicial do deslocamento vertical
delay_l <- vetor com "L" zeros
W <- vetor com "N" zeros
FOR i IN [1, 2,..., N]:
    W[i] <- delay_r[pickup] + delay_l[pickup]
    to_l <- -1 * AVERAGE(delay_r[delay_r.LENGTH - smoothing : delay_r.LENGTH]) *
sustain
    FOR i IN [delay_r.LENGTH - 2, delay_r.LENGTH - 3,...,0]:
        delay_r[i+1] <- delay_r[i]
        delay_r[0] <- -1 * delay_l[0]
        FOR i IN [0,1,...,delay_l.LENGTH - 2]:
            delay_l[i] <- delay_l[i+1]
            delay_l[L-1] <- to_l
OUTPUT W

```

Quadro 2: Pseudocódigo do algoritmo Digital Waveguides

Fonte: Elaboração própria



Figura 11: Excitação: 0.1 - Captação: 0.1  
Fonte: Elaboração própria

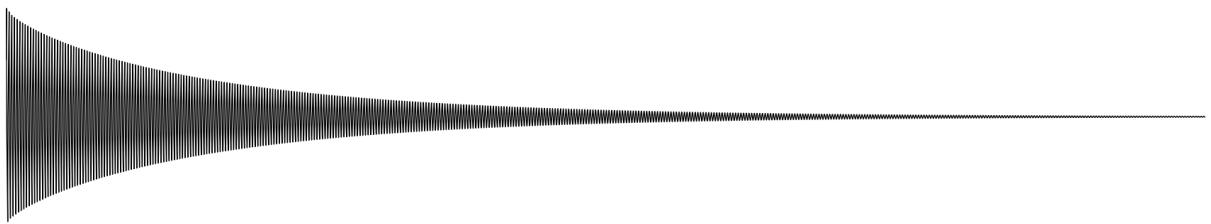


Figura 12: Excitação: 0.1 - Captação: 0.5  
Fonte: Elaboração própria

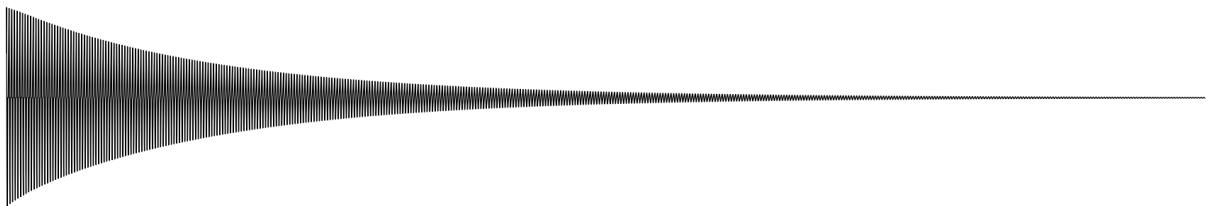


Figura 13: Excitação: 0.3 - Captação: 0.7  
Fonte: Elaboração própria

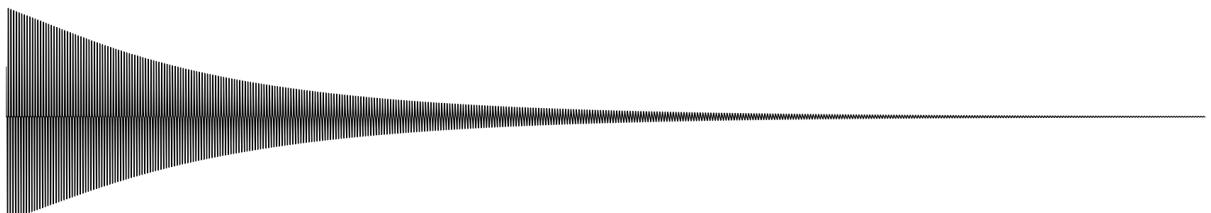


Figura 14: Excitação: 0.5 - Captação: 0.1  
Fonte: Elaboração própria

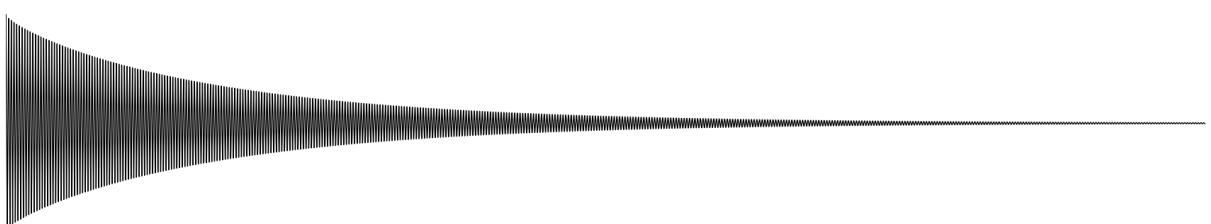


Figura 15: Excitação: 0.5 - Captação: 0.5  
Fonte: Elaboração própria

### 2.7.3 Digital Waveguides X Diferenças Finitas

Para a comparação dos dois algoritmos, alguns exemplos de ondas equivalentes, geradas por ambos, foram preparados, a fim de basear um julgamento subjetivo. A figura 16

compara o primeiro frame e último das simulações da evolução do pulso em uma corda para os dois métodos.

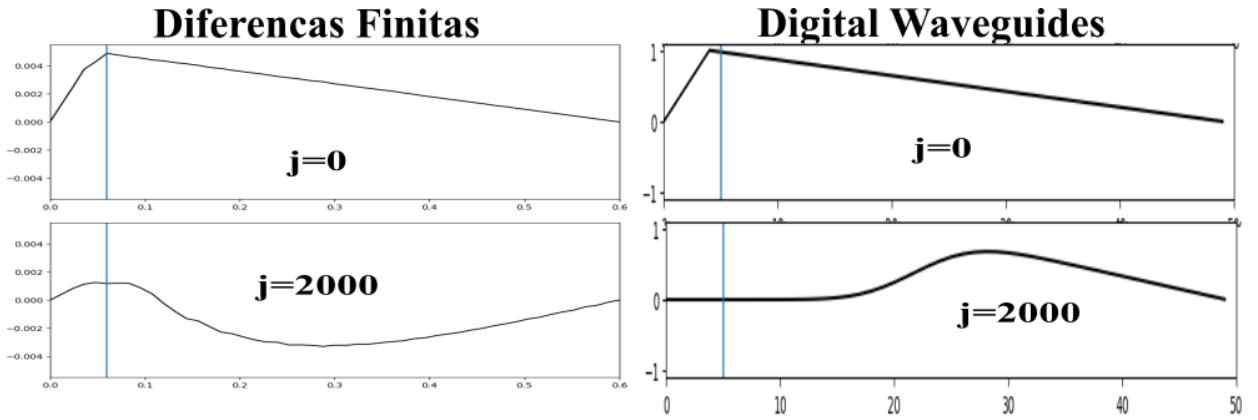


Figura 16: Animação da Corda - Diferenças Finitas x Digital Waveguide

Fonte: Elaboração própria

A movimentação da onda produzida a partir do método das diferenças finitas tem um caráter mais orgânico, sendo mais propensa a desvios da condição de estabilidade, enquanto a simulação a partir do algoritmo de *digital waveguides* é mais robusta, porém mais mecânica. Em relação às eficiências, abaixo são elencados os tempos que foram necessários para a geração dos sons apresentados anteriormente. Todas as amostras têm *framerate* igual a 44100 FPS e duração de 1 segundo. Os tempos são apresentados em segundos.

Tabela 1: Latência em segundos para uma onda com 44100 *samples*

Excitação	Captação	Digital Waveguides	Diferenças Finitas
0.1	0.1	13.811148881912231	16.9521381855011
0.1	0.5	13.020658254623413	19.620429277420044
0.3	0.7	11.332738161087036	24.256462335586548
0.5	0.1	12.350087404251099	17.89053726196289
0.5	0.5	12.122233390808105	18.002467393875122

### 3 REFERENCIAL TEÓRICO

O objetivo deste capítulo é oferecer uma definição mais formal das principais ferramentas utilizadas neste trabalho.

#### 3.1 TRANSFORMADA DISCRETA DE FOURIER

A transformada de Fourier, em sua forma contínua, é um tipo de transformada integral, como abaixo:

$$X(f) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{\infty} x(t) e^{-2\pi f t} dt$$

$$x(t) = \mathcal{F}^{-1}\{X(f)\} = \int_{-\infty}^{\infty} X(f) e^{2\pi f t} df$$

Tendo sido formulada por Fourier, enquanto investigava o fenômeno da transferência de calor (Stein e Shakarchi, 2003), trata-se de um dos algoritmos mais utilizadas na investigação de sistemas físicos (Lyons, 2010; Olson, 2017).

Esta ferramenta parte da premissa de que qualquer sinal contínuo e periódico pode ser representado a partir da soma de um número arbitrário de senoides (Smith, 1999) permitindo que a descrição de um sinal  $x(t)$ , em relação ao tempo, seja transformada em uma representação deste mesmo sinal em termos das frequências que o compõe, na forma  $X(f) = \mathcal{F}\{x(t)\}$ ;  $x(t)$  e  $X(f)$  são denominados um par de Fourier.

A versão discreta da transformada de Fourier, apresentada abaixo, é uma das ferramentas mais importantes e utilizadas no campo de processamento de sinais (Gazi, 2017). Contribui para essa penetração a existência de um algoritmo eficiente, denominado de *Fast Fourier Transform*, que diminuiu consideravelmente o número de operações necessárias ao cômputo da transformada discreta de Fourier (Pereyra e Ward, 2012). Considerado:

$$X[m] = \sum_{n=0}^{N-1} x[n] e^{-\frac{2\pi mn}{N} i}$$

$$x[n] = \frac{1}{N} \sum_{m=0}^{N-1} X[m] e^{\frac{2\pi mn}{N} i}$$

Onde  $x$  e  $X$  são sequências de  $N$  números complexos. Utilizando a fórmula de Euler,  $e^{x i} = \cos(x) + \sin(x) i$ , podemos alterar a definição acima, da forma exponencial, para a trigonométrica, reorganizando o expoente de forma a evidenciar algumas características importantes, na forma abaixo.

$X[m] = \sum_{n=0}^{N-1} x[n] \cos\left(m \frac{2\pi}{N} n\right) - x[n] \sin\left(m \frac{2\pi}{N} n\right) i$  ma acima torna mais fácil uma interpretação geométrica do algoritmo. Para cada valor de  $m \in 0, 1, 2, \dots, N-1$  é feita a multiplicação, elemento a elemento, das  $N$  posições do impulso  $x$  com  $N$  posições de uma onda em forma de cosseno ( $C(m)$ ), pura, de “frequência”  $m$ , na parte real da equação. Analogamente, em sua parte imaginária, o mesmo processo ocorre: cada posição medida do impulso  $x$  também é multiplicada pela posição equivalente, desta vez, de uma onda em formato de seno ( $S(m)$ ), com “frequência”  $m$ .

Note-se que esta “frequência”, e por isso a insistência nas aspas, refere-se a quantos ciclos das ondas ( $C(m)$ ) e ( $S(m)$ ) acima definidas estão contidos no espaço dos  $N$  pontos utilizados na transformada e não corresponde diretamente à frequência física da onda; essa relação é regida por  $f_l = \frac{f_r}{FPS.N}$ , onde  $f_l$  é a frequência local, e  $f_r$  sua frequência real, física.

Para o caso de sinais que, no domínio do tempo, consistem de uma sequência finita de números reais, como é o caso dos sinais considerados neste trabalho, pode-se, à luz da definição acima, derivar uma interessante e útil propriedade do algoritmo: A descrição do sinal no domínio da frequência é conjugado-simétrica em torno de sua mediana, de forma que  $X[m] = X^*[N-m]$ , com  $X^*$  denotando o conjugado complexo de  $X$ . Dessa forma, sendo  $N$  par, precisamos de apenas de  $N/2 + 1$  termos de  $X[m]$  para descrever completamente o pulso no domínio da frequência, enquanto  $(N+1)/2$  são suficientes caso  $N$  seja ímpar. A figura 17 ilustra geometricamente essa propriedade.

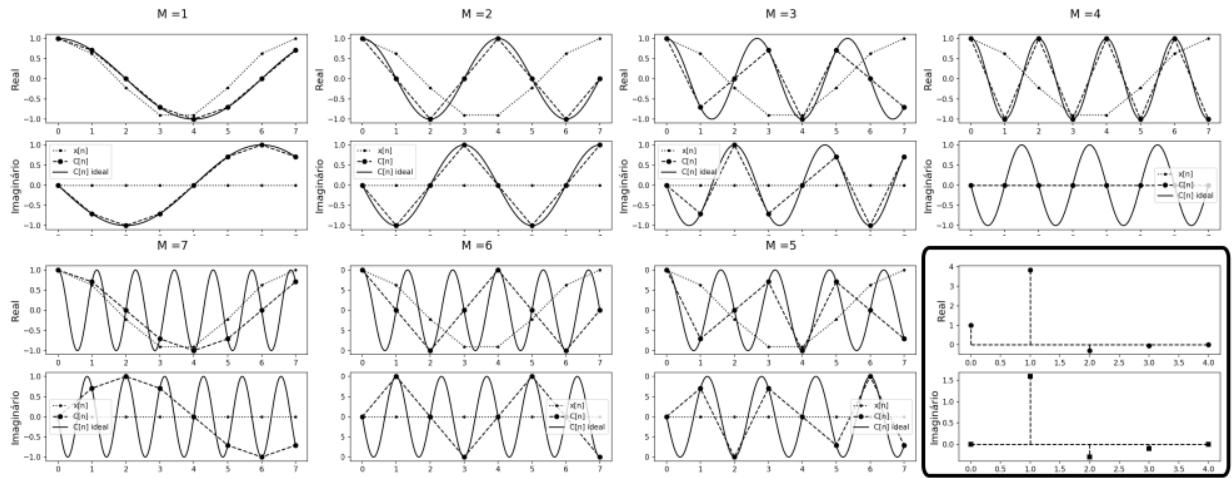


Figura 17: Transformada Discreta Real de Fourier - Interpretação Geométrica da Simetria  
Fonte: Elaboração própria

Na figura 17 é apresentado o resultado da transformada discreta de uma onda composta por 8 amostras, que pode ser vista no retângulo inferior da imagem. Os outros retângulos ilustram os respectivos passos e a simetria entre passos equidistantes de  $n/2$ . O motivo da simetria, causado pela perda de informação ao tomar apenas alguns pontos da onda contínua, torna-se explícito quando observado do ponto de vista geométrico: os 8 pontos equidistantes tomados como amostra, por exemplo, de uma senoide com frequência igual a 2 coincidem com os retirados de uma senoide com frequência igual a 6, já que partes do ciclo são ignoradas.

O código em Python utilizado para gerar os quadros da figura 17, serve para elucidar numericamente o exposto, e está disponível no repositório elaborado como acompanhamento ao presente trabalho, no caminho “Demonstração Transformada de Fourier/fourier.py”. É interessante notar que a interpretação geométrica da simetria da transformada discreta de Fourier aplicada em ondas puramente reais, até onde alcança o conhecimento do autor, é apresentada pela primeira vez neste trabalho. Cabe observar que implementações mais eficientes, como o referido algoritmo FFT, utilizadas na prática, tiram proveito da simetria da transformada, entre outras alterações mais sofisticadas, para economizar operações (Sorensen *et al.*, 1987).

### 3.2 A EQUAÇÃO DA ONDA

A equação diferencial que descreve o movimento de uma onda em duas dimensões pode ser derivada a partir das leis de Newton, assumindo algumas simplificações (Garrett, 2017). Considerando a figura 18:

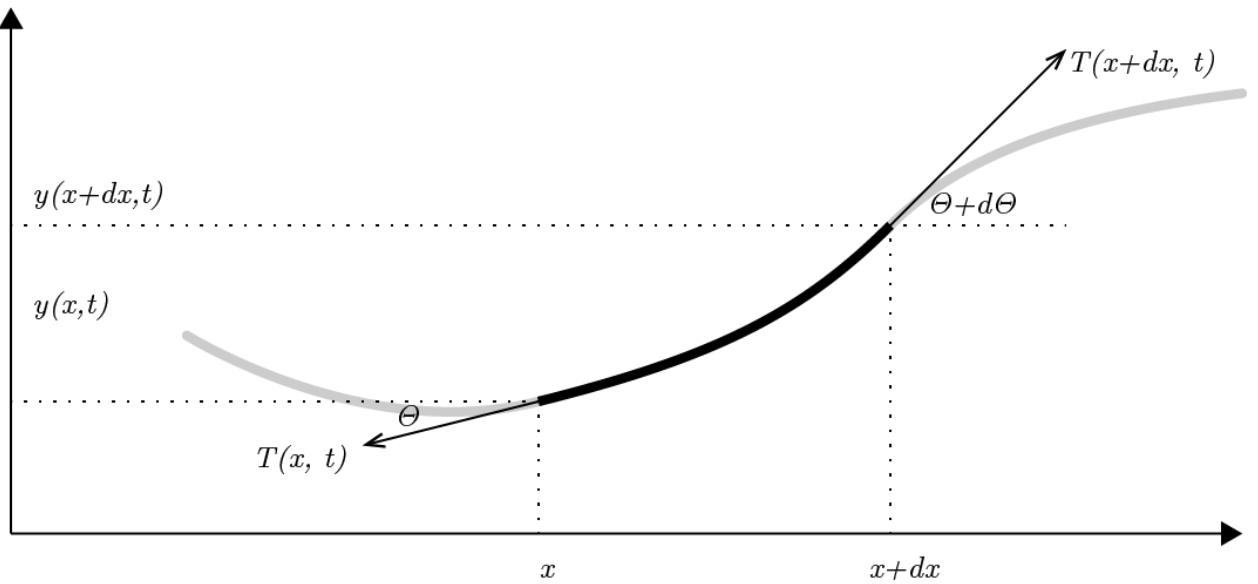


Figura 18: Representação de um segmento de corda  
Fonte: Elaboração própria

e as seguintes definições:

$y(x, t)$  a distância entre um ponto qualquer de uma corda e o eixo horizontal no tempo  $t$  e na posição  $x$ ;

$\theta(x, t)$  o ângulo entre a tangente da corda na posição  $x$  e no momento  $t$  e a direção horizontal;

$\vec{T}(x, t)$  o vetor representando a tensão na corda, no ponto  $x$  e no momento  $t$ ;

Assumindo ainda que qualquer ponto da corda move-se somente na posição vertical temos que a resultante das forças é também vertical, e pode ser descrita como:

$$\begin{aligned} F(x, t) &= T(x + dx, t) \sin(\theta + d\theta) - T(x, t) \sin(\theta) \\ &= T(x + dx, t) \cos(\theta + d\theta) \tan(\theta + d\theta) - T(x, t) \cos(\theta) \tan(\theta) \end{aligned}$$

se considerarmos somente pequenos deslocamentos verticais da corda em relação à sua posição de equilíbrio, temos que  $\theta \ll 1$  e, portanto  $\cos(\theta + d\theta) \approx \cos(\theta) \approx 1$ .

Além disso, lembrando que as tangentes podem ser escritas como a derivada parcial da função de deslocamento em relação à posição, temos:

$$F(x, t) = T(x + dx, t) \left( \frac{\partial y(x + dx)}{\partial x} \right) - T(x, t) \left( \frac{\partial y(x)}{\partial x} \right)$$

Assumindo tensão uniforme e invariável em toda a extensão da corda, condições bastante razoáveis no contexto de instrumentos musicais, podemos escrever  $T(x + dx, t) = T(x, t) = T$  e  $F(x, t)$  toma a forma  $T \left( \frac{\partial y(x+dx)}{\partial x} - \frac{\partial y(x)}{\partial x} \right)$  quando colocamos  $T$  em evidência.

Se assumirmos que a massa do segmento infinitesimal da corda pode ser escrita na forma

$\rho dx$ , onde  $\rho$  é a massa da corda por unidade de comprimento e  $dx \approx \sqrt{dx^2 + dy^2}$  para oscilações pequenas, aplicando a segunda lei de Newton para força resultante na corda temos  $F(x, t) = dx$ .

Assim, temos  $T \left( \frac{\partial y(x+dx)}{\partial x} - \frac{\partial y(x)}{\partial x} \right) = \rho dx \frac{\partial^2 y}{\partial t^2}$  que podemos reorganizar como  $\frac{\partial^2 y}{\partial t^2} = \frac{T}{\rho} \frac{\left( \frac{\partial y(x+dx)}{\partial x} - \frac{\partial y(x)}{\partial x} \right)}{dx}$ . Notando que o termo à direita é a segunda derivada de  $y$  em relação a  $x$ , chegamos à equação da onda:

$$\frac{\partial^2 y(x, t)}{\partial t^2} = \frac{T}{\rho} \frac{\partial^2 y(x, t)}{\partial x^2}$$

onde  $v = \sqrt{\frac{T}{\rho}}$  é a velocidade de propagação da onda na corda.

Uma solução para essa equação, quando não há uma fonte de excitação, foi proposta por d'Alembert (Chaigne e Kergomard, 2016), na forma:

$$y(x, t) = F(x + vt) - G(x - vt)$$

Essa equação pode ser interpretada como dois pulsos viajando em sentidos opostos em uma corda ideal com velocidade  $v$ .

Em  $t = 0$  podemos escrever:

$$\begin{aligned} y_0(x) &= y(x, 0) = F(x) - G(x) \\ v_0(x) &= y'(x, 0) = vF'(x) - vG'(x) \\ vF(x) - vG(x) &= \int_{-\infty}^x v_0(\epsilon) d\epsilon \end{aligned}$$

O que nos dá um sistema de equações que, resolvido, nos permite reescrever a equação de d'Alembert em função das condições iniciais de deslocamento e velocidade na corda, como abaixo:

$$y(x, t) = \frac{1}{2} (y_0(x + vt) - y_0(x - vt)) + \frac{1}{2v} \left( \int_{x-vt}^{x+vt} v_0(\epsilon) d\epsilon \right)$$

Essa formulação é importante já que em muitos casos pode-se trabalhar apenas com o deslocamento inicial da corda, o que torna o termo à direita zero.

Alternativamente, pelo método da separação de variáveis, pode-se obter a solução em termos de uma soma infinita de senoides estacionárias (Gracia e Sanz-Perela, 2016):

$$y(x, t) = \sum_{n=1}^{\infty} (a_n \cos(2\pi f_n t) + b_n \sin(2\pi f_n t)) \sin\left(\frac{n\pi x}{L}\right)$$

onde considera-se uma corda de comprimento  $L$  fixa em suas extremidades.  $f_n$  são as frequências parciais da corda e são dados pela equação  $\frac{nv}{2l}$ . Além disso, os coeficientes  $a_n$  e  $b_n$  são dados pelos coeficientes da transformada de Fourier das condições iniciais em  $t = 0$  ( $y_0(x)$  e  $v_0(x)$ ) (Salsa, 2016)

essas duas interpretações dão origem às duas escolas distintas de modelagem acústica, como referido anteriormente, uma focada no domínio do tempo e outra no domínio da frequência. Repare, no entanto, que ambas as soluções modelam uma corda ideal, não levando em conta a rigidez encontrada em cordas reais.

No domínio do tempo, Smith, (1992) aponta que a rigidez implica no fato de que ondas com diferentes frequências viajam através da corda em velocidades diferentes, relação regida por  $c(w) = c_0 \left( \frac{1+kw^2}{2Kc_0^2} \right)$  com  $k = \frac{E\pi r^4 \pi}{4}$  onde  $E$  é o modulo de Young e  $r$  é o raio da corda.

No domínio da frequência Rigaud, David e Daudet, (2013) demonstram que a rigidez pode ser incorporada ao modelo através da substituição do termo que descreve as frequências de cada parcial por  $fn = f_0 \sqrt{1 + Bn^2}$  onde  $f_0 = \frac{1}{2l} \sqrt{\frac{T}{\rho}}$  e  $B = \frac{\pi^3 Ed^4}{64 T l^2}$  onde  $E$  é o modulo de Young e  $d$  é o diâmetro da corda.

A maneira mais simples de acomodar o decaimento devido a forças dissipativas, completando a teoria necessária ao presente trabalho, é modular as amplitudes em ambas as soluções a partir de um termo da forma  $e^{-\alpha t}$

### 3.3 FORMALIZAÇÃO DE UMA REDE NEURAL - FORWARD PASS

Definindo-se de maneira esquemática uma camada de uma rede neural artificial conforme a figura 19:

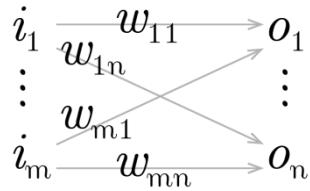


Figura 19: representação de uma camada de uma rede  
Fonte: Elaboração própria

Onde:

$I_{1 \times m} := [i_1 \dots i_m]$  é um vetor de entradas à rede, e  $O := [o_1 \dots o_n]$  um vetor de saídas, de forma que para cada um dos vetores de entrada a rede produz um vetor de saída, com o

objetivo de aproximar um vetor de alvos  $T_{1 \times n} := [t_1 \dots t_n]$ , com a mesma dimensão da saída da camada.

$W_{m \times n} := \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \dots & w_{mn} \end{bmatrix}$  é uma matriz representando os pesos das ligações entre

os neurônios de entrada e saída, de forma que  $w_{ij}$  seria o peso da ligação entre o  $i$ -ésimo neurônio de entrada e o  $j$ -ésimo neurônio de saída.

$A_{1 \times n} := [a_1 \dots a_n] \mid a_y := w_{1y} i_1 + \dots + w_{ny} i_n$  representa cada elemento da multiplicação entre o vetor de entrada e os pesos da camada, antes da aplicação, elemento à elemento, da função de ativação;

$O_{1 \times n} := [o_1 \dots o_n] \mid o_y := f(a_y) = f(w_{1y} i_1 + \dots + w_{ny} i_n)$  é um vetor representando as saídas finais da camada;

$E_{1 \times n} := [e_1 \dots e_n] \mid e_y := \varepsilon(o_y, t_y) = \varepsilon(f(w_{1y} i_1 + \dots + w_{ny} i_n), t_y)$  representa os erros individuais da rede para cada uma das saídas motivadas por cada uma das entradas.

Cabe notar que  $\varepsilon(o_y, t_y) = (o_y - t_y)^2$  é a forma mais usual de cômputo dos erros individuais; define-se ainda o escalar  $q := e_1 + \dots + e_n$  como a soma dos erros individuais para cada um dos vetores de erros.

De posse das definições acima, somos capazes de representar de forma compacta, em notação matricial, o movimento de predição (*forward pass*) de uma camada, como será visto. E, por extensão, de uma rede, na medida em que estas podem ser obtidas pela justaposição de um número arbitrários de camadas.

O mesmo pensamento, de forma simétrica, presta-se à definição do movimento que atualizará os pesos (*backward pass*), e que é formalizado na próxima seção. Assim, derivaremos a seguir a forma básica do algoritmo de *backpropagation*.

Além de introduzir um referencial matemático para o algoritmo, isso permitirá que suas extensões, como Adam e Adagrad, utilizadas adiante, possam ser melhor compreendidas.

### 3.4 FORMALIZAÇÃO DE UMA REDE NEURAL - BACKPROPAGATION

Tem-se que o gradiente do tensor de erros em relação a um peso arbitrário de uma camada pode ser escrito como na equação 2, com aplicação direta da regra da cadeia, e observando a seguinte notação para uma função qualquer  $h(x) : \frac{\partial}{\partial x} h(x) = h'(x)$ .

$$\frac{\partial q}{\partial w_{xy}} = \varepsilon'(o_y, t_y) f'(a_y) i_x \quad (2)$$

A matriz de incrementos para cada um dos pesos, a cada iteração, pode ser definida como abaixo, com a adição de uma taxa de aprendizado  $\alpha$  com sinal negativo. Assim é, uma vez que o gradiente acima aponta para sentido de maior crescimento dos erros no espaço dos pesos: minimizar os erros implica, portanto, em mover os pesos em sentido oposto.

$$\Delta W_{m \times n} := \begin{bmatrix} \Delta w_{11} & \dots & \Delta w_{1n} \\ \vdots & \ddots & \vdots \\ \Delta w_{m1} & \dots & \Delta w_{mn} \end{bmatrix} \mid \Delta w_{xy} := -\alpha \frac{\partial q}{\partial w_{xy}} = -\alpha \varepsilon'(o_y, t_y) f'(a_y) i_x$$

Resta definir uma forma de propagar os erros às entradas da rede, permitindo assim que um número arbitrário de camadas sejam conectadas e treinadas. Para tanto, convém considerar uma camada anterior à rede em tela, com a forma e notação apresentadas na figura 20:

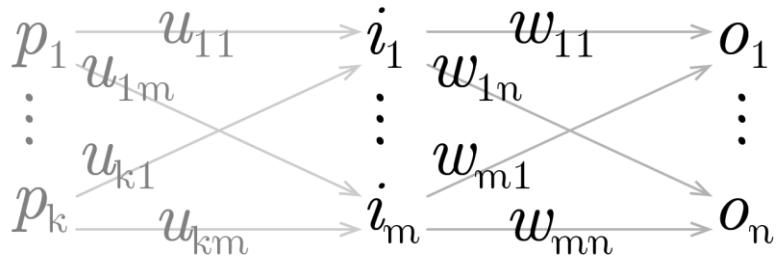


Figura 20: Duas camadas adjacentes  
Fonte: Elaboração própria

E, de forma análoga:

$$\begin{aligned} Z_{1 \times m} &:= [z_1 \dots z_m] \mid z_x := u_{1x} p_1 + \dots + u_{kx} p_k \\ I_{1 \times m} &= [i_1 \dots i_m] \mid i_x := g(z_x) = g(u_{1x} p_1 + \dots + u_{kx} p_k) \end{aligned}$$

$$\begin{aligned} \frac{\partial q}{\partial u_{rx}} &= [\varepsilon'(o_1, t_1) f'(a_1) w_{x1} g'(z_x) p_r + \dots + \varepsilon'(o_n, t_n) f'(a_n) w_{xn} g'(z_x) p_r] \\ &= [\varepsilon'(o_1, t_1) f'(a_1) w_{x1} + \dots + \varepsilon'(o_n, t_n) f'(a_n) w_{xn}] g'(z_x) p_r \end{aligned}$$

Comparando com  $I$  é fácil observar que o somatório na primeira parte da última expressão corresponde a derivada do erro na primeira equação. Assim, podemos definir o tensor do erro propagado da forma apresentada abaixo:

$$E_{1 \times m}^b := [e_1^b \dots e_m^b] \mid e_x^b := \varepsilon'(o_1, t_1) f'(a_1) w_{x1} + \dots + \varepsilon'(o_n, t_n) f'(a_n) w_{xn}$$

Fixamos, dessa forma, todos os termos necessários à uma arquitetura modular: para camadas ocultas, para as quais não há como aferirmos diretamente o erro, o mesmo é propagado a partir das camadas posteriores. Em forma vetorial, pode-se sintetizar o algoritmo como na equação 3, observando a conveniência de introduzirmos o tensor  $H_{1 \times n}$  para evitar redundância nos cálculos, e a notação  $\odot$  denotando o produto de Hadamard (elemento a elemento) entre dois tensores:

$$\begin{aligned} O &= f(IW) \\ H_{1 \times n} &\coloneqq E \odot f'(A) \\ \Delta W &= -\alpha I^t H = -\alpha I^t (E \odot f'(A)) \\ E^b &= HW^t = (E \odot f'(A))W^t \end{aligned} \quad (3)$$

## 4 METODOLOGIA

### 4.1 AMOSTRAS SONORAS UTILIZADAS

Para a busca da coleção de amostras sonoras a serem utilizados para o treinamento da rede, algumas características são essenciais. Em primeiro lugar, uma licença permissiva, que não limite a utilização, modificação e posterior divulgação dos resultados obtidos deve ser explicitamente atribuída pelo autor.

Além disso, para o caso da bateria, todas as peças de um *kit* ordinário devem estar presentes, e idealmente, organizadas de maneira inteligível, enquanto que para instrumentos temperados é bastante útil algum tipo de indicação da frequência fundamental de cada uma das amostras.

É desejável também que, para cada peça, amostras representando várias dinâmicas tenham sido gravados, com especial atenção à intensidade com que cada peça é golpeada. Interessante ainda é a presença dos chamados *round-robin*s: gravações redundantes de cada uma das amostras, que apresentam uma ideia de como fatores aleatórios influenciam no som produzido pelo instrumento.

Com isso em mente, procedeu-se à busca, em sites e blogs especializados, de indicações sobre trabalhos disponíveis que pudessem enquadrar-se nas condições citadas. No caso do *kit* de bateria, dois trabalhos em especial foram considerados:

O primeiro é um esforço de disponibilizar, através da plataforma *Github*, uma coleção livre de sons de bateria. O trabalho leva o nome de The Open Source Drumkit (“The Open Source Drumkit”, 2015) e consiste em uma coleção de amostras em formato .wav, separadas em pastas e nomeadas de acordo com a intensidade do golpe, e as peças específicas de um kit de bateria convencional, apresentando uma média de 10 articulações por peça.

O segundo é um esforço para a elaboração de um instrumento virtual de bateria denominado DrumGizmo (“DrumGizmo Wiki”, 2018), e apresenta samples gravadas a partir de 5 kits físicos de bateria. O número de articulações é bem maior, girando em torno de 20 por kit, e as gravações apresentam múltiplos microfones, com pelo menos um microfone por peça e são, em geral, de melhor qualidade. Além disso, a maioria dos *kits* adere à licença *Creative Commons Attribution 4.0 International* que permite o livre uso e adaptação do material disponibilizado em trabalhos derivados.

No caso das amostras de piano, em “Ivy Audio”, (2015) pode ser encontrado, entre outros, um DMI baseado em samples de um Steinway Model B Grand piano, com 5

dinâmicas e 2 *round-robbins* para cada tecla, além de um total de 4 condições de microfonação. Detalhes dos termos de uso são discutidos na página de FAQ do site.

No site da University of Iowa Electronic Music Studios (“University of Iowa Electronic Music Studios”, 2011) um conjunto bastante semelhante de amostras do mesmo modelo de piano pode ser encontrado, com a vantagem de uma atribuição mais livre de condições de uso; as amostras, no entanto, diretamente hospedadas no site, devem ser baixadas uma por vez. Deve-se preferência, no treinamento da rede que presta-se à emulação de um piano, a este conjunto de amostras.

Uma vasta gama de amostras sonoras de instrumentos clássicos, com variadas dinâmicas, é oferecida no site da Philharmonia Orchestra (“Philharmonia Orchestra”, 2018); algumas das quais foram utilizadas para o treinamento de um instrumento híbrido.

## 4.2 REDES DENSAS NO DOMÍNIO DO TEMPO

É natural que a investigação tenha início com uma análise da arquitetura *Feed-Forward*, com redes compostas de um número arbitrário de camadas totalmente conectadas. Isso possibilita formar uma ideia de como essa arquitetura básica se comporta em um ambiente de alta dimensionalidade, pavimentando o caminho para a subsequente incorporação de novas arquiteturas e formulações na medida em que limites de aplicabilidade forem sendo encontrados.

Convém lembrar que o trabalho tem como foco a síntese sonora em tempo real, de forma que o tamanho, no sentido do número de neurons de uma rede neural, e sua complexidade, aspecto relacionado à arquitetura, tornam-se fatores potencialmente limitantes, e passam a requerer especial atenção. Redes convolucionais, por exemplo, devem ser tratadas com extrema cautela, por não possuírem um histórico de implementações eficientes.

Em primeiro lugar, procedeu-se ao tratamento dos samples: foram escolhidos, para esta investigação inicial, 5 tambores (os 4 tons disponíveis, e o bumbo esquerdo) do kit denominado *Aasimonster* disponibilizado pela iniciativa DrumGizmo, de cada um dos quais 20 dinâmicas foram escolhidas, gerando um total de 100 amostras, cada um com 16 canais (1 por microfone utilizado na gravação).

Foram extraídos os canais pertinentes à cada peça, resultando em 100 samples monofônicos em formato .wav, com *framerate* de 44100 frames por segundo. As dinâmicas para cada peça foram normalizadas em grupos de 4, de forma a emular 4 samples redundantes (*round-robbins*) para 5 níveis diferentes de intensidade por peça. Cada uma das amostras foi truncada, para possuir exatamente um segundo de duração, e um efeito *fade-off* foi aplicado às

amostras que estendiam-se além desse intervalo. Esse efeito consiste na introdução de um decaimento suave que diminui a amplitude da onda gradativamente, até zero no último ponto amostrado.

Esse conjunto de amostras, de forma bastante direta, representou os alvos da rede neural, na forma de um vetor de dimensão 100 x 44100, como cada uma de suas linhas consistindo em uma representação digital da onda sonora de cada amostra. A conversão foi efetuada utilizando o módulo Pywave da linguagem Python, e cada vetor foi normalizado no espaço [-1,1]. Os samples foram renomeados de forma a serem mais descritivos.

Exemplificando, temos que a amostra de rótulo 154, corresponde à quarta amostra redundante referente à primeira peça (o tom mais agudo), golpeada com intensidade 5 (maior intensidade) seria representada pelo vetor de entrada {0.00, 1.00, 1.00}. Assim, as amostras com rótulos 321, 322, 323 e 324, por exemplo, correspondem à quatro variações do mesmo evento: cada um deles representa a terceira peça golpeada com intensidade 2 e seriam representadas, respectivamente, pelos vetores de entrada {0.50, 0.25, 0.00}, {0.50, 0.25, 0.33}, {0.50, 0.25, 0.66} e {0.50, 0.25, 1.00}. Os detalhes dos dados de entrada são elencados na tabela 2.

Tabela 2: Nomenclatura utilizada

<b>Peça</b>	<b>Peça Norm.</b>	<b>Intensidade</b>	<b>Intensidade Norm.</b>	<b>Amostra #</b>	<b>Amostra # Norm.</b>
1	0.00	1	0.00	1	0.00
2	0.25	2	0.25	2	0.33
3	0.50	3	0.50	3	0.66
4	0.75	4	0.75	4	1.00
5	1.00	5	1.00	-	-

O que pretende-se é forçar uma associação de cada uma das características da onda à cada uma das 3 dimensões do vetor de entrada. Esse *rationale* tem algumas implicações importantes: critérios de validação, embora possíveis de construir, perdem bastante do seu significado e foram deixados de lado em favor de um julgamento subjetivo da qualidade das amostras geradas.

### 4.2.1 Algoritmos de Otimização

Utilizando as 100 amostras processadas como descrito, buscou-se investigar os hiperparâmetros de uma rede neural densa mais adequados ao aprendizado e reprodução das ondas; trata-se de uma etapa importante, na medida em que vários parâmetros aqui definidos serão assumidos como ótimos e utilizados em etapas posteriores, em arquiteturas diferentes.

Essa necessidade tem origem nas restrições de tempo e poder computacional, que impedem a realização de novos testes dessa magnitude para cada uma das arquiteturas futuramente investigadas. A plataforma escolhida para a implementação das redes foi o Keras, rodando com o Tensorflow como *backend*. Essa escolha permite aproveitar a velocidade de experimentação conferida pela arquitetura em nível mais alto da biblioteca Keras sem comprometer muito a flexibilidade.

Em primeiro lugar, é conveniente investigar o melhor algoritmo de atualização dos pesos: embora, conceitualmente, mesmo a versão pura da descida em gradiente, utilizando uma taxa de aprendizado adequada, leve à eventual convergência da rede (Ruder, 2016), na prática a velocidade dessa convergência varia consideravelmente dependendo do método utilizado, e um algoritmo eficiente permite a realização de mais testes no mesmo intervalo de tempo.

A literatura disponível, embora esclarecedora, não é unânime ao apontar uma técnica universalmente superior às outras, de forma que uma investigação empírica faz-se necessária, e é descrita a seguir.

Testou-se, em uma rede com uma camada oculta composta por 100 neurons, em lotes (*batches*) de 100 vetores de entrada e durante 500 épocas os seguintes algoritmos de otimização: SGD com uma *learning rate* de 0,01, o algoritmo experimental RMSProp (“RMSProp: Divide the gradient by a running average of its recent magnitude”, 2012), os otimizadores Adam (Kingma e Ba, 2014; Reddi, Kale e Kumar, 2018), Adagrad (Duchi, Hazan e Singer, 2011), Adadelta (Zeiler, 2012), Adamax (Kingma e Ba, 2014) e Nadam (Dozat, 2016; Sutskever *et al.*, 2013). A função de ativação utilizada em todas as camadas foi a tangente hiperbólica, e a função de perda foi a média dos quadrados dos erros. Foi tomado cuidado para que todos os testes iniciassem em condições pseudorrandômicas idênticas.

SGD e Adadelta ofereceram resultados bastante semelhantes, e aquém dos outros. Todos os outros convergiram de forma semelhante, sendo o Adagrad o mais rápido de todos, e Nadam o mais eficaz na diminuição da função de perda.

A comparação por tempo relativo, disponível a partir do Tensorboard, permite visualizar a superioridade do algoritmo Nadam: com 2,57 minutos de treino, o tempo em que o treinamento foi concluído com o algoritmo Adagrad, ele proporcionou o menor valor da função objetivo.

Convém investigar, dessa forma, o comportamento da rede treinada com ambos os algoritmos, já que o parâmetro mais importante é a rapidez com que a rede alcança um nível de erro capaz de gerar samples com qualidade satisfatória. Após essa comparação exploratória, mais profunda, poderemos identificar tanto o valor limite da função de perda, quanto o tempo em que a rede mais eficiente leva para alcançá-lo, utilizando essas informações como parâmetros para investigações posteriores, inclusive de outras arquiteturas.

#### **4.2.2 Busca em grade para topologia e parâmetros**

Tendo sido fixados os potenciais algoritmos de otimização, Adagrad e Nadam, procedeu-se a uma busca em grade, para investigar o número de camadas ocultas, respectivos neurons e o efeitos do tamanho dos lotes, tanto em relação ao impacto no tempo total de treinamento quanto à rapidez (e limite) de convergência.

##### **4.2.2.1 Adagrad**

Testou-se lotes de 1, 10, 50, 100, 150 e 200 samples para redes com 25, 50, 75 e 100 neurons em cada camada oculta, em um intervalo de 0 a 3 camadas ocultas, gerando um total de 82 redes distintas, que foram treinadas durante 500 épocas.

Para uma rede sem camadas ocultas, um lote igual a 1 foi o mais efetivo, em 500 épocas, porém o mais lento. Lotes de 100 e 50 foram os mais rápidos. O impacto no valor final da perda, no entanto, foi desprezível para todos os tamanhos de lotes. Para uma camada oculta com 25 neurons, lotes de 100 e 150 neurons ofereceram resultados idênticos, sendo os mais rápidos e mais eficientes.

A partir desse ponto, lotes de 100, 150 e 200 começaram a apresentar comportamentos semelhantes. Cabe notar que lotes unitários parecem piorar muito o tempo de execução na medida em que o número de neurons aumenta. Temos abaixo, na tabela 3, os melhores resultados por lote e seus respectivos tempos. Todos eles encontram-se nas redes com 3 camadas ocultas.

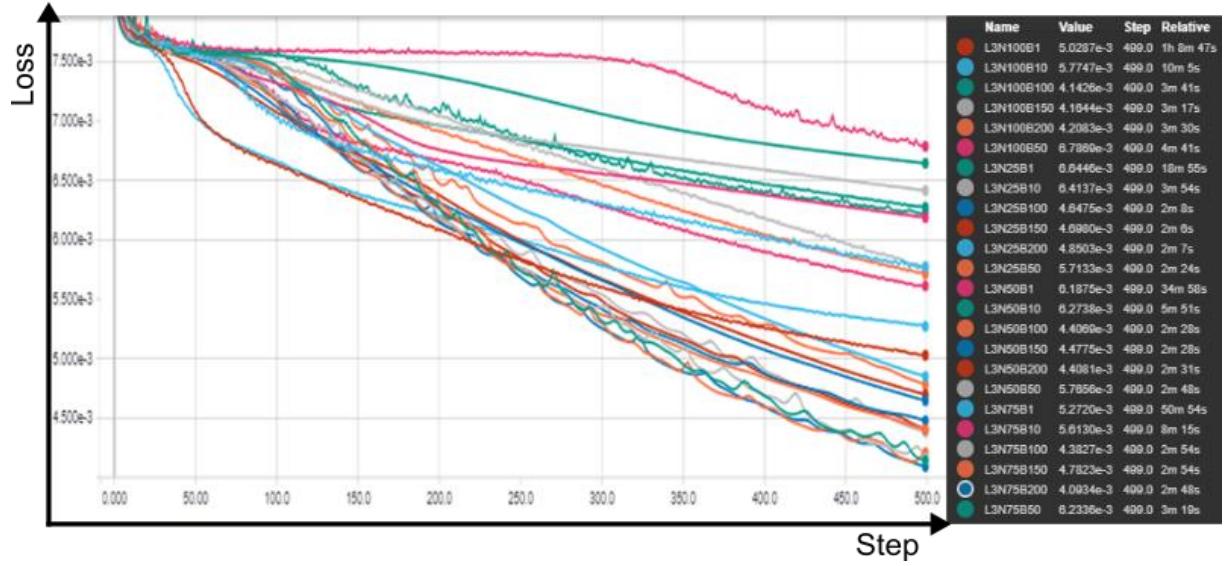


Figura 21: Adagrad: Loss x Step - 3 camadas ocultas

Fonte: Elaboração própria

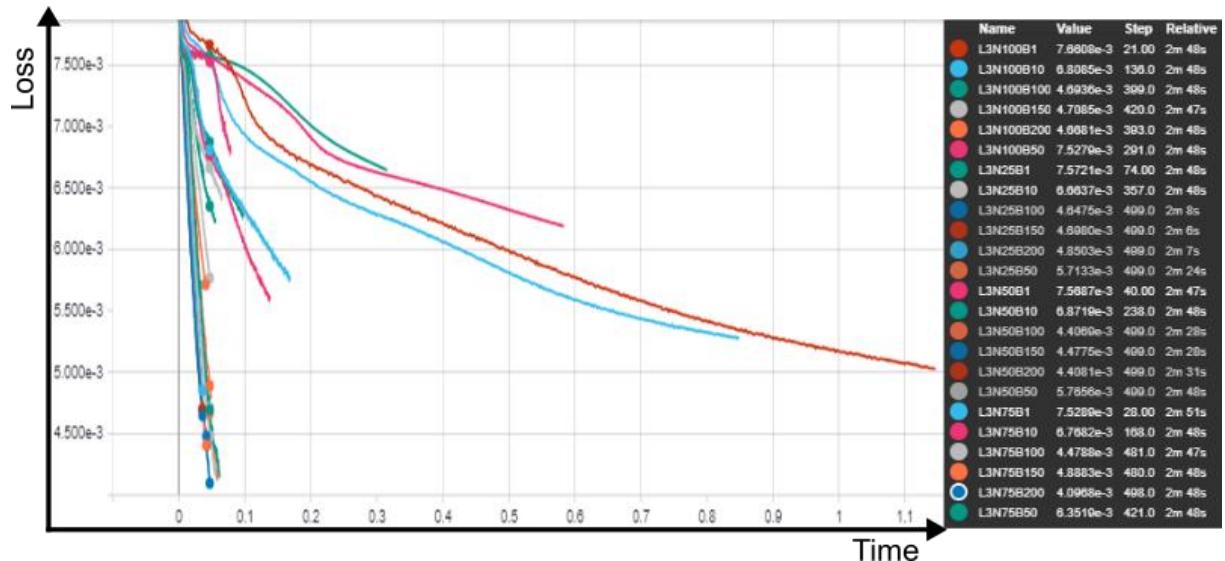


Figura 22: Adagrad: Loss x Time - 3 camadas ocultas

Fonte: Elaboração própria

Tabela 3: Menores valores da perda após 500 iterações

Lotes	neurons por camada oculta	Loss (e-3)	tempo
200	75	4,0934	2m 48s
100	100	4,1426	3m 41s
150	100	4,1644	3m 17s
1	50	4,4069	2m 28s
10	75	5,6130	8m 15s
50	25	5,7133	2m 24s

Nesse ponto fica evidente que uma arquitetura formada por 3 camadas ocultas de 75 neurons é suficiente para aprender uma representação direta das ondas, e a mais eficiente sob o otimizador Adagrad. O gráfico da função de perda em função do tempo de treinamento, apresentado na figura 22, torna essa visualização mais fácil.

#### 4.2.2.2 Nadam

Para essa etapa lotes de 10, 50 e 100 amostras sonoras foram utilizados; dispensou-se lotes unitários, pelo comportamento ineficiente identificado, e os lotes acima de 100, já que não introduziram melhorias significativas. Foram testadas, para cada um desses lotes, redes com todas as combinações entre 25, 50, 75 e 100 neurons, até o limite de 3 camadas ocultas. Os resultados são apresentados nas figuras 23 e 24, para as arquiteturas com 3 camadas.

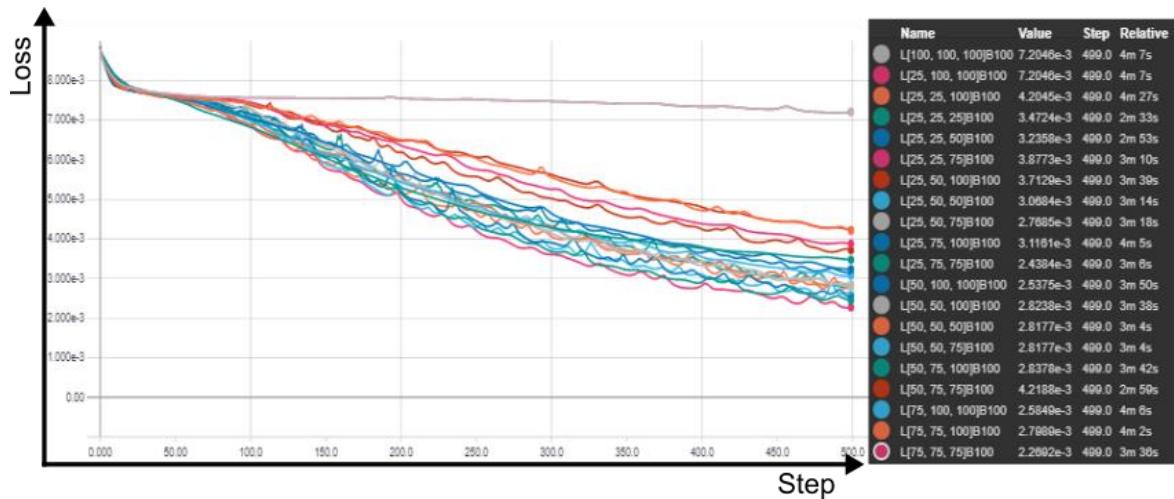


Figura 23: Nadam: Loss x Step - 3 camadas ocultas

Fonte: Elaboração própria

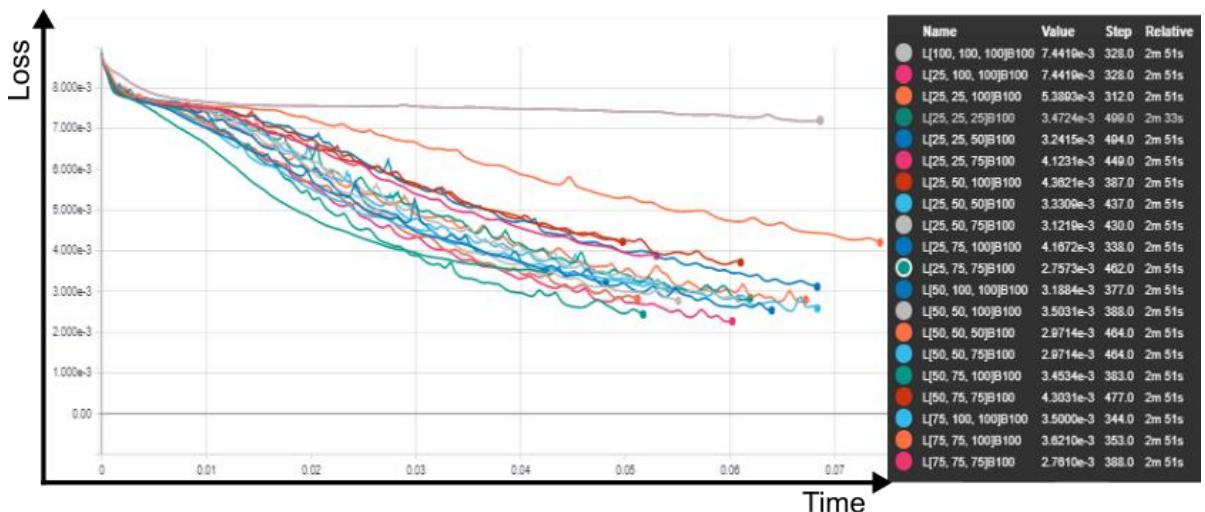


Figura 24: Nadam: Loss x Time - 3 camadas ocultas

Fonte: Elaboração própria

Tabela 4: Menores valores da perda após 500 iterações

Lotes	neurons por camada oculta	Loss (e-3)	tempo
10	75 100 100	0.44632	16m 59 s
10	50 50 100	1.5559	15m
100	25 75 75	2,4384	3m 6 s
50	75 75 75	2.2692	3m 36 s
100	75 75 75	2.2692	3m 38 s
50	75 100	4.9025	5m 4s

Uma rede *feedforward* com 3 camadas de 75 neurons parece oferecer um compromisso razoável entre tempo de treinamento e função de perda com ambos os otimizadores. Esses hiperparâmetros servirão da base para arquiteturas futuras desenvolvidas neste trabalho.

#### 4.2.2.3 Ativações e Otimizadores

A performance dos otimizadores Adagrad e Nadam, para 2000 *steps*, é apresentada na figura 25, onde também é comparado o desempenho da função de ativação softsign, em relação à tanh utilizada até agora.

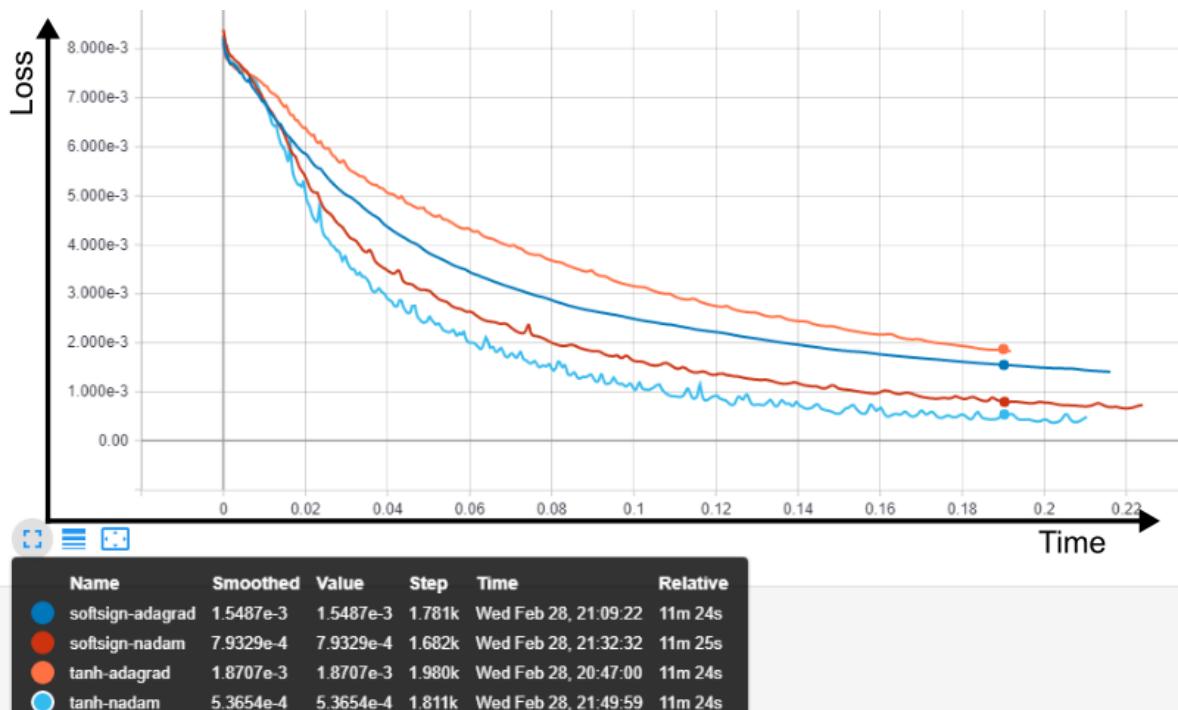


Figura 25: Comparação: Ativações e Otimizadores  
Fonte: Elaboração própria

Podemos ver que a combinação do otimizador Nadam com a função de ativação tanh é a mais eficiente.

#### 4.2.3 Análise dos resultados

Os testes acima proporcionam uma visão objetiva sobre o comportamento geral da arquitetura densa na resolução do problema em tela. Faz-se necessário, contudo, uma análise das amostras de som geradas pelas redes investigadas, de forma a entender, do ponto de vista sensorial, sua verossimilhança, além da capacidade de generalização da rede.

Tomando como métrica o valor da função de perda ao final do treinamento, podemos selecionar algumas redes para investigar seu comportamento.

Analisando a rede com maior valor de função de perda após o treinamento (a rede sem camadas ocultas treinada com Adagrad em lotes de 100), a rede escolhida como baseline (75 neurônios em cada uma de 3 camadas ocultas, otimizador Nadam, lotes de 100) e a rede com a menor perda após o treinamento (a rede com 75, 100 e 100 neurônios em suas camadas ocultas, treinada com Nadam em lotes de 10), desenvolveremos alguma intuição acerca da relação entre a função de perda e a verossimilhança das amostras produzidas. Os dados são sintetizados na tabela 5:

Tabela 5: Redes para análise das amostras sonoras

Nº	Camadas	Lote	Otimizador	Valor final da função de perda	Parâmetros
01	75 100 100	10	Nadam	4.4632 E-4	4,472,100
02	75 75 75	100	Nadam	2.2692 E-3	3,363,300
03	0	100	Adagrad	7.6421 E-3	176,400

O formato das ondas das amostras da primeira peça da bateria, segundos *round-robbins*, são mostrados na coluna à esquerda da figura 26, enquanto que as outras colunas mostram os resultados das redes. As amostras podem ser ouvidas no repositório preparado para o trabalho (“Repositório Dissertação”, 2018), no caminho “resources/01\_wavs\_FF/”.

Nas linhas da figura 26 em que não há um alvo, os resultados foram generalizados pelas redes. A primeira rede apresenta um balanço razoável entre acurácia e poder de generalização, enquanto a segunda, com um número um pouco menor de parâmetros, introduz uma quantidade considerável de ruído ao fim das amostras. A terceira, consideravelmente

menor, parece ter aprendido uma representação intermediária entre os exemplos, que varia apenas ligeiramente de peça para peça.

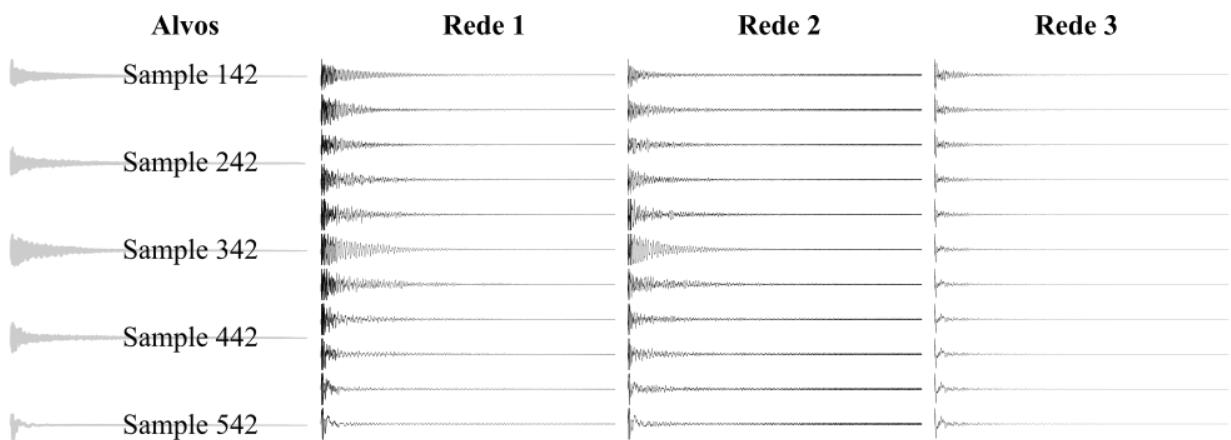


Figura 26: Resultados: Redes Densas  
Fonte: Elaboração própria

Cabe observar que a primeira rede, com pouco mais de 4 milhões de parâmetros, que ocupam em disco aproximadamente 50 MB, cobre apenas uma família de peças da bateria, implicando que um conjunto completo de redes girará em torno de 200MB, na medida em que uma rede ocupe-se dos tambores (incluindo o bumbo), como foi o caso, uma outra da caixa, um terceira do contratempo, e uma quarta para os pratos. Embora o tamanho seja razoável, sobretudo à luz dos instrumentos virtuais comerciais de ponta, como o Superior Drummer 3 da empresa Toontrack, com 230GB, estratégias podem ser utilizadas para aumentar a eficiência da rede.

Isso torna-se especialmente importante se lembrarmos que limitamos a duração das amostras ao tempo de 1 segundo. Para os tambores, caixa e a maior parte das articulações do contratempo, essa duração é razoável. Os pratos, contudo, principalmente quando alvos de um ataque mais forte, costumam apresentar um tempo de decaimento de aproximadamente 10 segundos, o que demandaria uma rede no mínimo 10 vezes maior.

Em outras palavras, essa arquitetura não se presta à generalização da duração de uma onda sonora, já que impõe um limite rígido à duração máxima que poderá ser gerada, e é inaplicável, ao menos de uma modo direto, à emulação de instrumentos que possuem o som sustentado pela injeção constante de energia da parte do instrumentista, como é o caso dos metais, como trombone, saxofone, etc., e instrumentos da família do violino, onde, respectivamente, o sopro e o arco injetam constantemente energia no sistema, e o som pode ser mantido por estendidos períodos de tempo.

### 4.3 REDES RECORRENTES

Os resultados obtidos com a utilização de redes densas encorajam a investigação do comportamento de redes recorrentes, já que possuem o potencial de diminuir o número de parâmetros, apresentando ainda a possibilidade de generalização do tempo de duração dos sons gerados. Uma escolha sensível é dividir os alvos em intervalos de 4410 samples, o que equivale a intervalos de 0.1 segundo para o *sampling rate* de 44100 FPS dos arquivos utilizados, gerando intervalos correspondentes, aproximadamente, à 5 vezes a resolução temporal mínima do ouvido humano (Fastl e Zwicker, 2007). Assim, a arquitetura recorrente apresentada na figura abaixo foi escolhida. A recorrência ocorre na quinta camada, que implementa uma arquitetura GRU (Gated Recurrent Unit).

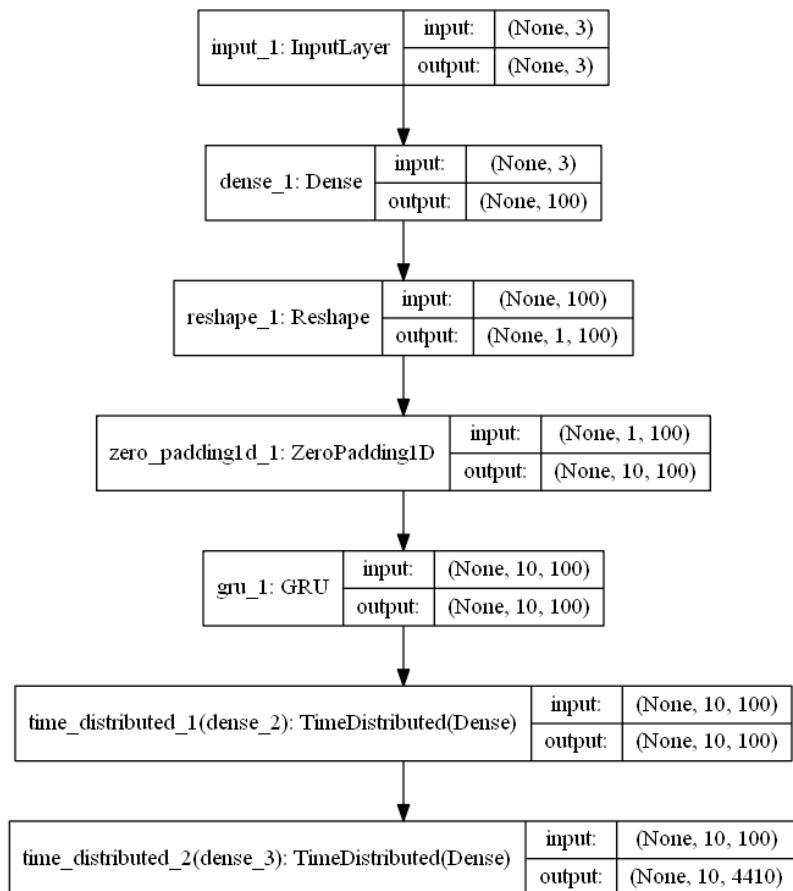


Figura 27: Topologia RNN  
Fonte: Elaboração própria

Com 516210 parâmetros treináveis (contra 4472100 da rede densa), essa topologia gera resultados semelhantes aos da rede totalmente conectada, tanto em relação à função de perda quanto à capacidade de generalização, além de possuir um tamanho em disco aproximadamente 10 vezes menor. A figura 28 ilustra as ondas geradas correspondendo às

amostras 454 e 554, e a generalização da rede para uma amostra entre essas duas, correspondente à um tom entre o tom mais grave e o bumbo. Esse intervalo foi escolhido justamente por apresentar a maior disparidade, tanto sonora quanto visual, entre as ondas.

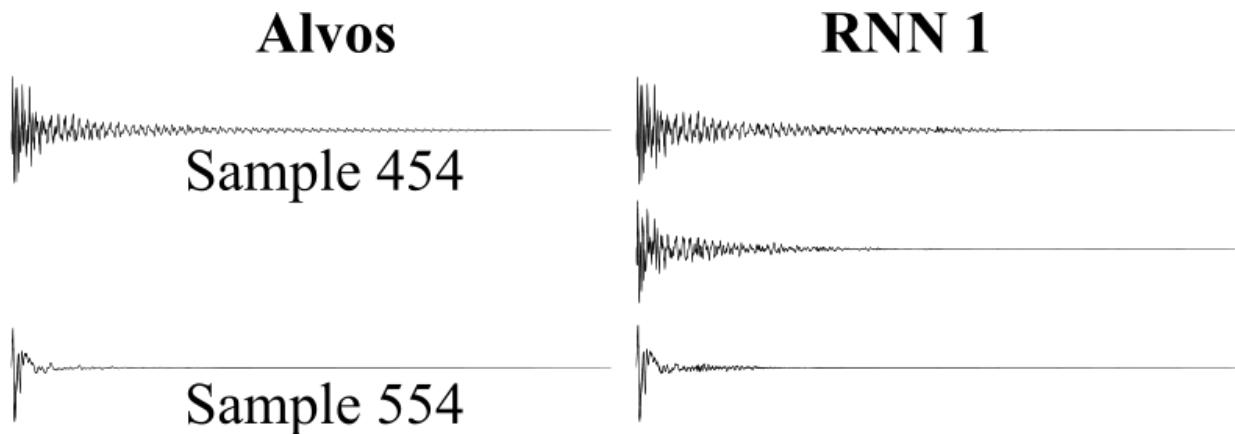


Figura 28: Resultados Rede Recorrente  
Fonte: Elaboração própria

Embora o valor final da função de perda seja de 1.1770e-4, a análise do som gerado pelas ondas revela que as pequenas descontinuidades entre duas previsões subsequentes (não visíveis na figura acima) traduzem-se em um ruído de batidas, o que compromete a qualidade do som gerado pela rede.

Uma estratégia adotada para contornar o problema da descontinuidade, inspirada no funcionamento das redes convolucionais, foi adotar uma interseção entre previsões subsequentes da rede; essa região seria ao fim multiplicada por uma função *smoothstep* decrescente, da forma  $1 - (3x^2 - 2x^3)$ , definida no intervalo entre 0 e 1, a fim de equacionar de forma suave as contribuições dos dois passos da rede, sem introduzir novas descontinuidades. A arquitetura proposta é ilustrada abaixo, e de fato é bastante eficaz em eliminar as descontinuidades diretas entre as previsões.

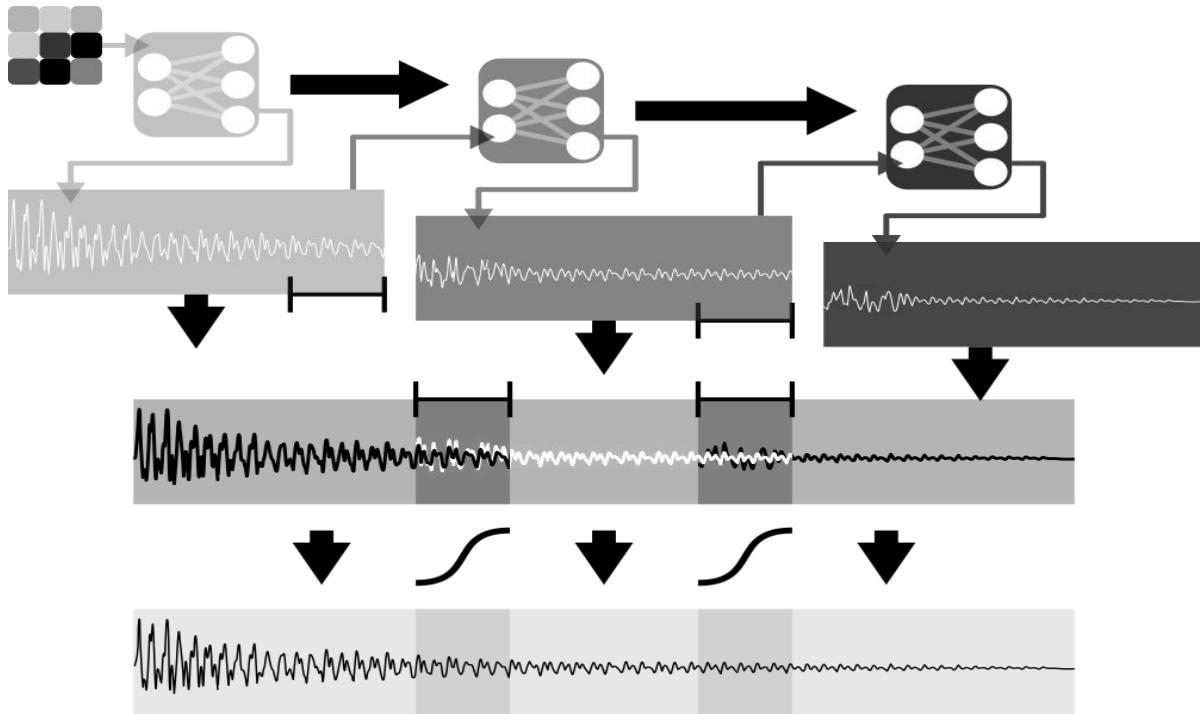


Figura 29: Arquitetura - Convolução Alisada  
Fonte: Elaboração própria

Os quadrados na região superior esquerda da figura representam os vetores de entrada iniciais fornecidos no primeiro passo. Nos passos subsequentes, a rede recebe seus próprios vetores de saída, e a partir deles gera a fração seguinte da onda, de forma recursiva, em uma configuração denominada *one-to-many* (na prática, como pode ser observado no código do programa disponibilizado no repositório dedicado ao trabalho (“Repositório Dissertação”, 2018), encontrando no caminho “resources/02\_RNN.py”, as entradas recebem o vetor de entrada em cada passo, ou um vetor nulo a partir do segundo passo. Após algumas épocas, no entanto, a rede aprende a ignorar essas entradas adicionais).

Todas as frações de onda geradas, com exceção da última, possuem uma interseção com o início da fração seguinte, e a influência de cada uma das previsões no resultado final dá-se a partir dos pesos retirados da função *smoothstep*: O primeiro ponto da região de interseção é determinado, portanto, inteiramente pela previsão mais antiga enquanto o último o é totalmente pela previsão mais recente. A influência majoritária nos pontos internos passa gradativamente, de maneira suave, da previsão mais antiga para a mais nova.

Contudo, nos resultados alisados persiste ainda um problema relacionado, principalmente, ao transiente no meio onde propaga-se a perturbação, uma etapa bastante malcomportada que ocorre logo após o impulso inicial ser aplicada à corda, membrana ou prato e durante a qual o regime (semi) periódico não foi ainda alcançado. Isso resulta, do

ponto de vista da onda, em uma etapa curta, que pode ser vista no início da figura 30, e que apresenta vibrações aberrantes em frequências diferentes e maiores do que frequência fundamental do sistema.

Essas frequências são capturadas pela rede recorrente, e equivocadamente aplicadas ao início de cada um dos seus passos (e não só ao primeiro). A figura 30, um detalhe aumentado da onda correspondente à amostra 111, onde é capturada a fronteira entre duas previsões subsequentes, ilustra esse efeito na coluna da direita, enquanto as descontinuidades não tratadas podem ser vistas na coluna da esquerda.

As consequências sonoras podem ser ouvido no caminho “resources/02\_wavs\_RNN/” do repositório preparado para o trabalho (“Repositório Dissertação”, 2018), nos dois arquivos .wav que possuem o prefixo “beating” e refletem-se em batidas, mais acentuadas no caso das ondas sem o tratamento.

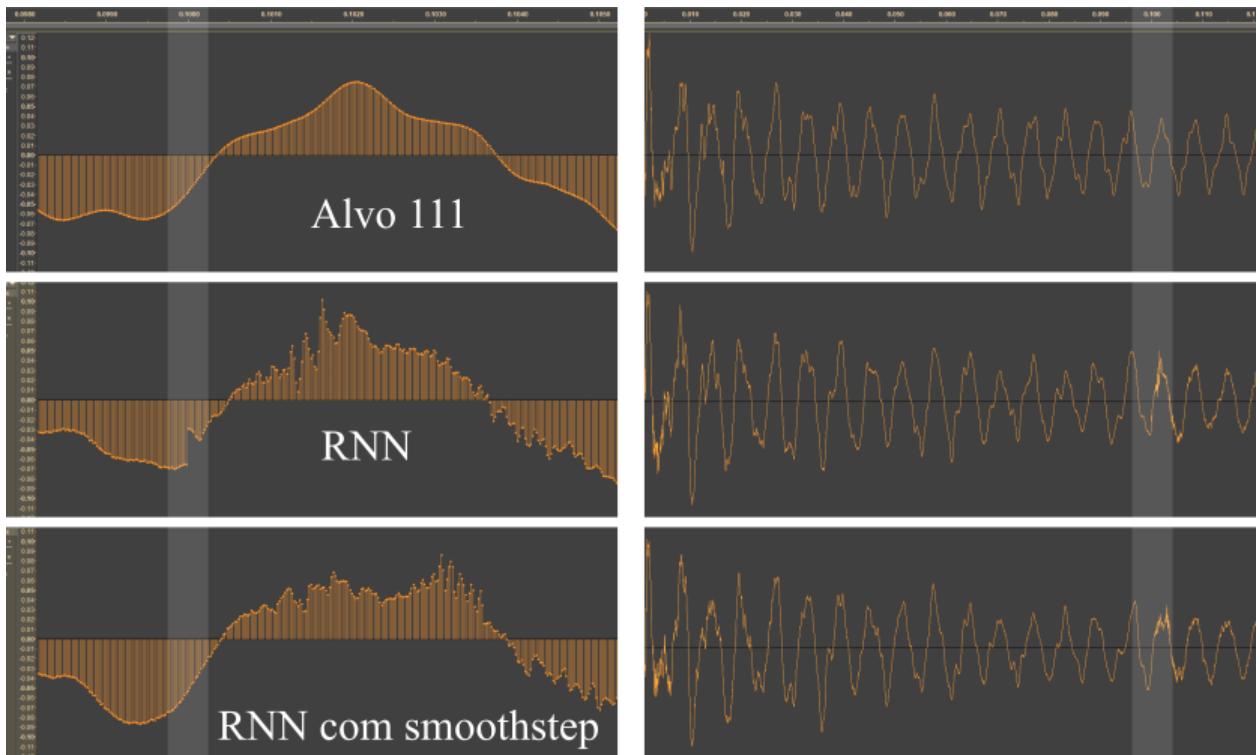


Figura 30: Descontinuidades e Transientes

Fonte: Elaboração própria

O problema é tratável, e um dos encaminhamentos possíveis, tomando como inspiração o algoritmo dos *banded digital waveguides* (Serafin, Huang e Smith, 2001) é distribuir o esforço de previsão entre mais de uma rede, por exemplo, ocupando cada uma de prever uma faixa de frequências; a saída final do modelo consistiria, então, na soma das contribuições de cada uma das redes. Observou-se, no caso dos samples utilizados até agora, que aproximadamente 4 faixas seriam necessárias, refletindo-se em 4 redes por família de

peças, em um total de 16 redes. As exatas fronteiras entre as faixas dependem tanto da arquitetura quanto das características do instrumento a ser emulado e requerem uma investigação empírica bastante extensa que deixou de ser desenvolvida neste trabalho, por limitações de tempo.

Uma outra abordagem seria reservar à uma rede densa a previsão dos transientes, deixando sob responsabilidade de uma rede recorrente o período estacionário, mais bem-comportado. É ainda possível uma abordagem experimental em que ao mecanismo de memória da rede sejam adicionados mecanismos que façam as vezes de filtros digitais passa-baixa, que tem como resultado amenizar as frequências mais altas das ondas.

#### 4.4 DOMÍNIO DA FREQUÊNCIA

Podemos tirar proveito do caráter periódico dos samples e representá-las diretamente no domínio da frequência. Utilizando a transformada de Fourier, como visto, temos à disposição uma forma de representar perfeitamente a onda utilizando um vetor de números complexos com metade da dimensão da onda original, explorando o fato de que estamos nos restringindo a representações temporais no domínio dos números reais.

A princípio, tal transformação de domínio não introduziria uma forma mais compacta de representação das ondas, do ponto de vista computacional, haja vista que números complexos são representados por pares de números reais. Levando em conta, contudo, que o ouvido humano não é capaz de perceber frequências fora da faixa entre 20 Hz e 20 KHz (Howard e Angus, 2017), identificamos uma das vantagens de trabalharmos no domínio das frequências: podemos truncar o resultado da FFT à este intervalo (tomando o cuidado de traduzi-lo em termos das frequências locais da transformada).

Além disso, trata-se de uma representação independente do tempo de duração da onda representada, o que nos permite trabalhar com uma arquitetura densa prevendo ondas de tamanhos variados. Mais à frente, serão ilustradas outras vantagens dessa abordagem, levando em conta características físicas do instrumento a ser emulado e propriedades da transformada.

A imagem a seguir compara a distribuição de frequências do som gerado por um prato de bateria e o som gerado pela Dó central de um piano (tecla 49 em um piano padrão), e ajuda a entender a diferença entre os dois instrumentos, e como, mais adiante, poderemos explorar o caráter bem-comportado dos sons produzidos por instrumentos harmônicos.

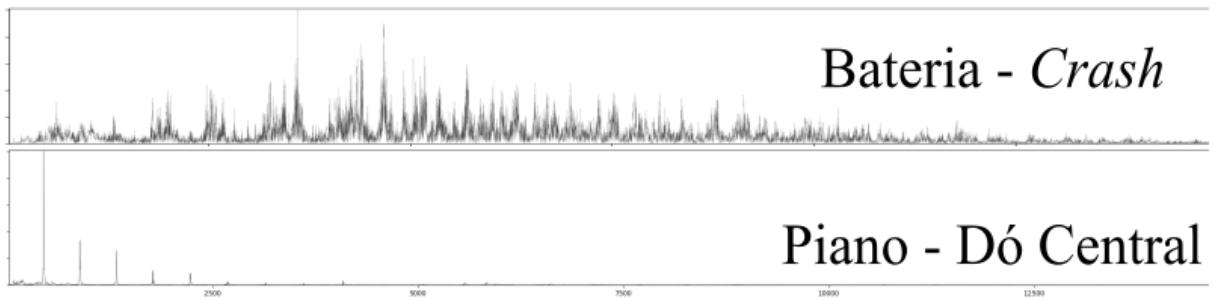


Figura 31: Espectro da intensidade das Frequências Audíveis

Fonte: Elaboração própria

Para a emulação de um *kit* completo de bateria 4 redes foram treinadas; as amostras sonoras foram divididas, de acordo com as afinidades de diferentes conjuntos de peças, e levando em conta também o grau de expressividade demandado, em quatro grupos: pratos, contratempo, caixa e tambores (tons e bumbo). As amostras utilizadas no treinamento, divididas em suas respectivas pastas, podem ser acessadas no repositório do trabalho (“Repositório Dissertação”, 2018) no caminho “resources/samples/”. De maneira semelhante, a implementação da rede está disponível no endereço “resources/02\_RNN.py”.

Somente para a caixa e tons foram utilizados *round-robin*, devido à alta propensão ao efeito *machine-gun* quando as peças são utilizadas em rápida sucessão, articulação comum em vários estilos musicais.

A arquitetura utilizada é como apresentada na figura 32 e é uniforme para todas as peças; variações no número de parâmetros ocorrem em função da diferença de duração das ondas no caso dos pratos. Para o caso da caixa, contratempo e dos tambores, incluído o bumbo, cada uma das redes é composta por 120008 parâmetros treináveis, incluindo pesos das conexões e *biases*, com um tamanho em disco de aproximadamente 1,4 Mb.

Os pratos, por possuírem um som com maior tempo de sustentação, demandaram uma rede com saídas maiores, totalizando 360 008 parâmetros treináveis e quase o triplo do tamanho em disco. Temos assim que todo o sistema possui em torno de 9 Mb de tamanho em disco. Foram utilizadas funções de ativação lineares, da forma  $f(x) = x$ , tanto para evitar *overfitting* quanto para permitir uma implementação menos intensiva, do ponto de vista computacional.

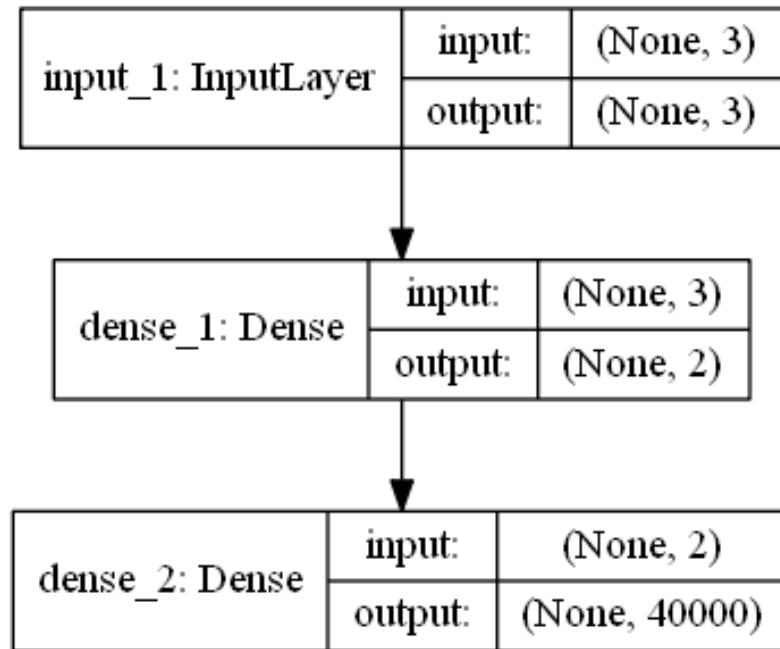


Figura 32: Arquitetura para a emulação da Caixa de uma Bateria  
Fonte: Elaboração própria

Os resultados sonoros são bastante realistas, tanto em termos de acurácia quanto de generalização. A rede não decora nenhuma das distribuições, como é esperado, mas aprende a generaliza-las no espaço bidimensional que tem como eixos o tipo da peça, que reflete a grosso modo uma variação entre agudo e grave e a intensidade da batida. Esses resultados podem ser ouvidos no caminho “resources/04\_freq\_domain/Exemplos das Predições - Bateria/” do repositório do trabalho (“Repositório Dissertação”, 2018).

A figura 33 ilustra uma onda onde foram concatenadas as saídas da rede para o contratempo aberto, golpeado com intensidade crescente. A onda concatenada pode ser ouvida no caminho “resources/04\_freq\_domain/Exemplos das Predições - Bateria/contratempo\_piano\_para\_forte.wav” do repositório, enquanto as amostras originais encontram-se em "resources/04\_freq\_domain/Exemplos das Predições - Bateria/Intensidade - Contratempo Aberto



Figura 33: Dinâmicas geradas para Contratempo Aberto - Pianíssimo até Molto Forte  
Fonte: Elaboração própria

No caso das redes treinadas com o uso de *round-robin*s pode-se imaginar um terceiro eixo no qual a rede generaliza variações percebidas nas amostras utilizados para o treinamento. A rigor, o efeito é determinístico, totalmente dependente dos inputs da rede.

Pode-se pensar nele, contudo, como um termo que adiciona certa aleatoriedade aos sons gerados, na medida em que pode-se alimentar a rede com um número arbitrário entre -1 e 1 para cada amostra gerada, introduzindo variações que emulam a aleatoriedade natural do instrumento. Essas variações são ilustradas na figura 34, para o caso do tom mais agudo. As ondas geradas pela rede podem ser encontradas em “resources/04\_freq\_domain/Exemplos das Predições - Bateria/Round-Robins - Tom 01” e a versão concatenada, exibida na figura 34, para maior comodidade, encontra-se em “resources/04\_freq\_domain/Exemplos das Predições - Bateria/tom\_roundrobin.wav”



Figura 34: 7 *Round-Robins* gerados para o Tom 1

Fonte: Elaboração própria

Um efeito colateral desse comportamento geral da rede é que ela não reproduz exatamente nenhuma das amostras utilizados no treinamento, e não há garantias do grau de semelhança entre os timbres gerados e os exemplos utilizados. Empiricamente, no entanto, observou-se que as características do espectro de frequências aprendidas e generalizadas pela rede imprimem ao resultado um timbre totalmente verossímil; em outras palavras, os tons soam como tons, não exatamente com o mesmo timbre que cada uma das peças utilizadas no treinamento, mas como uma amalgama entre elas. Assim é com os pratos, e sucessivamente para todas as peças.

Quando o esforço é o de emular duas dinâmicas diferentes de uma mesma peça, como é o caso da caixa, em que uma das variáveis é, como para todas as outras peças, a intensidade, e a outra é a centralidade da batida (mais próxima ou afastada do centro da caixa), observa-se resultados ainda mais orgânicos, como pode ser visto na figura 35. As ondas geradas podem ser ouvidas em “resources/04\_freq\_domain/Exemplos das Predições - Bateria/Centralidade - Caixa/”, enquanto que a versão concatenada, como apresentada na figura 35 está disponível em “resources/04\_freq\_domain/Exemplos das Predições - Bateria/caixa\_centro\_para\_bordas.wav”.



Figura 35: Dinâmica posição do golpe para caixa

Fonte: Elaboração própria

Para sons com características harmônicas, apresentando frequências parciais com caráter aproximadamente periódico, esse método não oferece bons resultados, na medida em que incorpora as características gerais de cada um dos sons utilizados no treinamento. É curioso observar que, por exemplo, treinando a rede para aprender a interjeição „ah“ cantada em diferentes notas, o resultado tem uma qualidade parecida com um coral.

As transformadas são exibidas na figura 36:, tanto dos sons originais quanto das previsões, para a nota mais baixa e mais alta. É possível reparar que a previsão encaminha-se para os picos em ambos os casos, mas incorporando uma grande quantidade de dados indesejáveis. Os áudios são encontrados no repositório do trabalho, no caminho “resources/04\_freq\_domain/Experimento - Voz/”

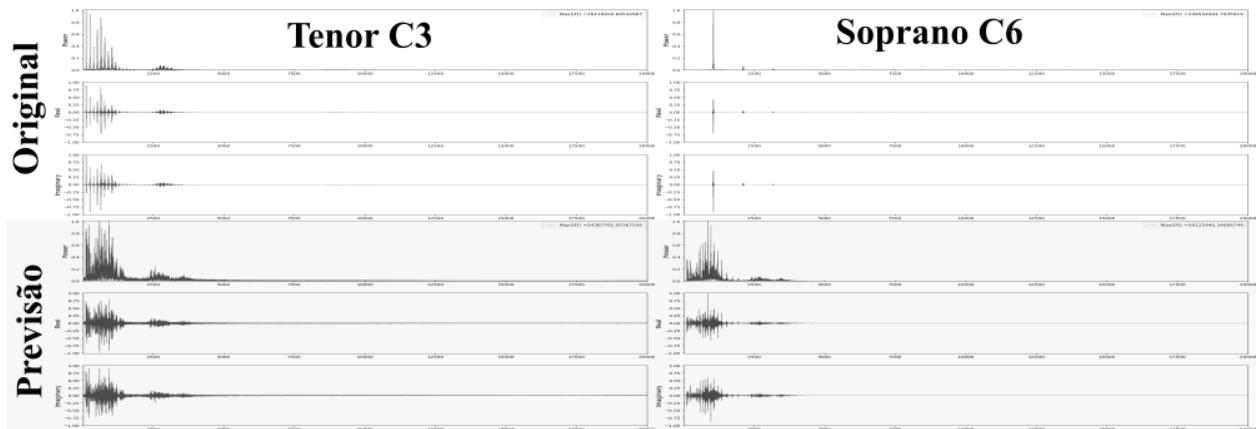


Figura 36: Experimento Emulação da Voz Cantada  
Fonte: Elaboração própria

#### 4.5 MODELO MISTO

Pudemos notar que instrumentos de caráter harmônico possuem uma distribuição de frequências bem-comportadas, consistindo basicamente de alguns picos em sua representação no domínio da frequência. Ademais, para que seu som seja considerado harmonioso, esses picos distribuem-se, forçosamente, de maneira ordenada; idealmente, são múltiplos inteiros da frequência fundamental da nota reproduzida.

É importante notar que a largura da „base“ desses padrões de picos no domínio da frequência é proporcional ao decaimento de uma senoide pura: no caso extremo, uma senoide perfeitamente periódica possui uma distribuição de frequências zero em todos os pontos, com exceção do ponto que representa sua frequência. A figura a figura 37 ilustra essa relação.

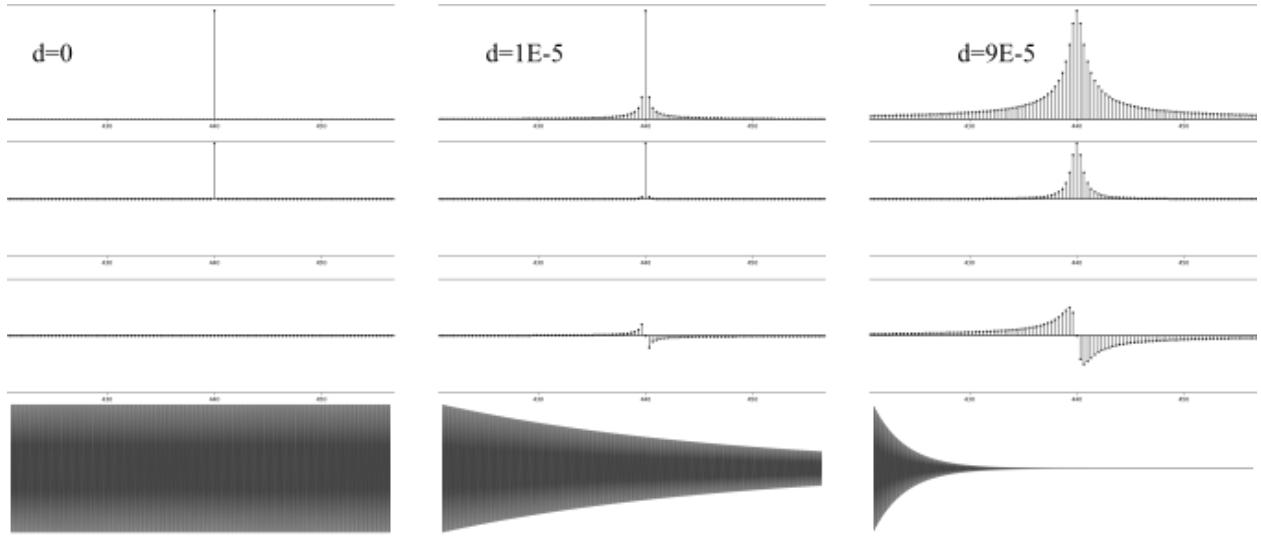


Figura 37: Efeito do Decaimento sobre a Transformada de Fourier: Senoide 440hz

Fonte: Elaboração própria

Podemos observar que o decaimento introduz novas frequências, ao redor da frequência principal, e mudanças de fase na representação no domínio da frequência; observa-se empiricamente que o „objetivo” primordial dessas novas frequências e fases é reproduzir o efeito de decaimento (ou, de forma mais ampla, o envelope) da onda.

Dessa observação, duas intuições importantes podem ser retiradas: A primeira é que, com um grau razoável de aproximação, podemos descrever um som harmônico gerado por uma excitação impulsiva em função da localização de algumas de suas frequências, suas respectivas intensidades e os seus decaimentos.

É natural supor também que a influência das fases dessas ondas pode ser ignorada, o que comprova-se empiricamente a partir da reconstrução de ondas com as fases originais e fases nulas ou aleatórias. No caminho “resources/05 Modelo Final/03\_waves\_from\_01\_info/piano/” do repositório podem ser encontradas ondas reconstruídas, descartando-se informações sobre as fases de cada uma das 100 frequências parciais, a partir das amostras utilizadas para o treinamento da rede, disponíveis em resources/05 Modelo Final/00\_samples/piano/. Comparando as amostras, pode-se perceber que a reprodução é bastante verossímil. A maior parte da diferença perceptual tem origem na quantidade de parciais consideradas, que não contempla as frequências presentes na fase transiente da onda, nas teclas mais graves.

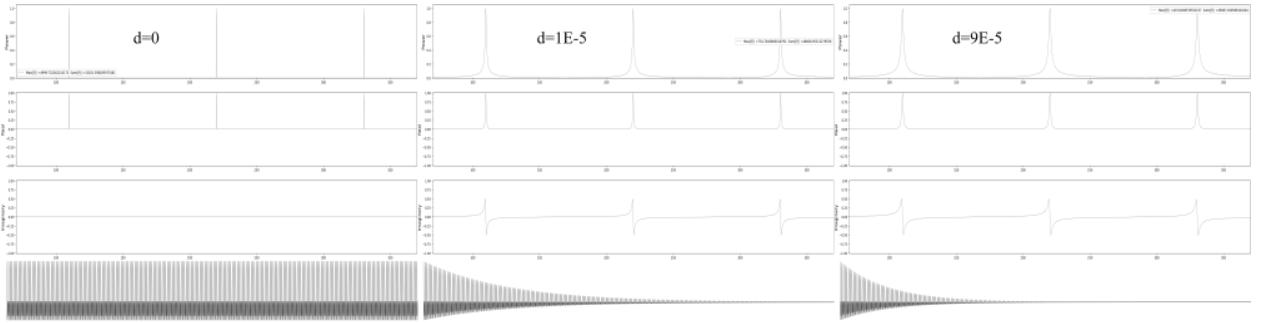


Figura 38: Onda composta por senoides a 110, 220 e 330hz

Fonte: Elaboração própria

Podemos, então, descartar as informações sobre a fase específica de cada parcial da onda, restando desenvolver um método para extração das frequências principais das amostras sonoras, e seus respectivos envelopes, que aqui assumiremos, simplificadamente, da forma exponencial, sendo caracterizados por uma taxa de decaimento e uma amplitude inicial.

Se lembrarmos que a transformada de Fourier oferece a intensidade de cada uma das frequências que compõe a onda (ou mesmo a amplitude, se normalizada de forma adequada) pode-se supor que o levantamento das duas primeiras informações sobre a onda torna-se trivial.

Algumas dificuldades, contudo, logo apresentam-se: elencando as frequências de maior intensidade de uma onda arbitrária, rapidamente notamos que muitas delas fazem parte de um mesmo pico. Além disso, as amplitudes referidas na transformada são as amplitudes médias da frequência em toda a onda que, além de não serem diretamente utilizáveis, não oferecem informação sobre o comportamento delas no tempo.

#### 4.5.1 Frequências

Atacando a questão das frequências, é conveniente lembrar de que, em uma onda perfeitamente harmônica, temos uma frequência fundamental  $f_0$ , da qual todas as outras frequências relevantes são múltiplos inteiros, da forma  $f_n = n f_0, n \in \mathbb{Z}^+$ . Além disso, a frequência fundamental é, geralmente, a de maior intensidade (mas nem sempre, como ilustrado na figura 39). Podemos fazer uso desta informação para encontrar os picos em uma onda.

No caso das teclas de um piano, a relação entre as 88 teclas e sua frequência fundamental pode ser encontrada a priori a partir da fórmula  $f_0 = 440 \times 2^{\frac{\text{tecla}-49}{12}}$ . Podemos, então, buscar os máximos nos intervalos apropriados. A figura 39 compara esse algoritmo com o simples levantamento dos maiores valores de intensidade, para uma onda

correspondente ao som emitido pela tecla 35 de um piano, da qual busca-se os 30 primeiros picos, correspondendo às 30 primeiras frequências parciais.

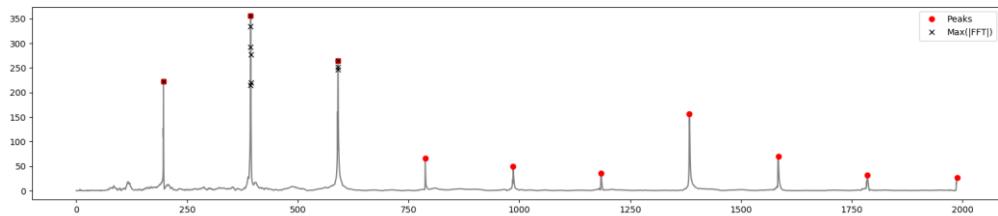


Figura 39: Picos e Valores Máximos - Piano, tecla 35  
Fonte: Elaboração própria

Repara-se que simplesmente elencando os maiores valores, todas as frequências identificadas gravitam em torno das 3 primeiras parciais. Com o algoritmo de busca de picos, as verdadeiras parciais da onda são capturadas, em intervalos bastante próximos à múltiplos inteiros de  $f_0$ . É interessante observar que a onda ilustrada em figura 39 representa um caso onde a frequência fundamental não corresponde a elevação mais alta.

#### 4.5.2 Decaimentos e Amplitudes

Uma abordagem possível para a investigação dos decaimentos relativos a cada uma das frequências parciais identificadas seria a aplicação de um método iterativo, seja a partir de um algoritmo de *curve fitting*, seja com a utilização de redes neurais. Ambos os métodos foram investigados e limitações críticas foram identificadas: O algoritmo iterativo usado para *curve fitting* não apresenta garantias de convergência.

Além disso, o processo iterativo deve ser utilizado para cada uma das ondas, e para cada uma das frequências parciais que deseja-se extrair de cada uma delas. Mesmo tratando-se de uma etapa prévia, que não fará parte do modelo em tempo real, os custos computacionais mostraram-se proibitivos (sobretudo para ondas com vários segundos de duração).

De maneira semelhante, o método baseado em redes neurais, embora não deixe de convergir, não apresenta garantias sobre a qualidade dos resultados. Ademais, os dois algoritmos são míopes, no sentido de que, na busca por minimizar suas funções objetivo geram, por vezes, senoides com amplitudes e decaimentos aberrantes.

Considerando, de forma aproximada, que todos os decaimentos são exponenciais, é possível estimá-los observando a diferença de intensidade da frequência de interesse entre intervalos arbitrários, como por exemplo a primeira e a segunda metades, da onda.

Tomando o cuidado de converter as frequências absolutas para as frequências locais, o somatório  $z(f) = \sum_{t=\frac{l}{2}}^{t+\frac{l}{2}} s[t] e^{-2\pi t f i} dt$  nos fornece essa informação: trata-se do caso discreto da integral da multiplicação ponto a ponto do sinal de interesse  $s(t)$  com o *kernel* da transformada de Fourier em função unicamente do tempo  $t$ , com  $f$  fixo na frequência local para a qual deseja-se a amplitude.  $z(f)$  fornece também a fase média no intervalo, que não nos é útil, como foi visto. Poderíamos utilizar um número arbitrário de intervalos; dois, no entanto, provaram-se suficientes empiricamente.

O decaimento pode ser estimado, então, a partir da relação  $d = 2\ln(a_1/a_2)/l$ . No caso específico de dois intervalos, temos que a amplitude  $a$  é  $a_1/e^{-dl/4} = a_2/e^{-d3l/4}$ . Na figura 40 são apresentadas as estimativas para as frequências mais presentes nos sons emitidos por algumas teclas de piano, além de um exemplo demonstrativo com um senoide puro.

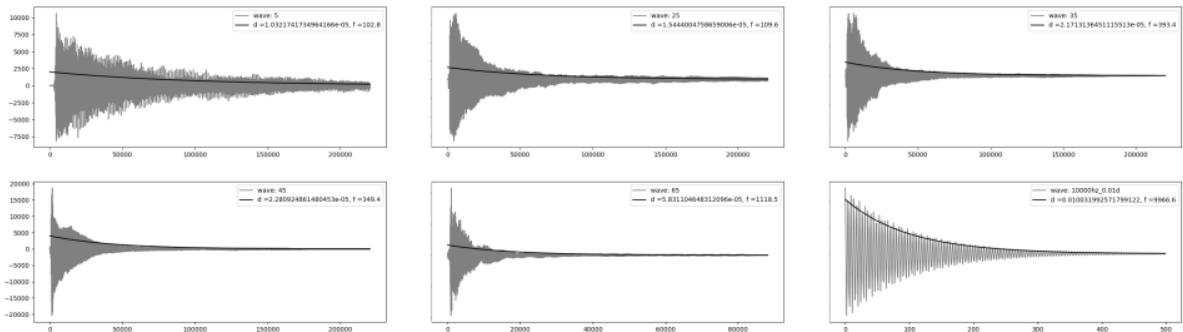


Figura 40: Decaimentos Estimados  
Fonte: Elaboração própria

Cabe observar que nos casos dos sons do piano ilustrados nos 5 primeiros retângulos da imagem o decaimento estimado é para uma das muitas frequências que compõe a onda - especificamente a de maior intensidade. No último retângulo, o decaimento coincide com o envelope da onda, já que esta é composta por apenas uma frequência, e podemos ter uma ideia melhor de sua acurácia. Estamos, desta forma, equipados para introduzir as redes neurais no modelo, que será usada para aprender a estimar e generalizar os parâmetros extraídos das amostras.

#### 4.5.3 Formulação Final

Podemos escrever as frequências, tanto a fundamental quanto as parciais, em função das teclas de um piano, da seguinte forma:

$$f(k, n) = 2^{\frac{k-49}{12}} 440(n+1), k \in 1, 2, \dots, 88, n \in 0, 1, 2, \dots$$

Essa abordagem é conveniente para uma vasta gama de instrumentos, já que essas 88 teclas vão desde a nota A0 até C8, cobrindo com folga o espectro de frequências da maioria dos instrumentos de interesse.

Para treinar a rede a partir de qualquer instrumento, basta rotular as amostras dos sons pertinentes com o número da tecla equivalente de um piano. Como pode ser ouvido nos exemplos apresentados ao fim deste trabalho, esse procedimento permitiu o treinamento de um instrumento híbrido, utilizando amostras de um baixo acústico, para o registro mais grave, um violoncelo para o meio do registro, e um violino para o mais agudo; todos esses instrumentos, ao mesmo tempo, cobriram até a tecla 75 do piano, somente.

Como foi visto, essa equação desconsidera as inharmonicidades presentes nos instrumentos, responsáveis por características importantes de seus timbres. Contudo, apresenta uma aproximação inicial bastante razoável que serve tanto para reforçar as características harmônicas básicas no modelo final quanto para aliviar o esforço de predição da rede, na medida em que podemos adicionar um termo de inharmonicidade na equação acima, a ser aprendido pela rede.

$$\text{Temos, assim: } f(k, n) = 2^{\frac{k-49}{12}} 440(n+1)i(k, n), k \in 0, 1, \dots, 88, n \in 0, 1, 2, \dots$$

Considerando tanto frequências fundamentais quanto parciais no espectro audível, teríamos um intervalo de 27 Hz, a frequência fundamental de A0, até um pouco mais de 20Khz, correspondendo, por exemplo, à quinta parcial de C8, caso tentássemos estimar diretamente as frequências parciais de um piano.

Escolhendo trabalhar com as inharmonicidades, por outro lado, reduzimos o intervalo para o limite entre 0 e 3. Além disso, o comportamento das inharmonicidades é razoavelmente previsível, com caráter ligeiramente exponencial, como ilustrado a na figura 41, para a primeira tecla de um piano, e a tecla 34, a última a ter todas as parciais no espectro audível.

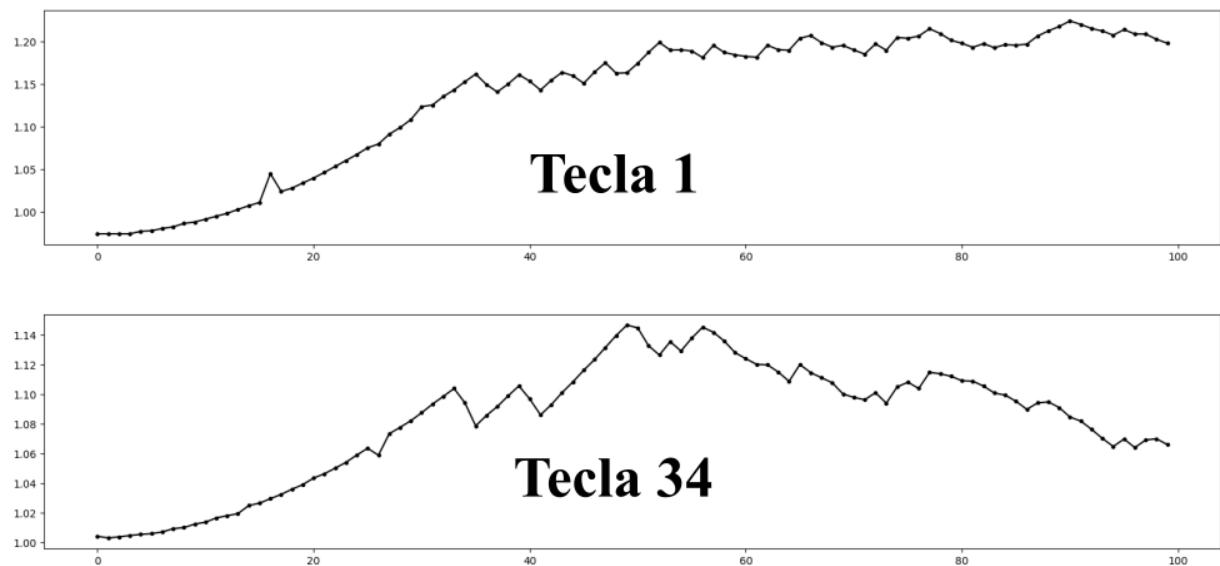


Figura 41: inharmonicidades

Fonte: Elaboração própria

Este mesmo *rationale* é aplicado aos decaimentos e amplitudes, que são previstos como uma fração dos valores máximos encontrados em cada uma das teclas (ou notas, mais genericamente, no caso de estarmos treinando um instrumento arbitrário), e encontram-se no intervalo fechado entre 0 e 1.

O comportamento das amplitudes e decaimentos máximos, por tecla, é apresentado na figura 42, e poderia ser, também, modelado, tanto a partir de redes neurais quanto por meio de modelos matemáticos. Empiricamente, contudo, nota-se que os valores médios dessas grandezas são suficientes para um bom resultado.

Isso é oportuno, pois reduz a complexidade do modelo final, além de conferir mais generalidade: na figura 42 abaixo notamos, por exemplo, que há uma descontinuidade nos valores dos decaimentos ocorrendo a partir da tecla 55, que deve-se provavelmente à mudança de cordas duplas para cordas simples.

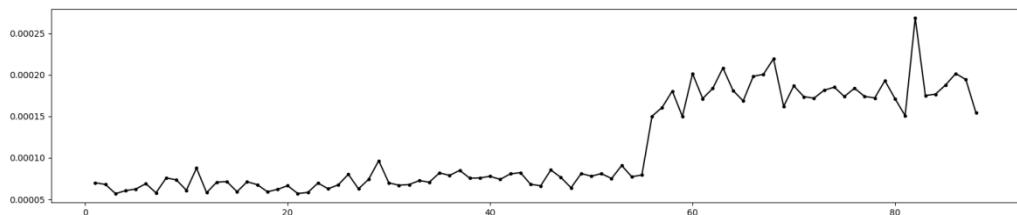


Figura 42: Decaimentos Máximos por Notas

Fonte: Elaboração própria

Esse fenômeno é particular dos pianos e, além disso, essa descontinuidade é arbitrária, variando de piano para piano. Teríamos, então, caso não fizéssemos uso da média, que formular uma aproximação para cada instrumento treinado (ou treinar uma rede adicional). As amplitudes, embora pareçam possuir uma tendência geral de diminuição na medida em que avançamos na escala do piano, possuem comportamento bastante arbitrário.

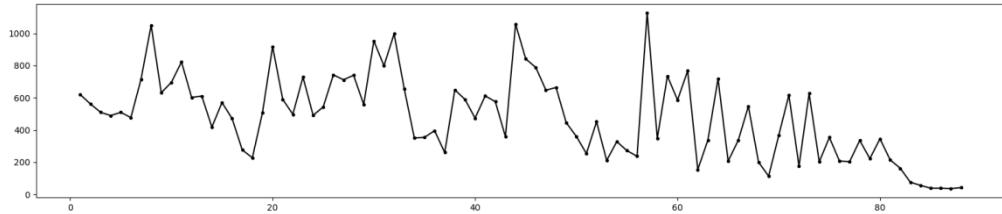


Figura 43: Amplitudes Máximas por Notas  
Fonte: Elaboração própria

Tendo sido observado que, de posse de uma estimativa do decaimento das frequências parciais, as fases de cada uma delas tem pouco impacto na reconstrução da onda, optou-se por randomizar as fases: Essa decisão apresenta a vantagem de conferir um resultado mais orgânico e variado à síntese executada pelo modelo final, na medida em que nenhuma onda gerada será exatamente igual a qualquer uma outra. Outra opção, menos rica do ponto de vista perceptual, seria ainda dar às fases um valor arbitrário (como zero, por exemplo).

Para a previsão de cada uma das 3 quantidades explicitadas, frequências parciais, amplitudes e decaimentos, uma rede densa, com 3 camadas, foi utilizada. A arquitetura das 3 redes é idêntica, com exceção do número de neurons em cada uma das camadas ocultas.

Optou-se por utilizar como função de ativação uma versão modificada da tangente hiperbólica, na forma  $a(x) = \tanh(6 * x - 3)/2 + 1/2$ , de forma a melhor cobrir o intervalo [0,1]. Além disso, o método de inicialização de pesos proposto por LeCun *et al.* (2012) e explorado em Klambauer *et al.* (2017) ofereceu uma melhora considerável no tempo de convergência das redes, e foi a inicialização utilizada na versão final do modelo, tanto para os *neurons* quanto para os *biases*. O método consiste em inicializar os pesos a partir de amostras retiradas de uma distribuição normal truncada, com valor esperado 0 e desvio padrão  $d = \sqrt{1/(dimensão\ do\ vetor\ de\ entrada)}$ .

As redes responsáveis pelas amplitudes e decaimentos possuem 50 neurons em cada uma das camadas ocultas, e um total de 5 301 parâmetros treináveis, com aproximadamente 100kb de espaço em disco. Sua arquitetura é apresentada abaixo. Para as frequências, o número de neurons foi reduzido para 10, para evitar *overfitting*, o que gera um total de 261

parâmetros treináveis e aproximadamente metade do tamanho em disco. A figura 44 a seguir compara os dados originais com os produzidos pelo modelo.

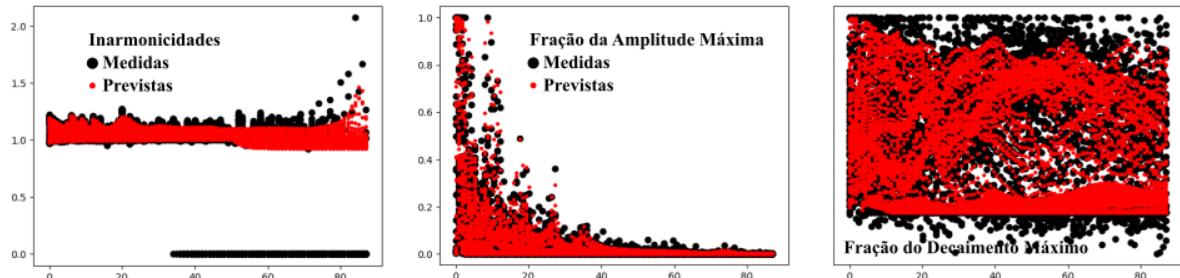


Figura 44: Previsões - Modelo Misto

Fonte: Elaboração própria

Na pasta resources/05 Modelo Final/02\_predictions/piano/ do repositório preparado para este trabalho podem ser observadas imagens semelhantes, para outros números de neurones em todas as camadas ocultas, incluindo alguns exemplos em que ocorre *overfitting*.

Podemos, assim, definir uma metodologia geral voltada à modelagem sonora de instrumentos harmônicos, batizada de Neuro-spectral Synthesis, resumida de maneira esquemática na figura 45.

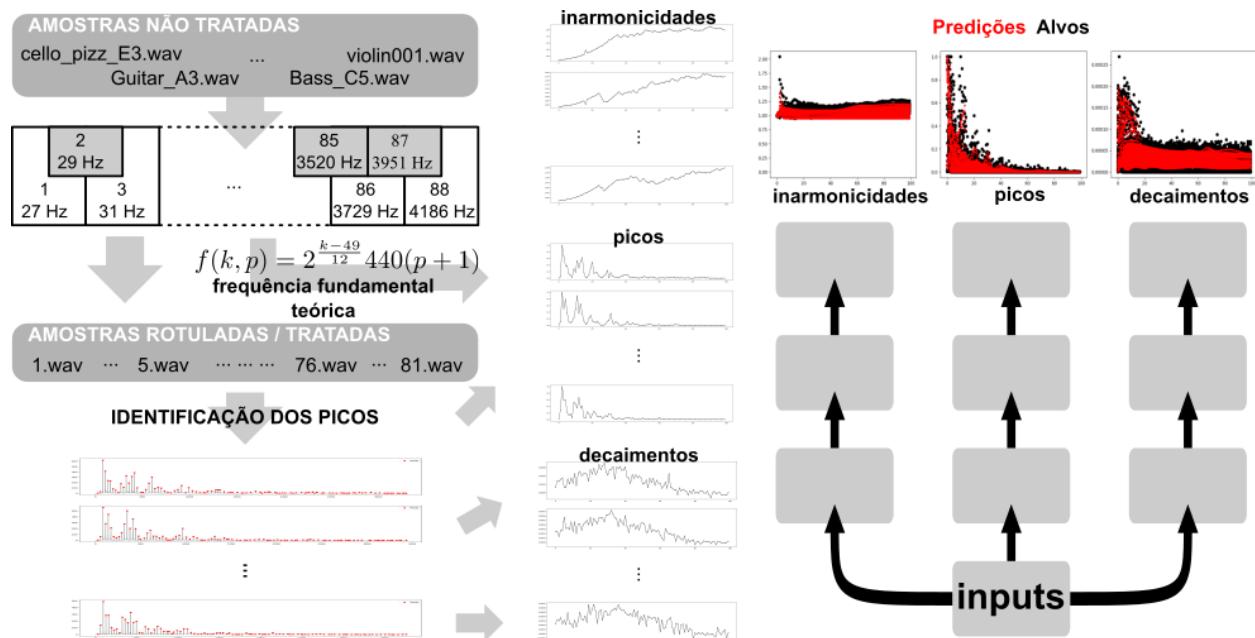


Figura 45: Visão Esquemática do Modelo Final

Fonte: Elaboração própria

## 5 RESULTADOS

O modelo final apresenta resultados mais realistas do que as implementações baseadas no modelo de *Digital Waveguides* e no método das diferenças finitas, e é ao menos uma ordem de grandeza mais eficiente, nas implementações aqui apresentadas.

Para efeito de comparação, levando em conta a geração de 5 ondas de 1 segundo, a 44100 FPS, pelas redes já treinadas, o método proposto foi, em média, 17 vezes mais rápido que o método baseado em Digital Waveguides e 26 vezes mais rápido do que o método das diferenças finitas. A tabela 6 abaixo elenca os resultados.

Tabela 6: Comparação Tempo de Geração(s)

	<b>Finite Differences</b>	<b>Digital Waveguides</b>	<b>Neuro-Spectral Synthesis</b>
1	16.95213819	13.81114888	0.5896253586
2	19.62042928	13.02065825	0.8354649544
3	24.25646234	11.33273816	0.7555158138
4	17.89053726	12.3500874	0.7475204468
5	18.00246739	12.12223339	0.7735056877
<b>Média</b>	<b>19.34440689</b>	<b>12.52737322</b>	<b>0.7403264523</b>
<b>Desvio Padrão</b>	<b>2.908692366</b>	<b>0.937300851</b>	<b>0.09102950916</b>

Treinadas as redes, uma implementação final do modelo é bastante simples, e é apresentada no repositório do trabalho, junto com os arquivos de definição da arquitetura das redes e seus respectivos pesos, em “resources/06 Tempo Real/”, e ocupam aproximadamente 250kb de armazenamento em disco. No repositório podem ser encontradas alguns áudios gerados pelo modelo.

A principal limitação da metodologia proposta consiste no fato de que todos os parâmetros a serem manipulados no modelo final devem, antes, ter sido incorporados ao processo de treinamento. Os dois paradigmas de modelagem física utilizados para comparação, tanto o método das diferenças finitas quanto os *digital waveguides*, permitem alguma manipulação em tempo real, a posteriori, de seus parâmetros: nos exemplos apresentados, o ponto de excitação pode ser mudado de onda para onda e, a qualquer tempo, o ponto de captação pode ser alterado, até mesmo durante a simulação, refletindo no timbre do som gerado.

Além disso, esses modelos prestam-se à incorporação razoavelmente trivial de uma fonte, contínua ou periódica, de excitação, e podem ser utilizados para a simulação de instrumentos de som continuo, como violinos acionados pelo arco (ao contrário do acionamento *pizzicato* aqui apresentado) e metais, por exemplo; esse não é o caso do modelo aqui proposto, e constitui uma possibilidade de desenvolvimento futuro interessante.

Uma outra barreira é que o modelo aprende características sonoras a partir de exemplos e não presta-se, de forma prática, à exploração sonora direta; esta deve ser efetuada a partir de outra ferramenta, e então incorporada ao modelo, à época de seu treinamento.

Para ilustração, algumas músicas criadas com os sons gerados pelos modelos propostos, treinados para emular um kit regular de bateria, um piano e um híbrido entre baixo acústico, violoncelo e violino podem ser ouvidos no endereço eletrônico “[soundcloud.com/carlos-tarjano/sets/spectral-neural-synthesis](https://soundcloud.com/carlos-tarjano/sets/spectral-neural-synthesis)”.

Do ponto de vista conceitual, o trabalho apresenta alguns encaminhamentos sobre a utilização da transformada discreta de Fourier no contexto das redes neurais que, até onde alcança o conhecimento do autor, não foram anteriormente explorados na literatura: a relação entre o envelope de uma onda, no domínio do tempo, e o formato de seus picos no domínio da frequência, por outro lado, possibilitou a formulação do modelo fisicamente informado, e pode ser explorada futuramente de maneira mais profunda, potencialmente baseando um método mais acurado de identificação de envelopes em ondas sonoras.

## 6 CONCLUSÃO

O presente trabalho, ao introduzir uma nova técnica de modelagem sonora, demonstra o potencial do uso de redes neurais à síntese de áudio, provando sobretudo a possibilidade desse uso em tempo real. O segundo modelo introduzido apresenta resultados mais verossímeis do que o algoritmo de modelagem acústica em tempo real mais utilizado e eficiente encontrado na literatura, à um custo computacional bastante inferior.

Uma outra vantagem frente aos modelos convencionais baseados em simulação física é que ele pode aprender características importantes do som que tem origem em partes do sistema difíceis de serem modeladas fisicamente, como a influência de ressonadores de geometria complexa, por exemplo.

O trabalho evidencia que arquiteturas densas e recorrentes são capazes de aprender representações internas que possibilitem a reprodução e generalização de amostras sonoras de maneira direta; a partir da introdução de uma representação compacta, fisicamente informada, de ondas sonoras harmônicas, o trabalho evidencia o potencial sinergético entre os desenvolvimentos na pesquisa sobre a acústica de instrumentos musicais e a utilização de redes neurais para basear modelos destinados à emulação desses instrumentos, ou famílias de instrumentos, específicos.

Além disso, através da utilização de funções de ativação especialmente elaboradas para acomodar os parâmetros da representação pertinentes e métodos de inicialização de pesos e vieses apropriados, essas arquiteturas podem ser simplificadas e o número necessário de parâmetros treináveis sensivelmente diminuído, tornando-as mais efetivas do ponto de vista da síntese sonora em tempo real.

Com a escolha apropriada da representação no domínio da frequência, os modelos prestam-se, com maior ou menor grau de eficiência, à emulação de uma vasta gama de instrumentos, respectivamente percussivos e harmônicos, dentro do escopo proposto para o trabalho; um kit de bateria, como o treinado durante o trabalho, representa por si só grande parte da família de instrumentos percussivos, em suas diferentes peças.

De maneira semelhante, o treinamento de um piano e um instrumento híbrido, a partir do modelo fisicamente informado apresentado representam de maneira bastante completa as aplicações em relação à instrumentos acústicos harmônicos de caráter impulsivo.

## 6.1 DESENVOLVIMENTOS FUTUROS

As possibilidades de desenvolvimentos futuros nesta área de intersecção entre redes neurais e acústica são inúmeras, haja vista, inclusive, a escassez de investigações semelhantes: Seria interessante, por exemplo, utilizar as saídas de um modelo elaborado a partir do método das diferenças finitas, que pode ser formulado de forma a simular características mais sofisticadas de um instrumento como, como rigidez, ressonância e vários termos de perda de um dado sistema acústico, ao custo de uma alta demanda de recursos computacionais, para treinar um modelo baseado em *digital waveguides* que tenha uma rede neural no ponto onde as perdas e demais cálculos são concatenados.

Devido ao alto grau de recursividade do algoritmo *digital waveguides*, o treinamento à partir do resultado final esperado para o modelo é bastante complexo de ser implementado; os vetores de saída de uma simulação baseada em diferenças finitas, no entanto, são plenamente compatíveis com este treinamento, e a inserção de uma rede neural poderia levar a um modelo que retenha ao menos parte da acurácia da simulação pelo método de diferenças finitas, com eficiência computacional próxima, ou até superior, à apresentada pelo algoritmo de *digital waveguides*. Em Gully *et al.* (2017), por exemplo, encontramos um exemplo de trabalho nesse sentido, que explora o uso de redes neurais para a identificação de parâmetros relevantes à uma simulação por *digital waveguides* do trato vocal humano.

Relaxar a simplificação adotada durante o trabalho em relação à decaimentos exponenciais é um outro desenvolvimento futuro com potencial interessante: para algumas categorias de som, como a voz humana por exemplo, o envelope do som apresenta importância maior do que as próprias frequências contidas em relação a características como inteligibilidade.

Estimar os envelopes com a técnica utilizada no segundo modelo, a partir de um número maior de pontos, e utilizar uma rede, possivelmente recorrente, para aprender as características desse envelope para um conjunto de sons de determinado instrumento (inclusive de excitação continuada), ou mesmo a voz humana, constitui outra interessante direção a ser investigada.

Além disso, um algoritmo eficiente de extração de envelope permitiria investigar a abordagem baseada em *autoencoders* que, alimentados com um senoide puro modelada pelo envelope da onda, teriam maior possibilidade de aprender a reconstruir a onda original de maneira verossímil, em uma abordagem inspirada nas técnicas de transferência de estilo utilizadas no campo da visão computacional.

Um outro desenvolvimento seria produzir uma implementação em uma linguagem de programação mais eficiente, como C ou C++, somada à uma interface visual e compatibilidade com controladores MIDI pode gerar uma linha de produtos comercialmente viável, a ser comercializado em formato *standalone* e/ou em formato de *plugin* VST, em que um módulo inicial pode ser oferecido e bancos adicionais, consistindo de novas arquiteturas e pesos, podem ser criados e comercializados mediante a demanda dos usuários ou o desenvolvimento da técnica.

## 7 REFERÊNCIAS

- ALJUMAH, A. e AHAMAD, T. A., **A novel approach for detecting DDoS using artificial neural networks**, International Journal of Computer Science and Network Security, v. 16, p. 132-138, 2016.
- BAHRAMPOUR, S. et al., **Comparative Study of Caffe, Neon, Theano, and Torch for Deep Learning**, CoRR, v. abs/1511.06435, 2015.
- BENSA, J. et al., **Computational modeling of stiff piano strings using digital waveguides and finite difference**, Acta Acustica united with Acustica, v. 91, p. 289-298, 2005.
- BENSA, J. et al., **The simulation of piano string vibration: From physical models to finite difference schemes and digital waveguides**, The Journal of the Acoustical Society of America, v. 114, n. 2, p. 1095-1107, 2003.
- BILBAO, S., **Numerical Sound Synthesis**, John Wiley & Sons, 2009.
- BISHOP, C. M., **Pattern Recognition and Machine Learning**, Springer-Verlag New York Inc., 2006.
- BÖCK, S. e SCHEDL, M., **Polyphonic piano note transcription with recurrent neural networks**, in Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), p. 121-124, 2012.
- BOULANGER-LEWANDOWSKI, N.; BENGIO, Y. e VINCENT, P., **High-dimensional sequence transduction**, in Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, p. 3178-3182, 2013.
- BOULANGER-LEWANDOWSKI, N. et al., **Phone sequence modeling with recurrent neural networks**, in Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), p. 5417-5421, 2014.
- BOVERMANN, T. et al., **Musical Instruments in the 21st Century**, Springer Singapore, 2016.
- BRUNETTE, E. S.; FLEMMER, R. C. e FLEMMER, C. L., **A review of artificial intelligence**, in Proceedings of the 4th International Conference on Autonomous Robots and Agents, p. 385-392, 2009.
- CHAIGNE, A. e KERGOMARD, J., **Acoustics of Musical Instruments**, Springer New York, 2016.
- CHOI, K.; FAZEKAS, G. e SANDLER, M., **Automatic Tagging Using Deep Convolutional Neural Networks**, in Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, p. 805-811, 2016.
- CHOI, K. et al., **Convolutional recurrent neural networks for music classification**, in Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), p. 2392-2396, 2017.

- COPPIN, B., **Artificial Intelligence Illuminated**, JONES & BARTLETT PUBL INC, 2004.
- COSTA, Y. M.; OLIVEIRA, L. S. e SILLA, C. N., **An evaluation of Convolutional Neural Networks for music classification using spectrograms**, Applied Soft Computing, v. 52, p. 28-38, 2017.
- DALGLEISH, M.; FOSTER, C. e SPENCER, S., **BLURRING THE LINES: AN INTEGRATED COMPOSITIONAL MODEL FOR DIGITAL MUSICAL INSTRUMENT DESIGN**, in Proceedings of the 9th Conference on Interdisciplinary Musicology – CIM14, 2014.
- DONAHUE, C.; MCAULEY, J. e PUCKETTE, M., **Synthesizing Audio with GANs**, in Proceedings of the Sixth International Conference on Learning Representations, 2018.
- DOZAT, T., **Incorporating Nesterov Momentum into Adam**, 2016.
- DUCHI, J.; HAZAN, E. e SINGER, Y., **Adaptive Subgradient Methods for Online Learning and Stochastic Optimization**, Journal of Machine Learning Research, v. 12, p. 2121-2159, 2011.
- DUMOULIN, V. e VISIN, F., **A guide to convolution arithmetic for deep learning**, CoRR, v. abs/1603.07285, 2016.
- ELMAN, J. L., **Finding Structure in Time**, Cognitive Science, v. 14, n. 2, p. 179-211, 1990.
- ENGEL, J. et al., **Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders**, CoRR, v. abs/1704.01279, 2017.
- ERICKSON, B. J. et al., **Toolkits and Libraries for Deep Learning**, Journal of Digital Imaging, v. 30, n. 4, p. 400-405, 2017.
- ESTEVA, A. et al., **Dermatologist-level classification of skin cancer with deep neural networks**, Nature, v. 542, n. 7639, p. 115-118, 2017.
- FASTL, H. e ZWICKER, E., **Psychoacoustics**, Springer Berlin Heidelberg, v. 22, 2007.
- FONTANA, F.; ROCCHESSO, D. e APOLLONIO, E., **Using the waveguide mesh in modelling 3D resonators**, in Proceedings of the International Conference on Digital Audio Effects (DAFx), p. 229-232, 2000.
- FORGEARD, M. et al., **Practicing a Musical Instrument in Childhood is Associated with Enhanced Verbal Ability and Nonverbal Reasoning**, PLoS ONE, v. 3, n. 10, p. e3566, 2008.
- FRANS, K., **Outline Colorization through Tandem Adversarial Networks**, CoRR, v. abs/1704.08834, 2017.
- GARRETT, S. L., **Understanding Acoustics**, Springer International Publishing, 2017.
- GATYS, L. A.; ECKER, A. S. e BETHGE, M., **Image Style Transfer Using Convolutional Neural Networks**, in Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 2414-2423, 2016.

- GAZI, O., **Understanding Digital Signal Processing**, Springer Singapore, 2017.
- GOLDBERG, Y., **A Primer on Neural Network Models for Natural Language Processing**, CoRR, v. abs/1510.00726, 2015.
- GOODFELLOW, I.; BENGIO, Y. e COURVILLE, A., **Deep Learning**, MIT Press, 2016.
- GRACIA, X. e SANZ-PERELA, T., **The wave equation for stiff strings and piano tuning**, Reports@SCM, v. 3, n. 1, p. 1-16, 2016.
- GRAVES, A.; MOHAMED, A. e HINTON, G., **Speech recognition with deep recurrent neural networks**, in Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, p. 6645-6649, 2013.
- GRINSTEIN, E. et al., **Audio style transfer**, CoRR, v. abs/1710.11385, 2017.
- GULLY, A. J. et al., **Articulatory Text-to-Speech Synthesis Using the Digital Waveguide Mesh Driven by a Deep Neural Network**, in Interspeech 2017, 2017.
- HAGAN, M. T. et al., **Neural Network Design**, Martin Hagan, 2014.
- HE, L. et al., **Multimodal Affective Dimension Prediction Using Deep Bidirectional Long Short-Term Memory Recurrent Neural Networks**, in Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge - AVEC '15, p. 73-80, 2015.
- HINTON, G. et al., **Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups**, IEEE Signal Processing Magazine, v. 29, n. 6, p. 82-97, 2012.
- HINTON, G. E.; SABOUR, S. e FROSST, N., **Matrix capsules with EM routing**, in Proceedings of the International Conference on Learning Representations, 2018.
- HORNIK, K., **Approximation capabilities of multilayer feedforward networks**, Neural Networks, v. 4, n. 2, p. 251-257, 1991.
- HORNIK, K.; STINCHCOMBE, M. e WHITE, H., **Multilayer feedforward networks are universal approximators**, Neural Networks, v. 2, n. 5, p. 359-366, 1989.
- HOROWITZ, M. C. et al., **Strategic Competition in an Era of Artificial Intelligence**, 2018.
- HOWARD, D. M. e ANGUS, J., **Acoustics and Psychoacoustics**, Routledge, 2017.
- HUTCHINGS, P., **Talking Drums: Generating drum grooves with neural networks**, in Proceedings of the First International Workshop on Deep Learning and Music joint with IJCNN, p. 43-47, 2017.
- HWANG, J. e ZHOU, Y., **Image colorization with deep convolutional neural networks**, 2016.
- IIZUKA, S.; SIMO-SERRA, E. e ISHIKAWA, H., **Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with**

**Simultaneous Classification**, ACM Transactions on Graphics (SIGGRAPH 2016), v. 35, n. 4, p. 110:1-110:11, 2016.

ISOLA, P. et al., **Image-to-Image Translation with Conditional Adversarial Networks**, CoRR, v. abs/1611.07004, 2016.

KARJALAINEN, M. e ERKUT, C., **Digital Waveguides versus Finite Difference Structures: Equivalence and Mixed Modeling**, Journal on Advances in Signal Processing (EURASIP), v. 2004, n. 7, 2004.

KARPATHY, A. et al., **Large-Scale Video Classification with Convolutional Neural Networks**, in Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, p. 1725-1732, 2014.

KHORRAMI, P. et al., **How deep neural networks can improve emotion recognition on video data**, in Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), p. 619-623, 2016.

KINGMA, D. P. e BA, J., **Adam: A Method for Stochastic Optimization**, CoRR, v. abs/1412.6980, 2014.

KLAMBAUER, G. et al., **Self-Normalizing Neural Networks**, in Advances in Neural Information Processing Systems 30, p. 971-980, 2017.

KULKARNI, T. D. et al., **Deep Convolutional Inverse Graphics Network**, in Advances in Neural Information Processing Systems 28, p. 2539-2547, 2015.

LAIRD, J. A., **The physical modelling of drums using digital waveguides**, 2001.

LARSSON, G.; MAIRE, M. e SHAKHNAROVICH, G., **Learning Representations for Automatic Colorization**, in Computer Vision (ECCV 2016), p. 577-593, 2016.

LECUN, Y. A. et al., **Efficient BackProp**, in Lecture Notes in Computer Science, p. 9-48, 2012.

LECUN, Y. et al., **Gradient-based learning applied to document recognition**, Proceedings of the IEEE, v. 86, n. 11, p. 2278-2324, 1998.

LESHNO, M. et al., **Multilayer feedforward networks with a nonpolynomial activation function can approximate any function**, Neural Networks, v. 6, n. 6, p. 861 - 867, 1993.

LYONS, R. G., **Understanding Digital Signal Processing**, ADDISON WESLEY PUB CO INC, 2010.

MAAS, A. L.; HANNUN, A. Y. e NG, A. Y., **Rectifier nonlinearities improve neural network acoustic models**, in Proceedings of the International Conference on Machine Learning, v. 30, n. 1, p. 3, 2013.

MCCULLOCH, W. S. e PITTS, W., **A logical calculus of the ideas immanent in nervous activity**, The Bulletin of Mathematical Biophysics, v. 5, n. 4, p. 115-133, 1943.

- MINSKY, M. e PAPERT, S. A., **Perceptrons: An Introduction to Computational Geometry**, MIT Press, 1969.
- MITAL, P. K., **Time Domain Neural Audio Style Transfer**, CoRR, v. abs/1711.11160, 2017.
- MIZUTANI, E.; DREYFUS, S. E. e NISHIO, K., **On derivation of MLP backpropagation from the Kelley-Bryson optimal-control gradient formula and its application**, in Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000). Neural Computing: New Challenges and Perspectives for the New Millennium, v. 2, p. 167-172 vol.2, 2000.
- MULLEN, J., **Physical modelling of the vocal tract with the 2D digital waveguide mesh.**, v. 2006, 2006.
- NGUYEN, A.; YOSINSKI, J. e CLUNE, J., **Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images**, in Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
- OLSON, T., **Applied Fourier Analysis**, Springer New York, v. 2017, 2017.
- OMAR, N.; JOHARI, Z. `A. e SMITH, M., **Predicting fraudulent financial reporting using artificial neural network**, Journal of Financial Crime, v. 24, n. 2, p. 362-387, 2017.
- VAN DEN OORD, A. et al., **WaveNet: A Generative Model for Raw Audio**, CoRR, v. abs/1609.03499, 2016.
- VAN DEN OORD, A.; KALCHBRENNER, N. e KAVUKCUOGLU, K., **Pixel Recurrent Neural Networks**, in Proceedings of the 33rd International Conference on Machine Learning, v. 48, p. 1747-1756, 2016.
- VAN DEN OORD, A. et al., **Conditional Image Generation with PixelCNN Decoders**, CoRR, v. abs/1606.05328, 2016.
- PANG, Y. et al., **Convolution in Convolution for Network in Network**, IEEE Transactions on Neural Networks and Learning Systems, v. 29, n. 5, p. 1587-1597, 2018.
- PARKER, D., **Learning-logic: Casting the Cortex of the Human Brain in Silicon**, 1985.
- PARVAT, A. et al., **A survey of deep-learning frameworks**, in Proceedings of the 2017 International Conference on Inventive Systems and Control (ICISC), p. 1-7, 2017.
- PATTERSON, J. e GIBSON, A., **Deep Learning: A Practitioner's Approach**, O'Reilly Media, Inc., 2017.
- PEREYRA, M. C. e WARD, L. A., **Harmonic Analysis: From Fourier to Wavelets**, American Mathematical Society, 2012.
- PFALZ, A., **Generating Audio Using Recurrent Neural Networks**, 2018.
- RASCHKA, S., **Python Machine Learning**, Packt Publishing, 2015.

REDDI, S. J.; KALE, S. e KUMAR, S., **On the Convergence of Adam and Beyond**, in Proceedings of the International Conference on Learning Representations, 2018.

REIFFENSTEIN, T., **Codification, patents and the geography of knowledge transfer in the electronic musical instrument industry**, The Canadian Geographer / Le Géographe canadien, v. 50, n. 3, p. 298-318, 2006.

RIGAUD, F.; DAVID, B. e DAUDET, L., **A parametric model and estimation techniques for the inharmonicity and tuning of the piano**, The Journal of the Acoustical Society of America, v. 133, n. 5, p. 3107-3118, 2013.

RISI, S. e TOGELIUS, J., **Neuroevolution in Games: State of the Art and Open Challenges**, IEEE Transactions on Computational Intelligence and AI in Games, v. 9, n. 1, p. 25-41, 2017.

ROBERTS, A. et al., **Learning Latent Representations of Music to Generate Interactive Musical Palettes**, in Proceedings of the 2018 ACM Workshop on Intelligent Music Interfaces for Listening and Creation, MILC@IJUI 2018, 2018.

ROBERTS, A. et al., **A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music**, CoRR, v. abs/1803.05428, 2018.

ROSENBLATT, F., **The Perceptron: A Perceiving and Recognizing Automaton**, v. 1957, n. 85-460-1, 1957.

ROSENBLATT, F., **The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain**, Psychological Review, p. 65-386, 1958.

RUDER, S., **An overview of gradient descent optimization algorithms**, CoRR, v. abs/1609.04747, 2016.

RUMELHART, D. E.; HINTON, G. E. e WILLIAMS, R. J., **Learning representations by back-propagating errors**, Nature, v. 323, p. 533 EP -, 1986.

RUSSELL, S. e NORVIG, P., **Artificial Intelligence: A Modern Approach, Global Edition**, Addison Wesley, 2016.

SABOUR, S.; FROSST, N. e HINTON, G. E., **Dynamic Routing Between Capsules**, in Advances in Neural Information Processing Systems 30, p. 3856-3866, 2017.

SAINATH, T. N. et al., **Deep Convolutional Neural Networks for Large-scale Speech Tasks**, Neural Networks, v. 64, p. 39-48, 2015.

SAK, H. et al., **Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition**, in Interspeech 2015, p. 1468-1472, 2015.

SALSA, S., **Partial Differential Equations in Action**, Springer International Publishing, 2016.

SANGKLOY, P. et al., **Scribbler: Controlling Deep Image Synthesis with Sketch and Color**, Computer Vision and Pattern Recognition (CVPR), 2017.

SARROFF, A. M. e CASEY, M., **Musical audio synthesis using autoencoding neural nets**, in Joint 40th International Computer Music Conference (ICMC) and 11th Sound & Music Computing conference (SMC), 2014.

SCHMIDHUBER, J., **Deep learning in neural networks: An overview**, Neural Networks, v. 61, p. 85-117, 2015.

SERAFIN, S.; HUANG, P. e SMITH, J., **The banded digital waveguide mesh**, in Workshop on Future Directions of Computer Music (Mosart-01), 2001.

SERRA, X., **State of the Art and Future Directions in Musical Sound Synthesis**, in Proceedings of the IEEE 9th Workshop on Multimedia Signal Processing, p. 9-12, 2007.

SHI, S. et al., **Benchmarking State-of-the-Art Deep Learning Software Tools**, in Proceedings of the 2016 7th International Conference on Cloud Computing and Big Data (CCBD), p. 99-104, 2016.

SIGTIA, S.; BENETOS, E. e DIXON, S., **An End-to-End Neural Network for Polyphonic Piano Music Transcription**, IEEE/ACM Transactions on Audio, Speech, and Language Processing, v. 24, n. 5, p. 927-939, 2016.

SMITH, J. O., **A Basic Introduction to Digital Waveguide Synthesis (for the Technically Inclined)**, Center for Computer Research in Music and Acoustics (CCRMA), 2006.

SMITH, J. O., **Digital Waveguide Architectures for Virtual Musical Instruments**, in Handbook of Signal Processing in Acoustics, p. 399-417, 2008.

SMITH, J. O., **Physical modeling using digital waveguides**, Computer Music Journal, v. 16, n. 4, p. 74-91, 1992.

SMITH, S. W., **The Scientist and Engineer's Guide to Digital Signal Processing**, 1999.

SOCHER, R., **Recursive deep learning for natural language processing and computer vision**, 2014.

SORENSEN, H. et al., **Real-valued fast Fourier transform algorithms**, IEEE Transactions on Acoustics, Speech, and Signal Processing, v. 35, n. 6, p. 849-863, 1987.

SOUTHALL, C.; STABLES, R. e HOCKMAN, J., **Automatic Drum Transcription Using Bi-Directional Recurrent Neural Networks**, in Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016, p. 591-597, 2016.

SPEED, M. D. A., **Voice synthesis using the three-dimensional digital waveguide mesh**, 2012.

STANLEY, K. O., **Compositional pattern producing networks: A novel abstraction of development**, Genetic Programming and Evolvable Machines, v. 8, n. 2, p. 131-162, 2007.

STAUDT, P., **Development of a Digital Musical Instrument with Embedded Sound Synthesis**, 2016.

- STEIN, E. M. e SHAKARCHI, R., **Fourier Analysis**, Princeton University Press, v. 1, 2003.
- SUTSKEVER, I. et al., **On the importance of initialization and momentum in deep learning**, in Proceedings of the 30th International Conference on Machine Learning, v. 28, n. 3, p. 1139-1147, 2013.
- THEIS, L. e BETHGE, M., **Generative Image Modeling Using Spatial LSTMs**, in Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, p. 1927-1935, 2015.
- TUOHY, D. R.; POTTER, W. D. e CENTER, A. I., **An Evolved Neural Network/HC Hybrid for Tablature Creation in GA-based Guitar Arranging**, International Computer Music Conference, 2006.
- VAN DUYNE, S. A. e SMITH, J. O., **Physical modeling with the 2-D digital waveguide mesh**, in Proceedings of the International Computer Music Conference, p. 40-47, 1993.
- VAN DUYNE, S. A. e SMITH, J. O., **The tetrahedral digital waveguide mesh**, in Proceedings of 1995 Workshop on Applications of Signal Processing to Audio and Acoustics, p. 234-237, 1995.
- VAUGHN, K., **Music and Mathematics: Modest Support for the Oft-Claimed Relationship**, Journal of Aesthetic Education, v. 34, n. 3/4, p. 149-166, 2000.
- VEIT, A.; WILBER, M. e BELONGIE, S., **Residual Networks Behave Like Ensembles of Relatively Shallow Networks**, in Proceedings of the 30th International Conference on Neural Information Processing Systems, p. 550-558, 2016.
- WERBOS, P., **Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.**, 1974.
- WIDROW, B. e HOFF, M. E., **Adaptive Switching Circuits**, 1960.
- WIDROW, B. e LEHR, M. A., **30 years of adaptive neural networks: perceptron, Madaline, and backpropagation**, Proceedings of the IEEE, v. 78, n. 9, p. 1415-1442, 1990.
- XU, W.; AULI, M. e CLARK, S., **CCG Supertagging with a Recurrent Neural Network**, in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), p. 250-255, 2015.
- YADAV, N.; YADAV, A. e KUMAR, M., **An Introduction to Neural Network Methods for Differential Equations**, Springer Netherlands, 2015.
- YANG, Y.; KROMPASS, D. e TRESP, V., **Tensor-Train Recurrent Neural Networks for Video Classification**, in Proceedings of the International Conference on Machine Learning, 2017.
- ZEILER, M. D., **ADADELTA: An Adaptive Learning Rate Method**, CoRR, v. abs/1212.5701, 2012.

ZHANG, R.; ISOLA, P. e EFROS, A. A., **Colorful Image Colorization**, CoRR, v. abs/1603.08511, 2016.

ZHANG, Y.; CHAN, W. e JAITLEY, N., **Very deep convolutional networks for end-to-end speech recognition**, in Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), p. 4845-4849, 2017.

ZHANG, Y. et al., **Towards End-to-End Speech Recognition with Deep Convolutional Neural Networks**, CoRR, v. abs/1701.02720, 2017.

ZHU, J.-Y. et al., **Generative Visual Manipulation on the Natural Image Manifold**, in Proceedings of European Conference on Computer Vision (ECCV), 2016.

ZWEIG, G. et al., **Advances in All-Neural Speech Recognition**, CoRR, v. abs/1609.05935, 2016.

**Caffe Deep Learning Framework**, [ online ], disponível em: <http://caffe.berkeleyvision.org/> , acesso em: 2018-06-30 .

**DrumGizmo Wiki**, [ online ], disponível em: <https://www.drumgizmo.org/wiki/doku.php> , acesso em: 30.06.2018 .

**Ivy Audio**, [ online ], disponível em: <http://www.ivyaudio.com/> , acesso em: 2018-06-30 .

**Keras: The Python Deep Learning library**, [ online ], disponível em: <https://keras.io/> , acesso em: 2018-06-30 .

**Magenta**, [ online ], disponível em: <https://magenta.tensorflow.org/> , acesso em: 2018-06-30 .

**Neon**, [ online ], disponível em: <https://ai.intel.com/neon/> , acesso em: 2018-06-30 .

**NSynthSuper**, [ online ], disponível em: <https://nsynthsuper.withgoogle.com/> , acesso em: 2018-06-30 .

**Philharmonia Orchestra**, [ online ], disponível em: [http://www.philharmonia.co.uk/explore/sound\\_samples](http://www.philharmonia.co.uk/explore/sound_samples) , acesso em: 2018-06-30 .

**Repositório Dissertação**, [ online ], disponível em: <https://github.com/tesserato/tesserato.github.io> , acesso em: 2018-06-30 .

**Rmsprop: Divide the gradient by a running average of its recent magnitude.**, [ online ], disponível em: <https://www.coursera.org/lecture/neural-networks/rmsprop-divide-the-gradient-by-a-running-average-of-its-recent-magnitude-YQHki> , acesso em: 2018-06-30 .

**TensorFlow**, [ online ], disponível em: <https://www.tensorflow.org/> , acesso em: 2018-06-30 .

**The Berkeley Artificial Intelligence Research Blog**, [ online ], disponível em: <http://bair.berkeley.edu/> , acesso em: 2018-06-30 .

**The Neural Network Zoo**, [ online ], disponível em: <http://www.asimovinstitute.org/neural-network-zoo/> , acesso em: 2018-06-30 .

**The Open Source Drumkit**, [ online ], disponível em: <https://github.com/crabacus/the-open-source-drumkit> , acesso em: 2018-06-30 .

**Theano 1.0.0 documentation**, [ online ], disponível em:  
<http://deeplearning.net/software/theano/> , acesso em: 2018-06-30 .

**Torch, a scientific computing framework for LuaJIT**, [ online ], disponível em:  
<http://torch.ch/> , acesso em: 2018-06-30 .

**University of Iowa Electronic Music Studios**, [ online ], disponível em:  
<http://theremin.music.uiowa.edu/MIS.html> , acesso em: 2018-06-30 .

**WaveNet: A Generative Model for Raw Audio**, [ online ], disponível em:  
<https://deepmind.com/blog/wavenet-generative-model-raw-audio/> , acesso em: 2018-06-30 .

## GLOSSÁRIO

Backpropagation	Algoritmo usado para a atualização de pesos de uma rede neural com o objetivo de aproximar suas previsões dos alvos desejados.
Backward pass	A etapa, durante o treinamento de uma rede neural, em que os erros são calculados e os pesos da rede atualizados.
Batches	Quantidade de vetores de entradas utilizados para o cálculo dos erros durante o treinamento de uma rede neural.
Biases	Pesos somados às saídas de uma camada de uma rede neural, com o intuito de evitar que os pesos das ligações entre neurônios fiquem presas no valor zero durante o treinamento.
Circular buffer	Estrutura de dados eficiente onde a última posição de uma lista, na memória, é adjacente à primeira.
Courant Number	Número adimensional que governa a qualidade e a convergência de algumas simulações baseadas em equações diferenciais parciais.
Creative Commons	Organização americana voltada à elaboração de licenças para uso livre de produções intelectuais.
Delay lines	Estruturas utilizadas no processamento digital de sinais onde ficam armazenados um número pré-definidos de samples por um período arbitrário de tempo.
SGD	Standard gradient descent, é uma técnica de atualização dos pesos de uma rede neural onde os pesos são movidos na no sentido oposto ao gradiente do erro calculado.
DFT	Discrete Fourier Transform, versão discreta da transformada de Fourier.
Digital Waveguides	Algoritmo de simulação acústica física criado por Julius O. Smith III.
DMI	Digital musical instrument, instrumento musical digital.
Efeito machine-gun	Efeito que ocorre quando a mesma amostra sonora é tocada em rápida sucessão, sem tratamento adicional, em um DMI, prejudicando o caráter natural do som gerado.
Epochs	No contexto de redes neurais, é o número de vezes, ou épocas, em que a rede utiliza todos os pares de vetores de entrada e alvos para atualizar seus pesos, durante a etapa de treinamento.
FFT	Fast Fourier Transform, Algoritmo que implementa de maneira eficiente a transformada discreta de Fourier.

Forward Pass	Movimento de predição de uma rede neural, onde seus vetores de entrada são transformados nos vetores de saída. Pode ocorrer durante o treinamento, a fim de aferir os erros, ou durante a utilização da rede treinada.
FPS	Frames per second, termo que denota a quantidade de dados reproduzidos por segundo em formatos de mídias digitais, como áudio e vídeo.
Framerate	Denota quantas amostras foram gravadas por segundo em formatos de mídias digitais, como áudio e vídeo.
GANs	Generative Adversarial Neural Networks, pares de redes neurais convencionais onde uma das redes gera saídas a serem classificadas pela outra, algumas vezes pré treinada, como autênticas ou não.
Grid search	Forma de investigação dos melhores parâmetros, em modelos onde sua relação não é independente. Idealmente, investiga-se todas as combinações possíveis de valores para cada um dos parâmetros.
Learning rate	Taxa de aprendizado de uma rede neural. Um valor que multiplica o gradiente dos erros, como forma de introduzir uma atualização suave dos pesos.
LMS	Least mean squares, mínima média dos quadrados, é uma métrica de erro muito utilizada, onde o objetivo do modelo é diminuir a média da soma dos quadrados dos erros.
LSTM	Long short-term memory são unidades utilizadas em redes recorrentes que tem como função regular as características que devem ser lembradas por longos períodos e as que devem ser esquecidas.
MIDI	Musical Instrument Digital Interface é uma linguagem introduzida para a normalização das conexões entre controladores e sintetizadores. É o padrão não formal de conexões entre DMIs e afins.
MNIST	Base de dados de dígitos escritos à mão, escaneados, processados e disponibilizados de forma gratuita. bastante utilizadas para a avaliação de modelos neurais da área de visão de computadores.
Overfitting	Ocorre quando um modelo perde a habilidade de generalização ao aprender de maneira muito próxima um exemplo simples em relação à sua arquitetura.
Pizzicato	É uma técnica em que o músico excita a corda, geralmente em instrumentos de arco, com o uso dos dedos.
Pooling	No contexto de redes convolucionais consiste na concatenação de dados de vários filtros; técnicas usuais são o max pooling, onde são

	utilizados os maiores valores e average pooling, onde as médias são utilizadas.
Round-robins	Amostras sonoras redundantes gravadas de uma mesma peça e articulação de um instrumento, com o objetivos de evitar o efeito machine gun durante a execução do DMI.
Samples	São os pontos de uma onda contínua amostrados, geralmente em intervalos de tempo regulares, durante o processo de digitalização.
Sampling rate	Denota quantas amostras foram gravadas por segundo em formatos de mídias digitais, como áudio e vídeo.
Smoothstep	Função sigmoidal utilizada para uma interpolação suave no contexto de gráficos digitais.
Tensorboard	Plataforma que permite a visualização do treinamento de redes neurais.
Transiente	Intervalo onde um sistema, do ponto de visto acústico, não alcançou a etapa periódica, e apresenta comportamento aberrante.
VST	Virtual Studio Technology, é uma especificação para o desenho de DMIs e efeitos a serem utilizados como plugins em softwares de manipulação e criação de áudio.