UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO (PPGCC)

LUCAS RAFAEL STEFANEL GRIS

Transcrição Automática de Sons Polifônicos de Guitarra na Notação de Tablaturas Utilizando Classificação Temporal Conexionista

Goiânia
2024

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

[X] Dissertação [ ] Tese [ ] Outro*:_____

*No caso de mestrado/doutorado profissional, indique o formato do Trabalho de Conclusão de Curso, permitido no documento de área, correspondente ao programa de pós-graduação, orientado pela legislação vigente da CAPES.

Exemplos: Estudo de caso ou Revisão sistemática ou outros formatos.

2. Nome completo do autor

Lucas Rafael Stefanel Gris

3. Título do trabalho

Transcrição Automática de Sons Polifônicos de Guitarra na Notação de Tablaturas Utilizando Classificação Temporal Conexionista

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:
a) consulta ao(à) autor(a) e ao(à) orientador(a);
b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Anderson Da Silva Soares**, **Professor do Magistério Superior**, em 25/10/2024, às 19:02, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Lucas Rafael Stefanel Gris**, **Discente**, em 29/10/2024, às 13:24, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4926930** e o código CRC **1228E25B**.

Referência: Processo nº 23070.038957/2024-86 SEI nº 4926930

LUCAS RAFAEL STEFANEL GRIS

Transcrição Automática de Sons Polifônicos de Guitarra na Notação de Tablaturas Utilizando Classificação Temporal Conexionista

Dissertação apresentada ao Programa de Pós–Graduação em Ciência da Computação do Instituto de Informática da Universidade Federal de Goiás, como requisito para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação.

Linha de pesquisa: Sistemas Inteligentes e Aplicações.

Orientador: Prof. Anderson da Silva Soares

Goiânia
2024

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Stefanel Gris, Lucas Rafael
Transcrição Automática de Sons Polifônicos de Guitarra na Notação de Tablaturas Utilizando Classificação Temporal Conexionista [manuscrito] / Lucas Rafael Stefanel Gris. - 2024.
LXXVIII, 75 f.: il.

Orientador: Prof. Dr. Anderson da Silva Soares.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Cidade de Goiás, 2024.
Bibliografia. Apêndice.
Inclui tabelas, lista de figuras, lista de tabelas.

1. Transcrição musical. 2. Transcrição Automática de Guitarra. 3. Recuperação da Informação Musical. I. da Silva Soares, Anderson, orient. II. Título.

CDU 5

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE DISSERTAÇÃO

Ata nº **22/2024** da sessão de Defesa de Dissertação de **Lucas Rafael Stefanel Gris**, que confere o título de Mestre em **Ciência da Computação**, na área de concentração em **Ciência da Computação**.

Aos vinte e três dias do mês de setembro de dois mil e vinte e quatro, a partir das nove horas,via sistema de webconferência, realizou-se a sessão pública de Defesa de Dissertação intitulada "**Transcrição Automática de Sons Polifônicos de Guitarra na Notação de Tablaturas Utilizando Classificação Temporal Conexionista**". Os trabalhos foram instalados pelo Orientador, Professor Doutor Anderson da Silva Soares (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professor Yuri de Almeida Malheiros Barbosa (UFPB), membro titular externo; Professor Doutor Gustavo Teodoro Laureano (INF/UFG), membro titular interno. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Anderson da Silva Soares, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos vinte e três dias do mês de setembro de dois mil e vinte e quatro.

TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Anderson Da Silva Soares**, **Professor do Magistério Superior**, em 23/09/2024, às 11:03, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Gustavo Teodoro Laureano**, **Professor do Magistério Superior**, em 23/09/2024, às 11:04, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Yuri de Almeida Malheiros Barbosa**, **Usuário Externo**, em 23/09/2024, às 12:45, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Lucas Rafael Stefanel Gris**, **Discente**, em 24/09/2024, às 15:36, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4737583** e o código CRC **47CD8627**.

Agradecimentos

Ao Prof. Dr. Anderson da Silva Soares, Prof. Dr. Arlindo Rodrigues Galvão Filho e Prof. Dr. Arnaldo Candido Junior pelos conhecimentos e experiências compartilhados, bem como todos os demais professores que colaboraram com meus conhecimentos nesta jornada.

A todos meus colegas do INF/UFG e o Centro de Excelência Artificial (CEIA) pelos conhecimentos e experiências compartilhados.

A minha família, que sempre me apoiou em meus estudos e momentos importantes da vida.

Every adversity, every failure, every heartbreak, carries with it the seed of an equal or greater benefit.

Napoleon Hill,
Think and Grow Rich.

Resumo

Gris, Lucas. **Transcrição Automática de Sons Polifônicos de Guitarra na Notação de Tablaturas Utilizando Classificação Temporal Conexionista**. Goiânia, 2024. 75p. Dissertação de Mestrado. Programa de Pós-graduação em Ciência da Computação (PPGCC), Instituto de Informática, Universidade Federal de Goiás.

A transcrição automática de guitarra, um ramo da transcrição musical automática, é uma tarefa com grande aplicabilidade para músicos de instrumentos trastejados como a guitarra elétrica e o violão. Frequentemente, os músicos desses instrumentos realizam a transcrição ou a leitura de músicas e peças musicais no formato de tablaturas, uma notação bastante utilizada para esse tipo de instrumento. Apesar da relevância, essa anotação ainda é feita de forma manual, sendo um processo bastante complexo, mesmo para músicos experientes. Nesse contexto, este trabalho propõe o uso de inteligência artificial para o desenvolvimento de modelos capazes de realizar a tarefa de transcrição de sons polifônicos de guitarra e violão de forma automática. Em particular, este trabalho investiga o uso de um método específico chamado Classificação Temporal Conexionista (CTC), um algoritmo que pode ser utilizado para treinar modelos de classificação de sequências sem a necessidade de alinhamento dos dados, aspecto fundamental para o treinamento de modelos mais robustos, uma vez que existem poucos conjuntos de dados disponíveis abertamente. Adicionalmente, este trabalho investiga o aprendizado multitarefa de predição de notas musicais além da predição de tablatura, obtendo melhorias consideráveis em relação ao aprendizado convencional. De uma forma geral, os resultados indicam que o uso do CTC é muito promissor para a transcrição de tablaturas, obtendo apenas 14,28% de queda relativa se comparado ao resultado obtido com dados alinhados.

Palavras–chave

Transcrição musical, Transcrição Automática de Guitarra, Recuperação da Informação Musical

Abstract

Gris, Lucas. **Automatic Guitar Transcription of Polyphonic Sounds in the Notation of Tablatures Using Connectionist Temporal Classification**. Goiânia, 2024. 75p. MSc. Dissertation. Programa de Pós-graduação em Ciência da Computação (PPGCC), Instituto de Informática, Universidade Federal de Goiás.

Automatic Guitar Transcription, a branch of Automatic Musical Transcription, is a task with great applicability for musicians of fretted instruments such as the electric guitar and acoustic guitar. Often, musicians on these instruments transcribe or read songs and musical pieces in tablature format, a notation widely used for this type of instrument. Despite its relevance, this annotation is still done manually, making it a very complex process, even for experienced musicians. In this context, this work proposes the use of artificial intelligence to develop models capable of performing the task of transcribing polyphonic guitar sounds automatically. In particular, this work investigates the use of a specific method called Connectionist Temporal Classification (CTC), an algorithm that can be used to train sequence classification models without the need for alignment, a fundamental aspect for training more robust models, as there are few openly available datasets. Additionally, this work investigates multi-task learning for note prediction alongside tablature prediction, achieving significant improvements over conventional learning. Overall, the results indicate that the use of CTC is very promising for tablature transcription, showing only a 14.28% relative decrease compared to the result obtained with aligned data.

Keywords

Automatic Music Transcription, Automatic Guitar Transcription, Music Information Retrieval

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

A música e a arte são manifestações presentes desde o início das primeiras civilizações [17]. Existem diversos instrumentos musicais que podem ser utilizados para produzir música, dentre eles, se destacam os instrumentos acústicos, em especial, a guitarra elétrica e a guitarra acústica (popularmente conhecida como violão), muito presentes na cultura brasileira [51].

As guitarras são instrumentos trastejados que produzem sons através da vibração de cordas, em que diferentes notas musicais são produzidas ao pressionar posições específicas do braço do instrumento. Frequentemente, os músicos precisam anotar ou verificar quais são as notas e os acordes a serem executados ou outras informações em um dado instante, para isso, utiliza-se notações musicais como as tablaturas e as partituras [1]. Segundo [6], a capacidade de transcrever música é um fascinante exemplo de inteligência humana, pois envolve percepção sonora, conhecimento de representações e estruturas musicais e capacidade de inferir e testar hipóteses.

As tablaturas e as partituras podem ser consideradas complementares, a primeira se caracteriza por fornecer informações a respeito da posição dos dedos para a execução dos sons, enquanto a segunda, se caracteriza por fornecer informações como notas musicais e ritmo de uma obra musical de forma precisa, permitindo que o músico tenha uma visão melhor da música a ser tocada [58]. Apesar de mais completa, a notação de tablatura pode ser considerada mais eficaz na transmissão do conteúdo musical para instrumentos musicais trastejados de corda, principalmente porque informam precisamente a posição dos dedos no instrumento para a execução das notas musicais [43], sendo amplamente adotadas por guitarristas [32].

Apesar da ampla adoção das tablaturas como forma de notação, a maioria das transcrições ainda é feita manualmente, onde os músicos precisam ouvir cuidadosamente a performance musical, talvez de forma repetitiva, inferindo a sequência de notas que estão sendo tocadas. Em geral, esse processo pode ser extremamente complexo e trabalhoso quando feito manualmente, mesmo para músicos experientes [10, 55]. Nesse sentido, a Transcrição Automática de Guitarra (AGT), ou Transcrição de Tablatura de Guitarra (GTT), tem a capacidade de tornar a prática de instrumentos musicais mais acessível ao

público, além de ter uma grande aplicabilidade no setor musical.

A AGT pode ser considerada um ramo da Transcrição Musical Automática (AMT), mas com foco no reconhecimento de sons produzidos pela guitarra. Atualmente, a maioria dos modelos para transcrição musical são treinados com o uso de áudios alinhados com suas respectivas notas musicais [6]. Outra alternativa possível, é o treinamento de modelos generativos *sequence-to-sequence* [33], contudo, esse tipo de modelo necessita de uma maior quantidade de dados, o que muitas vezes não está disponível. Nesse contexto, técnicas que possibilitem o treinamento de modelos sem a necessidade de dados alinhados se torna fundamental.

Uma das técnicas mais eficazes para o mapeamento de sequências de tamanhos variados como o áudio e o texto é a Classificação Temporal Conexionista (CTC) [20]. Esse algoritmo se mostrou bastante eficaz no contexto do transcrição de fala, pois permitiu o treinamento de modelos de inteligência artificial sem a necessidade de alinhamento de classes em cada janela de áudio, permitindo que o próprio modelo aprenda a gerar a saída alinhada e a predição correspondente. Nesse contexto, a utilização da CTC para a tarefa de transcrição musical pode se mostrar bastante promissora, uma vez que a obtenção de dados alinhados para o treinamento de modelos clássicos de classificação pode se mostrar extremamente custosa e demorada, além de possibilitar a geração de conjuntos de dados segmentados por meio da técnica de alinhamento forçado [35, 53].

Esse tipo de técnica ainda é pouco explorada em sistemas de transcrição de guitarra para tablaturas. Logo, o uso da CTC pode significar um grande avanço na área ao possibilitar o treinamento de dados sem alinhamento. Consequentemente, os modelos treinados com CTC podem ser utilizados para a criação de novos conjuntos de dados com o uso da técnica de alinhamento forçado. Nesse contexto, este trabalho propõe a investigação do uso da CTC para a tarefa de transcrição de guitarra no formato de tablaturas, comparando o uso dessa técnica com o treinamento utilizando dados alinhados.

1.1 Objetivos geral e específicos

O objetivo geral deste trabalho é avaliar o uso da CTC para a transcrição automática de sons polifônicos de guitarra no formato de tablaturas. Esse objetivo geral pode ser dividido nos seguintes objetivos específicos:

- Criar uma arquitetura de redes neural para a tarefa de transcrição;
- Comparar o desempenho dos modelos utilizando classificação com rótulos alinhados e a CTC sem alinhamento;
- Avaliar a aplicação da transcrição multi-rótulo para a transcrição de notas musicais;
- Demonstrar a possibilidade de utilização da CTC para a tarefa de alinhamento forçado em áudios não alinhados.

CAPÍTULO 2

Fundamentação teórica

Este capítulo apresenta alguns conceitos relacionados a pesquisa realizada. Inicia-se com conceitos iniciais de áudio e música, em seguida são apresentados conceitos de aprendizado profundo, o método CTC e a tarefa de reconhecimento musical. Por fim, são discutidos trabalhos relacionados a esta pesquisa.

2.1 Áudio e música

A música é propagada por meio do áudio, e o áudio consiste em vibrações no ar. Quando um objeto vibra, este produz uma sucessão de aumentos e diminuições na pressão do ar, que são então percebidas pelo ouvido humano. Esses eventos caracterizam uma onda sonora [7]. A percepção das ondas sonoras no decorrer do tempo pelo ouvido humano é chamado de som [36]. Segundo [59], o som é caracterizado como uma onda periódica ou como um ruído. O som que apresenta um padrão repetitivo é chamado de onda periódica, e o ruído é a ausência desse padrão.

A onda sonora apresenta propriedades importantes. Algumas dessas propriedades são: intensidade, altura, frequência, período, comprimento e duração. A intensidade corresponde a amplitude da onda, isto é, à diferença de pressão entre um pico e um vale da onda. A repetição de uma onda sonora é chamada de ciclo, e o número de ciclos que ocorrem por segundo, medida em *Hertz*, é definido como frequência [59]. A altura (em inglês, *pitch*) é definida como a percepção humana em relação a frequência de um som, e indica se um som é grave ou agudo. Em alguns cenários, principalmente na música, os sons apresentam uma frequência principal que se destaca. A frequência principal recebe o nome de frequência fundamental, e as demais frequências são chamadas de harmônicas [18]. Essa frequência fundamental está fortemente relacionada a percepção das notas musicais pelo ser humano. A música também apresenta alguns elementos importantes, tais como ritmo, melodia e harmonia, que se relacionam entre si [43].

Outro aspecto muito presente na música é o timbre. O timbre, uma característica perceptual complexa e subjetiva do som, é descrito em termos de sua cor ou qualidade tonal [40]. O timbre é um dos atributos mais abstratos e complexos da música, mas tam-

bém, um dos aspectos mais percebidos pelos ouvintes. Em geral, cada instrumento possui um timbre particular e mesmo instrumentos parecidos podem apresentar características de timbre distintas [40].

Em certos contextos se torna mais adequado representar o áudio visualmente. Uma forma de representação é o gráfico no domínio do tempo, chamado de oscilograma. Esse gráfico mostra a amplitude da onda no decorrer do tempo [59]. Outra forma de representar o áudio é por meio de espectrogramas. Os espectrogramas são representações visuais das frequências do som, ou seja, o som no domínio da frequência. A Figura 2.1 apresenta um oscilograma e um espectrograma de uma mesma nota musical produzida por um piano e um violino. É possível perceber diferenças de timbre dos dois instrumentos a partir do oscilograma e do espectrograma mostrado. Em especial, é possível notar diferenças consideráveis nas frequências produzidas pelos instrumentos e também na duração do som gerado. No campo da música, os espectrogramas podem facilitar o reconhecimento das notas, uma vez que são capazes de fornecer dados precisos da frequência em relação ao tempo [42].

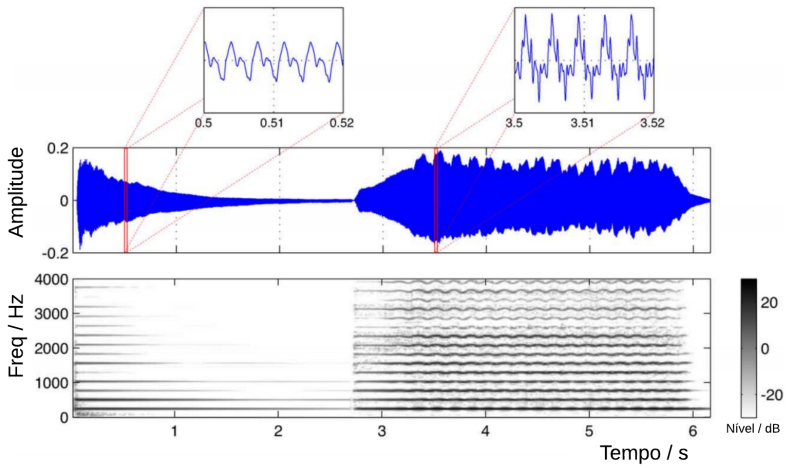


Figura 2.1: Representações no domínio da frequência de uma nota no piano e no violino. Na imagem, um som com a mesma frequência fundamental é produzido (mesma nota musical) primeiro no piano e depois no violino.

Fonte: Adaptado de [46].

2.1.1 Espectrogramas e *Constant-Q transform* (CQT)

O espectrograma pode ser considerado uma ferramenta básica de análise de áudio, principalmente porque fornece uma representação do áudio no domínio do tempo-frequência. O espectrograma é muito utilizado no campo do reconhecimento de voz [70]. Esse tipo de representação fornece características valiosas para algoritmos de aprendizado de máquina e de tomada de decisão [67].

O espectrograma pode ser calculado a partir da Transformada de Fourier de Curto Prazo (*Short-Time Fourier Transform*) (STFT), que é uma sequência de Transformadas Rápidas de Fourier (*Fast Fourier Transform*) (FFT) em segmentos específicos [70]. A FFT é um algoritmo eficiente para o cálculo da Transformada Discreta de Fourier (DFT), que é a transformada de Fourier utilizada em sinais discretos. A FFT captura o nível de energia em intervalos de frequência específicos (*bins*). Isso permite analisar como a energia de um sinal é distribuída em diferentes bandas de frequência, facilitando a identificação de componentes como tons, ruídos ou padrões periódicos em sinais complexos. A DFT é definida como na Equação 2-1, em que $X(k)$ é o valor da DFT na frequência k e representa a amplitude e a fase da componente de frequência k do sinal, $x(n)$ é o valor do sinal no tempo discreto no instante n e N é o número total de amostras do sinal. O termo $e^{-j\frac{2\pi kn}{N}}$ representa uma rotação complexa que decompõe o sinal em diferentes componentes e $X(k)$ é um número complexo que contém informações sobre a magnitude e a fase da componente de frequência k[1].

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi kn}{N}}, \quad k = 0, 1, \ldots, N-1 \tag{2-1}$$

A STFT (Equação 2-2) é fundamental para o cálculo do espectrograma, pois permite a análise de sinais não estacionários, como o áudio, que apresentam variações de frequência ao longo do tempo. Ela funciona dividindo o sinal contínuo em segmentos menores e aplicando a FFT em cada segmento, obtendo assim as componentes de frequência em diferentes instantes temporais. Essa abordagem é necessária porque, em sinais como o áudio, as características frequenciais mudam ao longo do tempo, e a STFT possibilita acompanhar essas mudanças, fornecendo uma representação tempo-frequência detalhada do sinal. Na Equação 2-2, $w(n)$ é a função janela[2] aplicada ao sinal, m é o índice que representa a posição temporal da janela, e R é o passo entre as janelas. Esses parâmetros podem ser ajustados para obter espectrogramas com diferentes características e resoluções de frequência ou tempo. O resultado desse cálculo é transposto para um gráfico tridimensional do tempo, frequência e intensidade. A Figura 2.2 ilustra o processo do algoritmo de STFT, em que uma série de transformadas de Fourier são calculadas em todo o áudio utilizando um tamanho de janela e passo específicos.

$$X(m,k) = \sum_{n=0}^{N-1} x(n) \cdot w(n - mR) \cdot e^{-j\frac{2\pi kn}{N}} \tag{2-2}$$

[1] Frequentemente utiliza-se apenas a magnitude para análise de sinais de áudio.

[2] Um dos tipos mais comuns de função de janelamento é a Hann [21], devido a suas características que minimizam artefatos e proporcionam uma representação mais precisa do sinal no domínio tempo-frequência.

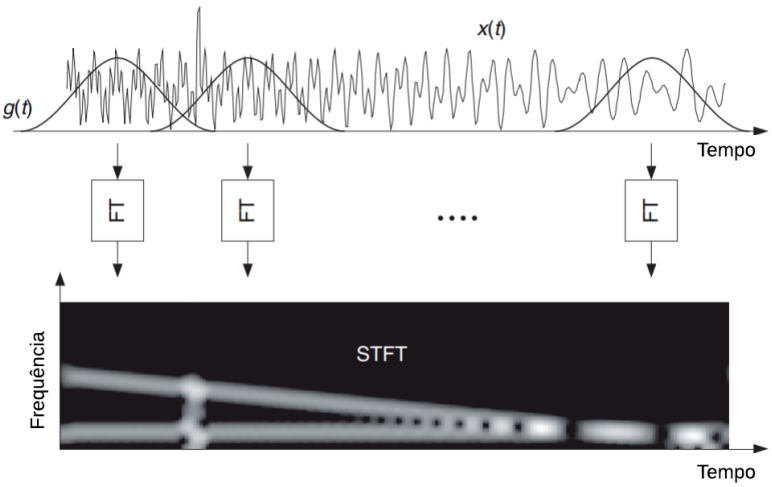


Figura 2.2: Processo de cálculo da STFT por meio de transformadas de Fourier. Na imagem, uma sequência de transformadas de Fourier é calculada em pequenos intervalos utilizando um janelamento Hann, gerando o gráfico do espectrograma.

Fonte: Adaptado de [29].

O espectrograma pode ser ajustado para diferentes escalas, permitindo que o eixo de frequências seja modificado para destacar melhor as faixas que o ouvido humano percebe com mais sensibilidade. Entre essas escalas, destacam-se a escala logarítmica e a escala de Mel, conforme mostrado na Figura 2.3. Nessas configurações, o eixo y dos espectrogramas é ajustado para seguir a escala de Mel ou a escala logarítmica, o que enfatiza as frequências mais baixas. A escala de Mel[3] é particularmente relevante, pois é baseada na forma como os seres humanos percebem as frequências[4] [71].

[3] O nome "mel"vem de *melody* (melodia), indicando que a escala é baseada nas percepções auditivas humanas.

[4] A propriedade da frequência percebida pelo ser humano é conhecida em inglês como *pitch*.

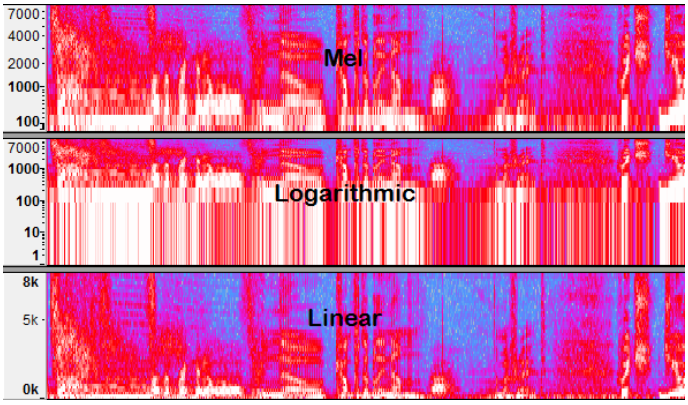


Figura 2.3: Espectrogramas em diferentes escalas.

Fonte: Extraído de [2].

Apesar de sua utilidade, a STFT possui uma limitação significativa: a resolução fixa no tempo e na frequência. Isso significa que, ao definir o tamanho da janela, há um compromisso entre a resolução temporal e a resolução frequencial. Janelas curtas oferecem boa resolução temporal, mas baixa resolução frequencial, e o oposto ocorre com janelas mais longas. Para lidar com essa limitação em contextos musicais, uma alternativa é a *Constant-Q Transform* (CQT). Diferentemente da STFT, que possui uma resolução uniforme, a CQT calcula as frequências em intervalos logarítmicos. Esse tipo de representação é especialmente útil no contexto da música, pois as notas músicais também são espaçadas em intervalos geométricos [66]. A CQT de um sinal discreto $x[n]$ é definida pela equação 2-3, onde $X_{CQ}(k,m)$ é o valor da CQT no tempo m e na frequência k, N_k é o tamanho da janela que varia com a frequência f_k para manter constante o fator Q, e $w_k[n]$ é a função janela aplicada.

$$X_{CQ}(k,m) = \sum_{n=0}^{N_k-1} x[n+m] \cdot w_k[n] \cdot e^{-j2\pi \frac{nf_k}{f_s}}, \quad k = 1,2,\ldots,K \tag{2-3}$$

O fator Q é definido como a razão entre a frequência central f_k e a largura de banda Δf_k de cada *bin* e pode ser definido como $Q = \frac{f_k}{\Delta f_k}$. Esse fator é mantido constante, o que significa que, à medida que a frequência aumenta, a largura de banda do *bin* também aumenta proporcionalmente, garantindo que a resolução frequencial seja ajustada de acordo com a escala logarítmica. As frequências f_k são calculadas conforme a equação 2-4, em que f_{min} é a frequência mínima a ser analisada e b é o número de *bins* por oitava.

$$f_k = f_{\text{min}} \cdot 2^{\frac{k}{b}}, \quad k = 0,1,\ldots,K-1 \tag{2-4}$$

Um dos grandes problemas ressaltados por [66] é o elevado custo computacional no cálculo da CQT se comparado a outras representações comumente usadas, como a DFT e o espectrograma de Mel. A Figura 2.4 apresenta um exemplo de CQT para um

trecho de áudio contendo a execução de um solo de guitarra. Apesar de bastante similar ao espectrograma e ao espectrograma de Mel, é possível perceber uma relação mais direta com as notas musicais do sistema ocidental.

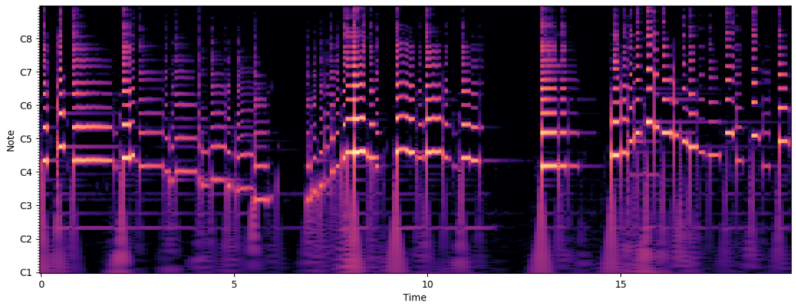


Figura 2.4: Exemplo de CQT para um trecho de áudio contendo a execução de um solo de guitarra.

2.1.2 Aumento de dados

Aumento de dados representam técnicas em que são produzidas instâncias artificiais a partir das instâncias existentes. O objetivo é aumentar um *dataset* e melhorar a capacidade de generalização de um modelo. O aumento de dados é especialmente útil no contexto do aprendizado profundo, uma vez que esse tipo de abordagem requer um treinamento requer com grande quantidade de dados para obter uma boa capacidade generalização [19].

As técnicas de aumento de dados são muito utilizadas para reconhecimento de padrões em imagens, e geralmente compreendem a rotação, distorção e translação dos dados. No campo do áudio, as técnicas são diferentes, e geralmente incluem a adição de ruido ou a alteração da entonação e da velocidade do áudio. Alguns exemplos podem ser visualizados na Figura 2.5.

Outra técnica muito utilizada é a de mascaramento de partes do espectrograma, como o SpecAugment [50] em que partes dos *bins* de frequência e *timesteps* são mascarados, forçando ao modelo a capacidade de encontrar informações úteis em diferentes regiões da entrada (Figura 2.6).

A aplicação de técnicas de aumento de dados também pode ser interessante no contexto da transcrição musical. No caso específico da guitarra, além das técnicas usuais de áudio, pode-se adicionar efeitos de guitarra, tais como o *delay*, *reverb* e a distorção (Figura 2.7). Esse tipo de técnica pode ser útil principalmente em conjuntos de dados limitados ou com poucas variações de timbre, simulando sons com diferentes características.

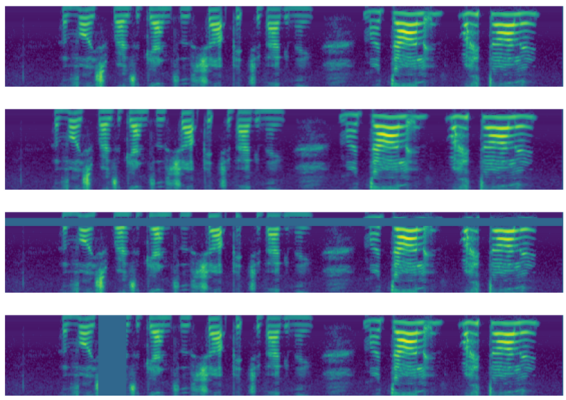
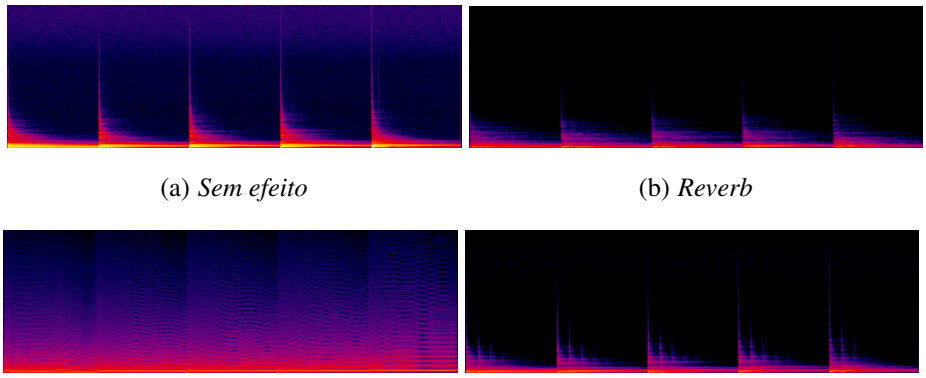


Figura 2.5: Exemplos de técnicas de aumento de dados aplicado ao espectrograma. De cima para baixo: (1) sem aumento de dados; (2) mudança de tempo em partes do áudio; (3) mascaramento de frequência e (4) mascaramento de tempo.

Fonte: Extraído de [50].

Figura 2.6: Técnica de aumento de dados SpecAugment.

Fonte: Adaptado de [50].



(a) *Sem efeito* (b) *Reverb*

(c) *Distorção* (d) *Delay*

Figura 2.7: Espectrogramas de alguns dos efeitos utilizados pelos guitarristas aplicados a um áudio.

2.1.3 Guitarra e violão

A guitarra e o violão podem ser considerados dois instrumentos de particular sucesso na música brasileira. O violão é vastamente utilizado na música popular [51]. Os dois instrumentos são visualmente semelhantes. Apresentam, na maioria das vezes, seis cordas. São trasteados, o que significa que possuem trastes que dividem as casas com

respectivas notas musicais no instrumento, e produzem sons por meio da vibração das cordas provocada pelos músicos por meio do uso de palhetas e/ou das mãos. A principal diferença é que os violões são compostos por uma caixa acústica que possibilita que o som seja transmitido pelo ar, enquanto as guitarras elétricas apresentam captadores que produzem uma saída elétrica a partir da vibração das cordas [41].

As cordas do instrumento apresentam características fisícas distintas, tais como material e espessura da corda, que possibilitam a execução de notas musicais com timbres variados. Cada posição entre um traste e outro no instrumento produz uma nota musical específica, contudo, devido as características do instrumento e dependendo da afinação adotada, a mesma frequência de som pode ser produzida em cordas diferentes. A Figura 2.8 apresenta a relação entre as notas musicais e suas respectivas frequências na guitarra usando a afinação padrão EADGBE. Apesar da diferença de timbre entre as cordas, essa diferença se torna mais sutil em cordas vizinhas, em que a espessura e o material são similares. Além disso, cada guitarra apresenta uma variação natural de timbre e encordoamentos distintos, o que aumenta ainda mais a possibilidade de timbres possíveis.

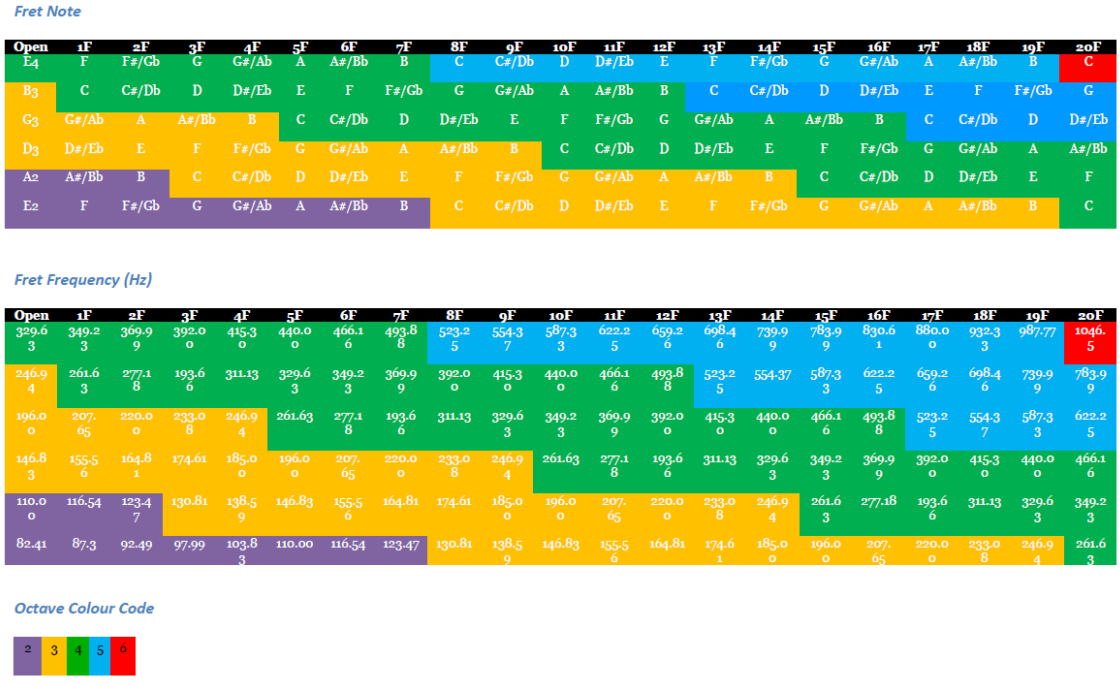


Figura 2.8: Notas na guitarra e no violão e suas respectivas frequências.

Fonte: Ower 89, CC BY-SA 3.0 <https://creativecommons.org/licenses/by-sa/3.0>, via Wikimedia Commons.

2.1.4 Notação musical

A notação musical determina as notas a serem executadas. A partitura é um tipo de notação musical que mostra o ritmo e o tempo das notas musicais e é a notação mais utilizada no mundo, sendo chamada de notação comum [65]. A partitura geralmente começa com algumas definições de notas e ritmos. Um exemplo de partitura pode ser visto na Figura 2.9.

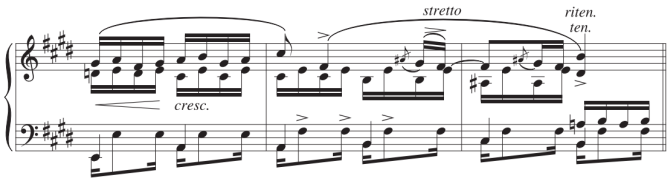


Figura 2.9: Exemplo de partitura

Fonte: Extraído de [1].

Apesar de amplamente utilizada, a tablatura é considerada mais adequada em alguns casos, principalmente porque em instrumentos como a guitarra e o violão, esse tipo de notação fornece informações precisas sobre as posições das execuções das notas no instrumento [43]. A tablatura é especialmente utilizada por músicos iniciantes, por ser uma forma simplificada de representar a música a ser tocada. Segundo Ramos (2016) [58], a tablatura é uma notação mais utilizada em instrumentos de corda, como a guitarra e o violão, pois se parece com o instrumento, como é possível observar na Figura 2.10 (parte inferior).

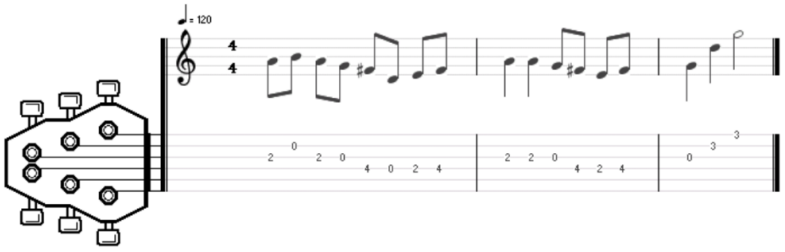


Figura 2.10: A tablatura como representação de instrumentos de cordas. Na figura, cada linha representa uma corda, e o número significa a posição dos dedos da mão no instrumento.

Fonte: Extraído de [58].

As tablaturas podem ser consideradas uma das melhores notações musicais para os instrumentos de cordas, entretanto, por mais simples que sejam, o processo de transcrição ainda é feito, na grande maioria das vezes, de forma manual. O problema de

transcrever tablaturas também é considerado uma tarefa difícil, principalmente devido a variedade de timbres de instrumentos [40] e da quantidade de notas disponíveis (Figura 2.8), além das oitavas presentes nos instrumentos [43]. Além disso, as tablaturas sugerem que existe uma certa relação em cada notação no sentido de facilitar a execução da música, isto é, a organização dos dedos deve ser feita da forma mais simples o possível, levando em consideração não somente a nota em si, mas a disposição dos dedos no instrumento. Segundo Ramos (2016) [58], a tarefa de transcrição para tablaturas é um processo complicado, porque a transcrição pode ser entendida como um problema combinatório, já que uma mesma nota de uma mesma frequência tem a possibilidade de ser executada em várias posições diferentes.

2.2 Redes neurais artificiais

As Redes Neurais Artificiais (ANNs) são um tipo de algoritmo específico de Aprendizado de Máquina (AM) com grande capacidade de generalização [19]. Em especial, o Aprendizado Profundo (AP), um tipo de abordagem na área de ANN em que muitas camadas ocultas são utilizadas com o intuito de promover a aprendizagem hierárquica de conceitos [19]. Dentre os tipos de ANNs existentes, pode-se citar redes do tipo *Multilayer Perceptron Feedforward* (MLP) e as Redes Convolucionais (CNNs), que se destacam pela sua simplicidade e sua capacidade de aprendizado.

As MLPs apresentam várias camadas em que a informação é propagada até a saída da rede neural [15, 19]. O principal componente das MLPs é o neurônio artificial (Figura 2.11), modelado como nas equações 2-5 e 2-6 [48], em que z_k é a saída da combinação linear do neurônio k, os valores de w_{kj} se referem aos pesos conectados aos sinais de entrada $x_1, x_2, ..., x_m$, e a saída a_k é obtida a partir da função de ativação ϕ e o potencial de ativação z_k. Adicionalmente, um *bias* b_k pode ser adicionado, deslocando o potencial de ativação. A função de ativação usualmente produz uma saída não-linear.

$$z_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \tag{2-5}$$

$$a_k = \phi(z_k) \tag{2-6}$$

Existem muitas funções de ativação disponíveis, tais como a função *Rectifier Linear Unit* (ReLU) [19, 47], muito utilizada devido a sua simplicidade e a possibilidade de desativar o neurônio dependendo do potencial de ativação; a função tangente hiperbólica, que possibilita a inclusão de valores negativos na saída do neurônio [22]; a *Exponential linear unit* (ELU) que também possibilita ativações negativas e busca tornar o processo de treinamento mais rápido [14]; além das funções Sigmoid e Softmax [8], muito utilizadas como funções de ativação na saída da rede. Cabe destacar também a Mish [44], uma fun-

ção de ativação não monotônica que apresentou desempenho superior a ReLU em vários cenários. A função Mish proporciona uma suavidade no gradiente que pode melhorar a convergência e estabilidade do treinamento em redes profundas.

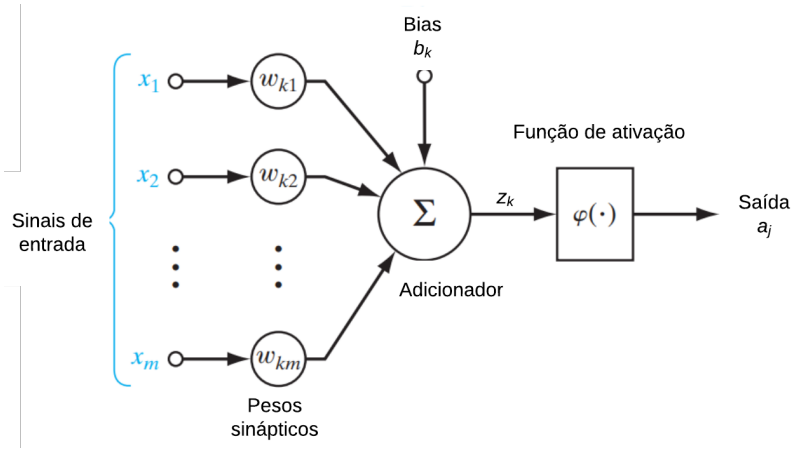


Figura 2.11: Neurônio artificial.

Fonte: Adaptado de [22].

2.2.1 Retropropagação de erros e o Gradiente Descendente

O algoritmo de retropropagação de erros (em inglês, *Backpropagation*) é o principal algoritmo utilizado para treinar uma ANN [19, 48], principalmente por ser relativamente simples e computacionalmente eficiente [38]. Esse algoritmo possibilita entender como uma pequena mudança em um dos parâmetros da rede afeta a sua saída, permitindo que cada parâmetro seja ajustado para que a saída seja o mais próximo do ideal possível [48]. Uma vez que os gradientes são calculados, o algoritmo do gradiente descendente pode realizar uma descida de encosta em direção ao ótimo local [48].

A função de custo desempenha um importante papel no processo de aprendizado da rede e é usualmente aplicada após a última camada da rede neural para fornecer o valor do erro obtido referente a entrada previamente fornecida. O algoritmo do gradiente descendente minimiza o custo C [48], já o cálculo do algoritmo de retropropagação de erros é realizado por meio da aplicação da regra da cadeia do cálculo, possibilitando a obtenção do valor escalar do gradiente em relação a qualquer nó na rede [19].

Existem quatro equações fundamentais, 2-7, 2-8, 2-9, 2-10, que possibilitam o cálculo do Backpropagation [48]. A Equação 2-7 se refere ao erro na última camada da rede e fornece o primeiro passo em direção ao cálculo do algoritmo de retropropagação de erros. A primeira parte da expressão, $\delta^S = \nabla_a C$, representa o vetor de derivadas parciais do custo com respeito às ativações na camada de saída, a segunda parte da expressão se refere à taxa de mudança da função de ativação na última camada. Uma vez que o erro δ

da última camada S é calculado, os erros das camadas anteriores L, $L-1$, ..., podem ser computados recursivamente (Equação 2-8) multiplicando a matriz de pesos transposta da camada $L+1$ pelo seu respectivo erro δ^{L+1}. Por fim, as equações 2-9 e 2-10 fornecem a derivada do custo com relação aos parâmetros da rede.

$$\delta^S = \nabla_a C \odot \phi'(z^S) \tag{2-7}$$

$$\delta^L = ((w^{L+1})^T \delta^{L+1}) \odot \phi'(z^L) \tag{2-8}$$

$$\frac{\partial C}{\partial b_j^L} = \delta_j^L \tag{2-9}$$

$$\frac{\partial C}{\partial w_{jk}^L} = a_k^{L-1} \delta_j^L \tag{2-10}$$

O Gradiente Descendente realiza a descida de encosta por meio de uma atualização direta nos parâmetros de pesos e *biases* da rede e toma como base o gradiente obtido no algoritmo de retropropagação de erros [45, 48]. A atualização do peso é feita como na Equação 2-11, e a atualização do *bias*, como na Equação 2-12. O valor do escalar do gradiente é multiplicado por uma constante especial η, denominada taxa de aprendizado, que determina o tamanho do passo em direção ao ótimo local [45]. Alguns autores recomendam taxas de aprendizado adaptativas, já que as camadas aprendem em velocidades distintas [22, 19, 79].

$$w_k' = w_k - \eta \frac{\partial C}{\partial w_k} \tag{2-11}$$

$$b_l' = b_l - \eta \frac{\partial C}{\partial b_l} \tag{2-12}$$

2.2.2 Redes neurais convolucionais

Em se tratando de detecção de padrões em imagens, as MLPs comuns apresentam uma grande limitação: elas são invariantes a ordem das características de entrada [81]. Contudo, em detecção de padrões em imagens, por exemplo, é essencial que as redes neurais sejam capazes de capturar e analisar características de pixels adjacentes. As Redes Neurais Convolucionais (*Convolutional Neural Networks*) (CNNs) [37] foram desenhadas para resolver esse problema. Esse tipo de rede neural introduz um conceito de receptor local, capaz de extrair características locais em estruturas espaciais [81].

As CNNs são redes que empregam operações conhecidas como convoluções, um tipo especial de operação linear [19]. Os principais conceitos introduzidos pelas CNNs

são: campos receptivos locais, que capturam características em uma região específica; pesos compartilhados, que possibilitam a varredura em todo o espaço e o *Pooling*, que diminui a dimensionalidade de um tensor [19]. A convolução das CNNs é discreta, como na Equação 2-13, na qual a função x representa a entrada enquanto w representa o *kernel*. A convolução $s(t)$ produz como resultado um mapa de características [19]. A Equação 2-13 é uma generalização da Equação 2-14 usada em dados bidimensionais, como imagens e espectrogramas. Um exemplo de como a convolução 2D funciona pode ser visualizada na Figura 2.12.

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{2-13}$$

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i-m,j-n) \tag{2-14}$$

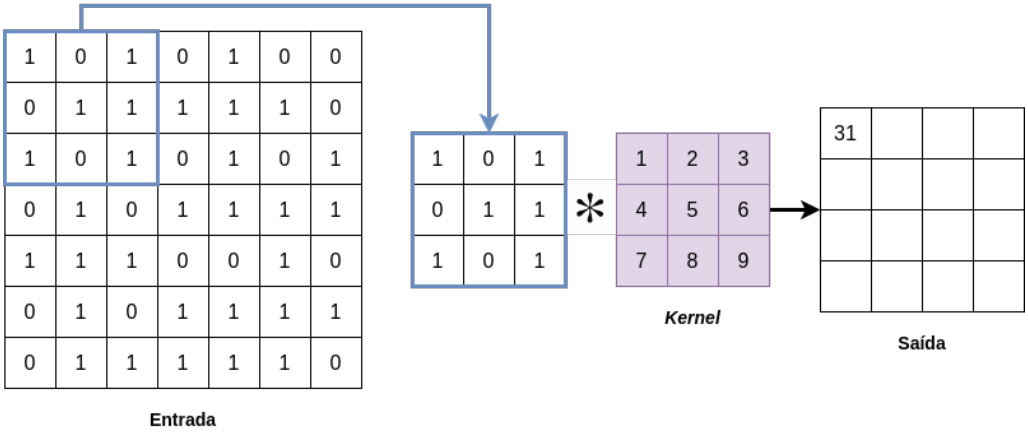


Figura 2.12: Exemplo de convolução 2D em uma CNN. A entrada sofre uma convolução a partir de um *kernel* aprendido e produz uma saída que corresponde a um mapa de características.

As camadas de *pooling* são muito utilizadas em conjunto com as camadas convolucionais pois simplificam as informações e diminuem a dimensionalidade do mapa de características [48], além de permitirem que a representação se torne invariante à translação [19]. Isso é útil quando a localização exata das propriedades na entrada não são tão relevantes, o que abrange a maioria dos problemas reais, como a localização dos olhos para a detecção das faces [19]. O *pooling* pode ser realizado de diversas formas [64]. Uma das técnicas mais comuns de *pooling* é o *max pooling* (Figura 2.13) [83], em que o maior valor de ativação da janela em análise é considerado.

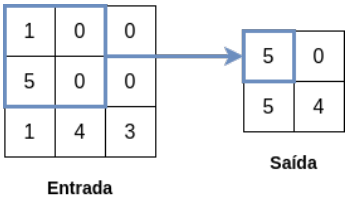


Figura 2.13: *Max pooling*: uma operação de *pooling* utilizada em redes convolucionais.

As CNNs apresentam grande aplicabilidade também no campo do processamento do áudio, uma vez que elas podem ser utilizadas para detectar padrões locais no áudio bruto ou a partir de características bidimensionais, como os espectrogramas (Figura 2.14).

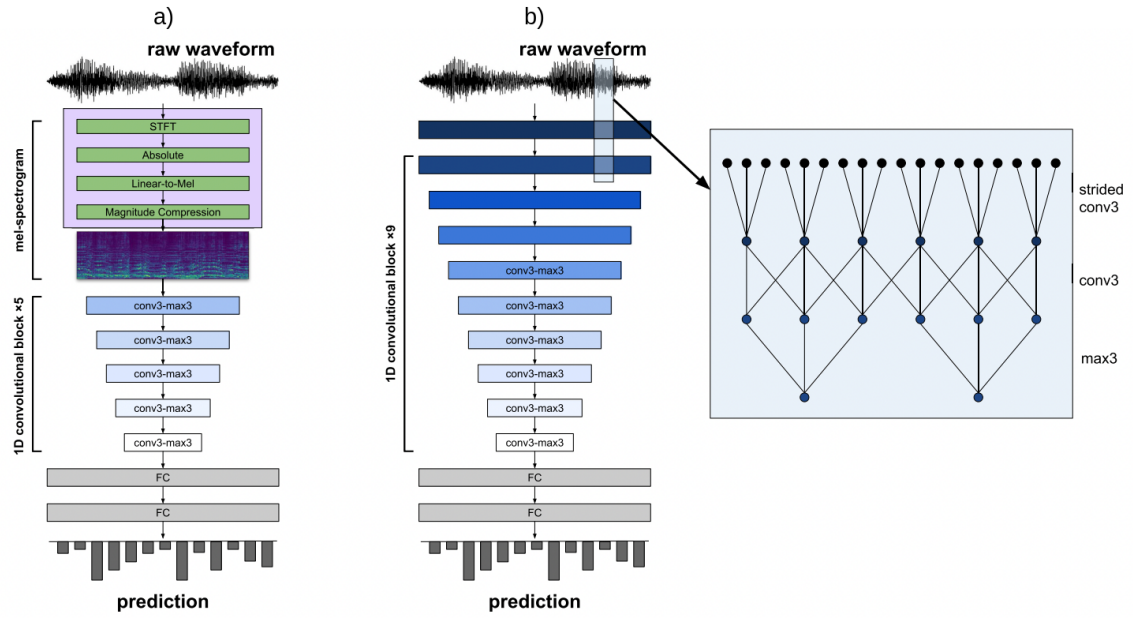


Figura 2.14: Exemplo de CNNs para classificação de áudio: em (a), o mel espectrograma é extraído a partir do áudio e fornecido a camadas convolucionais bidimensionais, e em (b), o áudio bruto é fornecido diretamente a camadas convolucionais unidimensionais. A figura também destaca, na direita, o uso de camadas convolucionais com *stride* e *max-pooling* que possibilitam a captura de mais informações do áudio.

Fonte: Extraído de [39].

2.2.3 Mecanismos de atenção

O mecanismo de atenção foi inicialmente proposto para permitir que o modelo de rede neural fosse capaz de dar mais importância a partes da sentença de entrada na tarefa de tradução de máquina [3]. Em linhas gerais, codificar toda a entrada em um vetor de

tamanho fixo para uma posterior decodificação apresenta alguns desafios, principalmente em sequências longas, uma vez que partes da informação de entrada podem ser mais relevantes que outras. O mecanismo de atenção soluciona esse problema ao permitir que a informação de entrada tenha pesos de importância correspondentes a uma determinada *query* [3].

O mecanismo de atenção (Figura 2.15) é definido como na Equação 2-15 [81], em que $\alpha(\mathbf{q}, \mathbf{k}_i) \in \mathbb{R}(i = 1, \ldots, m)$ são pesos de atenção, e $\mathbf{q}$ (*query*), $\mathbf{k}$ (*key*) e $\mathbf{v}$ (*value*) são todos vetores. A saída da função de atenção é o resultado de uma soma ponderada dos valores após a aplicação dos pesos de atenção calculados em função de $\mathbf{q}$ e a chave $\mathbf{k}$ correspondente [73]. Os pesos também são aplicados em uma função Softmax, com o intuito de evitar valores negativos e obter uma distribuição de probabilidades [81].

$$\text{Atenção}(\mathbf{q}, \{(\mathbf{k}_1, \mathbf{v}_1), \ldots (\mathbf{k}_m, \mathbf{v}_m)\}) \stackrel{\text{def}}{=} \sum_{i=1}^{m} \alpha(\mathbf{q}, \mathbf{k}_i)\mathbf{v}_i \tag{2-15}$$

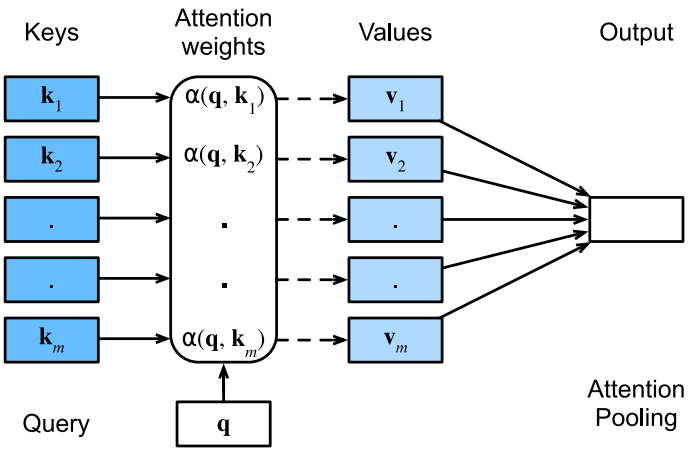


Figura 2.15: Mecanismo de atenção.

Fonte: Extraído de [81].

O Transformer [73] é um tipo de arquitetura de rede neural construído com o uso de mecanismos de atenção para modelar relações entre sequências. Os autores propuseram o uso de várias cabeças de atenção que funcionam paralelamente (Figura 2.16). Esse tipo de abordagem se mostrou bastante eficiente, uma vez que permite que várias dependências sejam capturadas pelo modelo ao mesmo tempo [81]. Outro grande avanço explorado por [73] foi o uso da auto-atenção (*self-attention*), que pode ser entendida como a atenção aplicada levando em consideração apenas a própria informação de entrada ($q = k = v$), permitindo o aprendizado de relações em uma mesma sequência. Além desses avanços, os autores também implementam o *Scaled Dot-Product Attention*,

uma função computacionalmente eficiente baseada no produto escalar que realiza um escalonamento baseado na dimensão da entrada ao calcular a atenção.

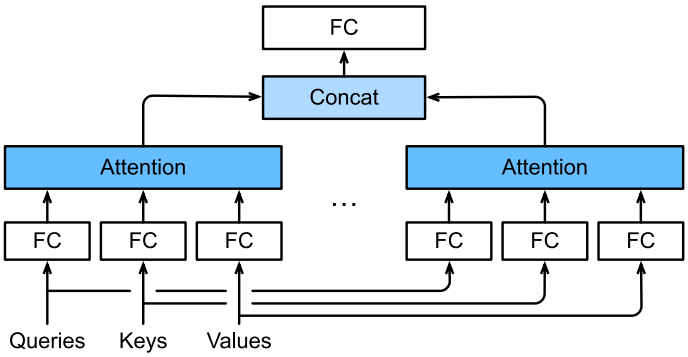


Figura 2.16: Mecanismo *Multihead Attention*.

Fonte: Extraído de [81].

Uma limitação do *self-attention* é o fato de não ser capaz de atribuir pesos de importância com base na localização da informação por si só [73, 81]. Isso ocorre porque a atenção calcula todos os pesos de forma paralela, diferentemente de redes recorrentes, por exemplo, em que a informação é computada sequencialmente [19]. Uma forma de resolver esse problema é utilizar um método de codificação posicional que adiciona uma informação temporal a informação original de forma que o modelo entenda a posição de cada item (*token*) da sequência [81]. A arquitetura Transformer é um exemplo, em que uma função baseada em cosenos e senos é utilizada para adicionar um valor a cada *token* da sequência na entrada do modelo.

2.3 Classificação Temporal Conexionista (CTC)

A tarefa de fornecer uma sequência de entrada para predizer outra sequência, como em reconhecimento de fala (ASR), é uma tarefa bastante desafiadora [11]. No ASR por exemplo, a maioria dos dados disponíveis para treinamento não são anotados de forma segmentada a nível de cada unidade da fala, o que inviabiliza a utilização de funções de perda em problemas de classificação comuns, como a entropia-cruzada [54]. Uma solução para esse problema é a utilização do algoritmo Classificação Temporal Conexionista (em inglês, *Connectionist Temporal Classification*) (CTC), introduzida por Graves et. al (2006) [20]. A Figura 2.17 apresenta um exemplo de como a CTC pode ser útil no contexto de predição de sequências de caracteres. É possível perceber que, independentemente do tamanho da imagem ou do áudio de entrada, o algoritmo é capaz de gerar a mesma saída.

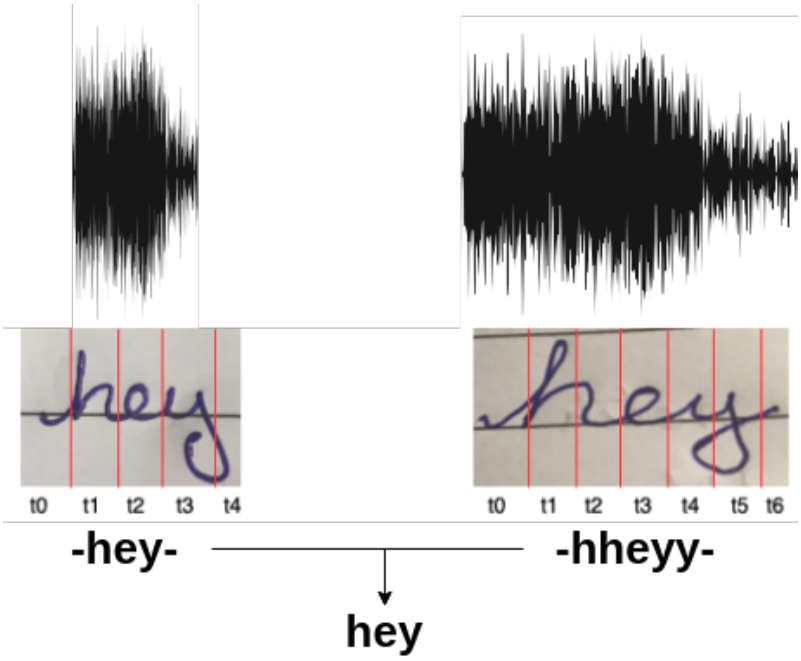


Figura 2.17: O método CTC para geração de sequências. Na figura, dois exemplos de imagem ou áudio com tamanhos diferentes apresentam a mesma saída esperada (*hey*).

Fonte: Adaptado de [4].

Rotular dados sequenciais é um problema presente em muitas tarefas como reconhecimento de voz (ASR) e reconhecimento de escrita (OCR). Isso ocorre porque os dados rotulados em problemas reais são discretos, como sequências de letras e palavras [20], mas eles não apresentam uma segmentação clara com o dado de entrada que é contínuo. Nesse contexto, a CTC aparece como uma alternativa viável pois dispensa a necessidade de dados pré-alinhados no treinamento [20].

O algoritmo CTC se baseia em uma saída do tipo Softmax contendo o vocabulário de saída com um caractere especial *blank*, que tem a função de permitir a repetição de caracteres ou indicar a presença de "silêncio" em um *timestep* específico. A saída Softmax fornece uma distribuição probabilística dos rótulos no tempo. Com estas saídas obtidas, é possível traçar caminhos que correspondem ao rótulo desejado. Isso permite que a CTC calcule o custo do modelo a partir de rótulos não alinhados [54]. A CTC basicamente possibilita o cálculo das probabilidades dos alinhamentos de cada caractere, dado uma determinada entrada e sua respectiva saída. O mapeamento $B : a \rightarrow y$ do alinhamento a para a saída y pode ser obtido realizando uma busca sobre o espaço de probabilidades seguido pelo colapsamento das repetições (Figura 2.18) [28].

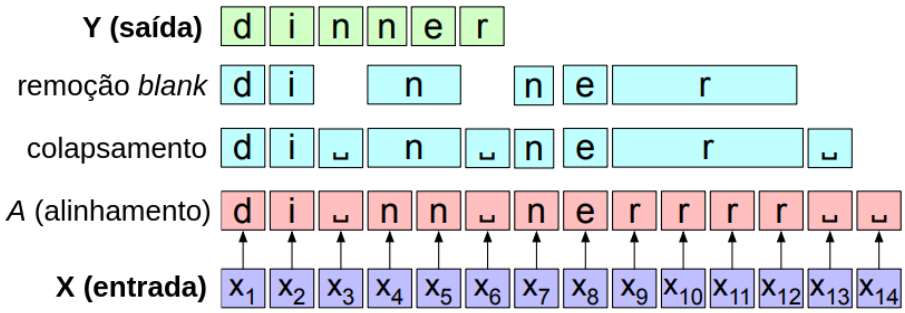


Figura 2.18: Exemplo de colapsamento das saídas repetidas com CTC após o alinhamento da sequência de entrada com a sequência de saída. Os caracteres repetidos consecutivamente e o caractere especial *blank* em A são removidos para gerar a saída Y.

Fonte: Adaptado de [28].

O método CTC assume algumas propriedades: (1) primeiro, a função de colapsamento das saídas repetidas para a saída desejada é *many-to-one*, o que significa que a entrada X é maior ou igual a saída Y, e (2), a relação entre a entrada e a saída é monotônica, isto é, as sequências de entradas seguintes sempre irão corresponder a saída atual ou a saídas seguintes, nunca a saídas anteriores. Isso é satisfeito em problemas como o ASR, mas não em outros problemas *sequence-to-sequence*, como a tradução, em que a geração do texto de saída pode depender de uma entrada ainda não processada.

A probabilidade de um alinhamento em particular $\hat{A} = \hat{a}_1, \ldots, \hat{a}_T$ é resultado de um produtório em todos os *timesteps* dada uma entrada X (Equação 2-16). Como pode ser observado, a CTC não assume uma dependência entre as saídas em diferentes *timesteps*, portanto é possível escolher o melhor alinhamento $\hat{A}$ simplesmente selecionando o rótulo de máxima probabilidade em cada *timestep* t (Equação 2-17) [28]. Esse algoritmo é conhecido como busca gulosa (*greedy search*). Contudo, em alguns casos uma busca mais adequada pode ser mais adequada. Uma possível solução é realizar um somatório de todos os alinhamentos possíveis da inversa do mapeamento B^{-1} da saída Y (Equação 2-18). O custo computacional da busca sobre todo o espaço de alinhamentos é bastante elevado, por isso o algoritmo *beam-search*, que limita o espaço de busca, é comumente utilizado [28].

$$P_{\text{CTC}}(A \mid X) = \prod_{t=1}^{T} p\left(a_t \mid X\right) \tag{2-16}$$

$$\hat{a}_t = \underset{c \in C}{\operatorname{argmax}} p_t(c \mid X) \tag{2-17}$$

$$
\begin{aligned}
P_{\text{CTC}}(Y \mid X) &= \sum_{A \in B^{-1}(Y)} P(A \mid X) \\
&= \sum_{A \in B^{-1}(Y)} \prod_{t=1}^{T} p\left(a_t \mid h_t\right) \\
\hat{Y} &= \underset{Y}{\operatorname{argmax}} P_{\text{CTC}}(Y \mid X)
\end{aligned}
\tag{2-18}
$$

A função de custo CTC pode ser entendida como o log da verossimilhança negativa (*negative log likelihood*) da probabilidade de uma saída Y dada uma entrada X (Equação 2-19). Novamente o custo computacional neste caso é bastante elevado, sendo praticamente intratável. Para solucionar esse problema, o algoritmo *forward-backward* é utilizado [20]. Esse algoritmo se baseia em programação dinâmica e guarda os resultados parciais das probabilidades já calculadas, otimizando o cálculo da função de custo para o treinamento do modelo.

$$
L_{\text{CTC}} = \sum_{(X,Y) \in D} -\log P_{\text{CTC}}(Y \mid X) \tag{2-19}
$$

Na maioria das tarefas, como ASR e OCR, a sequência de saída gerada após a busca e o colapsamento dos *tokens* corresponde ao objetivo da tarefa, isto é, em ASRs e OCRs, o objetivo é obter a sequência de saída, não o alinhamento. Contudo, o alinhamento gerado apresenta outras aplicações. Uma delas é o alinhamento forçado, técnica que pode ser utilizada para alinhar conjuntos de dados não rotulados. Outra aplicação reside no campo da transcrição musical, em que a duração e o *timestep* que cada saída ocorre pode ser utilizado para definir o momento que uma nota começa e termina, por exemplo.

2.3.1 Alinhamento forçado com CTC

Modelos de transcrição baseados em aprendizado profundo necessitam de uma grande quantidade de dados para treinamento [54]. Mesmo utilizando algoritmos de treinamento que dispensam o alinhamento, por exemplo usando a CTC, ainda é necessário segmentar áudios longos em trechos mais curtos, preservando a respectiva parte da transcrição em cada trecho [35]. Isso é necessário porque o treinamento de modelos de aprendizado profundo com áudios longos se torna computacionalmente ineficiente, principalmente em arquiteturas baseadas em mecanismos de atenção, em que o custo computacional cresce quadraticamente em relação ao tamanho do áudio [53]. O alinhamento forçado é uma tarefa que realiza o alinhamento dos dados com base em algum algoritmo específico, possibilitando a geração de conjuntos de dados com áudios segmentados [35, 53] (Figura 2.19). Em relação a outras técnicas, o alinhamento forçado com CTC apresenta uma vantagem: a possibilidade de alinhar sequências com erros, como em audiolivros em

que o áudio contém mais informação que o texto narrado [35]. A Figura 2.20 ilustra o processo de alinhamento a partir da saída de probabilidades do modelo acústico treinado com CTC.

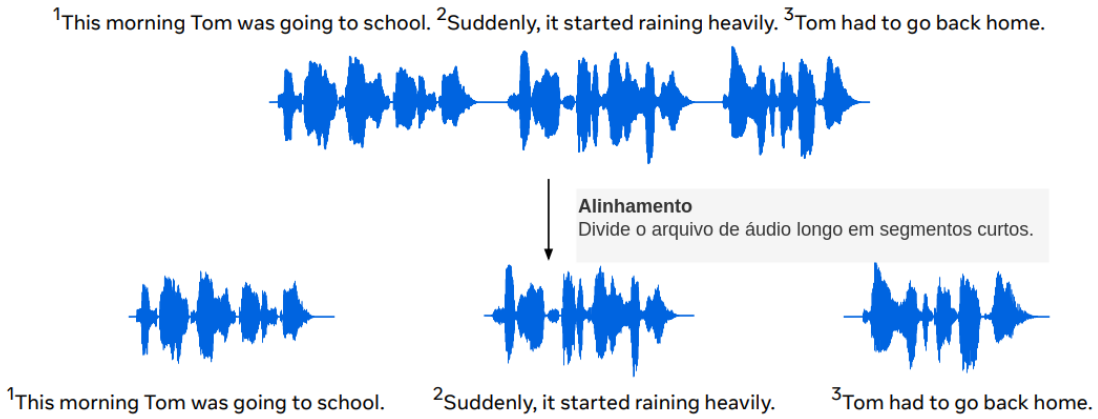


Figura 2.19: Ilustração da segmentação de um áudio com alinhamento forçado.

Fonte: Adaptado de [53].

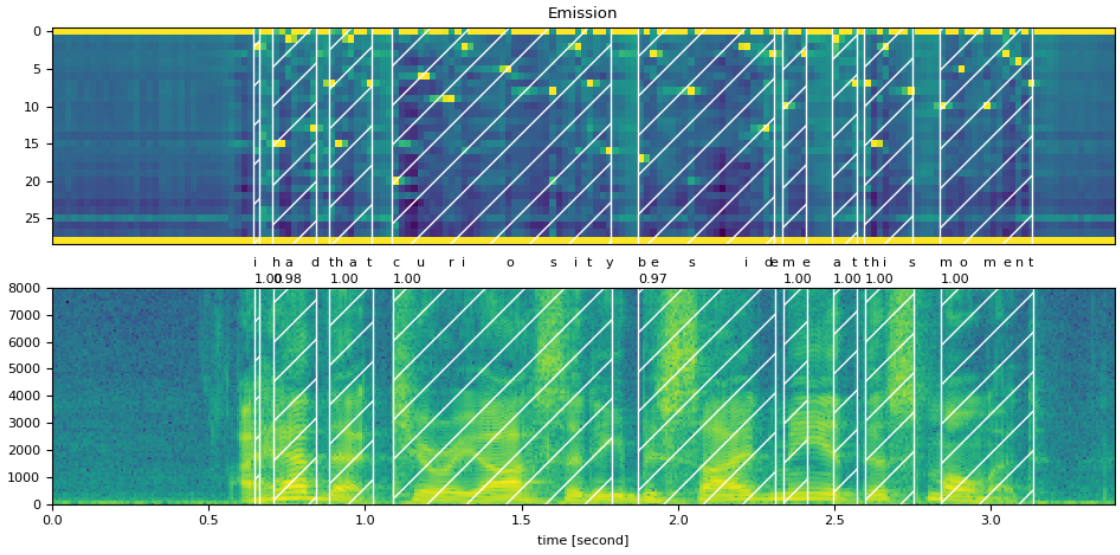


Figura 2.20: Ilustração de alinhamentos obtidos com CTC. Na figura, a imagem superior corresponde a saída probabilística do modelo, e a imagem inferior ao espectrograma do áudio. A figura mostra o início e a duração de cada palavra e caractere da sentença.

Fonte: Extraído de [82].

Para que o alinhamento forçado com CTC funcione, é necessário que um modelo com o algoritmo CTC tenha sido previamente treinado. Em algumas tarefas, como a tarefa de reconhecimento de fala, isso não é uma limitação, uma vez que esse tipo

de técnica tem sido explorado a mais tempo e há uma quantidade razoável de dados segmentados disponíveis, exceto para idiomas com poucos recursos [53]. Em [53], os autores realizaram a criação de um conjunto de dados massivo multilingual para a tarefa de ASR, escalando o reconhecimento de fala para mais de 1.000 idiomas diferentes. A tarefa de alinhamento é desafiadora: os autores treinaram dois modelos de alinhamento, um com dados até então disponíveis e um segundo com os novos dados alinhados. Esse tipo de técnica apresenta grande aplicabilidade em cenários de escassez de recursos, possibilitando a geração de conjuntos de dados para o treinamento de modelos robustos [53].

2.3.2 CTC multirótulo (MCTC)

Uma das grandes limitações da CTC é o fato do algoritmo necessitar de um espaço de probabilidades apenas multi-classe, contudo, em algumas situações, é necessário realizar a classificação multi-rótulo, isto é, uma classificação em que várias classes possam ser preditas em cada *timestep*. Um exemplo é a classificação da escrita chinesa ou a transcrição musical. Na escrita chinesa, a CTC multi-rótulo poderia permitir a transcrição dos componentes de cada caractere (Figura 2.21), e na transcrição musical, a CTC multi-rótulo poderia permitir a transcrição de várias notas ao mesmo tempo (Figura 2.22), o que é bastante comum em vários instrumentos polifônicos.

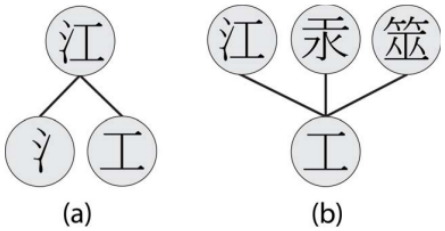


Figura 2.21: Exemplo de classificação multi-label para caracteres chineses: (a) caracteres chineses podem ser decompostos em componentes únicos; (b) um radical pode aparecer em vários caracteres.

Fonte: Extraído de [76].

É possível treinar o CTC para uma predição multi-rótulo por meio do produto cartesiano de todos os rótulos. Contudo, essa abordagem se torna intratável uma vez que o número de rótulos aumenta substancialmente [76]. Outra solução é a utilização de várias funções CTC para cada tipo de categoria C, os autores chamam essa abordagem de *Separable CTC* (SCTC), mas ressaltam que esse tipo de abordagem apresenta várias limitações, uma vez que cada categoria é tratada como distinta uma da outra. O *Multi-label Connectionist Temporal Classification* (MCTC) [76] é uma variação da CTC que

busca resolver as limitações desse algoritmo para problemas multi-rótulo e que permite o treinamento do modelo para a predição de várias classes, obtendo resultados estado-da-arte em várias tarefas.

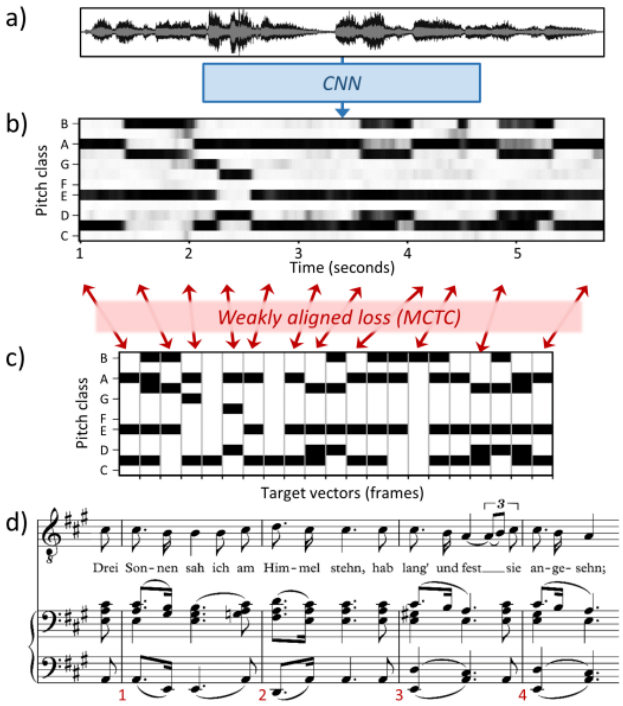


Figura 2.22: Exemplo de classificação multi-label para transcrição musical: (a) áudio; (b) saída do modelo; (c) saídas não alinhadas; (d) partitura; o MCTC calcula o custo a partir da saída probabilística do modelo e a saída não alinhada esperada.

Fonte: Extraído de [74].

O MCTC é definido como na Equação 2-20, em que $(l \mid X)$ é a probabilidade do rótulo l dado a entrada X e é definido pelo somatório de todos os caminhos possíveis no espaço de probabilidades fornecido. Todos os rótulos do problema são adicionados em um conjunto de categorias $C = \{C_1, C_2, ..., C_n\}$ que pode conter o rótulo especial *blank*. Os autores propõe duas versões do MCTC para definir o caractere *blank* do MCTC ($blank_{\text{MCTC}}$): MCTC:NE define que o $blank_{\text{MCTC}}$ é obtido se em um *timestep* específico todos os componentes forem *blank*, e MCTC:WE que adiciona mais um "rótulo"que corresponde a saída *blank* ou não-*blank*, simplificando a operação. O MCTC é uma generalização da CTC em que $|C| = 1$ (apenas um componente), C_1 = alfabeto $\cup$ *blank* e $blank_{\text{MCTC}} = blank$ [76].

$$P(l \mid X) = \sum_{A \in \mathcal{B}^{-1}(l)} \prod_{t=1}^{T} \prod_{i=1}^{|C|} \begin{cases} p\left(a_t \mid X\right)_{i,a_i^t}^{t}, & \text{if } a_i^t \neq \varepsilon \\ 1, & \text{caso contrário} \end{cases} \tag{2-20}$$

A Figura 2.23 ilustra os alinhamentos obtidos para o SCTC e as diferentes

versões do MCTC para um exemplo multi-tarefa de transcrição e detecção de entidades contendo três categorias. É possível observar as limitações do SCTC em relação a predição independente das classes.

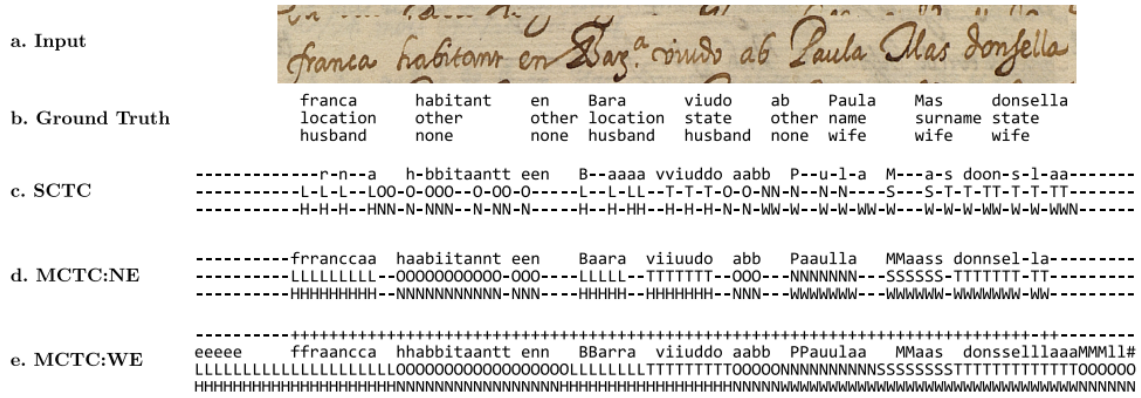


Figura 2.23: Ilustração do SCTC e das variações do MCTC. Os rótulos foram abreviados em localização (L), outro (O), nome (N), *wife* (W) e *husband* (H). O caractere − significa *blank*.

Fonte: Extraído de [76].

No contexto de transcrição musical, o MCTC pode ser modelado de forma que cada nota musical corresponde a uma categoria específica, o que permite que várias notas (categorias) sejam preditas ao mesmo tempo, além do silêncio ($blank_{\text{MCTC}}$).

2.4 Transcrição Musical Automática (AMT)

A AMT é um campo de pesquisa que visa desenvolver algoritmos e sistemas capazes de converter áudio musical em notação musical. Usualmente um sistema de AMT tem como objetivo a geração de representações das alturas (*pitches*) no tempo, podendo também ter como objetivo a geração de notação musical na sua forma mais completa (Figura 2.24) [6].

Segundo Benetos et al. (2008) [6], a área de AMT apresenta vários desafios. Primeiro, sons polifônicos são formados por várias notas e sons sendo executados simultaneamente, e boa parte dos instrumentos musicais produzem sons dessa forma. Além disso, em alguns casos, a AMT pode ser empregada em áudios contendo mais de um instrumento, o que aumenta ainda mais o grau de polifonia do áudio. Segundo, a sobreposição dificulta a obtenção da relação harmônica entre os sons. Terceiro, o tempo da execução das notas musicais é, em geral, um fator determinante para um AMT. Quarto, a anotação de dados para o treinamento de modelos supervisionados é uma tarefa complexa

e custosa. Além desses fatores, a transcrição de instrumentos pode apresentar desafios específicos desses instrumentos, além de presença de ruído e baixa qualidade de áudio.

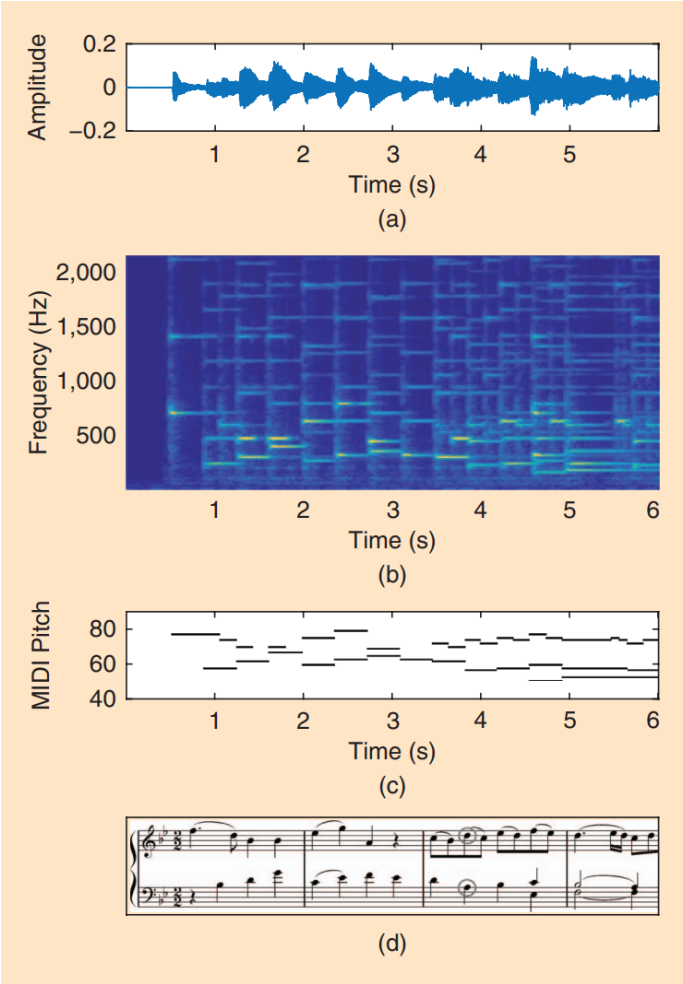


Figura 2.24: Representações envolvidas no processo de AMT: (a) forma da onda, (b) representação no domínio da frequência, (c) representação *piano-roll* (*multi-pitch*) e (d) notação musical (partitura).

Fonte: Extraído de [6].

Existem três tipos de transcrição: a transcrição a nível de *frame*, que corresponde a tarefa de estimar a altura dos sons em cada *frame* de áudio; a transcrição de nota, que também procura obter o tempo exato em que as notas são executadas; e por fim a transcrição a nível de *stream*, que corresponde a tarefa de transcrição de vários instrumentos ao mesmo tempo e a separação das fontes de forma adequada [6]. Para a transcrição de um único instrumento, apenas a transcrição a nível de *frame* e a nota podem ser aplicadas, sendo que a transcrição a nível de *frame* é mais comum na literatura. A Figura 2.25 ilustra as diferenças entre esses dois tipos de transcrição.

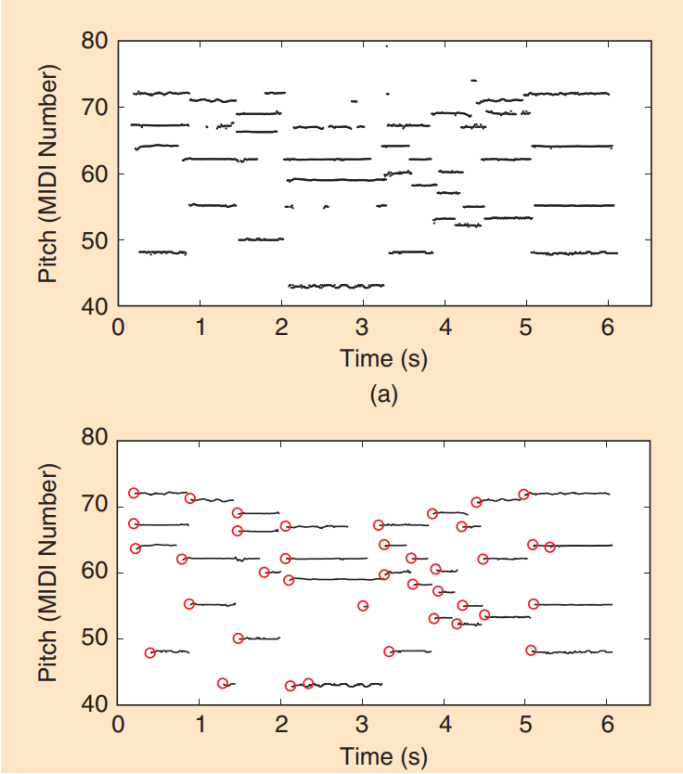


Figura 2.25: Transcrição a nível de *frame* (a) e (b) nota na AMT.

Fonte: Adaptado de [6].

Os primeiros trabalhos na área foram muito limitados quanto a quantidade e tipo de instrumentos utilizados. Uma das primeiras propostas de sistemas AMT procuraram obter a nota diretamente a partir da análise da frequência fundamental no espectrograma. [52] propuseram um sistema AMT de análise de áudio contendo apenas um único instrumento e com frequências fundamentais claras [63].

Algumas das abordagens relacionadas ao desenvolvimento de sistemas AMT incluem a utilização de ANNs, algoritmos de Fatoração de Matrizes Não-Negativas (*Non-negative matrix factorization*) (NMF), modelagem probabilística, abordagens bayesianas, e técnicas clássicas de processamento de sinais. Apesar de existirem muitos métodos em AMT, as principais abordagens aplicadas na última década foram os algoritmos de ANNs e NMFs [6].

Com relação ao uso de ANNs, alguns trabalhos apresentam resultados promissores [30, 31, 69]. Esses trabalhos diferem quanto a abordagem utilizada, o contexto de uso, instrumentos e limitações, mas compartilham o uso de ANNs como método principal para a realização do processo de transcrição. [69] apresentam um modelo híbrido de RNN treinado de ponta a ponta composto por um modelo acústico e um modelo de linguagem, que, juntos, são capazes de reconhecer as notas musicais no tempo. Segundo [69], as limitações do trabalho se referem principalmente a quantidade de dados para treinamento.

Uma área específica da AMT é a Transcrição Automática de Guitarra (AGT). Nessa tarefa a entrada do áudio corresponde a sons produzidos por instrumentos como guitarra, baixo e violão. Por natureza, esses instrumentos produzem sons polifônicos. Uma característica adicional desse tipo de tarefa é que, devido as características do instrumento, a notação de tablatura é mais comumente utilizada do que a representação de partitura.

Assim como outras áreas da AMT, a AGT também pode ser realizado a nível de *frame* ou de nota. A forma de tablaturas é normalmente considerada como transcrição a nível de nota, mas alguns autores caracterizam essa forma como uma categoria a parte [25], devido a características especiais desse tipo de tarefa.

CAPÍTULO 3

Trabalhos correlatos

A AGT é uma tarefa mais específica da AMT em que o áudio de entrada corresponde a instrumentos da família das guitarras. Comparado a área da AMT, os trabalhos relacionados a esse tipo de tarefa podem ser considerados ainda em fase inicial na literatura, mas já existem alguns trabalhos relevantes que podem ser estudados.

Em Paleari et al. (2008) [49], os autores combinaram a utilização de vários sistemas para realizar o processo de transcrição, incluindo módulos para rastreio da posição das mãos no instrumento por meio de vídeo, além da análise do áudio. Os resultados do trabalho mostraram-se promissores, mas limitados, principalmente quanto à realização de diferentes técnicas musicais e à execução de vários sons simultaneamente (polifonia). Em Barbancho et al. (2011) [5], os autores procuraram reconhecer os acordes tocados por meio de técnicas de HMM. Outro trabalho a ser citado é o de Shibata et al. (2019) [68], em que os autores propõem um sistema AMT para transcrição de múltiplos instrumentos ao mesmo tempo, também por meio de HMMs.

Em Burlet et al. (2017) [9], os autores propõem um modelo de ANN do tipo *Deep Belief Network*, que são redes gerativas probabilísticas compostas por várias camadas [23], similares a Máquinas Restritas de Boltzman, e realizam um pré-treinamento do modelo para aprender características relevantes no domínio do problema, que depois são utilizadas para realizar a classificação de forma discriminativa. A grande diferença nesse trabalho com relação aos modelos estado-da-arte é a utilização das DBNs para a representação de características do áudio. O trabalho apresenta resultados positivos, mas aponta limitações com relação ao *dataset* utilizado.

Em uma das primeiras abordagens end-to-end baseadas em DNN para AGT, Humphrey et al. (2014) [26] propôs o treinamento de um modelo de reconhecimento de acordes usando tablaturas de guitarra como rótulos e usando a CQT como representação de entrada. Motivado por este e outros trabalhos, Wiggins et al. (2019) [75] explorou o desenvolvimento de modelos de transcrição de guitarra usando o conjunto de dados GuitarSet [77]. No estudo, é utilizado um modelo CNN que é composto por várias camadas convolucionais com CQT como entrada e uma camada Softmax 6-dimensional para a previsão da posição do dedo em cada corda. Esse trabalho se tornou bastante

relevante na área por ser um dos primeiros modelos a propor um modelo de ponta-a-ponta para a realização da tarefa, além de propor métricas para a avaliação de sistemas AGT.

Este estudo aponta a dificuldade da AGT para a notação de tablatura ao invés de notas MIDI. A maioria dos sistemas depende de métodos estatísticos (como modelos de linguagem n-gram) ou algoritmos de busca para gerar tablaturas válidas, dadas as alturas detectadas [9, 56, 62, 72]. No entanto, esses métodos podem produzir arranjos diferentes da música original executada pelo mesmo músico. Wiggins et al. (2019) [75] não abordam esse problema diretamente, contando apenas com a capacidade de um modelo de ponta a ponta para distinguir entre diferentes características timbrais, obtendo cerca de 89% das notas corretamente identificadas com o posicionamento correto dos dedos.

Mais recentemente, destacam-se os trabalhos de Cwitkwitz et al. (2022) [16], Kim et al. (2022) [32] e Huang et al. (2023) [25]. Em Kim et al. (2022) [32], os autores desenvolveram um sistema baseado em atenção para modelar dependências de curto e longo prazo. O modelo é capaz de transcrever a nível de *frame* e nota. Da mesma forma, Cwitkwitz et al. (2022) [16] é focado na estimativa de afinação, mas também transcreve o som no nível da nota por meio de uma cabeça inicial que é integrada ao modelo ponta a ponta proposto. Outras obras que merecem destaque são Sarmento et al. (2021) [61] e Chen et al. (2022) [12]. O primeiro apresenta um novo conjunto de dados chamado DATAGP, um conjunto de dados de músicas do GuitarPro construído para previsão de modelagem de sequência e tarefas relacionadas. Em Chen et al. (2022) [12], os autores também apresentam um novo método para coletar dados e um conjunto de dados chamado EGDB composto por execuções de 240 tablaturas. O trabalho também investiga a transcrição de sons de guitarra para a notação de partitura usando uma arquitetura baseada em Transformers [73].

Kim et al. (2022) [32] propõem uma arquitetura (Figura 3.1) baseada em atenção e convolução que realiza a transcrição em nível de *frame* e de nota simultaneamente. A arquitetura recebe como entrada a informação de *Beats Per Minute* além do CQT do trecho de áudio. Essa informação é útil pois está fortemente relacionada com o *onset* das notas tocadas e portanto, pode ajudar a prever as probabilidades a nível de nota com maior precisão. Segundo os autores, a saída dupla a nível de *frame* e nota também colabora para uma aprendizagem mais robusta, uma vez que o modelo realiza um processo de *multi-task* learning.

Huang et al. (2023) [25] apresentam o objetivo de transcrição a nível de nota de sons reais de solos de guitarra, isto é, trechos de áudio contendo música e vozes de acompanhamento. Essa tarefa é bastante desafiadora, pois além dos desafios presentes na tarefa de AGT, o sistema ainda precisa separar as fontes adequadamente. Outro ponto a destacar desse trabalho é que os autores propõem a detecção de técnicas musicais,

frequentemente ignorada pelos trabalhos na área. A arquitetura proposta pelos autores é baseada na U-Net e também apresenta mecanismos de atenção (Figura 3.2).

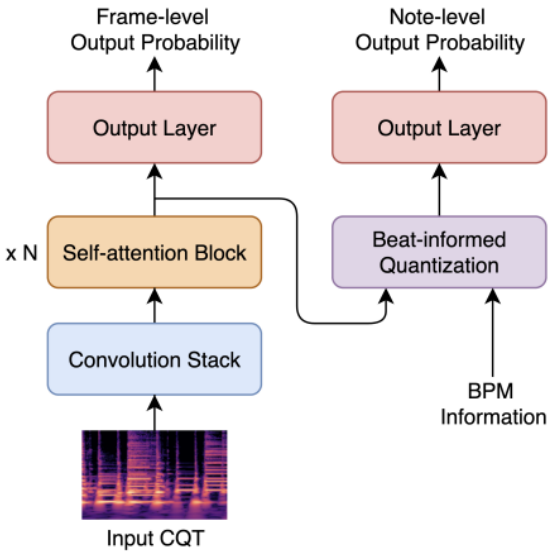


Figura 3.1: Arquitetura proposta em "Note-level automatic guitar transcription using attention mechanism".

Fonte: Extraído de [32].

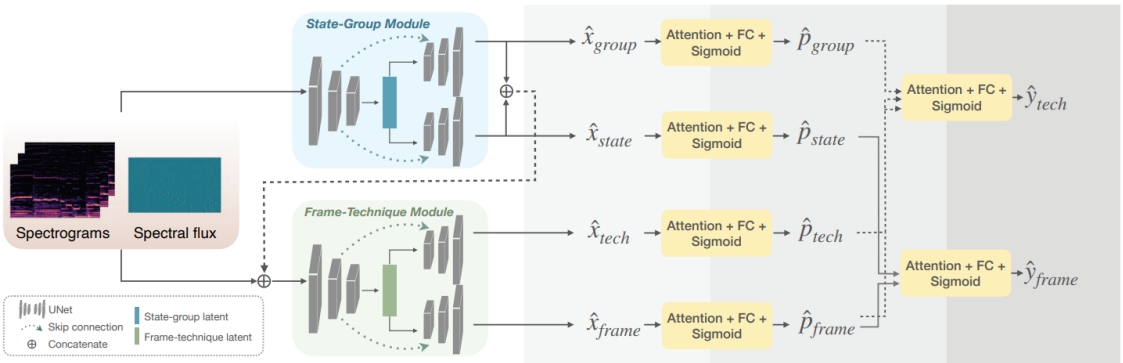


Figura 3.2: Arquitetura proposta em "Note and playing technique transcription of electric guitar solos in real-world music performance".

Fonte: Extraído de [25].

O trabalho Huang et al. (2023) [25] apresenta algumas contribuições notórias: (1) utiliza uma característica denominada Spectral Flux, que é a derivada de primeira ordem do mel-espectrograma e tem a função de fornecer uma característica mais precisa para a detecção de *onsets*; (2) utilizam a arquitetura U-net como base, que segundo os autores tem demostrado resultados promissores em tarefas de MIR e (3) apresentam um novo *dataset* contendo ambas anotações a nível de nota e de técnicas musicais.

3.1 Trabalhos correlatos com o uso da CTC

Apesar da grande relevância desse algoritmo em outras áreas, como a transcrição de fala, existem poucos trabalhos que propõem a utilização do algoritmo CTC para a transcrição musical, e em especial, de guitarra em tablaturas. Existem algumas razões possíveis, tais como a dificuldade em adaptar o uso da CTC para a transcrição multi-rótulo de notas musicais [76] e o foco na utilização de dados alinhados. Ainda que o alinhamento esteja disponível em *datasets* públicos, a investigação de técnicas que dispensam o alinhamento podem significar um avanço na área ao possibilitar o treinamento de modelos com mais dados.

Uma das primeiras tentativas em se utilizar a CTC para transcrição musical foi investigada por Roman et al. (2018) [60], em que os autores propõem o uso da CTC para o treinamento de um modelo de ponta-a-ponta para transcrição diretamente no formato de partituras. Apesar de promissor, o trabalho apresenta duas grandes limitações: primeiro, o modelo é treinado para a tarefa de transcrição de áudios monofônicos, e segundo, o modelo é treinado com áudios sintéticos.

Em se tratando de transcrição musical multi-rótulo, pode-se destacar Weiss et al. (2021) [74], em que os autores propõem o uso do MCTC para a transcrição de notas musicais em diferentes cenários, comparando com o cenário de transcrição com dados alinhados. Os resultados se mostram bastante próximos do cenário com dados alinhados, demonstrando um grande potencial de aplicação. A Figura 3.3 apresenta uma comparação das saídas dos modelos treinados por Weiss et al. (2021) [74]. É possível perceber que o MCTC apresenta uma grande semelhança com o resultado obtido no modelo treinado com alinhamento.

Mais recentemente, Kim et al. (2023) [33] propõem a utilização da CTC para a tarefa de transcrição de guitarra por meio da *tokenização*[1] de representações na saída. Esse tipo de abordagem apresenta uma grande vantagem, pois possibilita o alinhamento de uma saída única. A arquitetura proposta (Figura 3.4) é bastante promissora, pois é baseada em Transformer. Os autores também investigam o aprendizado multi-tarefa, obtendo uma maior performance.

[1] "Tokenização" é um termo utilizado para se referir ao processo de dividir uma sequência em unidades menores chamadas *tokens*.

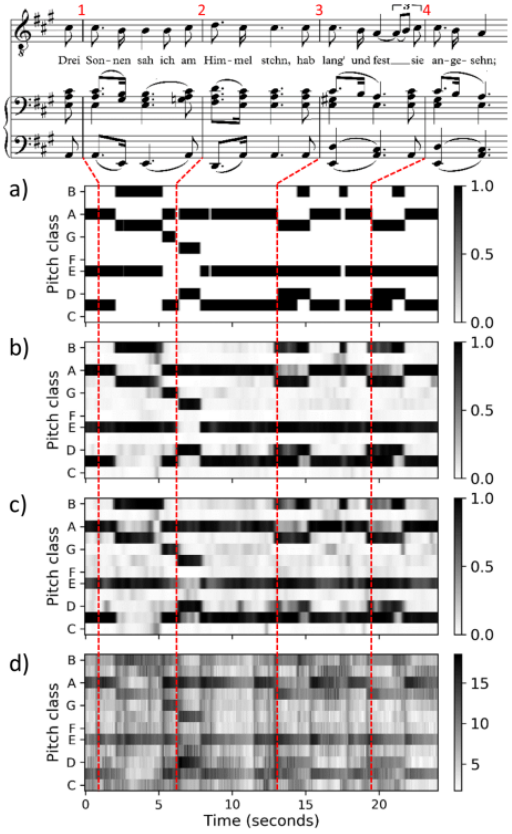


Figura 3.3: Exemplo de saída obtida com MCTC e dados alinhados para transcrição musical: (a) Dados anotados e alinhados. (b) Predições do modelo treinado com alinhamento. (c) Predições modelo treinado com MCTC. (d) Características CQT.

Fonte: Extraído de [74].

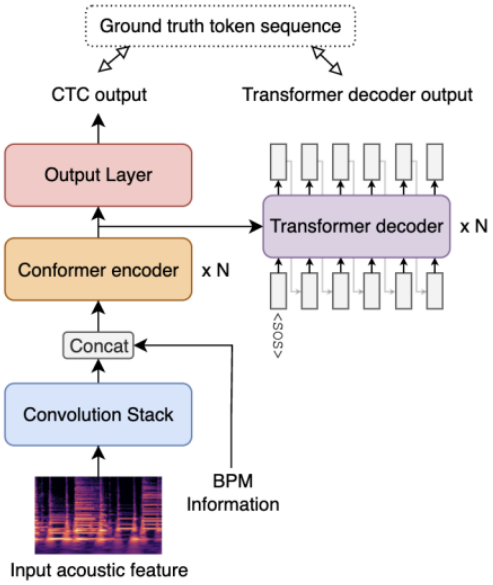


Figura 3.4: Arquitetura proposta em "Sequence-to-Sequence Network Training Methods for Automatic Guitar Transcription With Tokenized Outputs".

Fonte: Extraído de [33].

A metodologia adotada neste estudo difere de Kim et al. (2023) ao explorar o uso da CTC para transcrever simultaneamente os sons de todas as cordas diretamente, além de comparar os resultados com o uso de dados alinhados. Em relação ao estudo de Weiss et al. (2021), a transcrição de tablaturas não foi abordada. A Tabela 3.1 apresenta um resumo dos trabalhos discutidos em comparação com a pesquisa realizada neste trabalho.

Trabalho	**Polifonia**	**Tipos de transcrição**	**Multi-tarefa**	**Arquitetura**
Roman et al. [60]	Não	Transcrição musical	Não	CRNN
Weiss et al. [74]	Sim	Transcrição musical	Não	CNN
Kim et al. [33]	Sim	Tablatura (tokenizado)	Sim	Encoder-decoder (Conformer+Transformer)
Este trabalho	*Sim*	*Notas musicais e tablatura*	*Sim*	*Encoder (CNN+Atenção)*

Tabela 3.1: Comparação entre trabalhos correlatos que utilizam CTC na área de AMT.

CAPÍTULO 4

Proposta de pesquisa

A proposta desta pesquisa consiste na investigação da CTC para a transcrição automática de áudios polifoônicos de guitarra na notação de tablaturas. Para avaliar a viabilidade da CTC nesta tarefa, propõe-se a comparação de dois cenários:

1. Com alinhamento prévio, isto é, o modelo é treinado com o alinhamento conhecido;
2. Sem alinhamento prévio, ou seja, o modelo é treinado por meio da CTC com apenas a sequência de rótulos conhecida.

Para realizar esta comparação, propõe-se uma arquitetura baseada em CNN e a utilização de *datasets* alinhados, desconsiderando o alinhamento para o treinamento dos modelos com CTC. Os parâmetros de pré-processamento de áudio utilizados foram os mesmos em todos os experimentos, por meio da extração de segmentos de 5 segundos para o treinamento dos modelos. As seções seguintes detalham a arquitetura proposta (Seção 4.1), os *datasets* utilizados (Seção 4.2), os experimentos (Seção 4.3) e, por fim, os métodos de avaliação propostos (Seção 4.4).

4.1 Arquitetura

A arquitetura proposta pode ser visualizada na Figura 4.1. Alguns componentes dessa arquitetura foram baseados no trabalho de Wiggins et al. (2019) [75], mas com algumas modificações importantes, principalmente quanto a utilização de atenção e o módulo para a predição de notas. A arquitetura consiste na utilização de uma camada de extração de características, que converte o áudio bruto em espectrogramas do tipo CQT utilizando o *framework* nnAudio [13] com os parâmetros de 84 bins, tamanho de passo 512 e 12 *bins* por oitava. A característica CQT foi escolhida por ser uma representação mais compacta e que contém uma aproximação das notas musicais presentes no áudio. A extração do CQT é seguida por 6 camadas convolucionais de 64 filtros cada de tamanho 3x3 e com normalização de *batch*. Não há camadas de *pooling*. Após as camadas convolucionais, uma nova camada convolucional é adicionada (módulo *AggregatorStrings*), que tem o objetivo de agregar as características (64 mapas) em 6 novos mapas correspondentes as cordas

do instrumento. Após o agregamento das características, o modelo contém mais duas camadas totalmente conectadas. Por fim, a arquitetura contém um módulo para a predição da tablatura contendo 6 camadas lineares sem *bias* paralelas com 24 neurônios de saída (um pra cada posição do instrumento). Todas as camadas ocultas do modelo utilizam a função de ativação Mish, que durante testes preliminares, demonstrou uma performance ligeiramente superior em comparação com a ReLU.

A arquitetura proposta apresenta duas variações: a primeira, com o uso de 6 cabeças de atenção após as camadas totalmente conectadas, e a segunda, com a adição de um módulo para a predição de notas musicais. Na primeira variação, o objetivo está em facilitar o aprendizado de relações entre as características de outras cordas, permitindo que o modelo seja capaz de discernir os sons parecidos com maior robustez ou ainda aprendendo relações de sequências entre notas musicais. Na segunda variação, o modelo apresenta um aprendizado multi-tarefa de predição de notas musicais (48 no total) além da tablatura. A predição de notas musicais é feita com o uso da função de custo MCTC, no caso dos experimentos sem alinhamento, ou do uso da entropia binária cruzada para a classificação multi-rótulo contendo o alinhamento. Adicionalmente, as notas preditas são fornecidas para as camadas lineares de predição da tablatura, com a finalidade de diminuir o erro de predição de notas musicais. A programação do erro do custo da predição de tablaturas para os módulos de predição de notas também é opcional.

A utilização de 6 camadas lineares paralelas para a predição de tablatura apresenta um desafio, que é a suposição de que não há relação entre elas, o que não é necessariamente verdade. Por exemplo, quando um músico toca um acorde, há uma clara relação entre as notas. Essa relação não é explorada pela arquitetura atual, porém espera-se que o modelo seja capaz de classificar os sons de cada corda de forma independente. Apesar dessa limitação, a escolha de 6 camadas paralelas têm a função de permitir que a mesma nota com som similar possa ser predita em várias cordas ao mesmo tempo, o que não pode ser modelado usando apenas uma saída, como na predição de notas. A predição de notas tem a finalidade de facilitar o aprendizado do modelo. Apesar da predição de notas ser uma tarefa bastante relevante no contexto de transcrição musical, a predição de tablaturas necessitaria de um processo de busca adicional para encontrar a melhor combinação de notas tocadas [58]. Como o objetivo do trabalho é a predição de tablaturas diretamente, a utilização das 6 camadas paralelas se faz necessária.

Apesar da CTC ser comumente utilizado para a predição de sequência de caracteres, o alinhamento do modelo pode ser utilizado para definir o início de cada classe predita, possibilitando assim a geração da tablatura. Adicionalmente, o alinhamento gerado fornece uma predição razoável da duração das notas, que apesar de não ser crucial para a tarefa, fornece uma informação adicional para a geração de tablaturas visualmente agradáveis.

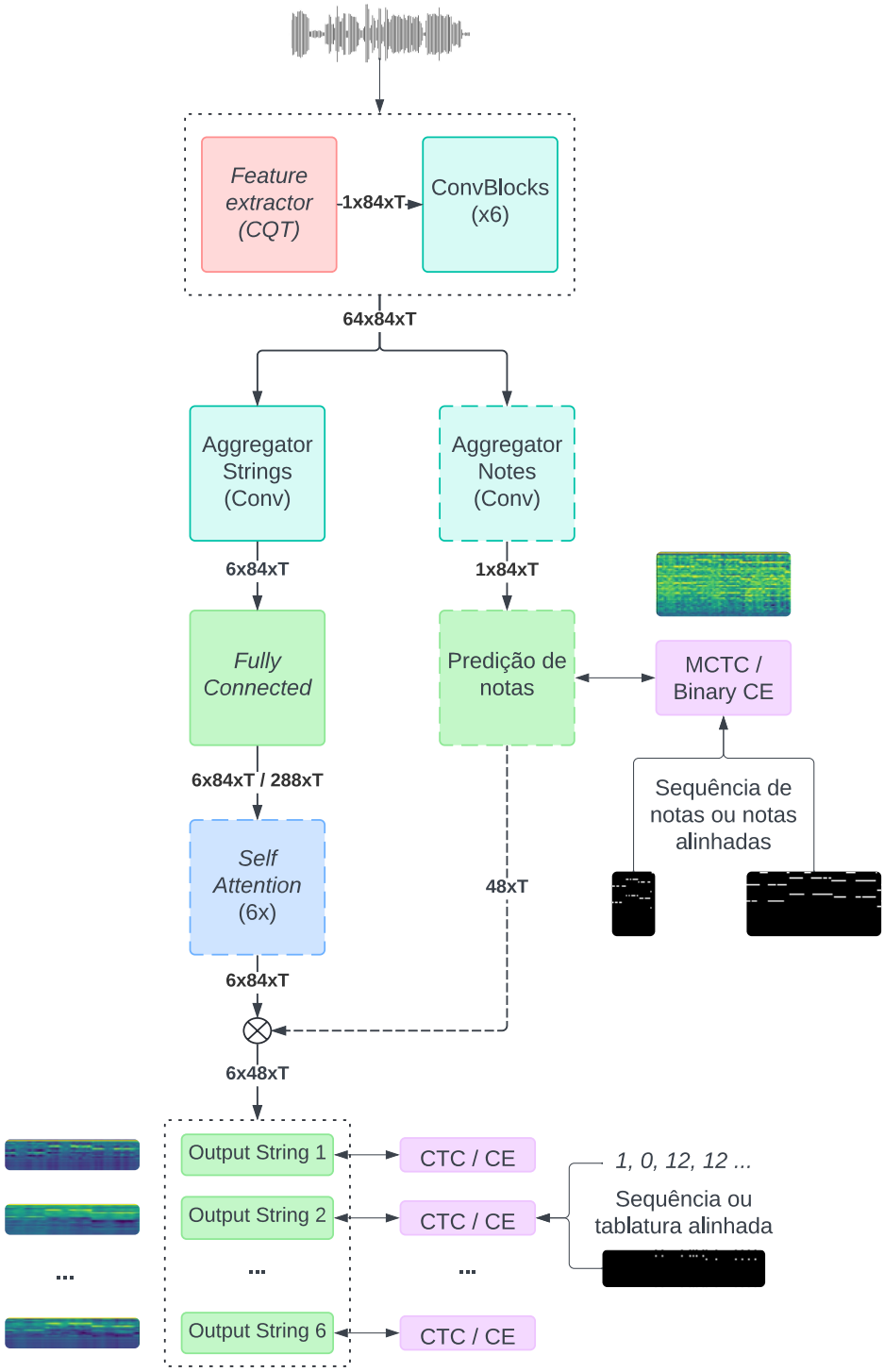


Figura 4.1: Arquitetura proposta. Na imagem, o CQT com 84 *bins* é extraído do áudio bruto (em vermelho) e em seguida processado por 6 blocos de convolução 2D (em ciano), produzindo um mapa de características que é fornecido ao(s) agregador(es) de características, também convolucional. A predição de notas e o *self-attention* (em azul) são componentes opcionais, assim como a propagação do erro da predição de notas (linha tracejada). As saídas e camadas totalmente conectadas estão em verde, e as funções de custo, em roxo.

4.2 *Datasets*

Dentre os *datasets* disponíveis, dois foram selecionados para a realização do trabalho devido a sua relevância: o GuitarSet [77], *dataset* de referência da área do AGT, e o SynthTab [78], *dataset* contendo muitas horas de áudios sintetizados.

4.2.1 GuitarSet

O GuitarSet[1] é um *dataset* construído principalmente para a criação e elaboração de sistemas de transcrição automática para guitarras. O *dataset* é cuidadosamente anotado, apresentando cerca de 3 horas de áudio no total. O GuitarSet é de código aberto e pode ser utilizado para pesquisas na área da transcrição musical para guitarras e também no estudo de outras áreas como segmentação harmônica e estimação de acordes [77].

O GuitarSet apresenta áudios de gravações reais feitas por 6 guitarristas profissionais (30 gravações por guitarrista). Cada corda do instrumento apresenta uma notação distinta, o que pode facilitar o desenvolvimento de sistemas de transcrição. Os músicos foram instruídos a tocar vários estilos e tempos diferentes, possibilitando uma variação de grande de performances. As anotações estão no formato JAMS [27], que são arquivos JSON contendo informações específicas do áudio, como tempo, tom e estilo, batidas musicais, acordes tocados e notas transcritas (incluindo a posição da corda e do traste) e o contorno da entonação para cada nota (*pitch*). A Figura 4.2 mostra graficamente as anotações do *dataset*.

Os áudios do *dataset* estão separados em dois diretórios principais, um para as gravações realizadas com captadores e outra para as gravações realizadas com microfone. A utilização depende do propósito do problema. O primeiro apresenta maior qualidade, mas o segundo é mais fiel a utilização em ambientes reais e por isso é mais adequado para a realização dos experimentos. Adicionalmente, há uma divisão em relação a polifonia das execuções, que são as execuções das performances *solo*, contendo menos polifonia, e *comp*, contendo mais acordes e polifonia. Essa divisão pode possibilitar uma análise mais detalhada do modelo em conseguir fazer a predição de notas isoladas ou várias notas sendo tocada ao mesmo tempo.

Os áudios do GuitarSet não apresentam aplicação de efeitos de guitarra, o que significa que o áudio está em sua forma mais pura e fiel ao som produzido diretamente pelas cordas, contudo, os áudios foram gravados a partir de guitarras acústicas (violões) de cordas de aço, o que pode significar uma limitação na capacidade de generalização para timbres diferentes.

[1] https://guitarset.weebly.com/

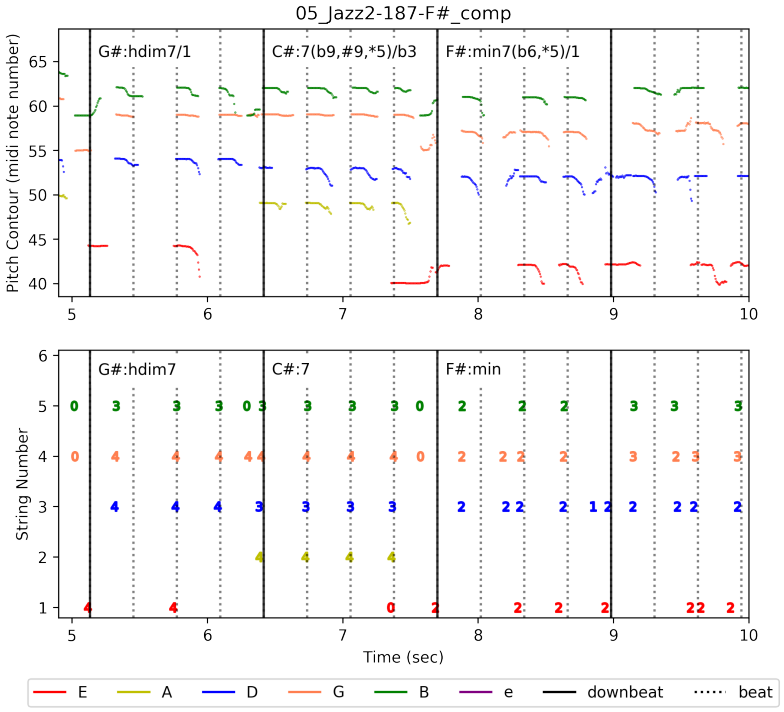


Figura 4.2: Anotação do *dataset* GuitarSet: cada corda está representada por uma cor diferente, a notação também mostra as batidas musicais no tempo (linhas verticais nos gráficos). O gráfico de cima mostra o respectivo valor da nota sendo executada no tempo em números MIDI, nesse caso, é possível perceber a nota não apresenta um valor constante no tempo. O gráfico de baixo mostra a respectiva posição da nota sendo executada no braço.

Fonte: Extraído de [77].

4.2.2 SynthTab

Outro *dataset* para a tarefa de transcrição é o SynthTab [78]. O conjunto de dados apresenta um total de 6.700 horas de áudios gerados sinteticamente em 6 timbres diferentes. Considerando que o tamanho do *dataset* é bastante elevado, optou-se pela utilização do conjunto acústico, que abrange um total de 3.200 horas de áudio e 10.650 músicas, sendo uma boa escolha para a realização de pré-treinamento de modelos.

O *dataset* SynthTab foi gerado por meio da converção de anotações do DadaGP [61] em áudios por meio de instrumentos virtuais. O processo de síntese inclui a aplicação de técnicas de modulação de *pitch*, como *bends* e *vibrato*, além de outros elementos expressivos, como *slides* e *palm-muting*, com o objetivo de simular músicas reais. Diferentes variações timbrais, estilos de execução e configurações de captação (ponte, meio e braço) foram gerados. Cada faixa foi renderizada com qualidade de 24 bits e uma taxa de amostragem de 44.100 Hz.

Os experimentos apresentados por Zang et al. (2024) [78] demonstram que o pré-treinamento em SynthTab pode melhorar significativamente a performance dos

modelos em cenários de *cross-dataset*, mitigando problemas de *overfitting* e aumentando a capacidade de generalização para outros *datasets* contendo gravações de guitarras reais.

4.2.3 Pré-processamento e divisão dos dados

Todos os áudios foram segmentados em trechos de 5 segundos e re-amostrados para 24 kHz para a realização dos experimentos com a finalidade de economizar recursos computacionais, porém mantendo um tempo e qualidade adequados para o aprendizado dos modelos. Para o *dataset* GuitarSet, foi utilizado um passo de 2,5 segundos para a segmentação, produzindo segmentos com sobreposição. Para o *dataset* SynthTab apenas uma segmentação simples foi utilizada, uma vez que a quantidade de segmentos desse conjunto de dados já é bastante elevada.

Em relação a divisão dos dados, optou-se pela divisão do GuitarSet em treino, validação e teste levando-se em consideração o guitarrista que realizou a gravação do áudio. Nesse sentido, os guitarristas 0 e 1 foram selecionados para teste e validação, respectivamente, ao passo que os demais foram destinados para treinamento. Para o SynthTab, utilizou-se todo o conjunto acústico para treinamento e o conjunto Dev[2], um subconjunto contendo algumas músicas sintetizadas em diferentes timbres, para validação. A Tabela 4.1 apresenta o total de segmentos de cada conjunto gerado.

Conjunto	**Dataset**	**Quantidade de segmentos**
Treinamento	GuitarSet	18305
	SynthTab	2029630
Validação	GuitarSet	635
	SynthTab	9469
Teste	GuitarSet	654

Tabela 4.1: Divisão dos dados de treino, validação e teste.

4.3 Experimentos

Os experimentos propostos são apresentados em detalhes na Tabela 4.2. Os modelos treinados com entropia cruzada como função de custo utilizando dados alinhados iniciam com a letra ‘A’ e os experimentos treinados sem alinhamento apresentam o prefixo “CTC”. Todos os modelos foram treinados com o módulo de atenção, exceto os experimentos com o sufixo “nA”. Adicionalmente, foram treinados modelos com a tarefa de predição de notas além da tarefa de transcrição de tablaturas. Por fim, apenas um

[2] https://synthtab.dev/

experimento com aumento de dados foi realizado[3], o experimento "CTC-AUG", treinado com o *dataset* SynthTab. Esse modelo foi utilizado como base para os experimentos de ajuste fino (sufixo "FT").

Experimento	**Descrição**
A (*baseline*)	Modelo treinado com alinhamento
A-FT	Ajuste fino com alinhamento
A-nA	Modelo treinado com alinhamento e sem atenção
A-PE	Modelo treinado com alinhamento e codificação posicional
A-NOTES	Modelo treinado com alinhamento com predição notas
A-NOTESG	Modelo treinado com alinhamento e predição notas com propagação do erro das notas
A-NOTESG-PE	Modelo treinado com alinhamento e codificação posicional e predição notas com propagação do erro das notas
CTC	Modelo treinado com CTC
CTC-AUG	Modelo treinado com CTC e aumento de dados (SynthTab)
CTC-FT	Ajuste fino com CTC
CTC-nA	Modelo treinado com CTC e sem atenção
CTC-PE	Modelo treinado com CTC e codificação posicional
CTC-NOTES	Modelo treinado com CTC com predição notas
CTC-NOTESG	Modelo treinado com CTC e predição notas com propagação do erro das notas
CTC-NOTESG-PE	Modelo treinado com CTC e codificação posicional e predição notas com propagação do erro das notas

Tabela 4.2: Experimentos propostos.

Em relação aos hiper-parâmetros de treinamento, todos os modelos foram treinados utilizando o otimizador Adam [34] com taxa de aprendizado de 10^{-4} com *warm-up* de 1000 passos seguido por dois decaimentos automáticos de 10^{-1} após 10.000 e 100.000 passos, respectivamente. Para os modelos treinados com duas funções de custo (predição de tablatura e notas), o custo final foi computado como na Equação 4-1, sendo α e β parâmetros de peso de cada custo. Para os experimentos com alinhamento $\alpha = \beta = 0,5$, para os experimentos sem alinhamento, foram usados $\alpha = 0,9$ e $\beta = 0,1$, para equilibrar o custo obtido pela função MCTC[4]. Na Equação 4-1, C_{tab} é definido como a média do somatório dos custos obtidos em cada saída do módulo de predição de tablaturas (Equação 4-2).

$$C = \alpha C_{tab} + \beta C_{notes} \tag{4-1}$$

[3]Não foram obtidos resultados relevantes com aumento de dados no treinamento com o *GuitarSet*. O aumento de dados foi necessário para o *SynthTab* devido a limitação de timbre do *dataset*, que foi gerado sinteticamente.

[4]O MCTC gera valores de custo muito maiores em comparação ao custo CTC, por isso foi necessário realizar esse ajuste.

$$C_{tab} = \frac{1}{6} \sum_{s=1}^{S=6} C_s \tag{4-2}$$

Os modelos foram treinados por um máximo de 10.000 épocas com um tamanho de *batch* igual a 92[5]. Com relação a regularização, os modelos foram treinados com Dropout de 0,25 aplicado em todas as camadas ocultas. Também foi aplicado SpecAugment na entrada do modelo, mascarando no máximo 5% da entrada[6]. Para a realização dos testes, os melhores *checkpoints* foram selecionados com base no menor custo obtido na validação. Todos os experimentos foram realizados em uma NVIDIA TESLA V100 com 32GB de VRAM, com duração aproximada de 3 dias por experimento[7].

Em relação as técnicas de aumento de dados aplicadas ao experimento "CTC-AUG" tem-se a utilização de dois grupos de técnicas: a utilização de perturbações no áudio com a biblioteca `audiomentations`[8] com probabilidade de 25%, e a simulação de efeitos de guitarra com a biblioteca `pedalboard`[9], também com probabilidade de 25%, ambos aplicados durante o carregamento dos segmentos na fase de treinamento.

4.4 Avaliação

Para a avaliação dos resultados propõe-se a utilização de métricas comuns em problemas de classificação, em especial, de transcrição musical, tais como a revocação (*recall*), precisão (*precision*) e medida F. Para realizar as avaliações, utiliza-se a biblioteca `mir_eval` [57], bastante utilizada para a avaliação de diferentes tarefas de MIR, como a transcrição musical.

Basicamente, há duas formas de realizar a avaliação da transcrição musical: considerando o *onset* (momento que a nota começa) e o *offset* (momento que a nota termina), ou apenas considerando o *onset*. Em transcrições mais precisas, por exemplo, para a geração de partituras, a primeira forma é mais recomendada, no entanto, para a geração de tablaturas, em que a duração das notas não é um aspecto crucial, a segunda forma é suficiente. Neste trabalho serão apresentados as duas métricas em comparação para a tarefa de transcrição de notas. Para a tarefa de transcrição da tablatura, apenas o resultado com *onset* será discutido.

[5] A utilização do tamanho de *batch* igual a 92 possibilitou o treinamento de dois modelos em paralelo em uma mesma GPU, mas é possível executar os experimentos com *batches* menores.

[6] Não foram investigados outros parâmetros de mascaramento.

[7] Apesar do *hardware* robusto, é possível executar os experimentos com *batches* menores e menos tempo de processamento, uma vez que a convergência dos modelos ocorre rapidamente, como pode ser observado no Apêndice A.

[8] https://github.com/iver56/audiomentations

[9] https://github.com/spotify/pedalboard

Adicionalmente, por se tratar de um problema similar a transcrição de fala, propõem-se a utilização de uma nova métrica de avaliação. Em ASRs, há duas métricas comumente utilizadas para avaliar a taxa de erro de palavras e caracteres em uma sentença: *Word Error Rate* (WER) e *Character Error Rate* (CER), respectivamente. Na Equação 4-3 a fórmula geral do cálculo dessas medidas é apresentada, em que S é o número de ocorrências substituídas, I é o número de ocorrências inseridas, D é o número de ocorrências deletadas e N é o total de ocorrências da sentença original. Para o WER, considera-se ocorrência como uma palavra inteira, para o CER, considera-se apenas o caractere. Quanto menor o erro, melhor o desempenho do modelo.

$$\text{WER/CER} = \frac{S + I + D}{N} \tag{4-3}$$

Uma vez que a saída do módulo de predição de tablaturas também pode ser transformado em uma sequência de caracteres, torna-se trivial o cálculo do CER para a verificação da taxa de erros de predição de posições preditas pelo modelo. Neste trabalho, a métrica CER será chamada de *Fret Error Rate* (FER).

O FER apresenta uma visão independente dos resultados das cordas, mas pode ser interessante analisar o desempenho do modelo em cada momento que existe uma nota tocada. Para considerar o aspecto dependente das predições, por exemplo, durante a execução de um acorde, sugere-se o TER, a variação do FER considerando todas as notas em um dado *timestep*.

Na prática, as métricas FER e TER são análogas ao CER e ao WER dos problemas de ASR, com a distinção de que o modelo faz apenas a predição de caracteres isolados. A "palavra" neste caso, é a concatenação das saídas em um determinado *timestep* (Figura 4.3). Os cálculos podem ser realizados de forma similar utilizando bibliotecas comuns. A grande vantagem da utilização desse tipo de métrica é o fato de não ser necessário ter uma tablatura alinhada para realizar a avaliação, o que pode ser necessário caso o método CTC seja utilizado para treinar modelos com dados não alinhados. Nos experimentos propostos, a tablatura alinhada está disponível, mas esse pode não ser o caso em outras situações.

```
|--1--------- ... --|
|--1--------- ... --|
|-----3------ ... --|
|------------ ... --|
|---------4-- ... --|
|------------ ... -5|

Palavras geradas: 11---- --3--- ----4- ... -----5
```

Figura 4.3: Exemplo de como a tablatura pode ser adaptada para uma sequência de caracteres e palavras.

Para realizar as avaliações, todos os segmentos de cada áudio foram inferidos e segmentados para gerar um único resultado. Posteriormente, a média dos resultados dos áudios foi realizada. Os resultados são apresentados no Capítulo 5.

CAPÍTULO 5

Resultados

Neste capítulo são apresentados os resultados obtidos para os experimentos propostos. A Seção 5.1 discute os resultados obtidos para a tarefa de transcrição de tablaturas, a Seção 5.2 apresenta os resultados para a tarefa de predição de notas. Por fim, a Seção 5.3 apresenta alguns exemplos de alinhamento obtidos. Os *logs* de treinamento podem ser visualizados no Apêndice A.

5.1 Transcrição de tablatura

Os resultados obtidos para a avaliação da transcrição utilizando a biblioteca `mir_eval` são apresentados nas tabelas 5.1, 5.2 e 5.3. De modo geral, o desempenho dos experimentos executados com alinhamento foram superiores aos modelos treinados sem alinhamento. Isso é esperado, uma vez a tarefa de classificação sem alinhamento representa um desafio adicional durante o treinamento. O melhor resultado obtido em todo o conjunto de teste (Tabela 5.1) para os experimentos sem alinhamento foi de 0,336 de F1-score com o experimento "CTC-NOTESG-PE", apresentando uma melhoria relativa de 14,67% no F1-score em comparação com o experimento *baseline* "A" utilizando dados alinhados. Em comparação com o melhor modelo treinado com alinhamento, "A-NOTESG-PE", o modelo apresentou uma queda relativa de apenas 14,28%.

Os resultados dos experimentos com predição de notas sugerem que o aprendizado multi-tarefa foi essencial para prover informação adicional ao módulo de predição de tablaturas, diminuindo o erro da predição e aumentando a performance geral do modelo. Em relação aos experimentos com ajuste fino, ambos os modelos "A-FT" e "CTC-FT" ajustados a partir do modelo "CTC-AUG" obtiveram desempenhos insatisfatórios na avaliação no GuitarSet. Apesar disso, o ajuste fino possibilitou uma melhoria considerável no desempenho, provavelmente devido a capacidade do modelo em detectar características e sons distintos.

Os experimentos com atenção sem codificação posicional também tiveram um desempenho inferior aos experimentos sem o uso de atenção, porém, ao utilizar a codificação posicional, o desempenho dos modelos apresentou melhorias. Esse resultado

ressalta a importância da informação de tempo nas sequências de entrada do módulo de atenção.

Experimento	**P**	**R**	**F**
A (*baseline*)	0,493 ± 0,203	0,235 ± 0,142	0,293 ± 0,157
A-FT	0,485 ± 0,166	0,309 ± 0,145	0,347 ± 0,144
A-nA	0,416 ± 0,166	0,320 ± 0,165	0,330 ± 0,161
A-PE	*0,488 ± 0,181*	0,328 ± 0,163	0,362 ± 0,156
A-NOTES	0,463 ± 0,176	0,293 ± 0,138	0,335 ± 0,148
A-NOTESG	0,483 ± 0,187	0,336 ± 0,152	0,369 ± 0,156
A-NOTESG-PE	**0,452 ± 0,148**	***0,381 ± 0,156***	***0,384 ± 0,140***
CTC	*0,438* ± 0,332	0,091 ± 0,098	0,136 ± 0,133
CTC-AUG	0,003 ± 0,021	0,000 ± 0,003	0,001 ± 0,006
CTC-FT	0,440 ± 0,213	0,193 ± 0,149	0,242 ± 0,160
CTC-nA	0,388 ± 0,270	0,094 ± 0,085	0,138 ± 0,114
CTC-PE	0,347 ± 0,293	0,067 ± 0,069	0,104 ± 0,101
CTC-NOTES	0,434 ± 0,200	0,217 ± 0,121	0,262 ± 0,139
CTC-NOTESG	0,430 ± 0,200	0,201 ± 0,106	0,248 ± 0,118
CTC-NOTESG-PE	**0,369 ± 0,160**	***0,376 ± 0,155***	***0,336 ± 0,150***

Tabela 5.1: Resultados de precisão (P), revocação (R) e medida F dos experimentos no conjunto de teste do GuitarSet. Em negrito, o melhor resultado de cada grupo de experimento em termos de medida-F. Em itálico, os melhores resultados em cada métrica.

Também nota-se o aumento de performance dos modelos treinados com predição de notas e propagação de erros. De alguma forma, a propagação do erro das notas preditas para o módulo de predição de tablaturas contribuiu para um melhor desempenho.

Outro ponto a ser notado é a tendência dos modelos em obter uma maior precisão e uma menor revocação. Esse comportamento pode ser observado em todos os experimentos, sugerindo uma presença considerável de falsos negativos na transcrição (notas erradas), possivelmente devido a complexidade das saídas paralelas do módulo de predição de tablaturas.

Comparando os resultados dos experimentos nos subconjuntos *solo* (Tabela 5.3) e *comp* (Tabela 5.2), observa-se que o desempenho dos modelos para a predição de sons com maior polifonia se torna ainda mais desafiador, o que é esperado, uma vez que mais notas executadas ao mesmo tempo dificultam a predição correta da tablatura. O mesmo efeito pode ser observado na Tabela 5.4, em que o erro do subconjunto *solo* é consideravelmente menor que o erro obtido no conjunto *comp*. Em geral, tanto os resultados dos subcojuntos *solo* quanto *comp* apresentaram comportamentos bastante similares, sendo os melhores resultados obtidos a partir dos modelos treinados para a tarefa de predição de notas com propagação dos erros.

Experimento	**P**	**R**	**F**
A (*baseline*)	0,378 ± 0,163	0,150 ± 0,104	0,198 ± 0,119
A-FT	0,389 ± 0,131	0,227 ± 0,087	0,266 ± 0,095
A-nA	*0,339 ± 0,153*	0,233 ± 0,128	0,253 ± 0,131
A-PE	*0,389 ± 0,134*	0,233 ± 0,110	0,274 ± 0,114
A-NOTES	0,389 ± 0,146	0,222 ± 0,110	0,263 ± 0,121
A-NOTESG	0,373 ± 0,149	0,246 ± 0,113	0,277 ± 0,117
A-NOTESG-PE	**0,382 ± 0,138**	***0,281 ± 0,106***	***0,301 ± 0,105***
CTC	0,287 ± 0,302	0,034 ± 0,044	0,057 ± 0,068
CTC-AUG	0,000 ± 0,000	0,000 ± 0,000	0,000 ± 0,000
CTC-FT	*0,339 ± 0,175*	0,123 ± 0,090	0,163 ± 0,108
CTC-nA	0,240 ± 0,210	0,041 ± 0,040	0,067 ± 0,060
CTC-PE	0,255 ± 0,234	0,054 ± 0,060	0,083 ± 0,088
CTC-NOTES	0,314 ± 0,138	0,168 ± 0,098	0,203 ± 0,104
CTC-NOTESG	0,291 ± 0,112	0,156 ± 0,086	0,186 ± 0,088
CTC-NOTESG-PE	**0,267 ± 0,099**	***0,295 ± 0,110***	***0,244 ± 0,092***

Tabela 5.2: Resultados de precisão (P), revocação (R) e medida F dos experimentos no subconjunto *comp* do conjunto de teste do GuitarSet. Em negrito, o melhor resultado de cada grupo de experimento em termos de medida-F. Em itálico, os melhores resultados em cada métrica.

Experimento	**P**	**R**	**F**
A (*baseline*)	*0,626 ± 0,170*	0,330 ± 0,136	0,394 ± 0,143
A-FT	0,589 ± 0,144	0,404 ± 0,135	0,445 ± 0,126
A-nA	0,509 ± 0,145	0,437 ± 0,159	0,432 ± 0,152
A-PE	0,574 ± 0,155	0,420 ± 0,135	0,447 ± 0,130
A-NOTES	0,576 ± 0,164	0,395 ± 0,132	0,435 ± 0,138
A-NOTESG	0,578 ± 0,161	0,420 ± 0,131	0,453 ± 0,137
A-NOTESG-PE	**0,516 ± 0,130**	***0,482 ± 0,136***	***0,458 ± 0,126***
CTC	0,575 ± 0,314	0,155 ± 0,138	0,218 ± 0,174
CTC-AUG	0,000 ± 0,000	0,000 ± 0,000	0,000 ± 0,000
CTC-FT	0,560 ± 0,187	0,264 ± 0,158	0,317 ± 0,159
CTC-nA	*0,579 ± 0,244*	0,158 ± 0,090	0,227 ± 0,114
CTC-PE	0,428 ± 0,331	0,102 ± 0,125	0,146 ± 0,148
CTC-NOTES	0,528 ± 0,200	0,267 ± 0,114	0,325 ± 0,132
CTC-NOTESG	0,508 ± 0,198	0,220 ± 0,105	0,286 ± 0,127
CTC-NOTESG-PE	**0,483 ± 0,144**	***0,445 ± 0,143***	***0,426 ± 0,149***

Tabela 5.3: Resultados de precisão (P), revocação (R) e medida F dos experimentos no subconjunto *solo* do conjunto de teste do GuitarSet. Em negrito, o melhor resultado de cada grupo de experimento em termos de medida-F. Em itálico, os melhores resultados em cada métrica.

Os resultados dos testes para as métricas TER e FER são apresentados na Tabela 5.4. Neste caso, quanto menor o valor, melhor. O comportamento dos testes segue o mesmo padrão obtido nas análises anteriores, com a vantagem de que essas métricas podem ser utilizadas mesmo em conjuntos de dados sem alinhamento. Como esperado, o resultado da métrica FER também é ligeiramente menor em comparação com a métrica TER, uma vez que o TER considera correto apenas se todas as notas em um instante específico forem preditas corretamente. Os modelos treinados sem alinhamento não apresentaram resultados muito satisfatórios na métrica TER, indicando uma grande limitação do módulo de predição de tablaturas em lidar com polifonia. Apenas os modelos treinados com MCTC obtiveram resultados mais próximos dos modelos treinados com alinhamento.

Experimento	**Guitaset Test**		**Guitaset Solo**		**Guitaset Comp**	
	TER	**FER**	**TER**	**FER**	**TER**	**FER**
A (*baseline*)	0,596 ± 0,190	0,480 ± 0,206	0,502 ± 0,199	0,416 ± 0,210	0,682 ± 0,132	0,537 ± 0,183
A-FT	0,532 ± 0,179	0,374 ± 0,183	0,467 ± 0,204	0,345 ± 0,210	0,610 ± 0,118	0,416 ± 0,156
A-nA	0,526 ± 0,168	0,351 ± 0,160	0,429 ± 0,151	0,272 ± 0,137	0,619 ± 0,111	0,421 ± 0,129
A-PE	0,523 ± 0,169	0,358 ± 0,169	0,438 ± 0,168	0,306 ± 0,172	0,603 ± 0,111	0,397 ± 0,142
A-NOTES	0,534 ± 0,168	0,384 ± 0,163	0,447 ± 0,174	0,342 ± 0,168	0,618 ± 0,119	0,428 ± 0,142
A-NOTESG	0,517 ± 0,168	0,363 ± 0,150	0,432 ± 0,166	0,320 ± 0,161	0,608 ± 0,119	0,415 ± 0,136
A-NOTESG-PE	***0,496 ± 0,159***	***0,314 ± 0,144***	***0,415 ± 0,158***	***0,266 ± 0,145***	***0,575 ± 0,116***	***0,360 ± 0,129***
CTC	0,840 ± 0,138	0,726 ± 0,253	0,774 ± 0,149	0,638 ± 0,262	0,898 ± 0,094	0,801 ± 0,215
CTC-AUG	0,956 ± 0,034	0,950 ± 0,062	0,959 ± 0,022	0,963 ± 0,030	0,952 ± 0,043	0,933 ± 0,082
CTC-FT	0,756 ± 0,149	0,507 ± 0,241	0,702 ± 0,166	0,487 ± 0,273	0,803 ± 0,120	0,510 ± 0,223
CTC-nA	0,800 ± 0,147	0,687 ± 0,226	0,735 ± 0,156	0,620 ± 0,238	0,866 ± 0,095	0,755 ± 0,188
CTC-PE	0,928 ± 0,225	0,716 ± 0,288	0,890 ± 0,196	0,738 ± 0,292	0,966 ± 0,248	0,690 ± 0,285
CTC-NOTES	*0,656 ± 0,152*	0,474 ± 0,205	0,593 ± 0,159	0,446 ± 0,208	*0,719 ± 0,102*	0,505 ± 0,178
CTC-NOTESG	0,679 ± 0,151	*0,366 ± 0,182*	0,629 ± 0,171	0,509 ± 0,229	0,721 ± 0,103	0,505 ± 0,190
CTC-NOTESG-PE	**0,694 ± 0,204**	**0,367 ± 0,183**	***0,592 ± 0,189***	***0,320 ± 0,203***	**0,803 ± 0,182**	***0,415 ± 0,170***

Tabela 5.4: Resultados de TER e FER dos experimentos no GuitarSet. Em negrito, o melhor resultado de cada grupo de experimento em termos de medida-F. Em itálico, os melhores resultados em cada métrica.

Por fim, a Figura 5.1 apresenta um exemplo de geração de tablaturas a partir dos melhores modelos obtidos nos experimentos. É possível observar que a qualidade da transcrição dos modelos “A-NOTESG-PE” e “CTC-NOTESG-PE” estão bem próximas, porém ainda há desafios para obter uma qualidade aceitável que viabilize a utilização desse sistema em cenários reais.

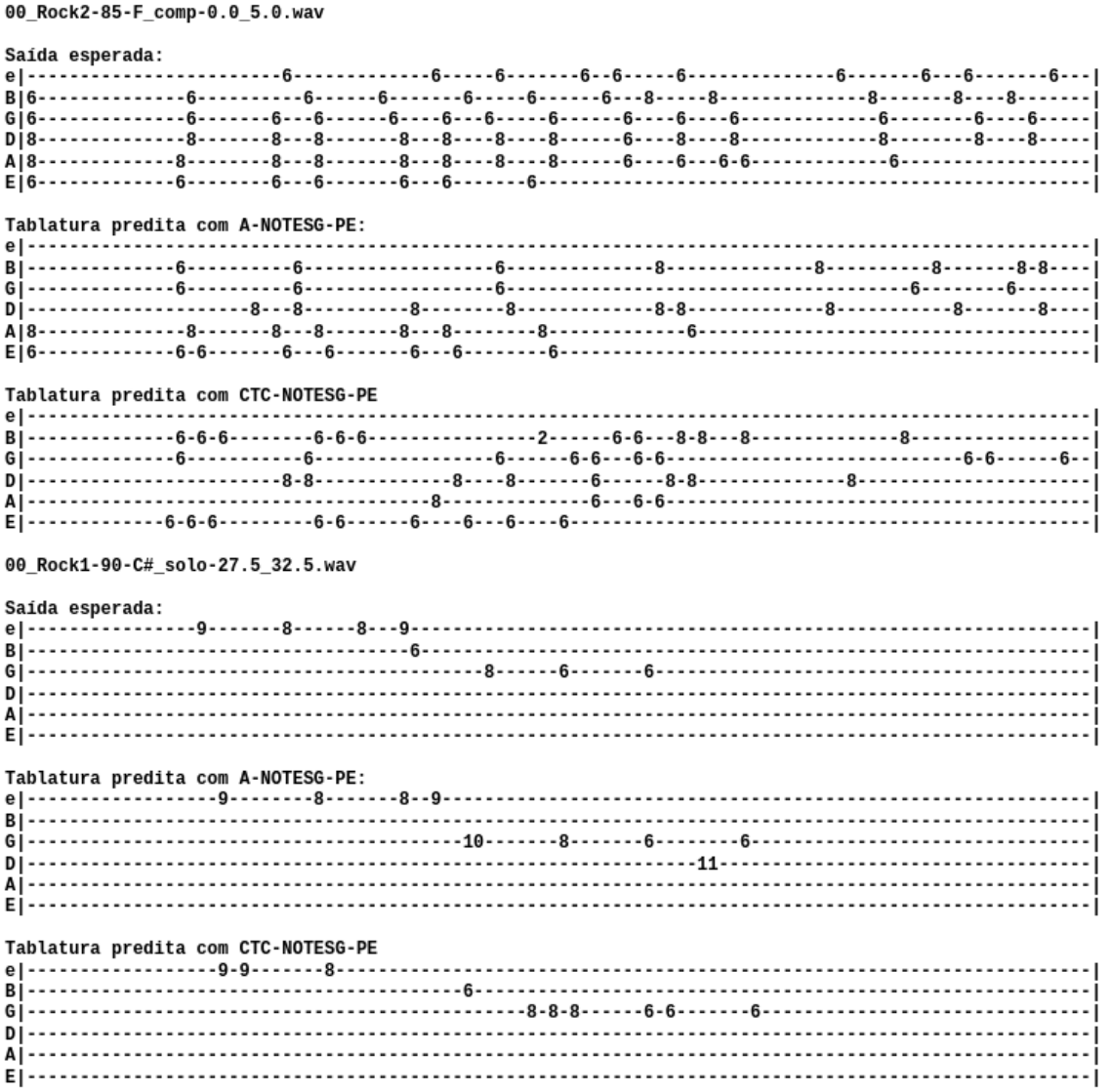


Figura 5.1: Exemplos de tablaturas geradas. Na parte superior, o segmento de áudio `00_Rock2-85-F_comp-0.0_5.0.wav` apresenta mais polifonia que o trecho de áudio `00_Rock1-90-C#_solo-27.5_32.5.wav`.

5.2 Transcrição de notas

Os resultados para a tarefa de predição de notas são apresentados na Tabela 5.5. Diferentemente da predição de tablaturas, que é baseada em uma saída Softmax, a saída de predição de notas é multi-rótulo, portanto, é necessário definir um limiar de decisão para estabelecer quando uma classe é predita ou não. Os limiares foram definidos testando-se valores entre 0,1 e 0,9 com passo de 0,1 usando o conjunto de validação. O melhor limiar para cada experimento em termos de F1-score (em torno de 0,3 e 0,4) foi então utilizado

para realizar os testes no conjunto de teste.

Experimento	Considerando *onset* e *offset*			Sem considerar *offset*		
	P	R	F	P	R	F
A-NOTES	***0,638 ± 0,130***	***0,782 ± 0,084***	***0,693 ± 0,089***	***0,782 ± 0,139***	**0,959 ± 0,058**	***0,851 ± 0,078***
A-NOTESG	0,559 ± 0,138	0,780 ± 0,074	0,643 ± 0,106	0,701 ± 0,146	*0,983 ± 0,025*	0,808 ± 0,097
A-NOTESG-PE	0,558 ± 0,139	0,775 ± 0,076	0,640 ± 0,106	0,701 ± 0,146	0,980 ± 0,030	0,807 ± 0,094
CTC-NOTES	0,328 ± 0,122	*0,704 ± 0,114*	0,439 ± 0,120	0,458 ± 0,138	*0,993 ± 0,014*	0,615 ± 0,118
CTC-NOTESG	0,366 ± 0,131	0,641 ± 0,140	0,458 ± 0,125	0,558 ± 0,142	0,983 ± 0,027	0,700 ± 0,101
CTC-NOTESG-PE	***0,368 ± 0,129***	**0,642 ± 0,138**	***0,461 ± 0,125***	***0,560 ± 0,136***	**0,981 ± 0,031**	***0,702 ± 0,098***

Tabela 5.5: Resultados de precisão (P), revocação (R) e medida F dos experimentos para a predição de notas no conjunto de teste do GuitarSet. Em negrito, o melhor resultado de cada grupo de experimento em termos de medida-F. Em itálico, os melhores resultados em cada métrica.

Em geral, a predição de notas apresenta um resultado consideravelmente mais satisfatório se comparado a predição das tablaturas. Novamente os resultados para os modelos com alinhamento mostram maior performance, porém os experimentos treinados com MCTC também demonstram resultados promissores. Também nota-se um melhor desempenho ao desconsiderar o *offset* das notas nos experimentos.

Um ponto interessante para os resultados da Tabela 5.5 é que, diferentemente da tarefa de predição de tablaturas, não há uma alta prevalência de falsos negativos na tarefa de predição de notas, o que sugere uma menor confusão do modelo em detectar as notas corretas, mas um maior desafio em detectar a posição correta do instrumento. Esse resultado demonstra que a tarefa de predição de tablaturas é mais desafiadora que a tarefa de predição de notas.

5.3 Exemplo de alinhamento

O alinhamento forçado apresenta algumas limitações para este trabalho. Primeiro, o alinhamento utilizando 6 saídas independentes não é uma tarefa trivial, uma vez que algumas saídas podem não apresentar predições. Outro problema é o fato de que o modelo pode confundir a corda em que a nota foi tocada, dificultando ainda mais o processo de alinhamento.

Adaptando o tutorial proposto pela documentação oficial do Pytorch [82] para lidar com 6 saídas simultâneas, pode-se obter os alinhamentos das saídas com trechos de áudios, como na Figura 5.2. Em geral, os testes realizados neste trabalho não apresentaram desempenho suficientes para a aplicação da técnica para a criação de novos *datasets*. Apesar do resultado ser razoável quando aplicado no *dataset* GuitarSet, os modelos não generalizaram suficientemente para dados fora de domínio. Além desses problemas, também é possível observar que o alinhamento do modelo tem um comportamento de baixa probabilidade durante a execução da nota. Esse problema é comum em modelos

treinados com o CTC [80]. Apesar desse comportamento não ser um grande problema para a transcrição de tablaturas, isso pode ser uma limitação caso a duração da predição tenha uma importância maior, como na transcrição musical em geral. Nesse caso, a utilização de funções CTC adaptadas para diminuir esse problema podem ser mais recomendadas, como a proposta por [24].

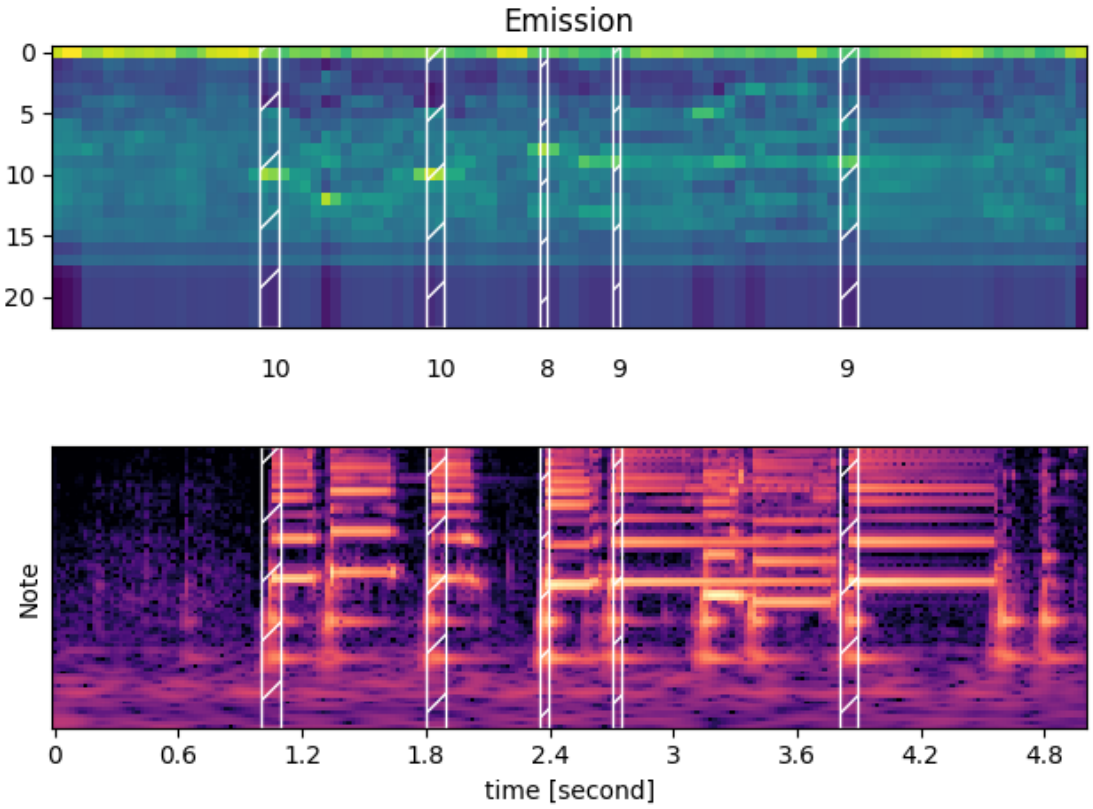


Figura 5.2: Exemplo de alinhamento obtido na corda 4 em um trecho de 5 segundos do áudio `03_SS2-88-F_solo` do *dataset* GuitarSet. Na parte superior, a saída probabilística da saída 4 do módulo de predição de tablaturas do modelo e suas respectivas predições alinhadas. Na parte inferior, o CQT de entrada.

CAPÍTULO 6

Conclusão

A Transcrição Automática de Guitarra apresenta vários desafios, tais como a presença de polifonia no som e a necessidade de identificar as posições tocadas corretamente, que podem ser ambíguas. O treinamento dos modelos normalmente é realizado com a presença de dados alinhados, ou seja, dados contendo a informação das notas tocadas, mas a obtenção desse tipo de *dataset* representa um desafio adicional, uma vez que a anotação desse tipo de dado é bastante custosa e ineficiente. Nesse sentido, este trabalho investigou o uso da técnica de Classificação Temporal Conexionista, muito utilizada em outros domínios, como o reconhecimento de fala, em que há uma abundância de dados não alinhados, com o objetivo de verificar a viabilidade do treinamento de modelos utilizando apenas segmentos de áudio com suas respectivas sequências. Mais especificamente, este trabalho realizou a investigação e a comparação da CTC para a tarefa de transcrição de tablaturas em dados não alinhados e a classificação com dados alinhados utilizando uma arquitetura baseada em redes neurais artificiais. Adicionalmente, este trabalho demonstrou o benefício do aprendizado multi-tarefa por meio da transcrição de notas em conjunto com a transcrição de tablatura com o uso do MCTC, uma variação da CTC para problemas multi-rótulo, obtendo resultados próximos aos obtidos nos experimentos com dados alinhados. Em relação a trabalhos similares que também utilizaram a CTC como função de custo, este trabalho realizou uma investigação distinta nos seguintes aspectos: (1) lidou com polifonia do som ao mesmo tempo que (2) investigou a transcrição da tablatura de ponta-a-ponta, realizando a transcrição de todas as cordas em paralelo.

A melhor abordagem utilizando CTC apresentou uma melhoria relativa de 14,67% no F1-score em comparação com o experimento *baseline* utilizando dados alinhados. Em comparação com o melhor modelo treinado com alinhamento, o modelo apresentou uma queda relativa de apenas 14,28%. Esses resultados indicam que a utilização da CTC é bastante promissora no contexto de transcrição musical, especialmente para a transcrição de tablaturas. Além disso, o treinamento de modelos mais robustos se tornam mais fáceis por meio de técnicas de alinhamento, que podem possibilitar a criação de novos conjuntos de dados massivos para a tarefa de transcrição.

Apesar dos avanços, os modelos treinados ainda não demonstram capacidade

suficiente de generalização para viabilizar a criação de novos *datasets* utilizando a técnica de alinhamento forçado, portanto, ainda se faz necessário a investigações de diferentes vias que possibilitem o treinamento de modelos mais robustos. Os experimentos também apontam algumas limitações na arquitetura e nos métodos propostos, que podem servir de base para a realização de trabalhos futuros, tais como:

- Limitação da função de custo e das saídas paralelas do módulo de predição de tablaturas: esse módulo considera que as saídas são independentes, dificultando a tarefa de aprendizado do modelo;
- Característica de entrada: é possível que outras características, como os espectrogramas lineares ou o áudio bruto, possam produzir resultados melhores, no entanto são dados que requerem maior tempo de treinamento e uma maior quantidade de dados;
- A investigação de adaptações da CTC: o CTC apresenta algumas limitações para a predição da duração das notas, e a investigação de algoritmos mais adaptados podem ser mais adequados para a tarefa de transcrição musical em geral;
- Outras arquiteturas: existem arquiteturas mais robustas para a tarefa de transcrição, por exemplo o Transformer, porém, essas arquiteturas necessitam de mais dados para que sejam capazes de aprender adequadamente.

Em suma, existem poucos *datasets* disponíveis para a tarefa de transcrição de guitarra. Apesar dos avanços recentes, os *datasets* disponíveis apresentam várias limitações como a baixa variedade de timbre, a baixa quantidade de áudios ou a baixa qualidade das transcrições. Essas limitações dificultam o treinamento de modelos robustos, ou ainda, o treinamento de modelos de alinhamento, que possam ser usados para a criação de *datasets* massivos. Nesse sentido, uma das principais vias que podem ser ressaltadas como trabalhos futuros é o treinamento de modelos de transferência de timbre, que possibilitem a geração de dados sintéticos próximos do real. Uma vez que esses dados sejam gerados, o treinamento de modelos de alinhamento se tornam mais factíveis, possibilitando a geração de *datasets* massivos e o treinamento de modelos robustos para a tarefa de transcrição de tablaturas.

Referências Bibliográficas

[1] ALDWELL, E.; CADWALLADER, A. **Harmony and voice leading**. Cengage Learning, 2018.

[2] AUDACITY. **Spectrogram view**, 2019.

[3] BAHDANAU, D.; CHO, K.; BENGIO, Y. **Neural machine translation by jointly learning to align and translate**. *arXiv preprint arXiv:1409.0473*, 2014.

[4] BANSAL, S. **Explanation of connectionist temporal classification**. https://sid2697.github.io/Blog_Sid/algorithm/2019/10/19/CTC-Loss.html, Oct 2019. Acesso em 04/08/2024.

[5] BARBANCHO, A. M.; KLAPURI, A.; TARDÓN, L. J.; BARBANCHO, I. **Automatic transcription of guitar chords and fingering from audio**. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(3):915–921, 2011.

[6] BENETOS, E.; DIXON, S.; DUAN, Z.; EWERT, S. **Automatic music transcription: An overview**. *IEEE Signal Processing Magazine*, 36(1):20–30, 2018.

[7] BENSON, D. **Music: A mathematical offering**, 2007.

[8] BRIDLE, J. S. **Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition**. In: *Neurocomputing: Algorithms, architectures and applications*, p. 227–236. Springer, 1990.

[9] BURLET, G.; HINDLE, A. **Isolated guitar transcription using a deep belief network**. *PeerJ Computer Science*, 3:e109, 2017.

[10] BURLET, G. D. **Automatic Guitar Tablature Transcription Online**. *McGill University Libraries*, (April), 2013.

[11] CHAN, W.; JAITLY, N.; LE, Q.; VINYALS, O. **Listen, attend and spell: A neural network for large vocabulary conversational speech recognition**. In: *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, p. 4960–4964. IEEE, 2016.

[12] CHEN, Y.-H.; HSIAO, W.-Y.; HSIEH, T.-K.; JANG, J.-S. R.; YANG, Y.-H. **Towards automatic transcription of polyphonic electric guitar music: A new dataset and a multi-loss transformer model**. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 786–790, 2022.

[13] CHEUK, K. W.; ANDERSON, H.; AGRES, K.; HERREMANS, D. **nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolutional neural networks**. *IEEE Access*, 8:161981–162003, 2020.

[14] CLEVERT, D.-A.; UNTERTHINER, T.; HOCHREITER, S. **Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)**. p. 1–14, 2015.

[15] COPPIN, B. **Artificial intelligence illuminated**. Jones & Bartlett Learning, 2004.

[16] CWITKOWITZ, F.; HIRVONEN, T.; KLAPURI, A. **Fretnet: Continuous-valued pitch contour streaming for polyphonic guitar tablature transcription**. *arXiv preprint arXiv:2212.03023*, 2022.

[17] DE CANDÈ, R. **História universal da música**. Número v. 1. WMF MARTINS FONTES, 2001.

[18] FONSECA, N. **Introdução à engenharia de som**. *Lisboa: FCA–Editora de Informática, Lda*, 2007.

[19] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. MIT press, 2016.

[20] GRAVES, A.; FERNÁNDEZ, S.; GOMEZ, F.; SCHMIDHUBER, J. **Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks**. In: *Proceedings of the 23rd international conference on Machine learning*, p. 369–376. ACM, 2006.

[21] HARRIS, F. J. **On the use of windows for harmonic analysis with the discrete fourier transform**. *Proceedings of the IEEE*, 66(1):51–83, 1978.

[22] HAYKIN, S. S.; OTHERS. **Neural networks and learning machines/Simon Haykin.** New York: Prentice Hall„ 2009.

[23] HINTON, G. E. **Deep belief networks**. *Scholarpedia*, 4(5):5947, 2009.

[24] HUANG, R.; ZHANG, X.; NI, Z.; SUN, L.; HIRA, M.; HWANG, J.; MANOHAR, V.; PRATAP, V.; WIESNER, M.; WATANABE, S.; OTHERS. **Less peaky and more accurate ctc forced alignment by label priors**. In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 11831–11835. IEEE, 2024.

[25] HUANG, T.-S.; YU, P.-C.; SU, L. **Note and playing technique transcription of electric guitar solos in real-world music performance**. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 1–5. IEEE, 2023.

[26] HUMPHREY, E. J.; BELLO, J. P. **From music audio to chord tablature: Teaching deep convolutional networks toplay guitar**. In: *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, p. 6974–6978. IEEE, 2014.

[27] HUMPHREY, E. J.; SALAMON, J.; NIETO, O.; FORSYTH, J.; BITTNER, R. M.; BELLO, J. P. **Jams: A json annotated music specification for reproducible mir research.** In: *ISMIR*, p. 591–596, 2014.

[28] JURAFSKY, D.; MARTIN, J. H. **Speech and language processing (3rd (draft) ed.)**, 2019.

[29] KEHTARNAVAZ, N. **Digital signal processing system design: LabVIEW-based hybrid programming**. Elsevier, 2011.

[30] KELZ, R.; DORFER, M.; KORZENIOWSKI, F.; BÖCK, S.; ARZT, A.; WIDMER, G. **On the potential of simple framewise approaches to piano transcription**, 2016.

[31] KIM, J. W.; BELLO, J. P. **Adversarial Learning for Improved Onsets and Frames Music Transcription**. jun 2019.

[32] KIM, S.; HAYASHI, T.; TODA, T. **Note-level automatic guitar transcription using attention mechanism**. In: *2022 30th European Signal Processing Conference (EUSIPCO)*, p. 229–233. IEEE, 2022.

[33] KIM, S.; TAKEDA, K.; TODA, T. **Sequence-to-sequence network training methods for automatic guitar transcription with tokenized outputs.** In: *ISMIR*, p. 524–531, 2023.

[34] KINGMA, D. P.; BA, J. **Adam: A method for stochastic optimization**. *arXiv preprint arXiv:1412.6980*, 2014.

[35] KÜRZINGER, L.; WINKELBAUER, D.; LI, L.; WATZEL, T.; RIGOLL, G. **Ctc-segmentation of large corpora for german end-to-end speech recognition**. In: *International Conference on Speech and Computer*, p. 267–278. Springer, 2020.

[36] LAZZARINI, V. E. **Elementos de acústica**. *apostila do Departamento de Artes da UEL, Londrina*, 1998.

[37] LECUN, Y.; JACKEL, L.; BOTTOU, L.; BRUNOT, A.; CORTES, C.; DENKER, J.; DRUCKER, H.; GUYON, I.; MULLER, U.; SACKINGER, E.; OTHERS. **Comparison of learning algorithms for handwritten digit recognition**. In: *International conference on artificial neural networks*, volume 60, p. 53–60. Perth, Australia, 1995.

[38] LECUN, Y. A.; BOTTOU, L.; ORR, G. B.; MÜLLER, K.-R. **Efficient backprop**. In: *Neural networks: Tricks of the trade*, p. 9–48. Springer, 2012.

[39] LEE, J.; KIM, T. **Cnn architectures for music classification**. https://mac.kaist.ac.kr/music_classification.html. Acesso em 04/08/2024.

[40] LOUREIRO, M. A.; PAULA, H. B. D. **Timbre de um instrumento musical: caracterização e representação**. *Per Musi*, 14:57–81, 2006.

[41] MARCONDES, J. **As características da guitarra**, 2018.

[42] MARTINS, A. L. **A guitarra elétrica na música experimental: composição, improvisação e novas tecnologias**. PhD thesis, Universidade de São Paulo, 2015.

[43] MED, B. **Teoria da música. 4ª edição revista e ampliada**. *Musimed Edições Musicais*, 1996.

[44] MISRA, D. **Mish: A self regularized non-monotonic neural activation function**. *arXiv preprint arXiv:1908.08681*, 4(2):10–48550, 2019.

[45] MITCHELL, T. M. **Machine learning. 1997**. *Burr Ridge, IL: McGraw Hill*, 1997.

[46] MULLER, M.; ELLIS, D. P.; KLAPURI, A.; RICHARD, G. **Signal processing for music analysis**. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1088–1110, 2011.

[47] NAIR, V.; HINTON, G. E. **Rectified linear units improve restricted boltzmann machines**. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, p. 807–814, 2010.

[48] NIELSEN, M. A. **Neural networks and deep learning**, volume 25. Determination press San Francisco, CA, USA:, 2015.

[49] PALEARI, M.; HUET, B.; SCHUTZ, A.; SLOCK, D. **A multimodal approach to music transcription**. In: *2008 15th IEEE International Conference on Image Processing*, p. 93–96. IEEE, 2008.

[50] PARK, D. S.; CHAN, W.; ZHANG, Y.; CHIU, C.-C.; ZOPH, B.; CUBUK, E. D.; LE, Q. V. **Specaugment: A simple data augmentation method for automatic speech recognition**. *arXiv preprint arXiv:1904.08779*, 2019.

[51] PEREIRA, M. F.; GLOEDEN, E. **De maldito a erudito: caminhos do violão solista no brasil**. *Revista Composição UFMS*, (10):68–91, 2012.

[52] PISZCZALSKI, M.; GALLER, B. A. **Automatic music transcription**. *Computer Music Journal*, p. 24–31, 1977.

[53] PRATAP, V.; TJANDRA, A.; SHI, B.; TOMASELLO, P.; BABU, A.; KUNDU, S.; ELKAHKY, A.; NI, Z.; VYAS, A.; FAZEL-ZARANDI, M.; OTHERS. **Scaling speech technology to 1,000+ languages**. *Journal of Machine Learning Research*, 25(97):1–52, 2024.

[54] QUINTANILHA, I. M. **End-to-End Speech Recognition Applied to Brazilian Portuguese Using Deep Learning**. PhD thesis, MSc dissertation, PEE/COPPE, Federal University of Rio de Janeiro, Rio de . . . , 2017.

[55] RADICIONI, D.; ANSELMA, L.; LOMBARDO, V. **A segmentation-based prototype to compute string instruments fingering**. *Proceedings of the Conference . . .*, (May), 2004.

[56] RADICIONI, D.; LOMBARDO, V. **Guitar fingering for music performance**. In: *ICMC*, 2005.

[57] RAFFEL, C.; MCFEE, B.; HUMPHREY, E. J.; SALAMON, J.; NIETO, O.; LIANG, D.; ELLIS, D. P.; RAFFEL, C. C. **Mir_eval: A transparent implementation of common mir metrics.** In: *ISMIR*, volume 10, p. 2014, 2014.

[58] RAMOS, J. V. **Diferentes abordagens evolutivas aplicadas no processo de transcrição automática de partituras musicais em tablaturas**. Dissertação (mestrado), Universidade Tecnológica Federal do Paraná, 2016.

[59] ROADS, C.; STRAWN, J.; OTHERS. **The computer music tutorial**. MIT press, 1996.

[60] ROMÁN, M. A.; PERTUSA, A.; CALVO-ZARAGOZA, J. **An end-to-end framework for audio-to-score music transcription on monophonic excerpts.** In: *ISMIR*, p. 34–41, 2018.

[61] SARMENTO, P.; KUMAR, A.; CARR, C.; ZUKOWSKI, Z.; BARTHET, M.; YANG, Y.-H. **Dadagp: A dataset of tokenized guitarpro songs for sequence models**. *arXiv preprint arXiv:2107.14653*, 2021.

[62] SAYEGH, S. I. **Fingering for string instruments with the optimum path paradigm**. *Computer Music Journal*, 13(3):76–84, 1989.

[63] SCHEIRER, E. D. **Music perception systems**. *Proposal for PhD dissertation, MIT Media Laboratory*, 1998.

[64] SCHERER, D.; MÜLLER, A.; BEHNKE, S. **Evaluation of pooling operations in convolutional architectures for object recognition**. In: *International conference on artificial neural networks*, p. 92–101. Springer, 2010.

[65] SCHMIDT-JONES, C. **Understanding basic music theory**. 2013.

[66] SCHÖRKHUBER, C.; KLAPURI, A. **Constant-q transform toolbox for music processing**. In: *7th sound and music computing conference, Barcelona, Spain*, p. 3–64, 2010.

[67] SEJDIĆ, E.; DJUROVIĆ, I.; JIANG, J. **Time–frequency feature representation using energy concentration: An overview of recent advances**. *Digital signal processing*, 19(1):153–183, 2009.

[68] SHIBATA, K.; NISHIKIMI, R.; FUKAYAMA, S.; GOTO, M.; NAKAMURA, E.; ITOYAMA, K.; YOSHII, K. **Joint transcription of lead, bass, and rhythm guitars based on a factorial hidden semi-markov model**. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 236–240. IEEE, 2019.

[69] SIGTIA, S.; BENETOS, E.; DIXON, S. **An end-to-end neural network for polyphonic piano music transcription**. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 24(5):927–939, May 2016.

[70] SMITH, J. O. **Mathematics of the discrete Fourier transform (DFT): with audio applications**. Julius Smith, 2007.

[71] STEVENS, S. S.; VOLKMANN, J.; NEWMAN, E. B. **A scale for the measurement of the psychological magnitude pitch**. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

[72] TUOHY, D. R.; POTTER, W. D. **Ga-based music arranging for guitar**. In: *2006 IEEE International Conference on Evolutionary Computation*, p. 1065–1070. IEEE, 2006.

[73] VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. **Attention is all you need**. *arXiv preprint arXiv:1706.03762*, 2017.

[74] WEISS, C.; PEETERS, G. **Training deep pitch-class representations with a multi-label ctc loss**. In: *International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[75] WIGGINS, A.; KIM, Y. E. **Guitar tablature estimation with a convolutional neural network.** In: *ISMIR*, p. 284–291, 2019.

[76] WIGINGTON, C.; PRICE, B.; COHEN, S. **Multi-label connectionist temporal classification**. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*, p. 979–986. IEEE, 2019.

[77] XI, Q.; BITTNER, R. M.; PAUWELS, J.; YE, X.; BELLO, J. P. **Guitarset: A dataset for guitar transcription.** In: *ISMIR*, p. 453–460, 2018.

[78] ZANG, Y.; ZHONG, Y.; CWITKOWITZ, F.; DUAN, Z. **Synthtab: Leveraging synthesized data for guitar tablature transcription**. In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 1286–1290. IEEE, 2024.

[79] ZEILER, M. D. **ADADELTA: an adaptive learning rate method**. *CoRR*, abs/1212.5701, 2012.

[80] ZEYER, A.; SCHLÜTER, R.; NEY, H. **Why does ctc result in peaky behavior?** *arXiv preprint arXiv:2105.14849*, 2021.

[81] ZHANG, A.; LIPTON, Z. C.; LI, M.; SMOLA, A. J. **Dive into deep learning**. Cambridge University Press, 2023.

[82] ZHANG, X.; HIRA, M. **Ctc forced alignment api tutorial**. https://pytorch.org/audio/main/tutorials/ctc_forced_alignment_api_tutorial.html. Acesso em 04/08/2024.

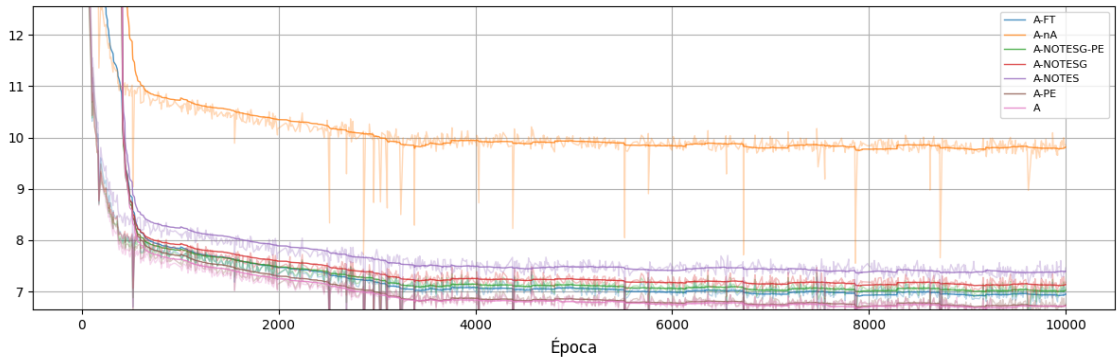
[83] ZHOU, Y.-T.; CHELLAPPA, R. **Computation of optical flow using a neural network**. In: *IEEE International Conference on Neural Networks*, volume 1998, p. 71–78, 1988.

APÊNDICE A

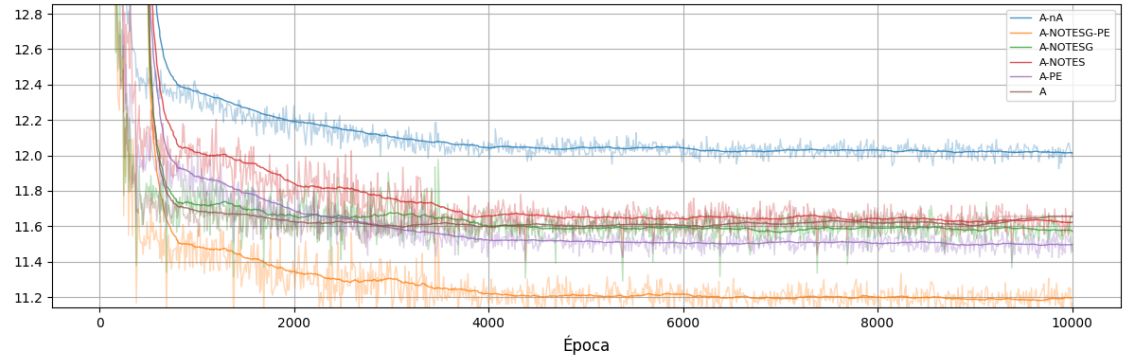
Logs de treinamento

Neste apêndice são apresentados os *logs* de treinamento dos experimentos realizados. As curvas foram suavizadas para facilitar a visualização.

A.1 Experimentos com alinhamento

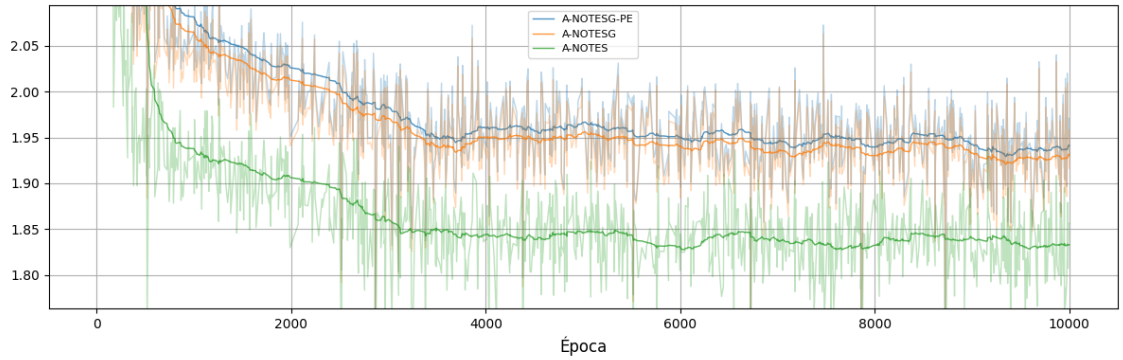


(a)

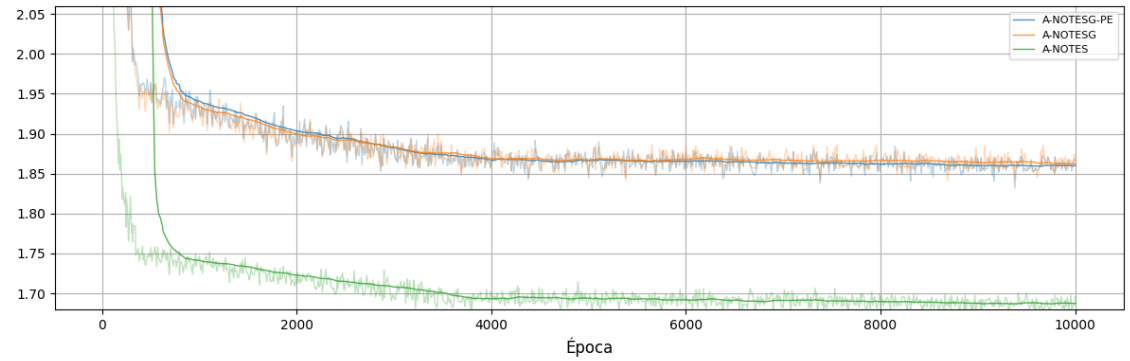


(b)

Figura A.1: *Loss* para a predição de tablatura nos conjuntos de treino (a) e validação (b).

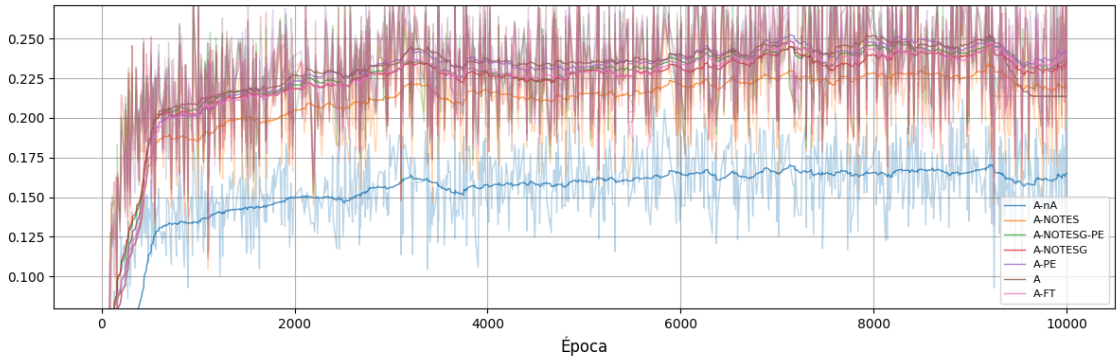


(a)

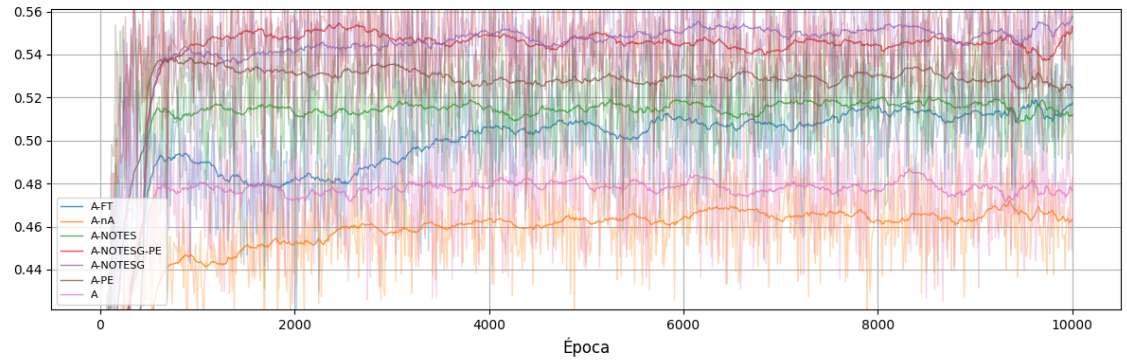


(b)

Figura A.2: *Loss* para a predição de notas nos conjuntos de treino (a) e validação (b).



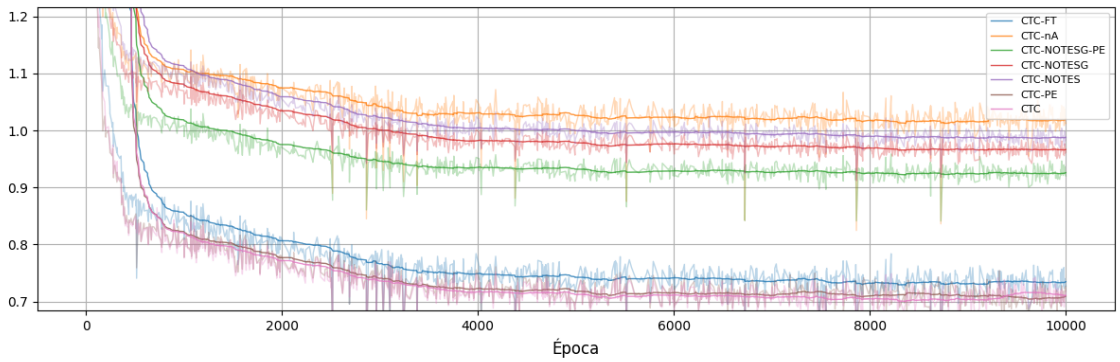
(a)



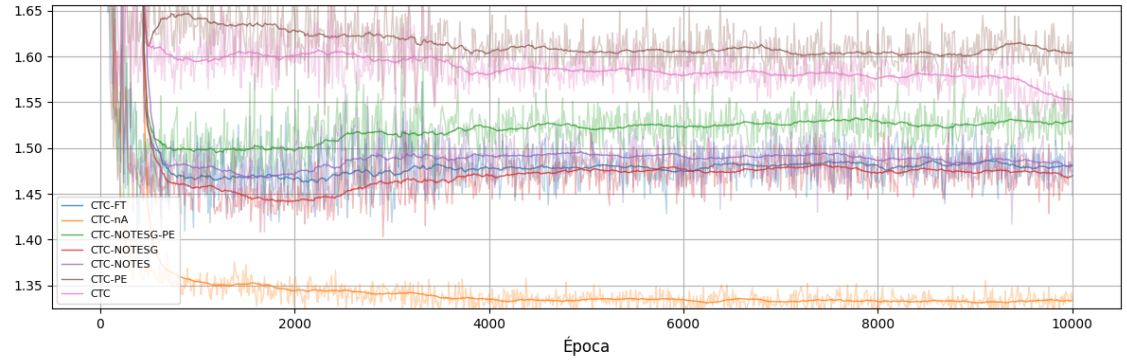
(b)

Figura A.3: Medida F nos conjuntos de treino (a) e validação (b).

A.2 Experimentos sem alinhamento (CTC)

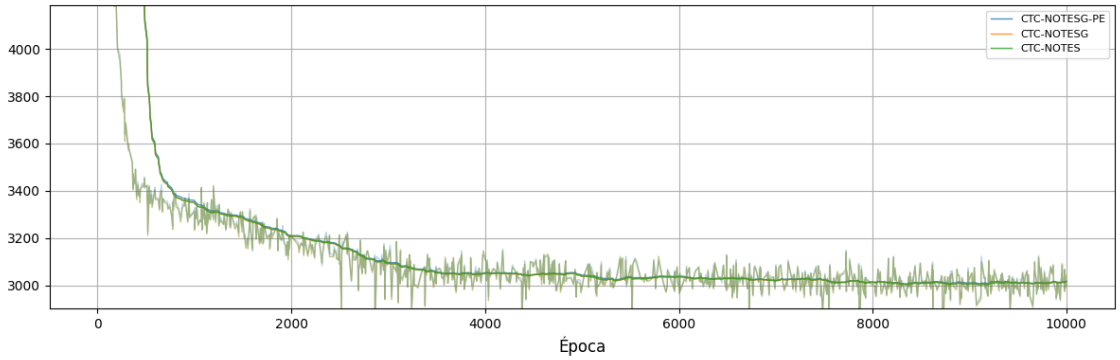


(a)

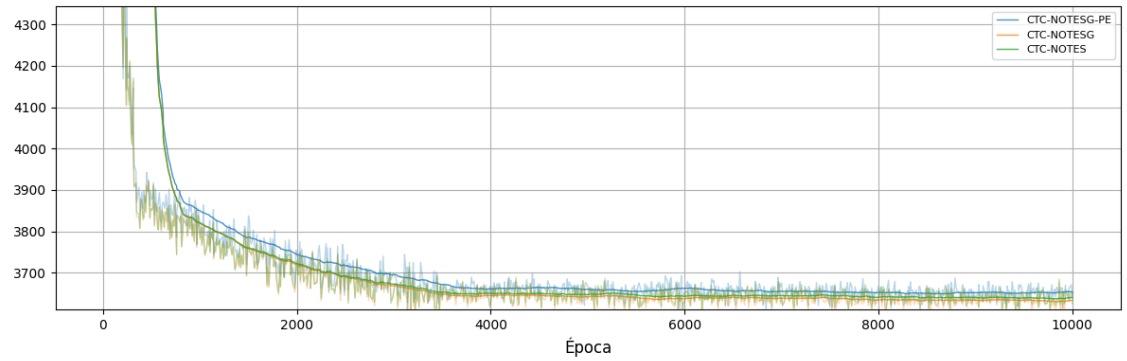


(b)

Figura A.4: *Loss* para a predição de tablatura nos conjuntos de treino (a) e validação (b).

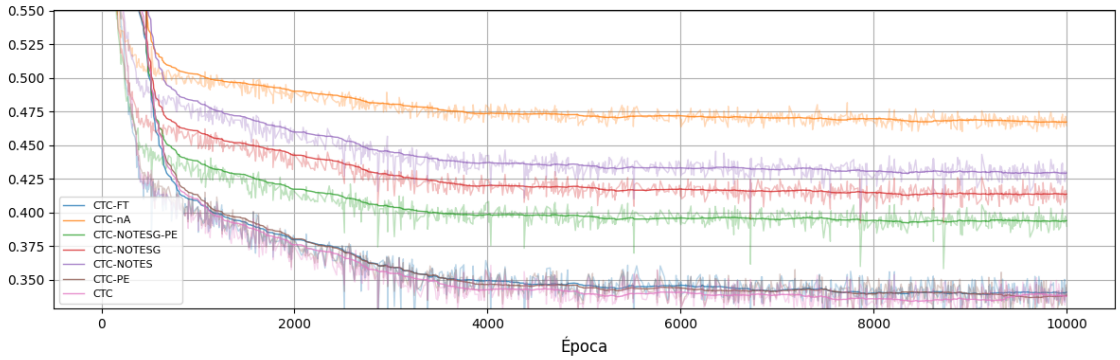


(a)

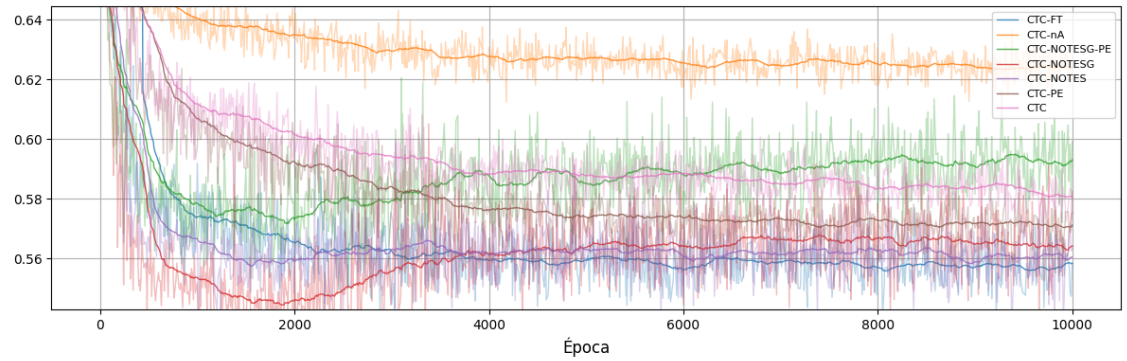


(b)

Figura A.5: *Loss* para a predição de notas nos conjuntos de treino (a) e validação (b).



(a)



(b)

Figura A.6: *Character Error Rate* (CER) nos conjuntos de treino (a) e validação (b).