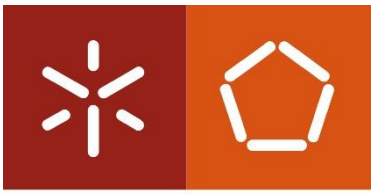

Universidade do Minho

Mestrado em Engenharia Informática

Reconhecimento de voz - Aprendizagem Profunda

Duarte Parente (PG53791)
Francisca Lemos (PG52693)
Santiago Domingues (PG54225)

31 de maio de 2024

Conteúdo

Capítulo 1

Introdução

Desde a última década, o *Deep Learning* (DL) surgiu como uma nova área atrativa de *Machine Learning* (ML) e desde então tem sido examinado e utilizado nuuma variedade de diferentes tópicos de pesquisa. Esta nova área de ML tem obtido resultados muito melhores em comparação com outras em várias aplicações, inclundo tarefas de voz, e, assim, tornou-se uma área de investigação atrativa. [1] As tecnologias de processamento de voz, como a conversação por voz (VC) e o reconhecimento automático de voz (ASR), melhoraram drasticamente na última década, graças a esses mesmos avanços do DL. No entanto, a tarefa de treinar esses modelos continua a ser um desafio em domínios com poucos recursos, pois eles sofrem de *overfitting* e não generalizam bem para aplicações práticas. [2]

O objetivo deste projeto é desenvolver um modelo *Transformers* para reconhecimento de voz utilizando datasets públicos. Para tal, o grupo decidiu utilizar um modelo pré-treinado da *Hugging Face* especificamente para esta tarefa e realizar o fine-tuning do mesmo. No caso deste projeto, foi necessário preparar o dataset para que fosse compatível com o modelo pré-treinado e, em seguida, ajustar os hiperparâmetros e otimizar o desempenho para obter boas métricas e evitar overfitting. Este relatório fornece um roteiro detalhado de todo o trabalho desenvolvido, desde as escolhas iniciais até as decisões finais sobre o modelo.

Capítulo 2

Descrição do Modelo

2.1 Escolha do Modelo

Para o desenvolvimento do nosso projeto de reconhecimento de voz, escolhemos utilizar o modelo pré-treinado **Wav2Vec2** da Hugging Face. Utilizar um modelo pré-treinado permite aproveitar a aprendizagem já realizada em grandes quantidades de dados de áudio. Isso permite economizar tempo de treino, mas também melhora a precisão inicial do modelo antes do fine-tuning com os nossos dados. A escolha deste modelo foi motivada por várias razões, mas principalmente, pela sua eficácia para a tarefa de reconhecimento de voz. Segundo [3], aprender representações poderosas apenas a partir de áudios, seguido de ajustes (*fine-tuning*) em transcrições de voz, pode superar os melhores métodos semi-supervisionados, e, ao mesmo tempo que é conceptualmente mais simples.

2.2 Arquitetura do Modelo

Nesta secção será explicada a arquitetura do modelo escolhido, retirado do artigo [3]. O modelo é composto por um codificador de características convolucionais de várias camadas, que recebe como entrada áudio bruto e produz representações de voz. Em seguida, elas são alimentadas a um Transformer para construir representações capturando informações de toda a sequência. A saída do codificador de características é discretizada com um módulo de quantização para representar os alvos na tarefa auto-supervisionada.

- **Feature Encoder:** consiste em vários blocos contendo uma convolução temporal seguida de normalização de camada e uma função de ativação GELU. A entrada de forma de onda bruta para o codificador é normalizada para média zero e variância unitária. O passo total do codificador determina o número de passos de tempo que são fornecidos ao Transformer.

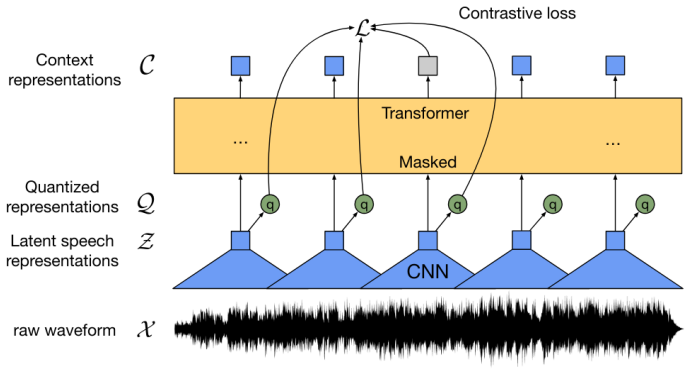

Figura 2.1: Arquitetura do Modelo *Wac2Vec2* [3]

- **Representações contextualizadas com Transformers:** A saída do codificador de características é alimentada a uma rede de contexto que segue a arquitetura do Transformer. Adicionamos a saída da convolução seguida de um GELU às entradas e, em seguida, aplicamos normalização de camada.

- **Módulo de quantização:** Para o treino auto-supervisionado, discretizamos a saída do codificador de características para um conjunto finito de representações de fala via quantização de produto. A quantização de produto consiste em escolher representações quantizadas de múltiplos codebooks e concatená-las. Isto torna o modelo mais eficiente em termos de memória e recursos computacionais.

- **Transformers:** O coração do Wav2Vec2 é uma stack de camadas Transformer. O Transformer aplica mecanismos de atenção para modelar dependências a longo alcance dentro do sinal de áudio. Isso é crucial para capturar o contexto necessário para a transcrição precisa do discurso.

Dentro dos modelos propostos, o escolhido pelo grupo foi **Wav2Vec2-Base-100h**. É um modelo pré-treinado e ajustado em 100 horas de áudio amostrado a 16kHz do dataset Librispeech. O modelo 'base' possui 12 camadas Transformer, cada uma com 8 *attention heads*. Após o pré-treino, o modelo pode ser ajustado (fine-tuning) para tarefas específicas

Capítulo 3

Preparação dos Dados

3.1 Dataset

Devido a limites computacionais, não foi possível utilizar o dataset inicialmente pensado, o *Librispeech*. Além disso, como vários datasets não permitem carregar apenas uma parte do mesmo , e como era totalmente impossível carregar a totalidade do dataset, optamos por um dataset menor. O dataset escolhido é o **'Hani89/medical_asr_recording_dataset'** e contém 8,5 horas de declarações de áudio combinadas com texto para sintomas médicos comuns. Essas gravações abrangem uma variedade de contextos e tópicos dentro da medicina, fornecendo uma ampla gama de material de treino para sistemas de reconhecimento automático de voz (ASR). A escolha deste dataset para o processo de fine-tuning é fundamentada pelas seguintes razões:

- O discurso médico muitas vezes envolve terminologias técnicas e conceitos altamente especializados. Ao treinar um modelo de ASR com essas gravações, ele é exposto a uma ampla gama de termos técnicos, o que ajuda a melhorar sua capacidade de reconhecer e interpretar corretamente esses termos em contextos variados;
- O dataset abrange uma diversidade de oradores de diferentes origens geográficas e culturais, o que torna o modelo de ASR mais robusto e capaz de lidar com uma variedade de vozes e sotaques.

3.2 Pré-Processamento dos Dados

Inicialmente, é essencial realizar um pré-processamento dos dados antes de alimentá-los no modelo. Isso inclui garantir que os áudios estejam amostrados a uma frequência de 16kHz. Além disso, é crucial remover caracteres considerados "especiais"e converter todos os caracteres para letras maiúsculas. Este processo de pré-processamento é aplicado tanto aos conjuntos de dados de treino quanto aos de teste. Uma vez concluído o pré-processamento, procede-se à divisão do

conjunto de dados em conjuntos separados para treino e validação. Essa divião é realizada utilizando a função ‘train_test_split‘, com uma proporção de 70% para treino e 30% para validação:

```
dataset = dataset.train\_test_split(test_size = 0.3)
```

Capítulo 4

Fine-Tuning do Modelo

Fine-tuning é a técnica de ajustar um modelo pré-treinado numa nova tarefa ou conjunto de dados específico. No nosso caso, iremos aplicar fine-tuning para adaptar o modelo às particularidades de um novo dataset, com um domínio diferente daquele em que foi originalmente treinado. Para a realização de fine-tune foi necessário algumas etapas:

Preparação dos Dados

Inicialmente, é necessário preparar os dados de áudio e transcrições para o modelo Wav2Vec2, convertendo o áudio em valores numéricos e codificando as transcrições em IDs de entrada, que são necessários para treinar o modelo de reconhecimento de voz (função **prepare_dataset**). Isto tudo aliado ao processamento já feito anteriormente.

Configuração do Treino

Configurar o treino do modelo envolve a definição de vários hiperparâmetros. Para isso, é necessário encontrar os melhores hiperparâmetros. Para essa procura utilizou-se a biblioteca de otimização automática de hiperparâmetros *Optuna*. Esta utiliza técnicas avançadas de amostragem e pruning para encontrar os melhores valores de hiperparâmetros de forma eficiente. Neste caso, o objetivo do modelo é minimizar a métrica escolhida, word error rate (WER), que iremos abordar mais à frente.

```
def hp_space_optuna(trial):
return {
"learning_rate": trial.suggest_float("learning_rate", 1e-6, 1e-4, log=True),
"per_device_train_batch_size": trial.suggest_categorical("per_device_train_batch_size", [4,8
"weight_decay": trial.suggest_float("weight_decay", 0.0, 0.3),
"num_train_epochs": trial.suggest_int("num_train_epochs", 3, 6)
}
```

É importante definir um otimizador para ajustar os parâmetros do modelo durante o treino. Neste caso, utlizou-se $AdamW$ por ser um método utilizado quando se deseja aplicar uma regularização correta e eficiente dos pesos do modelo, melhorando assim a capacidade de generalização e reduzindo o risco de overfitting.

```
optimizer = AdamW(model.parameters(), lr=learning_rate)
```

Após a procura de hiperparâmetros, foi possível identificar os melhores parâmetros para o treino do modelo Wav2Vec-base-100h. Através de múltiplas tentativas e ajustes automáticos e manuais, chegamos aos seguintes valores :

```
learning_rate: 4.321006599412334e-06
per_device_train_batch_size: 4
weight_decay: 0.10197235050164018
num_train_epochs: 6
```

Para mostrar a relação entre o erro do modelo em relação aos dados de treino e aos dados de validação ao longo do processo é mostrado um gráfico de perda de treino e validação. Com base na análise do gráfico seguinte(4.1), podemos concluir que o modelo apresenta um bom desempenho. A curva da perda de validação está baixa e segue uma tendência similar à curva da perda de treino, o que indica que o modelo está a generalizar bem para novos dados.

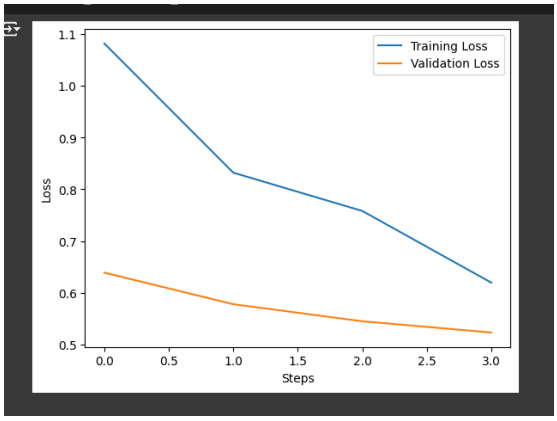

Figura 4.1: Gráfico da perda dos dados de treino e validação

Capítulo 5

Avaliação do Modelo

A métrica utilizada para avaliar a performance do modelo é Word Error Rate (WER). Esta calcula a diferença entre a transcrição automática produzida pelo modelo e a transcrição correta. Um WER mais baixo indica uma melhor performance do modelo, pois significa que a transcrição gerada está mais próxima da transcrição correta.

5.1 Modelo base - sem fine-tune

Com o objetivo de comparar a performance dos modelos antes e após o fine-tune, avaliamos o modelo nos dados de testes (executados pelo train_test_split) onde 70% corresponde aos dados de treino e 30% aos de teste/validação. A WER obtida foi de $37,5\%$. De relembrar que estes dados sofreram apenas o pré-processamento inicial.

5.2 Modelo com fine-tune

Após realizar o fine-tuning do modelo, carregamos conjuntos de dados de treino e teste específicos do próprio dataset e observou-se uma melhoria na performance do modelo de reconhecimento de voz. Os resultados obtidos mostraram que o WER foi reduzido para $22,8\%$. Na tabela 5.1 , é possível observar alguns dos resultados obtidos, i.e, a transcrição gerada pelo modelo e a original. Além disso, de forma a avaliar melhor o modelo, o grupo optou pela gravação de 3 áudios com temas variados. Em dois dos áudios é utilizado a voz do *Google Tradutor*. As transcrições previstas e originais , tal como a WER pode ser vista na tabela 5.2

Transcrição original	Transcrição prevista
i wake up at night feeling cold	i wake up at night feeling cold
i am having problems seeing things feel like a cloud on my eyes everything is blurry	i am having problem seeing things feel like a cloud on my eyes everything is blutry
when i play sports i have some burning sensation in my spine	when i play spolts i have some burnding sensation in my pine
i feel a clicking sensation in my knee each time i step	i feel a clicking sensation in my knee each time i step
i have eruptions on my face that come and go	i have eruptions on my face that come and go
i feel congestion in my chest	i feel conjestion in my chest

Figura 5.1: Resultados obtidos

Transcrição original	Transcrição prevista	WER
tragic end person kiddel at amsterdam airport was airline employee who intentionally climbed into jet engine	tragic end person killed at amster dam airport was alline imploy who intentionally climbed into jetengin	37.5%
best lung cancer drug halts tumours giving years to dying patients	best lung canser drug holds humors giving years to dying patience	36.5%
today I fell to the ground and twisted my foot	to day i fell to the groundand twisted my foot	40%

Figura 5.2: Resultados obtidos para as gravações do grupo

Capítulo 6

Interface gráfica

A interface web realizada pelo grupo serve, exclusivamente, para visualização dos resultados. É possível ouvir o áudio, ver o texto original e o previsto pelo modelo. Também é sublinhado as palavras erradas.

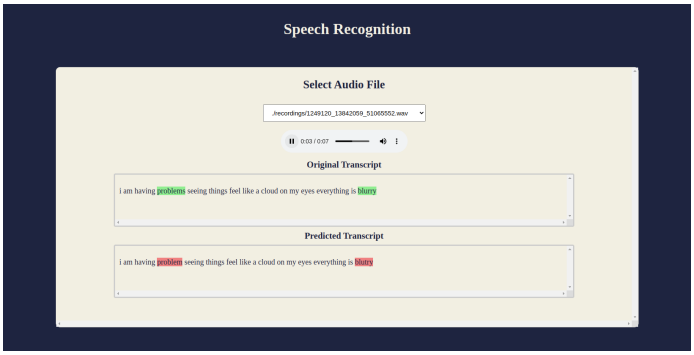

Figura 6.1: Página Inicial

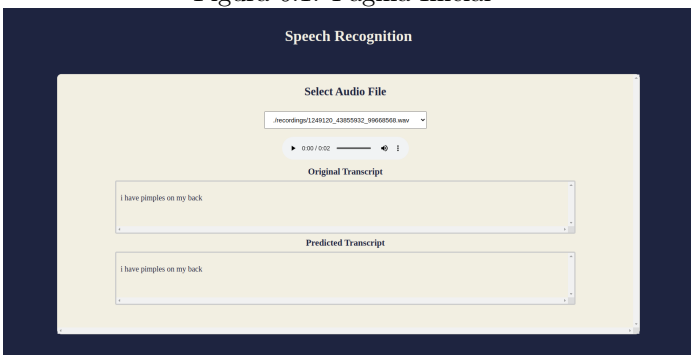

Figura 6.2: Página Inicial

Capítulo 7

Análise Crítica e Conclusão

Em conclusão, embora idealmente teria sido interessante trabalhar com um conjunto de dados diferente, os resultados obtidos com o conjunto de áudios médicos foram bastante satisfatórios. Devido às limitações impostas pelo dataset, o modelo não generalizou como gostariamos para os áudios colocados pelo grupo, principalmente, aqueles que foram gravados por nós. Além disso, uma limitação sentida é o limite computacional que o *Google Colab* apresenta, não conseguindo trabalhar com todos os dados do dataset. Daí, termos como objetivo futuro, trabalhar com a totalidade do dataset e, posteriormente, com outros datasets mais interessantes e maiores.

No que diz respeito à interface web, gostaríamos que não apresentasse apenas os resultados finais mas que fosse possível a gravação de áudio para posteriormente ser passado ao modelo. Com este projeto, foi possível observar o impacto positivo de modelos pré-treinados, pois é possível aproveitar o conhecimento sobre a linguagem e o contexto, o que faz com que se economize tempo e recursos. Além disso, esses conhecimentos podem ser transferidos para tarefas específicas com a técnica de fine-tune.

Bibliografia

[1] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech Recognition Using Deep Neural Networks: A Systematic Review,"in *IEEE Access*, vol. 7, pp. 19143-19165, 2019. doi:10.1109/ACCESS.2019.2896880.

[2] Mayank Kumar Singh, Naoya Takahashi, Onoe Naoyuk : "Iteratively Improving Speech Recognition and Voice Conversion,"2023 Sony Research India, Sony Group Corporations, Japan

[3] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, Michael Auli : "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations", 2020,2006.11477,doi:10.48550/arXiv.2006.11477