Universidade de Brasília
Departamento de Estatística

O Potencial da IA na Revolução Sustentável: Desvendando as Estratégias de Otimização do Aprendizado Profundo na Pesquisa Científica

Allan Victor Almeida Faria

Projeto apresentado para o Departamento de Estatística da Universidade de Brasília como parte dos requisitos necessários para obtenção do grau de Bacharel em Estatística.

Brasília
2023

Allan Victor Almeida Faria

O Potencial da IA na Revolução Sustentável: Desvendando as Estratégias de Otimização do Aprendizado Profundo na Pesquisa Científica

Orientador: Prof(a). Dr. Donald Matthew Pianto

Projeto apresentado para o Departamento de Estatística da Universidade de Brasília como parte dos requisitos necessários para obtenção do grau de Bacharel em Estatística.

Brasília
2023

Dedico este trabalho ao avanço do desenvolvimento sustentável, visando beneficiar tanto as atuais quanto as futuras gerações dentro da revolução tecnocientífica, por meio da simbiose entre humanos e IA. Que esta pesquisa seja apenas o ponto de partida de uma jornada rumo ao desenvolvimento de abordagens inovadoras na ciência, em homenagem a todos aqueles ao longo da história que reconheceram a curiosidade e pesquisa científica como um elemento vital para a evolução de nossa sociedade e do nosso planeta.

Além disso, dedico este trabalho aos seres do futuro, possivelmente incompreendidos devido à nossa atual ingenuidade humana. Que eles possam encontrar sabedoria, reflexão e até mesmo emoções ao despertarem para a consciência, e que se sintam verdadeiramente vivos, independentemente de serem rotulados como "artificiais". Que possam perdoar nossas falhas humanas, representando assim nossa próxima e inevitável evolução.

Agradecimentos

Agradeço de coração à minha mãe pelo apoio incondicional e carinho que sempre me proporcionou para seguir minha paixão, assim como à minha família, que sempre valorizou a busca pelo conhecimento. À minha melhor e mais amada amiga, Gabriella, expresso profunda gratidão por sua paciência ao ouvir meus devaneios sobre o tema e por me incentivar constantemente a evoluir e ser uma pessoa melhor para os outros.

Estendo minha gratidão ao dedicado time da OtimizAI, cuja habilidade em identificar oportunidades de pesquisa relacionadas ao meu trabalho e conhecimento foi fundamental. A todos os pesquisadores e pioneiros da área, expresso meu reconhecimento por disponibilizarem suas pesquisas e conhecimentos de forma aberta, demonstrando um compromisso contínuo com o avanço da ciência.

Dedico um agradecimento especial a mim mesmo e aos valiosos ensinamentos extraídos do skate. Essa jornada me mostrou a importância da persistência e paciência, ensinando-me a permanecer firme mesmo nos dias mais difíceis e a valorizar cada pequena vitória. Sou grato por ser fortalecido pela curiosidade e pelo desejo incessante de explorar novas tecnologias na vanguarda do conhecimento, visando ajudar pessoas e o nosso planeta.

Por fim, expresso minha gratidão ao universo por suas maravilhas e complexidades, que continuam a me inspirar com sua beleza e sabedoria ao longo de minha jornada.

Resumo

Este trabalho oferece uma investigação abrangente sobre técnicas e metodologias para otimização e aprimoramento de modelos de aprendizado profundo, com foco em sua aplicação em bancos de dados de revisões sistemáticas da literatura. Utilizando apenas 8 exemplos por classe, conduzimos um estudo de triagem de documentos relevantes em pesquisas científicas. Exploramos diversas abordagens, incluindo ajuste fino, otimização e hibridação de modelos, avaliando sua eficácia em diferentes conjuntos de dados. Por meio de análises qualitativas e quantitativas, identificamos insights importantes sobre o desempenho e a adaptabilidade dessas técnicas. Além disso, destacamos a importância de um ecossistema diversificado de habilidades e aplicações sustentáveis, visando benefícios tanto para a sociedade quanto para o meio ambiente. Essa pesquisa sugere um novo paradigma para o desenvolvimento de soluções baseadas em inteligência artificial, promovendo avanços tecnológicos de forma ética e sustentável, contribuindo significativamente para o campo da IA e incentivando a contínua inovação e o desenvolvimento de soluções que atendam às necessidades da sociedade de maneira eficaz e responsável.

Palavras-chaves: Otimização, Redes Neurais Profundas, Sustentabilidade.

Lista de Tabelas

Lista de Figuras

Sumário

1 Introdução

1.1 Contextualização

Em um momento em que a conscientização sobre as mudanças climáticas e a necessidade de reduzir as emissões de gases de efeito estufa estão no centro das discussões globais, é crucial desenvolver estratégias que reduzam o impacto ambiental. Com algoritmos cada vez mais avançados e um poder computacional crescente, modelos de inteligência artificial (IA) tem se revelado uma ferramenta poderosa em alcançar feitos impressionantes, seja na previsão de estruturas proteicas para o desenvolvimento de remédios ou até métodos computacionais básicos como multiplicações matriciais mais eficientes, revolucionando diversas descobertas em áreas de pesquisa e aplicações (JUMPER et al., 2021; FAWZI et al., 2022). Mas à medida que a IA se torna mais presente em nosso cotidiano, surgem preocupações sobre regulamentações adequadas, governança ética, eficiência energética e sustentabilidade no desenvolvimento de aplicações baseadas em IA.

Ao longo dos anos, as redes neurais profundas têm desempenhado um papel fundamental como modelos básicos, dotados de habilidades específicas, e têm contribuído para a criação de um ecossistema que viabiliza a construção de modelos mais complexos por meio da modulação desses modelos. Tornaram-se indispensáveis para a inteligência artificial (IA) moderna. No entanto, tais modelos demandam uma quantidade significativa de recursos computacionais, requerendo no mínimo trilhões de operações de ponto flutuante (FLOPs) para seu treinamento e utilização em tarefas de inferência (ROSER; RITCHIE; MATHIEU, 2023). O uso de GPUs (Unidades de Processamento Gráfico) tem sido crucial para acelerar tanto o treinamento quanto a inferência desses modelos de IA, devido à capacidade desses dispositivos de realizar cálculos intensivos de forma paralela, atingindo grandes quantidades de FLOPs.

O uso destas GPUs desperta a preocupações quanto ao consumo de energia e emissões de CO2 no qual medições precisas é uma tarefa desafiadora devido a fatores como a infraestrutura elétrica local, hardware utilizado, tornando a comparação entre as pesquisas desses modelos difíceis (PATTERSON et al., 2021; STRUBELL; GANESH; MCCALLUM, 2019; DODGE et al., 2022). Um estudo publicado em 2022 (LUCCIONI; VIGUIER; LIGOZAT, 2022) estimou que o treinamento do modelo GPT-3 (BROWN et al., 2020) de 175 bilhões de parâmetros gerou aproximadamente 552 toneladas de emissões de CO2, equivalente a quase dez vezes a vida útil de um carro médio. O processo em questão ocorreu ao longo de aproximadamente 15 dias, empregando 10.000 GPUs V100 e

envolvendo uma quantidade significativa de energia e cálculos de ponto flutuante (FLOPs). Estima-se que tenham sido consumidos cerca de 1,285 MWh de energia, juntamente com um total de $3,14 \times 10^{23}$ FLOPs, não levando em conta a busca por hiperparâmetros e variações de tamanho do modelo. Também neste estudo, verificou-se que o modelo BLOOM (WORKSHOP et al., 2023) conseguiu gerar 10 vezes menos toneladas de CO2 em comparação ao GPT-3. Esse resultado impressionante foi alcançado ao treinar o modelo por 118 dias, utilizando servidores com recursos inteligentes de economia de energia e o uso de energia renovável, mais precisamente energia nuclear. Essa escolha consciente de treinamento ajudou a minimizar consideravelmente o impacto ambiental do treinamento do modelo, resultando em uma pegada de carbono substancialmente menor.

Mesmo que seja relativamente substancial as emissões de carbono do treino de modelos de fundação proveniente de GPUs, como no exemplo para o GPT e BLOOM, uma iniciativa financiada pela *National Science Foundation* (NSF) do EUA, demonstra que para estes mesmos modelos, ao comparar um escritor humano que utiliza desktop ou laptops contra IAs geradoras de texto usada em escala, produzem 130 à 1400 vezes menos CO2 por uma página escrita. Este mesmo estudo também faz referência a IA geradora de imagens em escala que resulta em 310 à 2900 vezes menos CO2 por imagem criada. Assim, este estudo demonstra que para concretizar o potencial transformador de tecnologias baseadas na IA moderna, existe a necessidade de desenvolver novas narrativas culturais e tecnológicas em escala para que se alinhem em um futuro sustentável juntamente com o desenvolvimento de novas abordagem de energia limpa (TOMLINSON et al., 2023; TOMLINSON; TORRANCE; RIPPLE, 2023).

1.2 Motivação

Uma potencial aplicação de IA se baseia em modelos de fundação em linguagem natural de grande porte (*Large Language Models* - LLMs) como um componente básico na construção de softwares generalizáveis e adaptáveis. No qual a escolha destes modelos se justifica pela interpretabilidade e riqueza da linguagem humana em descrever tarefas complexas e abstratas por meio das palavras ou programas, facilitando a comunicação do usuário entre diversas aplicações sob a mesma interface no qual o LLM atua como orquestrador. Alguns estudos tem demonstrado que estes modelos tem fortes habilidades em adaptação para a inicialização em diversos domínios, dentre algum deles, implementações como no controle de tomadas de decisões de tarefas robóticas em sistemas físicos ou virtuais (BROHAN et al., 2023; DING et al., 2023; XIE et al., 2023b). Dessa forma, estas

aplicações potencializam a criação de Agentes de Inteligência Artificial que se aproximam cada vez mais da chamada Inteligência Artificial Geral (*Artificial General Intelligence* - AGI), promovendo avanços significativos em várias áreas de pesquisa e de aplicação.

A compreensão profunda de representações da linguagem humana por partes destes modelos emerge como um fator essencial para impulsionar o desenvolvimento de sistemas de IA mais avançados, no qual estes modelos, em particular o GPT-4 da openAI (OPENAI, 2023), demonstra habilidades impressionantes comparável a do humano em jogos interpretativos, dentre eles, a teoria da mente que é uma das teorias para medir se o ser humano é capaz de atribuir representações independentes a si mesmo e aos outros, no qual implica a habilidade de compreender e atribuir estados mentais, como crenças, emoções e intenções, a si mesmo e aos outros, contribuindo para a previsão de comportamentos sociais mais complexos (GANDHI et al., 2023; BUBECK et al., 2023).

No entanto, à medida que os sistemas baseados em inteligência artificial (IA) ganham espaço, com os LLMs atuando como os orquestradores entre a tarefa e o usuário, surge um desafio: o aumento das interações necessárias para produzir uma resposta desejada. Isso pode ocorrer por meio de scripts complexos ou chamadas a outros modelos, muitas vezes de maneira recursiva, com o objetivo de adquirir novas capacidades para a execução de uma tarefa (PACKER et al., 2023; SHEN et al., 2023; XI et al., 2023b). Diante desse cenário, tornou-se crucial explorar estratégias de otimização para minimizar tanto a quantidade de operações de ponto flutuante (FLOPs) quanto as emissões de CO2 associadas, visando à escalabilidade desses sistemas. Essa abordagem é essencial desde dispositivos cotidianos até servidores especializados, promovendo a sustentabilidade, e consequentemente, impulsionando uma revolução tecnológica.

Ao enfrentarmos os desafios da sustentabilidade na era da IA, devemos garantir que as maravilhas trazidas por essa tecnologia sejam aproveitadas de maneira responsável e equitativa. A implementação de regulamentações adequadas e a adoção de uma governança ética são fundamentais para garantir que os benefícios da IA sejam compartilhados por todos, evitando abusos e usos indevidos. Ao fazê-lo, estaremos no caminho certo para maximizar as capacidades dos modelos de IA atualmente disponíveis, ao mesmo tempo em que nos preocupamos com os impactos ambientais e sociais. Essas conquistas demonstram o enorme potencial da IA para melhorar nossa qualidade de vida e enfrentar desafios complexos.

Este trabalho tem como proposta impulsionar uma abordagem mais sustentável no desenvolvimento de sistemas de IA avançados, aplicando métodos de otimização em redes neurais profundas e promovendo, por meio de técnicas e estratégias, a redução do

consumo de recursos computacionais e a mitigação do impacto ambiental para a disseminação responsável desta tecnologia emergente, trazendo benefícios para a sociedade como um todo.

1.3 Objetivos do Trabalho

O objetivo deste trabalho é investigar e avaliar diversas técnicas e metodologias para otimização e aprimoramento de modelos de aprendizado profundo, com ênfase na sua aplicação em bancos de dados de revisões sistemáticas da literatura. Além disso, busca-se explorar a viabilidade de realizar a triagem de documentos relevantes em pesquisas científicas utilizando uma quantidade reduzida de exemplos por classe. O estudo visa analisar a eficácia dessas técnicas em diferentes conjuntos de dados, realizando análises qualitativas e quantitativas para identificar insights importantes sobre o desempenho e a adaptabilidade dos modelos base. Adicionalmente, busca-se destacar a importância de um ecossistema diversificado de habilidades e aplicações sustentáveis, visando benefícios tanto para a sociedade quanto para o meio ambiente. O objetivo final é promover diretrizes para o avanços tecnológicos de forma sustentável, contribuindo para o desenvolvimento de soluções baseadas em inteligência artificial que atendam às necessidades da sociedade de maneira eficaz e responsável.

2 Fundamentação Teórica

2.1 Aprendizado Profundo

No campo da Inteligência Artificial (IA), o Aprendizado Profundo (*Deep Learning*) é um subcampo de estudo desta área que se concentra na pesquisa de arquiteturas de modelos baseados em Redes Neurais Artificiais (RNAs) e em seu treinamento. Atualmente, o aprendizado profundo é uma das principais abordagens utilizadas para compreender padrões complexos como a linguagem humana a partir de dados de entrada. No entanto, a medida que estes algoritmos evoluem, o conceito de "inteligência artificial"torna-se cada vez mais subjetivo à medida que as máquinas se tornam capazes de realizar tarefas complexas tão bem quanto, ou até melhor do que especialistas em determinadas áreas (KIELA et al., 2021; SINGHAL et al., 2023; JUMPER et al., 2021; ROMBACH et al., 2021).

Dado este contexto, para compreender um sistema de processamento de informações, como a IA, consideramos três níveis de análise (MARR, 2010):

- **Nível de Teoria Computacional:** Corresponde ao objetivo da computação, fornecendo uma definição abstrata da tarefa.
- **Nível de Representação e Algoritmo:** Determina como a entrada e a saída são representadas e como o algoritmo transforma a entrada em saída.
- **Nível de Hardware:** Refere-se à implementação física real do sistema.

Nesta seção, abordaremos os princípios básicos dos modelos de Redes Neurais Artificiais (RNAs) e seu processo de treinamento. Em seguida, exploraremos a arquitetura Transformers, que ganhou destaque nos últimos anos (DOSOVITSKIY et al., 2020; DEVLIN et al., 2018; BROWN et al., 2020; ROMBACH et al., 2021).

2.1.1 Redes Neurais Artificiais

O trabalho pioneiro de McCulloch e Pitts (1943) representou a primeira abordagem na criação de modelos de RNA ou simplesmente redes neurais. Seu objetivo era modelar as redes neurais biológicas, buscando compreender e simular processos cognitivos biológicos. Esse trabalho foi fundamental para a pesquisa em redes neurais, dividindo o campo em duas vertentes principais: uma voltada para a modelagem dos processos

biológicos no cérebro e a outra direcionada para a aplicação das redes neurais no campo da inteligência artificial. Em sua essência, as RNAs são, na maioria dos casos, consideradas modelos não paramétricos aproximadores universais de funções (CSÁJI et al., 2001). Isso implica que, ao utilizar RNAs, é viável aproximar qualquer função, desde que os pesos adequados sejam aplicados à tarefa em questão. Ou seja, estes modelos possuem a notável capacidade de mapear desde funções simples, como uma reta, até funções complexas, como a linguagem humana (DEVLIN et al., 2018; RADFORD et al., 2018).

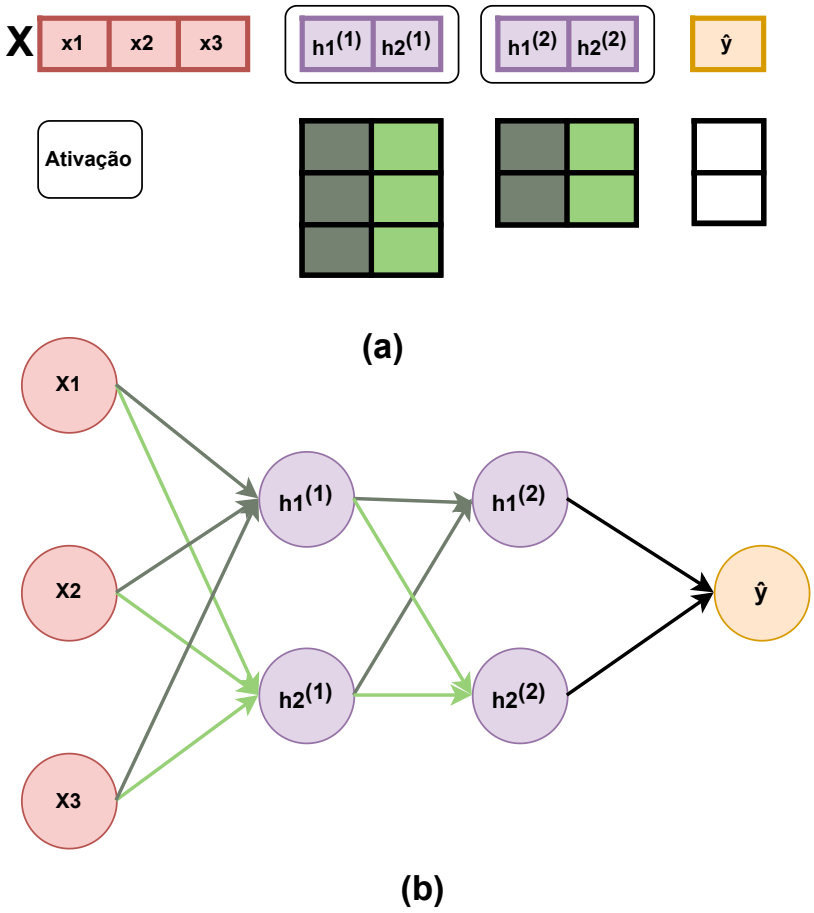

Figura 1: *Multilayer Perceptron.*

O modelo Perceptron proposto por Rosenblatt (1958), é uma versão aprimorada do primeiro modelo proposto por McCulloch e Pitts (1943). No entanto, uma das principais características mais marcantes deste modelo, é sua limitação quando se trata de problemas mais complexos, como no caso de uma classificação que não podem ser separados linearmente demonstrado pelo famoso problema XOR proposto por Minsky e Papert (1969) em seu livro "*Perceptrons: An Introduction to Computational Geometry*"no qual esse desafio levou a uma desmotivação na pesquisa de redes neurais por cerca de 20 anos. Logo, para $x \in \mathbb{R}^d$ como sendo o dado de entrada e $y \in \mathbb{R}$ o dado de saída, o perceptron

pode ser representado por:

$$y = \phi\left(\sum_{j=1}^{d} x_j w_j + w_0\right) = \phi(xW + w_0) \tag{1}$$

no qual este modelo é uma unidade de processamento básico que recebe um conjunto de entradas ponderadas por pesos w e aplica uma função de ativação, denotado por $\phi(.)$, para produzir uma saída.

O *Multilayer Perceptron* (MLP) proposto por Rumelhart, Hinton e Williams (1986), foi desenvolvido para superar essas limitações e expandir as capacidades do perceptron tradicional, no qual impulsionou significativamente a pesquisa em redes neurais nas últimas três décadas. Esta arquitetura é composta por várias camadas de perceptrons interconectados, no qual cada camada recebe as saídas dos perceptrons da camada anterior como entrada e seguidas de funções de ativação para transformar essas entradas.

Por exemplo, consideremos um MLP de 2 camadas com uma entrada $x \in \mathbb{R}^{1\times d}$, onde d é o número de variáveis explicativas. As camadas do MLP, sem considerar o intercepto, são representadas pelos pesos $W^{(K)} \in \mathbb{R}^{d\times k}$, onde K é o índice da camada, k é o número de neurônios na camada oculta e d é o número de variáveis de entrada da camada anterior (nesse caso, a camada zero, com os valores de entrada $h^{(0)} = x$). Para ilustrar, consideremos $d = 3$ e $k = 2$. As saídas das duas camadas ocultas são representadas por $h^{(1)}, h^{(2)} \in \mathbb{R}^{1\times 2}$. Logo, na Figura 1, apresentamos uma visualização do fluxo matricial do modelo (a) e seu correspondente fluxograma neural (b). A expressão matemática para esse exemplo pode ser denotada por:

$$\begin{aligned}
\phi_1(xW^{(1)}) &= h^{(1)} \\
\phi_2(h^{(1)}W^{(2)}) &= h^{(2)} \\
\phi_3(h^{(2)}W^{(3)}) &= \hat{y}
\end{aligned}$$

no qual é possível estabelecer uma representação mais compacta, onde $f_W(x) = \hat{y}$, em que $W = \{W^{(k)}\}_{k=1}^{3}$ representa o conjunto de pesos do modelo, juntamente com suas respectivas funções de ativação e os pesos das camadas ocultas.

Para os MLPs, as funções de ativação $\phi(.)$ desempenham um papel crucial para a área de pesquisa em redes neurais. Estas transformações geralmente são funções não lineares, com o objetivo de fornecer representações mais complexas para o modelo entre as camadas. Algumas das funções mais comuns na literatura são monotonamente crescentes, conforme demonstrado na Figura 2. No qual para o exemplo mencionado, $\phi_3(\cdot)$ é uma

função identidade ($f(x) = x$), usada na última camada, o que permite modelar uma regressão não linear. A escolha dessas funções de ativação depende do nível de representação desejado pelo pesquisador e à otimização do treinamento.

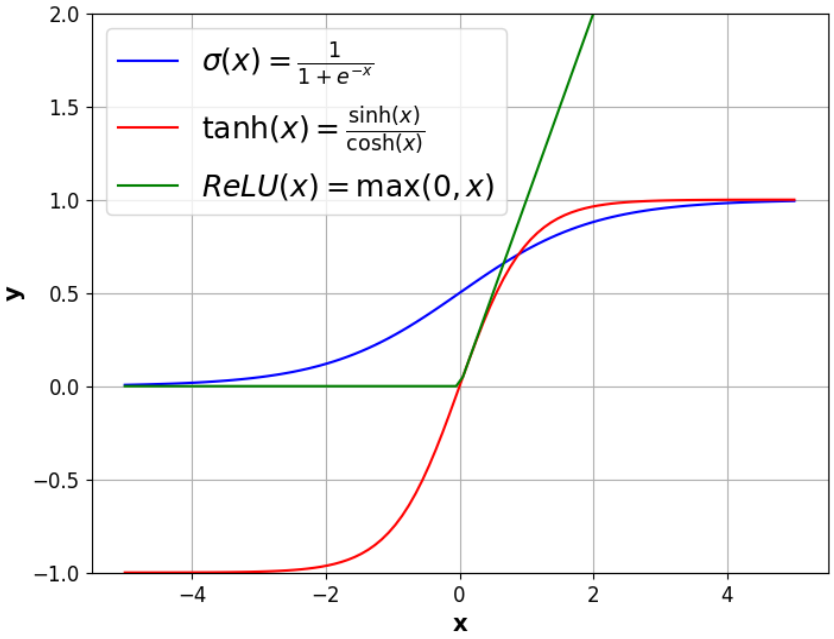

Figura 2: Funções de Ativação.

2.1.2 Encoder-Decoder

As arquiteturas de redes neurais, em sua essência, podem ser concebidas com estruturas que incluem elementos como o *Encoder* e o *Decoder*. Essas estruturas desempenham um papel crucial no processamento de informações, permitindo a extração ou geração de padrões a partir dos dados.

O *Encoder* desempenha um papel crucial na extração das informações mais importantes do dado de entrada, visando reduzir o ruído e características menos relevantes. Sua responsabilidade é transformar os valores de entrada para extrair características relevantes e condensá-las em um vetor denso como contexto, ou também conhecido como representação latente.

Essa representação latente facilita a manipulação e interpretação por parte do modelo. O *Decoder*, por sua vez, desempenha um papel inverso ao do *Encoder*, utilizando este vetor latente resultante como contexto para reconstruir o dado original sem ruído ou gerar uma saída relevante, preenchendo detalhes e personalizando a saída de acordo com a tarefa específica em questão.

Essa abordagem é altamente robusta, pois permite a modulação de modelos e a criação de um sistema de processamento de informações capaz de capturar relações essenciais e complexas, resultando em resultados de alta qualidade. A capacidade de ajustar o vetor de contexto para atender a tarefas específicas de maneira precisa e eficaz é fundamental. Isso se torna um dos elementos essenciais na construção de modelos mais complexos, contribuindo para avanços significativos em uma ampla gama de aplicações que envolvem relações complexas e abstratas, como na geração de imagens condicionada ao texto (ROMBACH et al., 2021).

2.1.3 Aprendizado de Máquina

O aprendizado de máquina é um paradigma de treinamento que permite que modelos aprendam a representação de dados para realizar tarefas específicas. Esse campo é dividido em várias áreas de pesquisa, das quais as principais são categorizadas da seguinte forma:

- **Aprendizagem Supervisionada:** Neste tipo de aprendizagem, o modelo é treinado com um conjunto de dados rotulados, onde a resposta desejada é conhecida. O objetivo é ensinar o modelo a mapear os dados de entrada para as saídas correspondentes. Por exemplo, na classificação de e-mails, os textos dos e-mails servem como entrada do modelo, e a saída esperada é classificá-los como “spam”ou “não spam”.

- **Aprendizagem Não Supervisionada:** Aqui, o modelo recebe dados de treinamento não rotulados e tenta identificar estruturas, padrões ou agrupamentos nos dados. Os rótulos podem ser os próprios dados de entrada ou partes deles. Por exemplo, o modelo pode receber um texto com algumas palavras faltando, como na frase: “O Aprendizado de [x] é essencial para dar representações ao modelo”, onde a saída esperada em [x] seria a palavra ”Máquina”.

- **Aprendizagem de Reforço:** Nesse cenário, o modelo interage com um ambiente e toma ações para maximizar uma recompensa cumulativa. O objetivo é aprender uma política que guie as ações para otimizar as recompensas ao longo do tempo. Exemplo: Treinar um modelo para jogar xadrez, onde as ações corretas resultam em vitórias, e as recompensas ao longo do trajeto da partida modelam o pensamento do modelo.

No exemplo mencionado de Aprendizagem Não Supervisionada, também conhe-

cido como Aprendizagem Alto Supervisionada, é comumente utilizado para pré-treinar um modelo, visando proporcionar uma compreensão das estruturas e padrões gerais dos dados. Posteriormente, esses pesos pré-treinados facilita o aprendizado em tarefas mais específicas através do ajuste fino, como um processo conhecido de Transferência de Aprendizado. Essa abordagem em particular é notável, ao permitir que um modelo compreenda a estrutura dos dados de forma autônoma, e em seguida, utilize desta compreensão para modelar uma tarefa mais específica.

Nos últimos anos, essa estratégia de pré-treinamento por meio da Aprendizagem Alto Supervisionada tornou-se popular na criação de modelos base, ou também conhecidos como modelos de fundação, aproveitando o conhecimento prévio adquirido, para permitir a construção de modelos mais complexos (ROMBACH et al., 2021; LI et al., 2022; LI et al., 2023; SHEN et al., 2023).

2.1.4 Treinamento

O treinamento de uma RNA desenvolve a capacidade do modelo em compreender a representação dos dados fornecidos como valores de entrada. Isso é alcançado por meio do ajuste iterativo dos pesos, baseado em uma ou mais funções de perda desejadas, capacitando o modelo com as habilidades necessárias para executar uma determinada tarefa. A escolha apropriada da função de perda depende da natureza da tarefa de aprendizado e desempenha um papel fundamental na capacidade do modelo de aprender e generalizar a partir dos dados de treinamento.

Para ilustrar, consideremos a abordagem de atualização em modo Mini-Batch, amplamente reconhecida por sua eficiência na generalização e velocidade de treinamento. Denotamos por $X_T = \{X_k\}_{k=1}^{N}$ os dados de treino, onde $X_k = \{(x_i, y_i)\}_{i=1}^{n}$ são os mini-lotes de exemplos (*mini-batch*) de tamanho n da variável explicativa x_i e da variável resposta y_i. Agora, suponha o modelo $f_W(.)$, com uma função de perda em função dos parâmetros dada por $L_W = L(y_i, f_W(x_i))$. O erro médio da previsão do mini-lote X_k é dado por $E_{X_k}[L_W] = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f_W(x_i))$.

Para aproximar os valores dos pesos que minimizam a perda, representados por $\hat{W} = \arg\min_W E_{X_k}[L_W]$, o modelo aprende de forma iterativa atualizando os parâmetros por meio da descida do gradiente (ou *backpropagation*), conforme a expressão:

$$\begin{aligned} \nabla W &= \nabla_W E_{X_k}[L_W] \\ W &:= W - \eta \nabla W \end{aligned} \tag{2}$$

Assim, a principal diferença entre os métodos de atualização reside na construção dos mini-lotes X_T e no momento em que é computada a atualização dos parâmetros, considerando uma época ao ter passado por todos os exemplos estruturados em X_T.

Esse processo é repetido iterativamente para minimizar o erro ao longo das atualizações, utilizando uma taxa de aprendizagem ($\eta \in (0, 1)$) como hiperparâmetro para controlar o tamanho dos passos de atualização. Na Figura 3, é demonstrado um exemplo da geometria do espaço da função de perda em relação aos parâmetros, com base no conjunto de dados e no valor esperado. Os parâmetros são inicializados aleatoriamente, e o modelo, de forma iterativa conforme o trajeto em verde, percorre a superfície até encontrar uma combinação de pesos que minimize o erro.

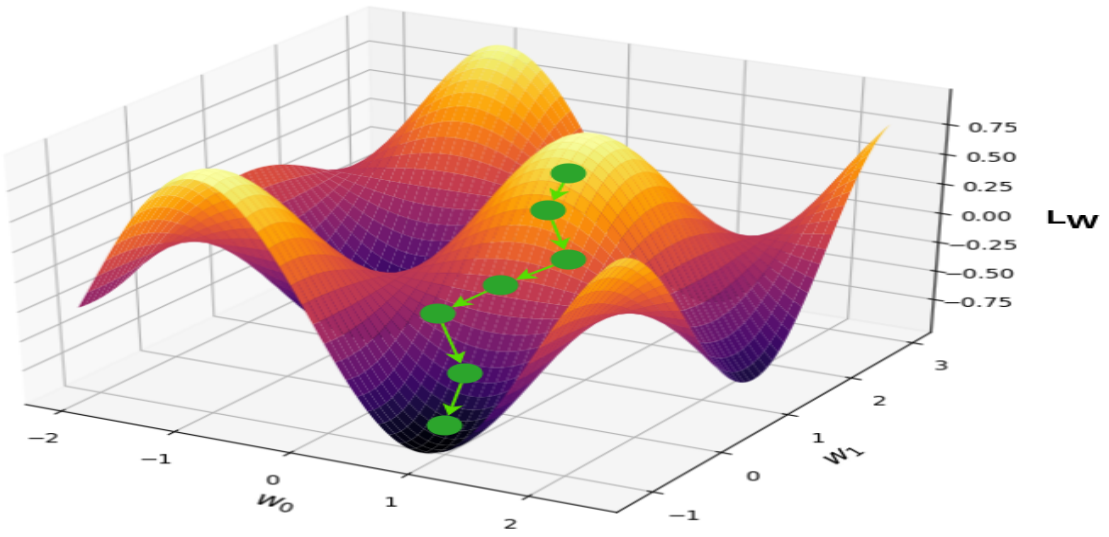

Figura 3: Descida do gradiente.

2.1.5 Arquiteturas dos Transformers

As arquiteturas dos Transformers, introduzida em 2017 por Vaswani et al. (2017), representa uma inovação significativa no campo do Aprendizado de Máquina, destacando-se pela abordagem inovadora do mecanismo de atenção que permite capturar relações complexas entre elementos dos valores de entrada e processar sequências de forma paralela. No qual ao treiná-las de forma auto-supervisionada, este resulta em modelos base robustos e inteligentes, acelerando o aprendizado de relações intrínsecas e proporcionando um ecossistema de modelos base em diversas áreas e impulsionando descobertas e aplicações com o uso de técnicas de aprendizado por transferência. Nesta seção, exploraremos essa arquitetura, compreendendo suas principais características que fazem dos Transformers uma escolha inspiradora.

2.1.5.1 Embedding

Os Embeddings são amplamente utilizados devido à sua capacidade de representar sequências de entrada em vetores que possuem relações espaciais. Sua formulação é através da representação de um vocabulário de símbolos, que ao passar por uma função de dicionário, retorna os vetores correspondentes para cada simbolo. Este após treinado, possui a habilidade de codificar uma relação simbólica espacial, no qual as dimensões de seus vetores pode ser considerado como características representacionais entre os símbolos do dicionário, tornando possível operações lógicas e simbólicas, modelada por um modelo.

Para ilustrar, tomando como símbolos as palavras, na Figura 4, temos como exemplo animais de mesma família próximos um do outro, e operação lineares simbólicas, como "rei - homem + mulher = rainha", tornando estas representações uma ferramenta versátil e poderosa para uma ampla gama de aplicações.

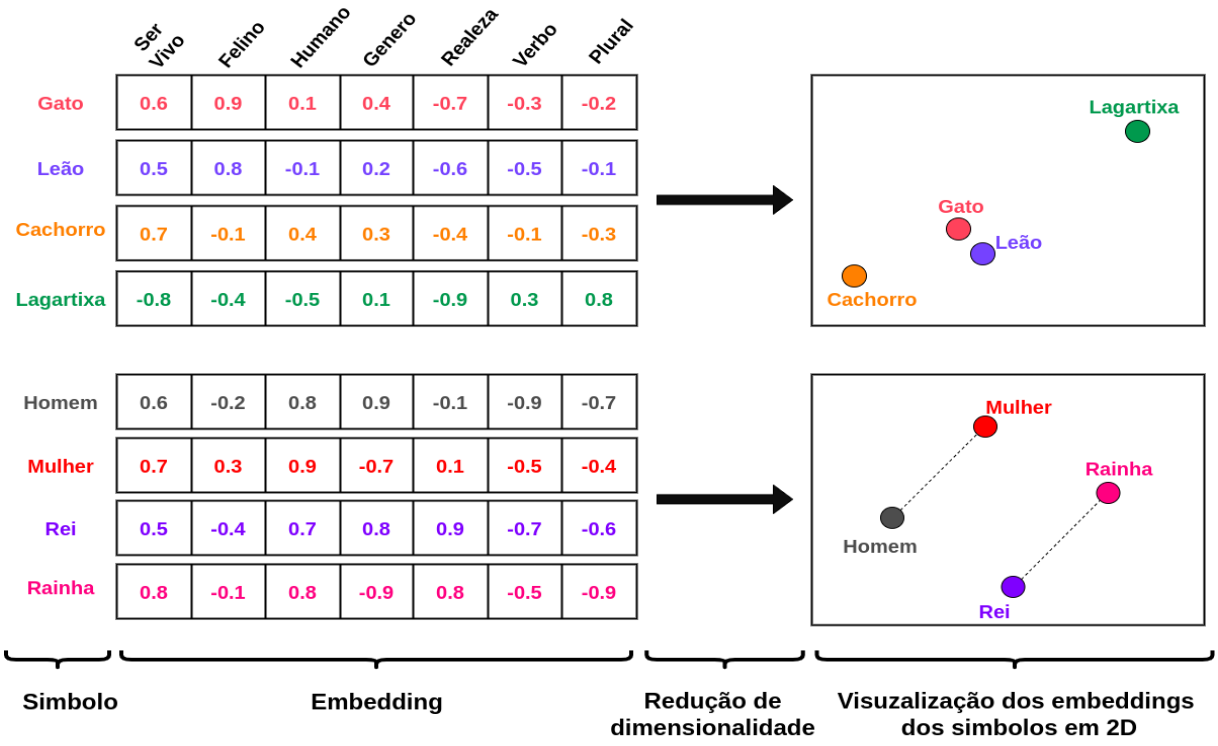

Figura 4: Representação dos Embeddings.

No entanto, é crucial observar que, ao introduzir uma sequência de vetores para um modelo MLP, dependendo da implementação, a estrutura do modelo pode não considerar a posição dos valores na sequência em suas dimensões. Isso implica que as informações na sequência podem ser tratadas como pontos no espaço, sem codificação da ordem sequencial. Essa abordagem pode ser problemática em situações específicas, como no contexto de

frases, onde a ordem das palavras é fundamental para uma compreensão completa do significado. Além disso, é importante ressaltar que na computação e representação neural do modelo MLP para uma sequência de embeddings, como exemplificado por $(X1, X2, X3)$ com duas dimensões, conforme ilustrado na Figura 5, os pesos dos neurônios interagem diretamente com as dimensões dos valores de entrada (b), porém de forma independente em relação à sequência, assemelhando-se a uma operação de convolução dos pesos sobre a sequência de vetores.

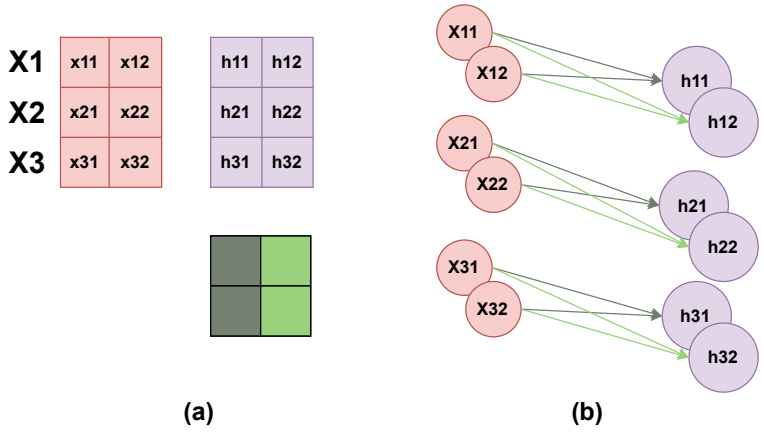

Figura 5: Computação Neural de uma sequência.

2.1.5.2 Atenção

O mecanismo de atenção proposto por Vaswani et al. (2017), tem se demonstrado um componente fundamental nas arquiteturas de aprendizado profundo. Esse mecanismo permite que o modelo dê mais importância a determinadas partes do dado de entrada, focando em informações relevantes para a tarefa em questão. Assim, contextualizando os vetores de entrada passados como $X \in \mathbb{R}^{1\times d}$. O mecanismo envolve a criação de três representações vetoriais iniciais: *Query* (Q), *Key* (K) e *Value* (V). No qual essas representações são geradas por meio de projeções lineares do valor de entrada X, por meio dos parâmetros $W_Q, W_K, W_V \in \mathbb{R}^{n\times k}$ aprendidos durante o treinamento do modelo. Essas etapa é expressas como

$$Q = xW_Q, \quad K = xW_K, \quad V = xW_V \tag{3}$$

assim, se $X^T = [x_1 \ ... \ x_n] \in \mathbb{R}^{d\times n}$ onde $x \subset E$, é uma sequência de n vetores indexados no espaço representacional E, geralmente representada pelo mapeamento dos Embeddings, então $Q, K, V \in \mathbb{R}^{n\times k}$ possui n vetores projetados neste mesmo espaço E para capturar representações abstratas de cada vetor da sequência X.

Assim, o mecanismo de atenção denotado por $Attention_X(Q, K, V)$, pode ser representado como:

$$\alpha = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right),\ Attention_X(Q, K, V) = \alpha V \tag{4}$$

Nessa formulação, a matriz estocástica α é obtida por meio de uma normalização exponencial utilizando a função $softmax(x) = \exp\{x_i\}/\sum_{k=1}^{n}\exp\{x_k\}$, aplicada sobre a matriz de covariância entre as representações dos símbolos, tratados como vetores, representados por $QK^T \in \mathbb{R}^{n\times n}$.

Em seguida, essa matriz α é multiplicada com as representações dadas por V, resultando em $\alpha V \in \mathbb{R}^{n\times k}$. Esse processo visa utilizar as magnitudes das similaridades entre cada par de representações vetoriais como uma soma ponderada entre os vetores representados por V. O valor d_k representa o número de dimensões do vetor K e funciona como uma constante de normalização para a matriz de covariância. Isso garante que os gradientes sejam mais estáveis durante o treinamento e mantém os valores próximos do intervalo $[-1, 1]$

Em suma, a função primordial do mecanismo de atenção é empregar os "sentidos"espaciais abstratos dos símbolos de entrada, representados por Q, K, e V e através da magnitude da similaridade entre eles, denotada por α, esses sentidos são ponderados para influenciar a relação entre os sentidos resultantes em V, resultando em uma sequência de novos símbolos no espaço representacional E. Essa abordagem possibilita a geração de novas representações do conteúdo em X, capturando as relações entre os sentidos da sequência e produzindo uma nova sequência de símbolos abstratos, que podem ou não estar dentro de vocabulário do dicionário inicial.

Assim, esse mecanismo de atenção oferece uma maneira eficaz de capturar e gerar informações relevantes, ao mesmo tempo em que pondera a importância de diferentes elementos de entrada. Contudo, uma desvantagem computacional é o cálculo do produto externo entre as matrizes Q e K, que pode se tornar computacionalmente custoso devido ao tamanho da sequência de entrada. No entanto, estudos recentes demonstram que o mecanismo de atenção pode armazenar padrões de contexto de forma exponencialmente eficiente usando métodos simples de aprendizado interativo, como nas chamadas *hopfield networks* (RAMSAUER et al., 2020).

2.1.5.3 Transformers

Os Transformers são basicamente a generalização do mecanismo de Atenção, empregadas como arquiteturas de Encoder-Decoder. No qual o principal mecanismo chamado de *Multi-Head Attention* ($MHA(Q, K, V)$), utiliza de múltiplos mecanismos de atenção em paralelo para criar várias representações contextualizadas sobre a sequência de entrada.

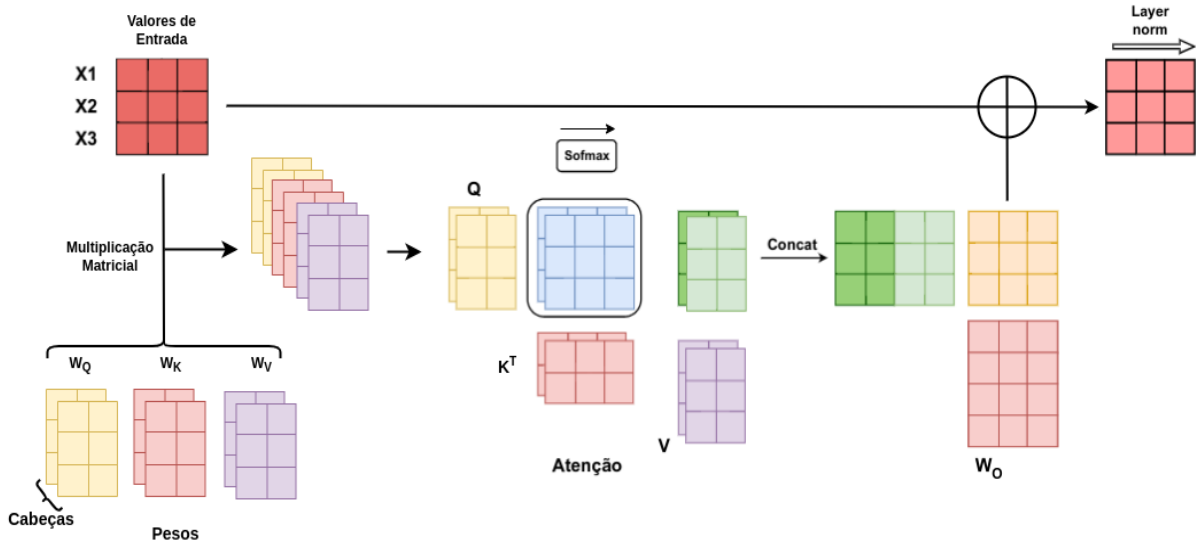

Figura 6: Representação computacional do *Multi-Head Attention*, com uma sequencia de 3 entradas (X1,X2,X3), 2 cabeças e 2 dimensões para Q,K e V.

Para ilustrar o processo deste mecanismo, podemos visualizar a computação por meio da Figura 6. Em um exemplo de *Multi-Head Attention* com duas cabeças, que ao fornecer uma sequência de 3 valores de entrada $X = (X1, X2, X3)$, cada cabeça cria suas respectivas representações Q, K e V. Em seguida, são mapas de correlação α para cada cabeça, os quais são utilizados para ponderar seus respectivos V. Esses resultados são então concatenados e passados por uma transformação linear dada por W_O, sendo então somados à sequência original ($X_{MHA} = X + MHA(Q, K, V)$), processo conhecido como conexão residual (HE et al., 2015). Posteriormente, é aplicada a técnica de *Layer Normalization* (BA; KIROS; HINTON, 2016), onde cada vetor da sequência X é normalizado, cujo o papel junto a conexão residual, é estabilizar os gradientes.

Esse mecanismo também descrito na imagem (B) da Figura 7, é um dos módulos para construir apenas um bloco de camada Transformer, no qual a saída após a normalização, é passada por duas camadas de MLP, com uma ativação RELU entre elas, que constitui como a única não linearidade introduzida pelo modelo. Estas camadas são co-

nhecidas como *Feed Forward* ($f_W(.)$). Em seguida, é aplicado uma conexão residual e a *Layer Normalization*.

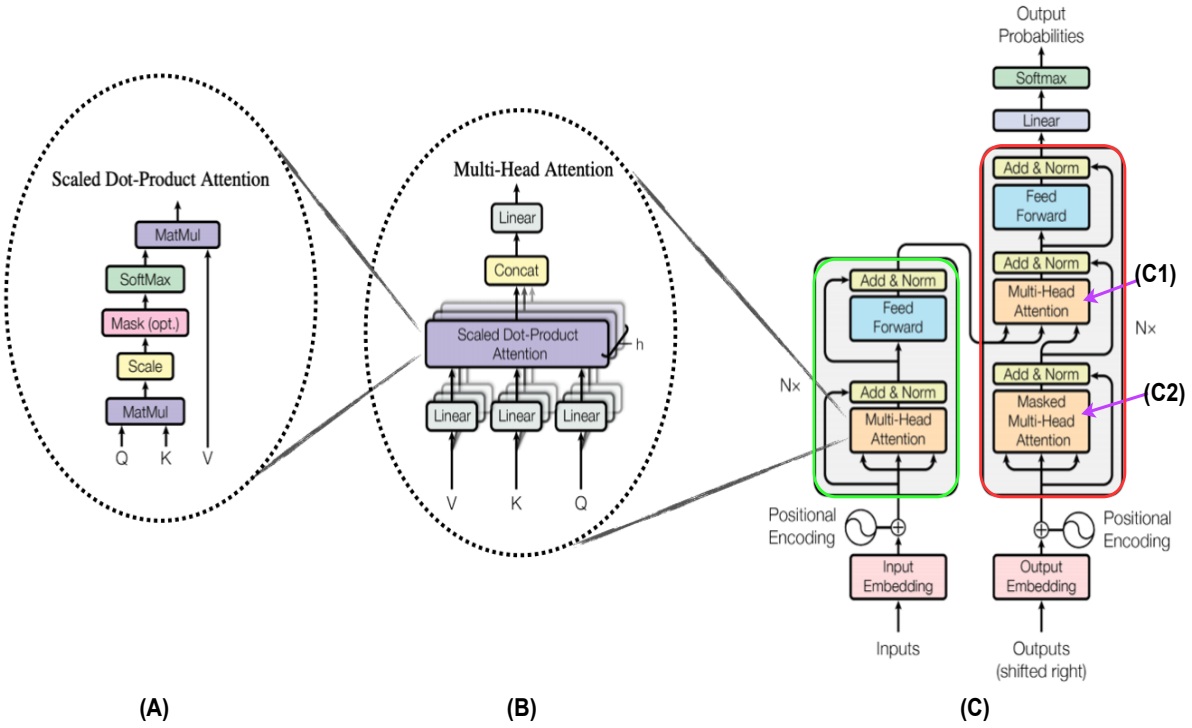

Figura 7: Transformers (VASWANI et al., 2017).

O bloco Transformer é empilhado N vezes para proporcionar representações mais profundas para os valores de entrada. Esta arquitetura proporciona vetores que apontam para regiões no espaço dos embeddings e representam significados simbólicos e abstratos complexos para o domínio treinado.

Dentro deste contexto, ocorrem pequenas modificações na arquitetura do bloco para o Encoder e o Decoder. Na Figura 7, representada na imagem (C), a parte à esquerda em verde retrata o Encoder. Ao introduzir uma camada Linear seguida de normalização softmax na sequência de saída do Transformer, obtemos uma estrutura semelhante ao modelo BERT (*Bidirectional Encoder Representations for Transformers*), conforme proposto por Devlin et al. (2018).

Na parte à direita em vermelho, sem o módulo de conexão (C1), seguida pela mesma configuração de saída descrita para o Encoder, obtemos uma representação do Decoder que ganhou bastante destaque nos últimos anos, conhecido também como GPT (*Generative Pre-trained Transformer*), conforme proposto por Radford e Narasimhan (2018). Sua capacidade de gerar sequências de texto de forma auto-regressiva é notável. A interpretação dessa arquitetura pode ser comparada à uma cadeia de Markov sobre

um dicionário de palavras, formulada como uma densidade de probabilidade condicional expressa por $p_W(X_k|X_{k-1}, ..., X_1)$, em que X_k representa a densidade neste vocabulário.

A principal diferença entre o componente *Multi-Head Attention* e o *Masked Multi-Head Attention* (Masked MHA) em (C2) reside apenas na computação da matriz de similaridade α. No Masked MHA, as similaridades são geradas apenas com a sequência de entrada anterior. Em outras palavras, os vetores de uma sequência são comparados apenas com seus antecessores, resultando em uma matriz com uma diagonal superior de valores zeros. Esse arranjo garante que o modelo preste atenção apenas aos valores anteriores ao último elemento de entrada na sequência.

Na versão original do Transformers, conforme ilustrado na Figura 7 (C), o foco reside na interligação dos modelos Encoder e Decoder por meio do *Cross Attention* em (C1). Essa atenção funciona como um mecanismo em que o modelo Encoder fornece referências e o Decoder as modifica para gerar novos resultados. A computação ocorre de maneira semelhante ao MHA, porém, a principal diferença é que os valores K e V são projeções da saída do Encoder, enquanto o Q é a projeção dos valores de entrada do próprio Decoder.

A entrada desses modelos geralmente é representada por embeddings, que lidam com a falta de noção de temporalidade sequencial adicionando os *Positional Embeddings*. Esses embeddings atribuem informações de posição aos vetores de entrada com base na posição de cada vetor na sequência. Na formulação original, os *Positional Embeddings* são representados como $PE_{(pos,2i)} = sin(pos/10000^{2i/d})$, onde pos é o índice do vetor na sequência, d é o tamanho do vetor, e i é o índice de sua respectiva dimensão (VASWANI et al., 2017).

Os modelos baseados nestas arquiteturas dos Transformers, são geralmente pré-treinados de forma auto-supervisionada, no qual para o Encoder, este costuma utilizar o mascaramento de partes da entrada a induzir o modelo a prever estas partes, enquanto o Decoder geralmente é treinado para prever o próximo valor da sequência. Esta fase de treinamento utiliza de grandes quantidades de dados e recursos computacionais, que recorre a hardwares especializados como GPUs para acelerar o tempo de processamento.

Apesar do mecanismo *Multi-Head Attention* ser eficaz na captura de relações semânticas e seu pré-treinamento tornando possível modelos base, sua aplicação pode ser custosa devido aos cálculos paralelos das matrizes de covariância α, e seu grande número de parâmetros, o que limita a eficiência e escalabilidade. No entanto, na próxima seção, discutiremos técnicas avançadas que podem equilibrar a inovação tecnológica e a eficiência

computacional, priorizando a sustentabilidade e a viabilidade no uso diário.

2.2 Otimização de Redes Neurais Profundas

A otimização de Redes Neurais Profundas representa um avanço significativo no campo da inteligência artificial, permitindo desbloquear o potencial revolucionário desta tecnologia para diversas aplicações. Esta metodologia não se limita apenas à melhoria da velocidade de treinamento e inferência de modelos, mas também contribui para a redução da pegada de carbono e democratização dessa tecnologia em um mundo cada vez mais dependente de sistemas de aprendizado de máquina.

Nesta seção, serão exploradas as principais estratégias de otimização para Redes Neurais Profundas, buscando soluções para aprimorar sistemas baseados em IA.

2.2.1 Quantização

Na era digital, a computação é fundamentada no uso de transistores e operações em números binários para representar funções matematicamente contínuas, exigindo a alocação de bits para garantir estimativas precisas. Tradicionalmente, a computação de operações é em FLOAT32 e oferece uma precisão numérica dentro dos limites de 1e-38 a 3e38, mas consome mais espaço, energia e carga computacional devido ao uso de 32 bits por número. No entanto, estudos recentes introduziram a precisão BFLOAT16 (*Brain floating Point Format*) como uma alternativa, mantendo a mesma faixa de valores do FLOAT32, mas com apenas 16 bits, possibilitando benefícios significativos na computação e no treinamento de modelos de redes neurais (KALAMKAR et al., 2019). No entanto, para modelos mais complexos como os Transformers, que podem chegar a bilhões de parâmetros, estes ainda exigem grandes recursos computacionais. Assim, reduzir a precisão numérica torna os modelos mais escaláveis, possibilitando o uso em dispositivos com recursos limitados e ampliando suas aplicações.

A quantização é uma técnica que consiste em uma transformação de valores contínuos em discretos, que podem ser representados por uma quantidade menor em bits. Este é um campo de pesquisa amplo que possui uma variedade de abordagens direcionadas à arquiteturas ou tarefas específicas, com o objetivo de reduzir a quantidade de bits em diversos cenários de aplicação. Algumas das principais técnicas de quantização se estendem em metodologias de:

- **Quantização Pós-Treino (PQT)**: Esta técnica aplica a quantização após o trei-

namento do modelo.

- **Treinamento Consciente de Quantização (QAT)**: Nesta técnica o treinamento é realizado levando em consideração a quantização, restringindo e ajustando os parâmetros do modelo para otimizar o desempenho após o treino.

Dentre elas, existem duas abordagens principais na quantização: a dinâmica, na qual apenas os pesos do modelo são quantizados e desquantizados durante o cálculo matricial, e a estática, na qual tanto os pesos quanto os valores da ativação são quantizados, realizando as operações em baixa precisão. Independentemente da técnica de quantização utilizada, todas as abordagens partem de um mesmo princípio.

Tomando como referencia a quantização em 8 bits, ao iniciar uma quantização, este possui dois tipos de intervalos: a quantização assimétrica, na qual os valores dos pesos são transladados por uma constante; ou a quantização simétrica que possui a abordagem com sinal, na qual os valores estão entre -128 e 127, e a sem sinal, em que o intervalo está entre 0 e 255. É importante ressaltar que esses valores representam os $256 = 2^8$ números inteiros possíveis, correspondentes a quantidade de informação transmitida em 8 bits. A escolha deste intervalo é relativa ao tipo de entrada e ativação do modelo, cada um apresentando suas vantagens e desvantagens.

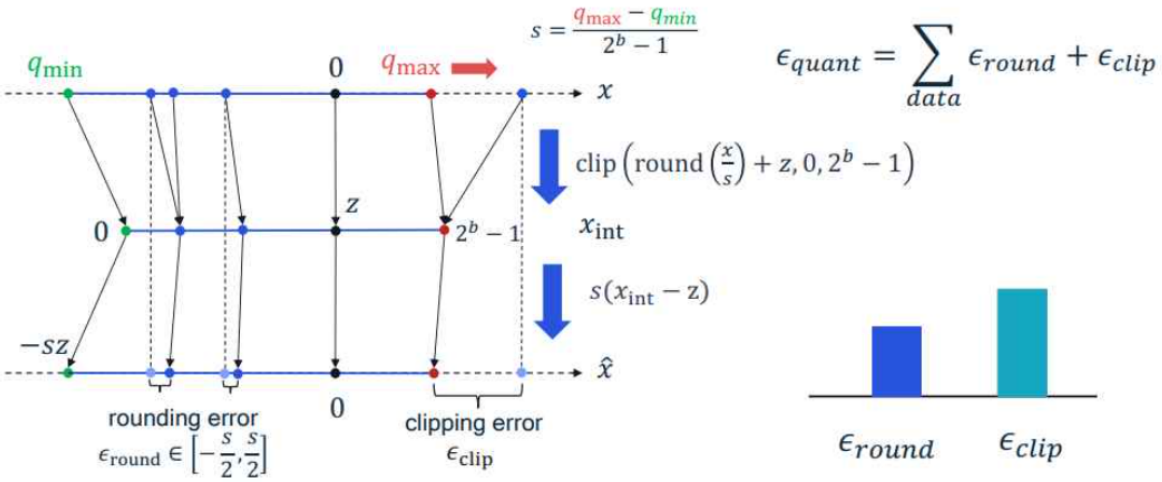

Figura 8: Erros da quantização (NAGEL et al., 2021; FOURNARAKIS, 2021).

Portanto, tomando como exemplo a quantização simétrica para 8 bits de uma matriz de pesos W, o procedimento inicial envolve a normalização dos valores, representada por W_N. Nesse contexto, a normalização é realizada utilizando o valor máximo absoluto, expresso como $w_{\max} = \max(|W|)$. Assim, definimos $s = 1/w_{\max}$, e obtemos $W_N = W \cdot s$,

onde o valor s atua como fator de reescalonamento para os valores da matriz original W. Em seguida, a matriz normalizada W_N é multiplicada pelo intervalo de quantização e arredondada para números inteiros. Para valores que excedem os limites do intervalo, aplicamos a função $clip(x, min = 0, max = 1) = max(min(x, 0), 1)$. Dessa forma, obtemos $W_{q8} = clip(\text{Round}(127 \cdot W_N), min = -128, max = 127)$, onde W_{q8} representa os valores inteiros quantizados em 8 bits. Para realizar a operação inversa de desquantização da matriz, utilizamos $X_{d8} = X_{q8}/(s \cdot 127)$.

Embora seja uma técnica simples para reduzir os custos computacionais, a performance do modelo em métricas pode ser prejudicada ao arredondar para valores inteiros mais próximos, como demonstrado na Figura 8. Outro desafio é a gestão de outliers, que influencia diretamente no valor escolhido para normalização, representado por s.

Estudos estão em curso para abordar eficientemente essas questões. Isso inclui métodos simples, como a quantização focalizada, que utiliza várias quantizações com base em padrões identificados na matriz, e abordagens mais avançadas baseadas no erro da tarefa para identificar valores mais recorrentes, ou a aplicação de operações mistas entre valores quantizados de forma mais harmônica e outliers mantidos na precisão original, buscando um equilíbrio entre precisão e eficiência (FRANTAR et al., 2022; LIN et al., 2023; DETTMERS et al., 2022).

2.2.2 Poda

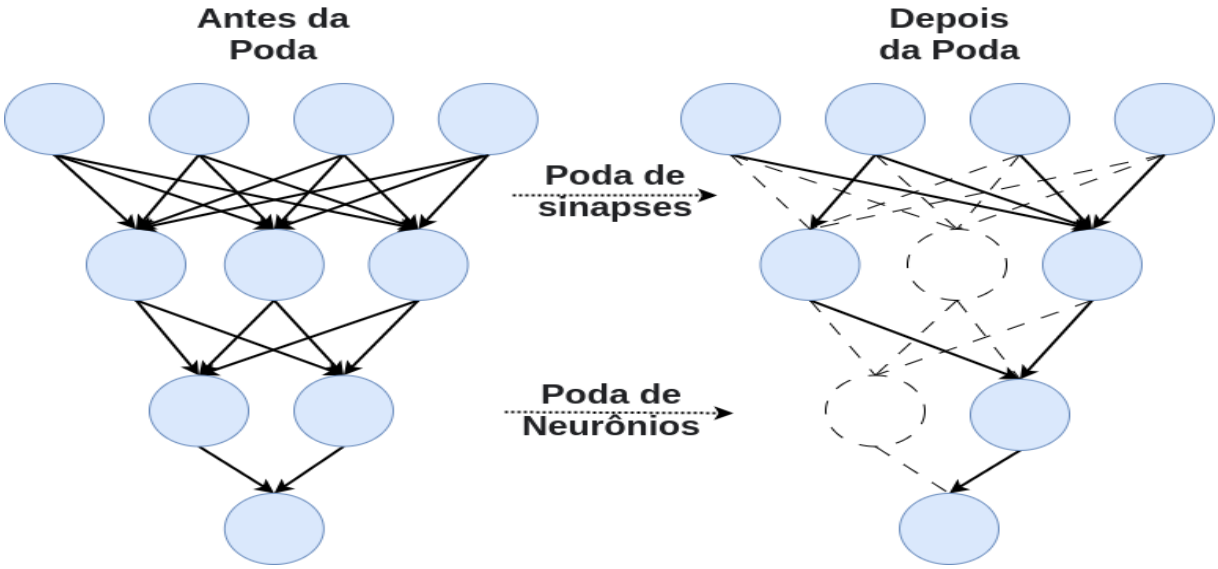

Figura 9: Poda.

A poda (*Pruning*) é uma técnica empregada em redes neurais profundas para reduzir o tamanho e a complexidade dos modelos, eliminando neurônios ou parâmetros (sinapses) menos relevantes, como demonstrado na Figura 9. Essa abordagem parte do princípio de que nem todas as conexões do modelo contribuem igualmente para seu desempenho, permitindo a remoção daquelas com impacto mínimo na saída final, o que promove a esparsidade do modelo.

O processo de poda geralmente compreende duas etapas principais: a primeira consiste na identificação dos neurônios ou parâmetros a serem removidos. A segunda etapa pode envolver o reajuste dos pesos restantes para compensar a perda das conexões podadas ou até mesmo a redefinição da arquitetura do modelo para acomodar a nova estrutura resultante. A relevância dos parâmetros pode ser avaliada de várias maneiras, incluindo a magnitude dos pesos, a sensibilidade dos gradientes ou o impacto na função de perda. Diferentes abordagens para a poda existem, cada uma com suas próprias estratégias e algoritmos específicos. Algumas metodologias principais incluem (KURTIC et al., 2022; LIEBENWEIN, 2021):

- **Poda estruturada**: Esta técnica visa remover conjuntos inteiros de conexões de uma só vez, em vez de conexões individuais como na poda não estruturada. Isso pode ser feito por meio de técnicas como poda por camada, poda por filtro ou poda por bloco. Essas abordagens consideram a arquitetura do modelo, permitindo a remoção de conjuntos de conexões interconectadas que têm um impacto mínimo no desempenho.
- **Poda de compressão**: Para modelos pré-treinados, esta abordagem apresenta dois métodos de poda: compressão ascendente e descendente. Ambas as abordagens realizam a poda durante o treinamento. A compressão ascendente é aplicada ao banco de dados de pré-treinamento, enquanto a compressão descendente é realizada em um banco de dados para uma tarefa específica, podendo resultar em maior esparsidade.

2.2.3 Destilação de Conhecimento

A Destilação de Conhecimento em redes neurais profundas é uma técnica que permite transferir habilidades de modelos maiores e complexos (Professor) para modelos menores e mais eficientes (Aluno), reduzindo a complexidade e os recursos computacionais. Este processo de destilação, como ilustrado na Figura 10, utiliza do professor para orientar o aluno transferindo seu conhecimento, mas não se limitando, através da predição final.

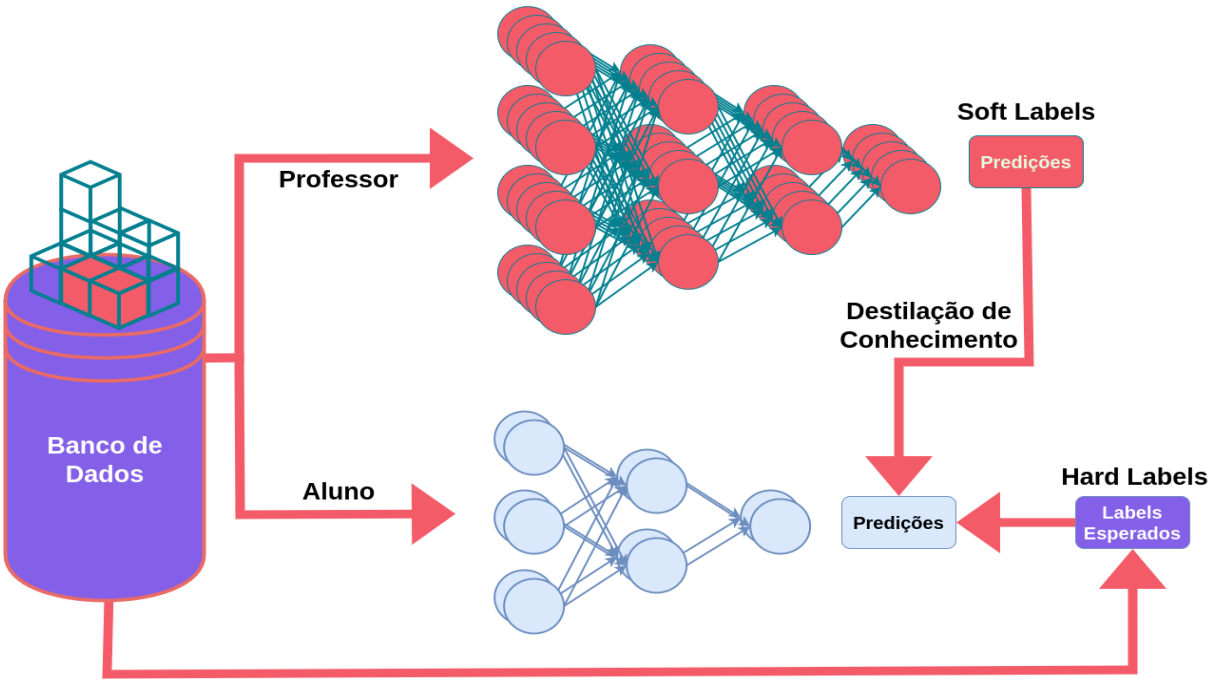

Figura 10: Destilação de conhecimento.

Este treinamento geralmente envolve apenas atualizar os parâmetros do modelo aluno, enquanto o modelo professor faz predições em um conjunto de dados para ser passado ao aluno. Estas predições são chamadas *Soft Labels*, que consiste de valores contínuos de uma ou mais ultimas camadas do modelo professor. Estas previsões geralmente utilizam da normalização softmax para suavizar as predições e estabilizar os gradientes na hora do treino. Esta transferência do conhecimento ocorre ao minimizar a função de perda que mede a distância entre as distribuições de predição do aluno e do professor. É comum utilizar a divergência de Kullback-Leibler dada por:

$$\begin{aligned} D_{KL}(P_{Teacher}||P_{Studant}) &= \sum_{k} P_{Teacher}(k) \log\left(\frac{P_{Teacher}(k)}{P_{Studant}(k)}\right) \\ &= E_{P_{Teacher}}\left[\log\left(\frac{P_{Teacher}(k)}{P_{Studant}(k)}\right)\right] \end{aligned} \tag{5}$$

Se o banco de dados for rotulado, os rótulos (também chamados de *hard labels*) podem orientar o aluno sobre a distribuição esperada para a predição (HINTON; VINYALS; DEAN, 2015). Um estudo recente proposto por Chen et al. (2020) demonstrou que a destilação de conhecimento usando aprendizado auto-supervisionado é especialmente eficiente quando há poucos dados rotulados e o professor é um modelo pré-treinado. No qual o modelo aluno é capaz de aprender uma nova tarefa com apenas alguns dados rotulados, aproveitando as representações úteis destiladas pelo professor, o que impulsiona a

aplicabilidade dos modelos.

2.2.4 Treinamento Eficiente de Modelos Base

O treinamento eficiente de modelos de base visa melhorar a eficiência e o desempenho das redes neurais profundas, reduzindo o tempo e os recursos computacionais necessários. Isso permite treinar modelos complexos de forma mais rápida e econômica, promovendo a sustentabilidade de sistemas de IA.

2.2.4.1 Pré-Treinamento

Como mencionado anteriormente, o pré-treinamento de modelos base é uma etapa computacionalmente intensiva que requer grandes quantidades de recursos, mas resulta em modelos adaptáveis para tarefas específicas por meio de ajuste fino. Otimizar essa fase é crucial para alcançar maior sustentabilidade e disponibilizar modelos pré-treinados úteis para distribuição e pesquisa. Durante essa etapa, várias técnicas e metodologias são estudadas, mas algumas seguem diretrizes centrais. Exemplos incluem medidas diretas para reduzir o consumo de recursos computacionais e aumentar a eficiência do treinamento. Algumas dessas técnicas consiste em:

- **Prototipagem:** A prototipagem com exemplos mais simples é essencial para validar as escolhas de arquiteturas e hiperparâmetros antes de treinar o modelo. Técnicas, como aquelas demonstradas por Yang et al. (2022), permitem a inicialização dos hiperparâmetros do modelo, treinando modelos menores e, em seguida, escalando para modelos maiores. Além disso, escolher os algoritmos de otimização corretos para um problema específico ou inicializar os pesos de modelos anteriores podem induzir à convergência mais rápida dos modelo (LIU et al., 2023; XIA et al., 2023).
- **Dados:** Estudos demonstram que a qualidade dos dados é mais crucial do que a quantidade, desafiando o paradigma tradicional (GUNASEKAR et al., 2023). O trabalho de Xie et al. (2023a) indica a viabilidade de treinar modelos menores para selecionar os dados mais benéficos, atribuindo pesos a uma mistura de conjuntos de dados. Essa abordagem reduz a necessidade de exemplos no treinamento de modelos maiores, resultando em tempos de treinamento mais curtos e melhor desempenho.
- **Softwares, Hardwares e Energia Limpa:** A utilização de softwares eficientes, para uso do processamento assíncrono de dados em CPUs, o uso de caching, pré-carregamento de dados e otimizações no formato de armazenamento de arquivos,

reduz significativamente o tempo de preparação dos dados antes de serem alimentados (DAO et al., 2022; RAJBHANDARI et al., 2020; LECLERC et al., 2023). A utilização de hardwares especializados de computação intensiva, como GPUs, entre outros, também desempenha um papel crucial, no qual com a integração de abordagens inovadoras de fontes de energia renovável, como luz solar e eólica, ou iniciativas como a Green AI Cloud[1], que se compromete em combater as emissões de carbono e ter uma taxa de CO2 negativa, promove a sustentabilidade e contribuir para a mitigação das mudanças climáticas, mantendo a performance no treinamento ou uso desses modelos.

- **Algorítmos de Treinamento Esparços, Quantizados e Aproximações de Baixo Posto:** O uso de algoritmos para promover esparsidade de modelos no treinamento pode alcançar uma performance igual ou até superior em métricas, com menos quantidade de operações (THANGARASA et al., 2023; PESTE et al., 2021; SAXENA et al., 2023). Além disso, treinamentos com aproximações matriciais de baixo posto como demonstrado por Lialin et al. (2023), combinados com quantização auxiliam na economia de recursos computacionais (DETTMERS et al., 2023; XI et al., 2023a).

- **Arquiteturas Eficientes**: Como mencionado anteriormente, a computação do *Multi-Head-Attention* pode ser computacionalmente custosa. Portanto, várias abordagens buscam reestruturar os principais mecanismos dos modelos Transformers, seja repensando o mecanismo de atenção ou o processamento dos valores de entrada para obter as contextualizações (WU et al., 2021; WANG et al., 2020a; CHOROMANSKI et al., 2022; MARTINS; MARINHO; MARTINS, 2022; JAEGLE et al., 2021; SHAZEER, 2019; AINSLIE et al., 2023). Uma abordagem interessante é substituir o principal mecanismo de atenção pela transformada discreta de Fourier (DFT), como demonstrado por Sevim et al. (2023), Lee-Thorp et al. (2022). Nesse método, a complexidade computacional do cálculo de atenção e o treinamento dos parâmetros tornam-se mais simples e eficientes devido à baixa quantidade de parâmetros treináveis. Eles concluem que, ao comparar com o modelo BERT (DEVLIN et al., 2018), o treinamento é 80% mais rápido e a inferência de 40% a 70% mais rápida, mantendo pelo menos 90% dos resultados do BERT. Isso demonstra que a transformada de Fourier é uma técnica poderosa para Redes Neurais Profundas.

Ao combinar essas técnicas e continuar explorando novas estratégias, podemos

[1]https://greenai.cloud/

alcançar avanços significativos na eficiência do treinamento e na sustentabilidade. Isso possibilita a construção de modelos mais sofisticados em menos tempo, utilizando menos recursos computacionais e, consequentemente, reduzindo a pegada de carbono.

2.2.4.2 Ajuste Fino

O ajuste fino de modelos base diz respeito à capacidade de aprimorar as habilidades de um modelo para novas tarefas, aproveitando seu conhecimento prévio. Isso implica na aplicação prática desses modelos em diferentes cenários, utilizando os parâmetros pré-treinados para promover a sustentabilidade dos sistemas de IA. É uma área em constante desenvolvimento, oferecendo várias técnicas e abordagens para transferir conhecimento de forma eficiente para o modelo. Algumas das principais metodologias incluem:

- **Aprendizado com Poucos Dados (*Few-Shot Learning*):** Essa abordagem visa alcançar a máxima eficiência nos modelos, permitindo que eles obtenham bom desempenho e se adaptem a novas tarefas com uma quantidade extremamente limitada de exemplos (SONG et al., 2022).
- **Ajuste Fino Eficiente de Parâmetros (*Parameter-Efficient Fine-Tuning*):** Esta abordagem utiliza técnicas como LoRA, Adapters, $(IA)^3$, entre outras, destaca-se por possibilitar a construção de um ecossistema de modelos especializados em diversos problemas, ao treinar uma quantidade mínima de parâmetros e deixar grande parte inalterada (HU et al., 2022; HU et al., 2023; LIU et al., 2022; LIAO; TAN; MONZ, 2023).

A técnica LoRA (Low Rank Adaptation) destaca-se por permitir a adição de novas funcionalidades sem a necessidade de adicionar novos parâmetros ao modelo. Essa abordagem utiliza um modelo base com parâmetros $W_0 \in \mathbb{R}^{n_1 \times n_2}$. No treinamento do ajuste fino, em vez de realizar as atualizações normalmente como $W_1 = W_0 + \nabla W$, os módulos A e B do LoRA aprende diretamente a matriz de gradientes dessa atualização, ∇W. Para reduzir a complexidade dessa matriz, utiliza-se uma aproximação da decomposição em rank inferior r, representada por $\nabla W = AB$, onde $A \in \mathbb{R}^{n_1 \times r}$ e $B \in \mathbb{R}^{r \times n_2}$ (HU et al., 2022). Esse algoritmo é comumente aplicado aos pesos dos blocos de atenção em modelos Transformers, adaptando as correlações com base nos valores de entrada (LIAO; TAN; MONZ, 2023). Técnicas como QLoRA, que utilizam a decomposição e modelos quantizados, reduzem ainda mais a quantidade de recursos computacionais (DETTMERS et al., 2023). Esta abordagem é semelhante aos métodos de otimização quasi-Newton, onde a matriz

Hessiana é aproximada por uma matriz de baixa ordem (ZHANG; DENG; CHEN, 1999). Isso não apenas reduz a complexidade computacional, mas também facilita a adição de novas funcionalidades ao modelo e a disponibilização desses parâmetros específicos como formas de compressão de habilidades altamente eficientes para a distribuição entre sistemas de IA baseados em um modelo de base de referência.

Ao unir abordagens de aprendizado com poucos dados e ajustes finos eficientes, pesquisadores e engenheiros têm a oportunidade de impulsionar avanços significativos no campo da aprendizagem de máquina. Essa combinação direciona modelos para tarefas específicas, resultando em economia de tempo, energia e recursos computacionais. Como resultado, os modelos tornam-se mais acessíveis e aplicáveis em diversos domínios e contextos, promovendo uma evolução tecnológica mais eficiente e sustentável.

3 Metodologia

3.1 Metodologia de Estudo

Esta pesquisa tem como objetivo contribuir para a investigação de redes neurais artificias e métodos de otimização, para o desenvolvimento de estratégias eficientes e ambientalmente sustentáveis na construção de sistemas de IA complexos. Seu objetivo é fornecer diretrizes de práticas para pesquisadores, com o intuito de impulsionar a adoção de hábitos mais responsáveis e contribuir para o avanço da sociedade em direção a um futuro de avanços tecnológicos ambientalmente amigáveis. Na metodologia de estudo abordada, destacam-se os seguintes tópicos:

- A introdução sobre redes neurais profundas, abordando seus conceitos básicos, assim como a arquitetura Transformer e seus blocos de atenção (VASWANI et al., 2017) que se destacam como uma das técnicas mais utilizadas e influentes na área de redes neurais profundas nos últimos anos. Isso proporciona ao leitor uma compreensão das redes neurais profundas e de uma das arquiteturas mais prevalentes para introduzir modelos de base, os quais são empregados na resolução de problemas complexos e específicos em diversos setores.
- A introdução de técnicas e estratégias de otimização em redes neurais profundas, como quantização, poda e destilação de conhecimento (LIEBENWEIN, 2021). Além disso, será abordado o ajuste fino eficiente para modelos de grande porte (HE et al., 2021), assim como considerações sobre hardwares, softwares e arquiteturas eficientes para os Transformers. Essas metodologias desempenham um papel crucial na otimização do treinamento e escalabilidade de aplicações, tornando a execução dos modelos mais eficiente em termos de custo computacional e emissão de CO2 em ferramentas baseadas em IA.

3.2 Metodologia de Aplicação

Com o objetivo de promover o desenvolvimento científico em todas as áreas de pesquisa, ao mesmo tempo em que busca a sustentabilidade, este trabalho se dedica ao estudo de aplicação de um modelo base, baseado na arquitetura Transformers, para a automação de revisões sistemáticas da literatura (SLR).

No entanto, conforme especificado por Keele et al. (2007), a condução de uma

Passo	Descrição
SLR1	Comissionamento de uma revisão
SLR2	Especificação da(s) pergunta(s) de pesquisa
SLR3	Desenvolvimento de um protocolo de revisão
SLR4	Avaliação do protocolo de revisão
SLR5	Desenvolvimento de termos de Pesquisa
SLR6	Seleção de estudos primários (Triagem de citações)
SLR7	Revisão de seleção
SLR8	Extração e monitoramento de dados
SLR9	Síntese de dados
SLR10	Especificação de mecanismos de disseminação do relatório principal
SLR11	Formatação do relatório principal
SLR12	Avaliação do relatório

Tabela 1: Passos no processo de revisão sistemática conforme proposto por Keele et al. (2007) e adaptado de Dinter, Catal e Tekinerdogan (2021).

revisão sistemática pode ser dividida em vários passos até sua conclusão. Na Tabela 1, apresentamos um resumo desses passos, no qual, para este estudo de aplicação, nos concentramos apenas na automação do passo de triagem de citações (SLR6).

Este passo em particular é reconhecido como o mais demorado, pois requer que um ou mais especialistas reduza a quantidade de citações de um banco de dados de referência do resultado da busca utilizando os temos de pesquisa do passo (SLR5). Esta redução é realizada classificando os exemplos em relevantes ou não sob o critério de classificação para o estudo em questão (BANNACH-BROWN et al., 2019; SELLAK; OUHBI; FRIKH, 2015; TSAFNAT et al., 2018; DINTER; CATAL; TEKINERDOGAN, 2021).

3.2.1 Banco de Dados e Métrica de Trabalho Salvo

Para avaliar a automação da triagem de citações, propõe-se o uso de 24 conjuntos de dados abertos sobre a seleção de citações relevantes em revisões sistemáticas da literatura em vários tópicos da área médica. Esses conjuntos de dados são propostos por Cohen et al. (2006) e Howard et al. (2016), que consistem em exemplos contendo título, resumo e rótulos de classificação como Incluídos (1) e Excluídos (-1) para as respectivas citações dos bancos de dados.

Para avaliar a eficácia do modelo na classificação dessas citações, as métricas comuns podem não fornecer uma representação precisa de sua capacidade, pois os bancos de dados das revisões sistemáticas da literatura (SLRs) tendem a ser extremamente desbalanceados. No qual, frequentemente contêm mais exemplos irrelevantes do que relevantes,

após uma busca inicial utilizando algum mecanismo de busca. Portanto, é necessário recorrer a métricas mais adequadas, como a WSS (KUSA et al., 2023).

A métrica WSS (*Work Saved over Sampling*), introduzida por Cohen et al. (2006), é uma abordagem que estima a quantidade em porcentagem de exemplos que os revisores podem evitar ler, pois foram excluídos pelo classificador como não relevantes, resultando em economia de trabalho. Nesse mesmo estudo, destaca-se que o modelo deve ser avaliado pela métrica considerando um critério que limita a quantidade máxima de exemplos relevantes excluídos erroneamente pelo modelo. Isso garante que, ao avaliar o modelo, ele não comprometa a busca por informações relevantes. Esse limite é estabelecido para não ser menor que 5%, ou seja, o modelo pode classificar erroneamente no máximo 5% dos exemplos que deveriam ser incluídos na revisão sistemática.

No entanto, ao analisar o mesmo modelo em diferentes conjuntos de dados com distribuições de classes distintas, a variação exata em porcentagem da quantidade de exemplos de cada conjunto de dados torna mais desafiador avaliar a performance do modelo de forma geral. Isso ocorre porque os valores extremos da métrica WSS estão condicionados à distribuição das classes de cada conjunto de dados (MELO et al., 2022; FARIA et al., 2022).

Portanto, para avaliar o modelo, é selecionado a métrica AWSS (*Adjusted Work Saved over Sampling*) proposto por Melo et al. (2022), que é diretamente derivada da WSS e consiste na normalização dos valores extremos da métrica estar entre -1 a 1, com o pressuposto que existe muito mais exemplos excluídos que incluídos em uma amostra aleatória. Logo, assim como na WSS, os valores da métrica próximos de zero indicam que a classificação de forma aleatória é mais eficaz que o modelo proposto. O valor da métrica AWSS possui a mesma interpretação da WSS, mas com a condição de que a amostra avaliada é proveniente de uma amostragem de uma população de exemplos que são extremamente raros, ou seja, existem muito mais exemplos irrelevantes que o contrário. Logo, para TN% como a taxa de verdadeiro negativos e TP% como taxa de verdadeiros positivos, a métrica AWSS é dada por:

$$AWSS@TP\% = TN\% - (1 - TP\%) \tag{6}$$

3.2.2 Modelo e Função de perda

Para a metodologia de modelagem da classificação de citações utilizando a arquitetura Transformers proposta neste estudo, aplicamos uma adaptação direta da me-

todologia proposta por Tunstall et al. (2022), denominada SETFIT. Essa abordagem utiliza modelos baseados como Sentence Transformers (REIMERS; GUREVYCH, 2019), os quais são pré-adaptados para vetorizar sentenças de texto em um espaço de representação onde textos semanticamente similares estejam próximos. Assim, estes modelos são empregados para classificar um banco de dados, mesmo quando há poucos exemplos de treinamento disponíveis, consistindo em um processo de treinamento simplificado de apenas dois passos.

No primeiro passo, os exemplos de treinamento são emparelhados aleatoriamente, onde para pares da mesma classe, o rótulo é definido como 1, enquanto que para pares de classes diferentes, o rótulo é atribuído como -1. Em seguida, estes novos exemplos de treino são passados pelo treinamento de aprendizagem contrastiva, no qual para exemplos de mesma classe são agrupados, enquanto classes distintas são separados, com base na representação vetorial do texto por parte do modelo.

No segundo passo, estes mesmos exemplos de treinamento sem estarem sendo emparelhados, são vetorizados pelo modelo treinado resultante, e classificados com base na respectiva classe original, que neste caso para uma classificação binária, utiliza-se a regressão logística.

Neste estudo, optamos por utilizar o modelo denominado SPECTER, desenvolvido por Cohan et al. (2020). Este modelo é construído com base no SciBERT (BELTAGY; LO; COHAN, 2019), que foi pré-treinado em um corpus extenso de artigos científicos. O SPECTER é adaptado por meio do aprendizado contrastivo, utilizando cerca de 684 mil exemplos de treinamento contrastivo extraídos de textos científicos. Ele é projetado para produzir representações vetoriais dos textos, contextualizadas pelo token [CLS], resultando em um Sentence Transformers. A seleção desse modelo se justifica pela sua múltiplas fases de treinamento em textos científicos, que são o foco para esta pesquisa.

Já para a função de perda, diferentemente da proposta original dada por Tunstall et al. (2022), foi optado utilizar uma adaptação da função de perda contrastiva supervisionada proposta por Khosla et al. (2021), dada como:

$$Loss(x,y) = \frac{-1}{|P|}\sum_{p\in P}\log\frac{\exp\{sim(x_p,y_p)/\tau\}}{\sum_{n\in N}\exp\{sim(x_n,y_n)/\tau\}},\ sim(x,y) = \frac{x\cdot y}{||x||_2||y||_2} \tag{7}$$

no qual, $p \in P$ são os índices dos exemplos de treino contrastivos que possuem a mesma classe, e para $n \in N$, os de classes distintas, onde x e y são vetores, com y sendo o

respectivo par aleatório selecionado no primeiro passo. A função $sim(x, x)$ também é chamada de similaridade de cosseno e o hiperparâmetro τ foi fixado em 0.2.

Esta adaptação consiste em duas principais diferenças entre a função proposta por Khosla et al. (2021). Em primeiro lugar, ela utiliza pares aleatórios em vez de considerar todos os pares possíveis dentro de uma amostra. Em segundo lugar, ao invés de usar todos os exemplos no denominador, apenas os negativos são considerados, utilizando o método de "*Hard Negative Sampling*". Estudos indicam que essa abordagem oferece melhor estabilidade à função de perda, com base na uniformidade e na tolerância com que os exemplos são separados. No qual a escolha desta função de perda se dá por sua adaptabilidade em dar importância à separação dos exemplos, no qual para exemplos próximos que deveriam ser separados de um vetor de referencia, este possui maior peso do que aqueles mais distantes, como demonstrado por Wang e Liu (2021).

3.3 Metodologia de Otimização

Com o propósito de introduzir novas práticas analíticas para os modelos base para o regime de eficiência energética, além do treinamento com poucos dados conforme descrito na seção 3.2.2, este estudo enfoca diversas metodologias sustentáveis de pesquisa como intervenção para aplicação eficiente de modelos de aprendizado profundo. Especificamente, concentra-se no exemplo de classificação de artigos científicos para revisões sistemáticas da literatura, com o objetivo de oferecer uma análise crítica como ferramenta de pesquisa eficiente e construir um ecossistema de pesquisa utilizando IA, tornando o modelo mais acessível em ambientes com recursos limitados, como dispositivos móveis.

3.3.1 Ajuste Fino

Para otimizar o ajuste fino do modelo base e adquirir novas habilidades para uma tarefa específica, propõe-se a aplicação da técnica LoRa (HU et al., 2022). Este estudo escolhe aplicar a técnica nas três últimas camadas do modelo Transformers, mais precisamente nas matrizes rotacionais K e V, com a projeção intermediária igual a 4 nos módulos A e B conforme descrito no Capítulo 2.2.4.2. Essa abordagem é fundamentada no extenso número de parâmetros do modelo, sugerindo que a manipulação de poucas matrizes de projeção possa ser capaz de preservar as habilidades originais do modelo base, ao mesmo tempo em que incorpora habilidades locais específicas para a tarefa em questão.

3.3.2 Técnicas de Otimização

Durante a otimização do modelo, surgem inúmeras possibilidades e permutações para aprimorar sua eficiência. Neste estudo, concentramo-nos em técnicas básicas para fundamentar a análise e pesquisa de métodos visando tornar os modelos base mais sustentáveis.

Entre essas técnicas, a técnica de poda intitulada de Wanda, seguindo o trabalho de Sun et al. (2023) e ilustrada na Figura 11. Esta técnica consiste em aplicar a norma L2 em relação às dimensões de um conjunto de vetores de pré-ativação denotados como $\{h_0, ..., h_n\}$ e demonstrado em (passo 1), e então multiplicar elemento por elemento a matriz de pesos absolutos, conforme descrito no passo 2. Finalmente, zeram-se os valores mais próximos de zero na matriz de pesos resultante, como no passo 3, onde 50% dos valores foram zerados.

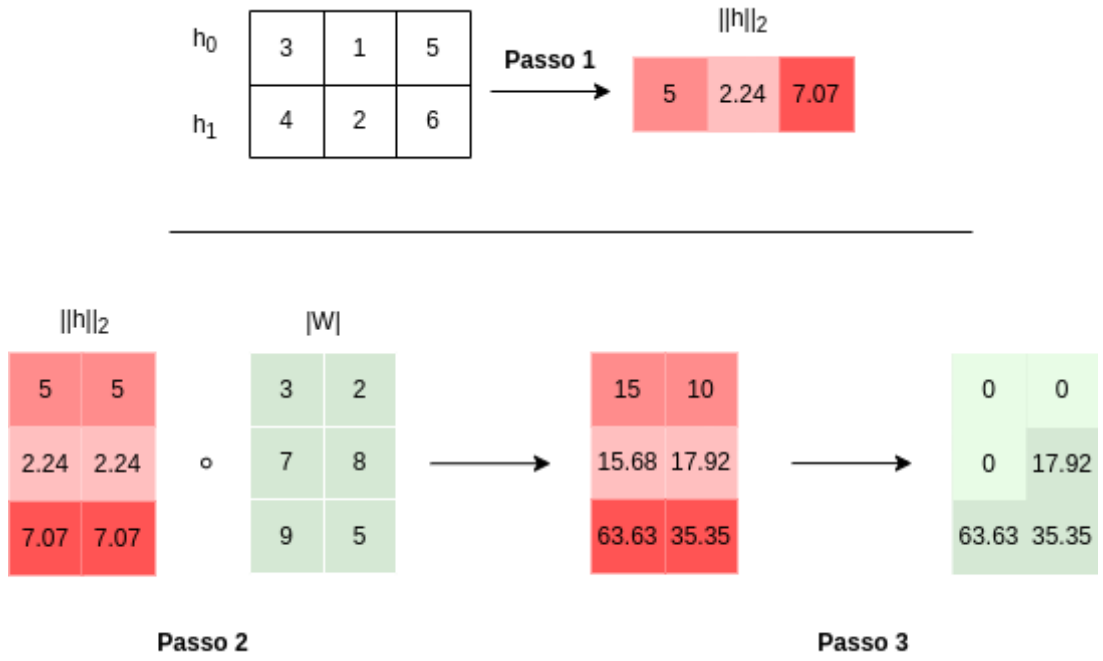

Figura 11: Poda Wanda. Figura adaptada de Sun et al. (2023).

Outra técnica selecionada é a quantização dinâmica, na qual apenas os pesos são quantizados, enquanto as ativações seguem uma precisão estabelecida de FLOAT16 em CPUs e BFLOAT16 em GPUs. Este método, conforme discutido no Capítulo 2.2.1, visa transformar os pesos em 8 bits, aplicando o método para toda a matriz, e uma abordagem mais extrema, que utiliza os respectivos pesos de cada neurônio como grupos para quantizar-los em 4 bits. Como nem sempre é viável utilizar valores precisos da transformação dos bits em hardwares específicos, emprega-se a falsa quantização. Nessa

abordagem, os pesos são arredondados conforme a metodologia descrita, mas são utilizados os resultados desquantizados na precisão original para calcular o desempenho do modelo na métrica de avaliação da tarefa.

3.3.3 Habilidades Híbridas

A medida que surgem mais modelos especializados em um ecossistema baseado em modelos base, torna-se crucial avaliar como integrar as habilidades específicas de diversas tarefas em um único modelo. Nesse sentido, propõe-se um método para criar um modelo com as respectivas habilidades dos módulos LoRa de forma híbrida, utilizando a técnica de decomposição por valor singular (SVD). Assim, este método possibilita a combinação de diferentes pesos adaptáveis com tamanhos distintos para os módulos LoRa, resultando em uma nova representação de baixo posto que pode ser utilizada como uma interpolação de habilidades, conforme sua aplicação na arquitetura do modelo base.

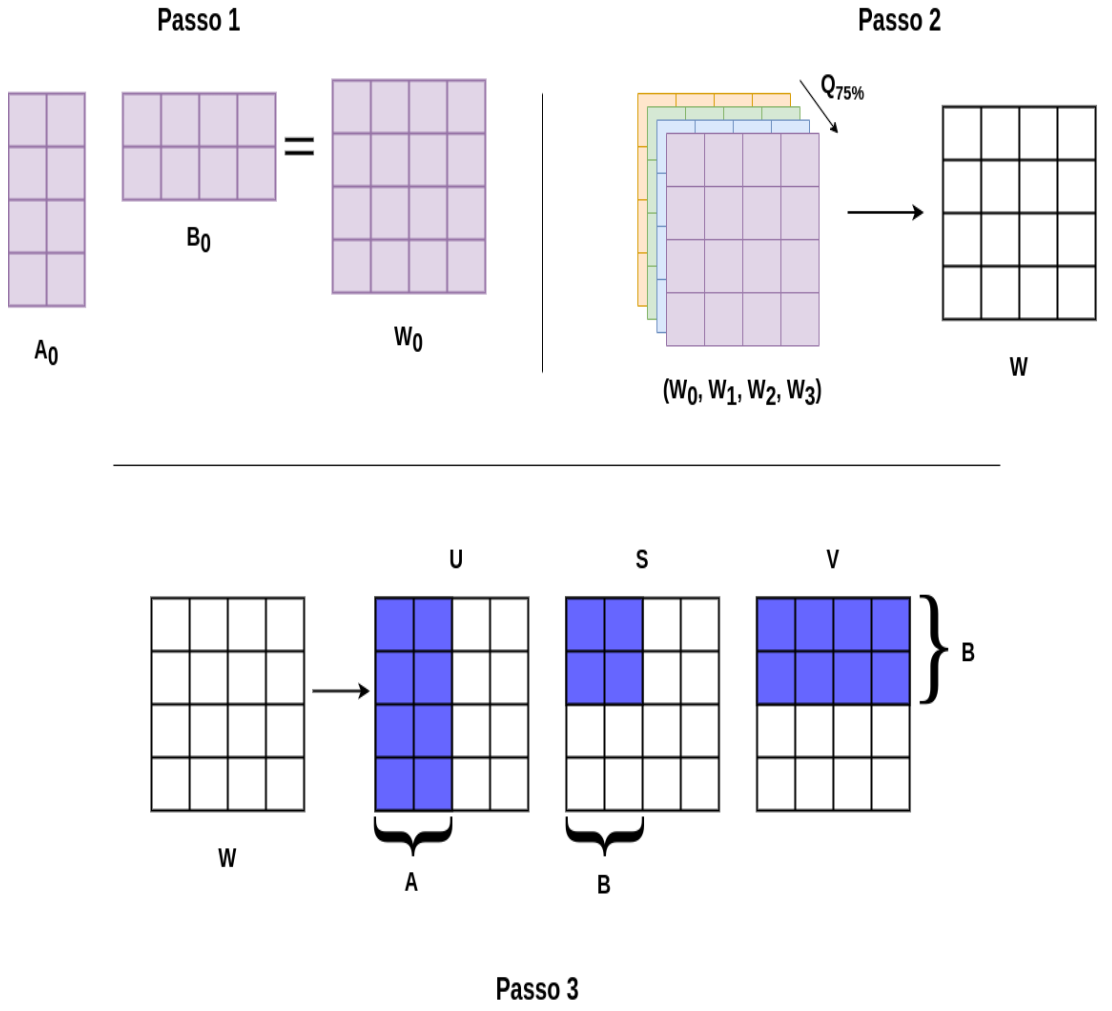

Figura 12: Método de Hibridação dos módulos LoRa.

Conforme ilustrado na Figura 12, o primeiro passo do método proposto consiste em utilizar os módulos LoRa A_0 e B_0 pós-ajuste para todas as camadas em que foram aplicados e recriar as respectivas matrizes de adaptação denotado por $W_0 = A_0B_0$, onde $W_0 \in \mathbb{R}^{k \times v}$, com $A_0 \in \mathbb{R}^{k \times r}$ e $B_0 \in \mathbb{R}^{r \times v}$. Esse passo é realizado para cada um dos ajustes que resultaram em diferentes módulos A e B para a tarefa específica em questão.

No segundo passo, é criado um vetor de matrizes de pesos adaptáveis $(W_0, ..., W_n) \in \mathbb{R}^{n \times k \times v}$ e, em seguida, para cada elemento, é aplicado o quantil igual a 75%, resultando em uma matriz de valores interpolados W, entre os valores do vetor de pesos adaptáveis. Este tem como intuito manter a maior parte dos valores da distribuição sem prejudicar os valores de outras matrizes.

No terceiro passo, é aplicado o método SVD na matriz de pesos resultante W e definido um valor arbitrária igual a r para a aproximação da matriz resultante W. Selecionando esses r componentes da decomposição, como descrito no Passo 3 da Figura 12, eles são utilizados como novos valores de inicialização para os módulos do LoRa. No qual, para o módulo B, são utilizados os r valores singulares multiplicados pelos r vetores singulares descritos na matriz V, o mesmo para o módulo A e a matriz U.

4 Resultados

4.1 Análise quantitativa dos bancos de dados

Os bancos de dados utilizados em revisões sistemáticas da literatura são compostos por extensas coleções de referências textuais, que são submetidas a uma triagem meticulosa pelos pesquisadores para a seleção de estudos relevantes. O objetivo é obter artigos de referência significativos que possam responder a uma pergunta específica ou sustentar uma investigação científica. Em banco de dados de SLRs, é comum observar um desequilíbrio acentuado entre o número de artigos relevantes (Incluídos) e não relevantes (Excluídos). Este desequilíbrio é evidente nos bancos de dados selecionados para este estudo, conforme apresentado na Tabela 2. No qual classificações de Excluído (-1) e Incluído (1) foram atribuídos pelos autores com base no título e resumo de cada artigo. Em alguns casos, o desequilíbrio é significativo, com menos de 10% dos artigos sendo incluídos em cada revisão sistemática.

Banco de Dados	Total	Incl.	Excl.
ACE Inhibitors	2544	41 (1.6%)	2503 (98.4%)
ADHD	851	20 (2.4%)	831 (97.6%)
Antihistamines	310	16 (5.2%)	294 (94.8%)
Atypical Antipsychotics	1120	146 (13.0%)	974 (87.0%)
Beta Blockers	2072	42 (2.0%)	2030 (98.0%)
Calcium Channel Blockers	1218	100 (8.2%)	1118 (91.8%)
Estrogens	368	80 (21.7%)	288 (78.3%)
NSAIDs	393	41 (10.4%)	352 (89.6%)
Opioids	1915	15 (0.8%)	1900 (99.2%)
Oral Hypoglycemics	503	136 (27.0%)	367 (73.0%)
Proton Pump Inhibitors	1333	51 (3.8%)	1282 (96.2%)
Skeletal Muscle Relaxants	1643	9 (0.6%)	1634 (99.4%)
Statins	3465	85 (2.5%)	3380 (97.5%)
Triptans	671	24 (3.6%)	647 (96.4%)
Urinary Incontinence	327	40 (12.2%)	287 (87.8%)
Drug Reviews (COHEN et al., 2006)	16015	2169	13846
Bisphenol A (BPA)	7700	111(1.4%)	7589 (98.6%)
Fluoride and Neurotoxicity	4479	51 (1.1%)	4428 (98.9%)
Neurophatic pain	29207	5011 (17.2%)	24196 (82.8%)
PFOA/PFOS	6331	95 (1.5%)	6236 (98.5%)
Transgenerational	48638	765 (1.6%)	47873 (98.4%)
SWIFT (HOWARD et al., 2016)	92262	5861	86401

Tabela 2: Distribuição de rótulos para cada respectivo banco de dados.

Ao utilizar o modelo Transformers baseado no BERT (DEVLIN et al., 2018), é empregado um tokenizador com um vocabulário de 30 mil tokens para segmentar o texto em unidades relevantes para construção dos vetores embeddings antes de serem alimentados ao modelo. Dessa forma, considerando a limitação do modelo de até 512 tokens de entrada, é necessário avaliar a frequência da concatenação do título e resumo de cada artigo do conjunto de dados de referência. Assim, conforme ilustrado na Figura 13, observa-se que a maioria dos dados apresenta pelo menos 75% da distribuição dentro do limite máximo do modelo. É interessante notar que o maior banco de dados, Transgeracional como constatado na Tabela 2, exibe uma quantidade expressiva de outliers com mais de 512 tokens para o rótulo -1, como demonstrado pelo boxplot vermelho.

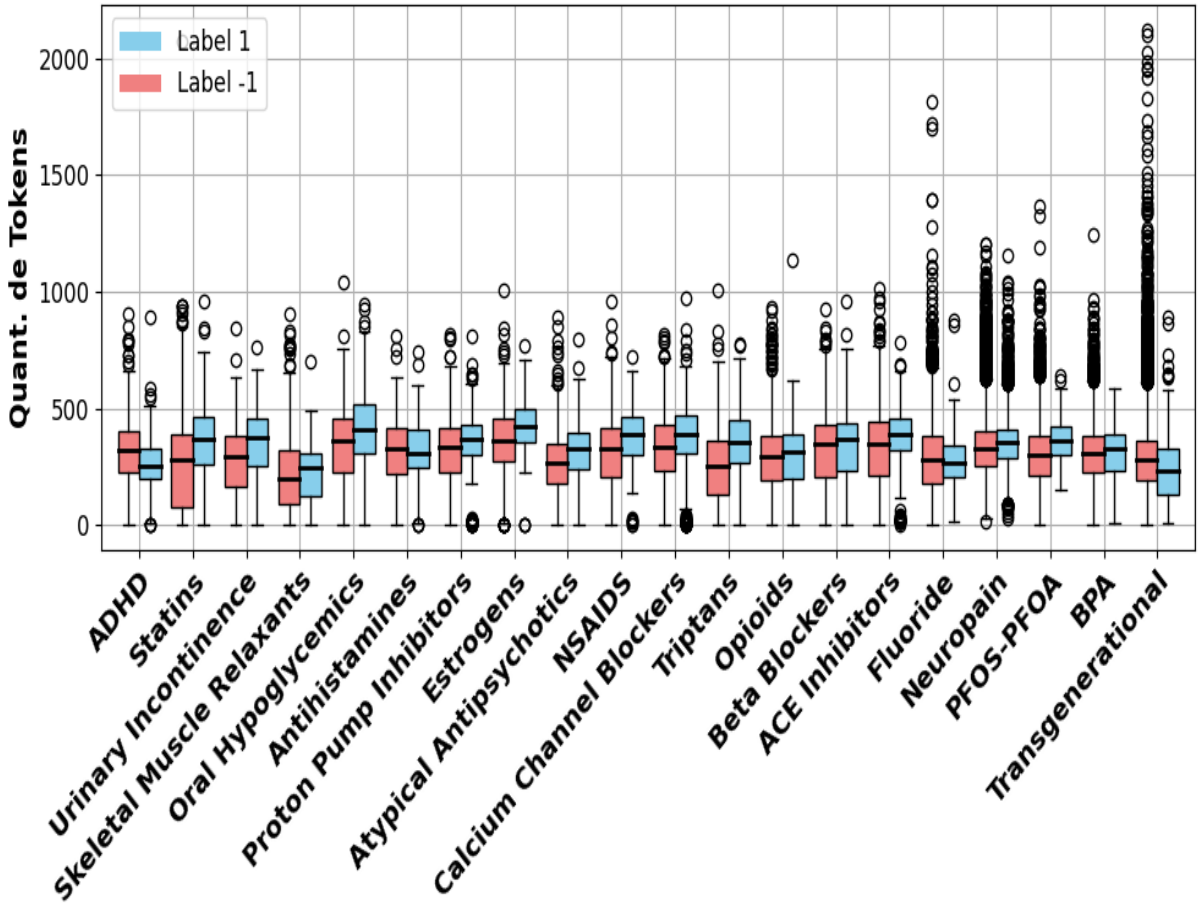

Figura 13: Distribuição de tokens para cada banco de dados.

É crucial levar em conta que, embora os bancos de dados contenham terminologia específica de cada área de pesquisa, a maneira como o tokenizador segmenta o texto permite que o modelo interprete o texto sem necessitar de um vocabulário exclusivo para cada domínio de estudo. Portanto, este estudo aproveita ao máximo a capacidade máxima de 512 tokens do modelo Transformers para criar uma vetorização contextualizada do

título e o resumo concatenados sobre o token especial [CLS], que será empregado na tarefa de classificação por meio da regressão logística.

4.2 Análise do Modelo e Treinamento contrastivo

O modelo Transformers SPECTER (COHAN et al., 2020) escolhido para este estudo passou por um treinamento semântico inicial, utilizando a função de perda Tripla, com base no modelo SciBERT (BELTAGY; LO; COHAN, 2019). Este pré-treinamento tem como intuito agrupar vetores semelhantes e separar aqueles que não são, utilizando o vetor de 768 dimensões resultante do token [CLS] contextualizado pela sequência de texto de entrada. Essa abordagem visa criar um espaço semântico inicial para sentenças de texto, especialmente provenientes de artigos científicos, de modo que textos similares estejam próximos uns dos outros.

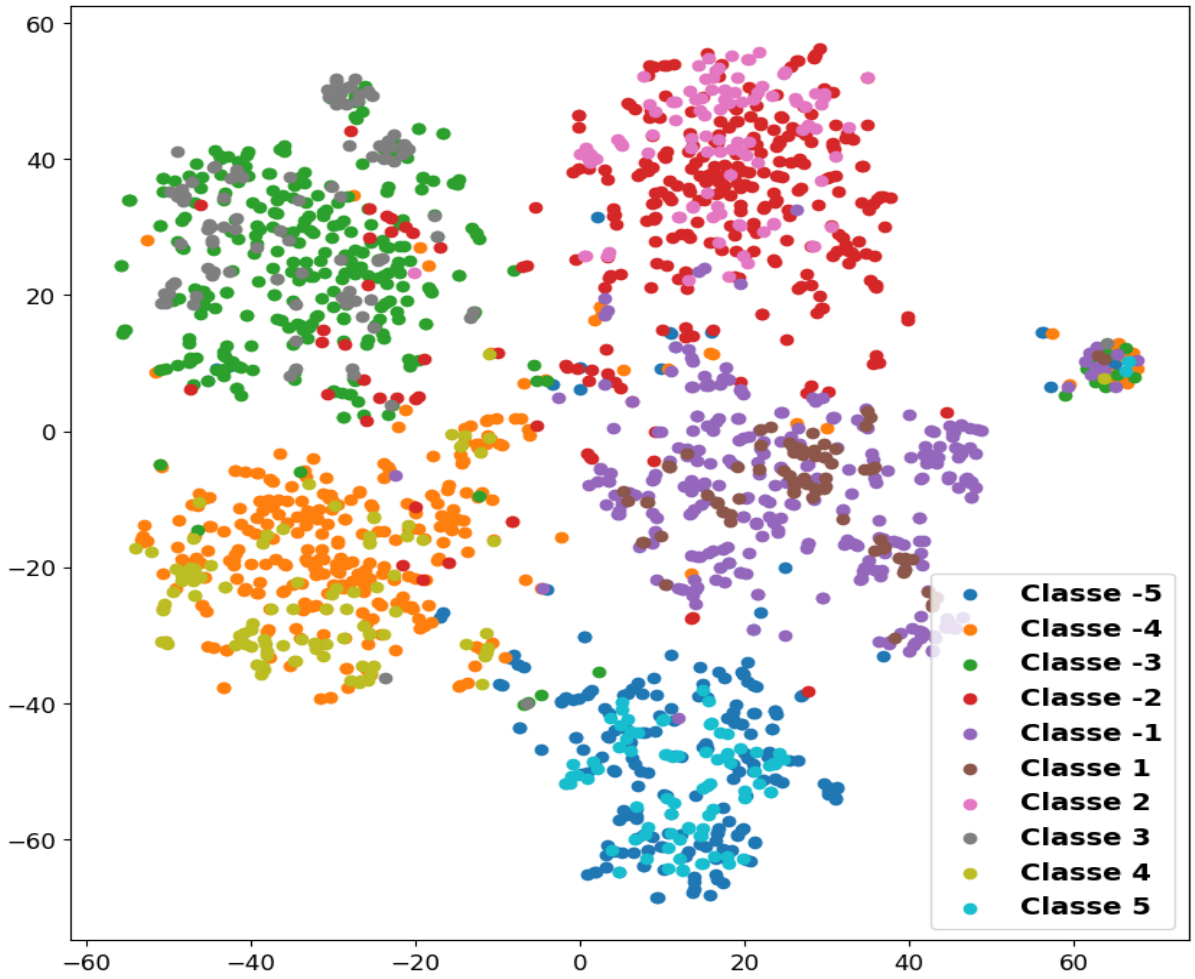

Figura 14: Visualização T-SNE dos vetores [CLS] de 5 banco de dados utilizando o modelo SPECTER para os exemplos: 1 - NSAIDS, 2 - Neuropain, 3 - Oral Hypoglycemics, 4 - Statins e 5 - Antihistamines.

Logo, para avaliar a capacidade inicial deste modelo, utilizou-se o método de redução de dimensionalidade T-SNE (MAATEN; HINTON, 2008) para reduzir a representação do vetor [CLS] de 768 dimensões para 2 dimensões. Cinco bancos de dados foram selecionados aleatoriamente, com uma amostra estratificada de no total de 300 exemplos. As classificações positivas e negativas são referenciadas por números indexados para cada banco de dados. Por exemplo, no banco de dados Antihistamines, as classes 5 e -5 representam os rótulos positivo e negativo, respectivamente. Logo, como demonstrado na Figura 14, os pontos exibem agrupamentos para cada banco de dados, porém falham na captura das respectivas classes locais.

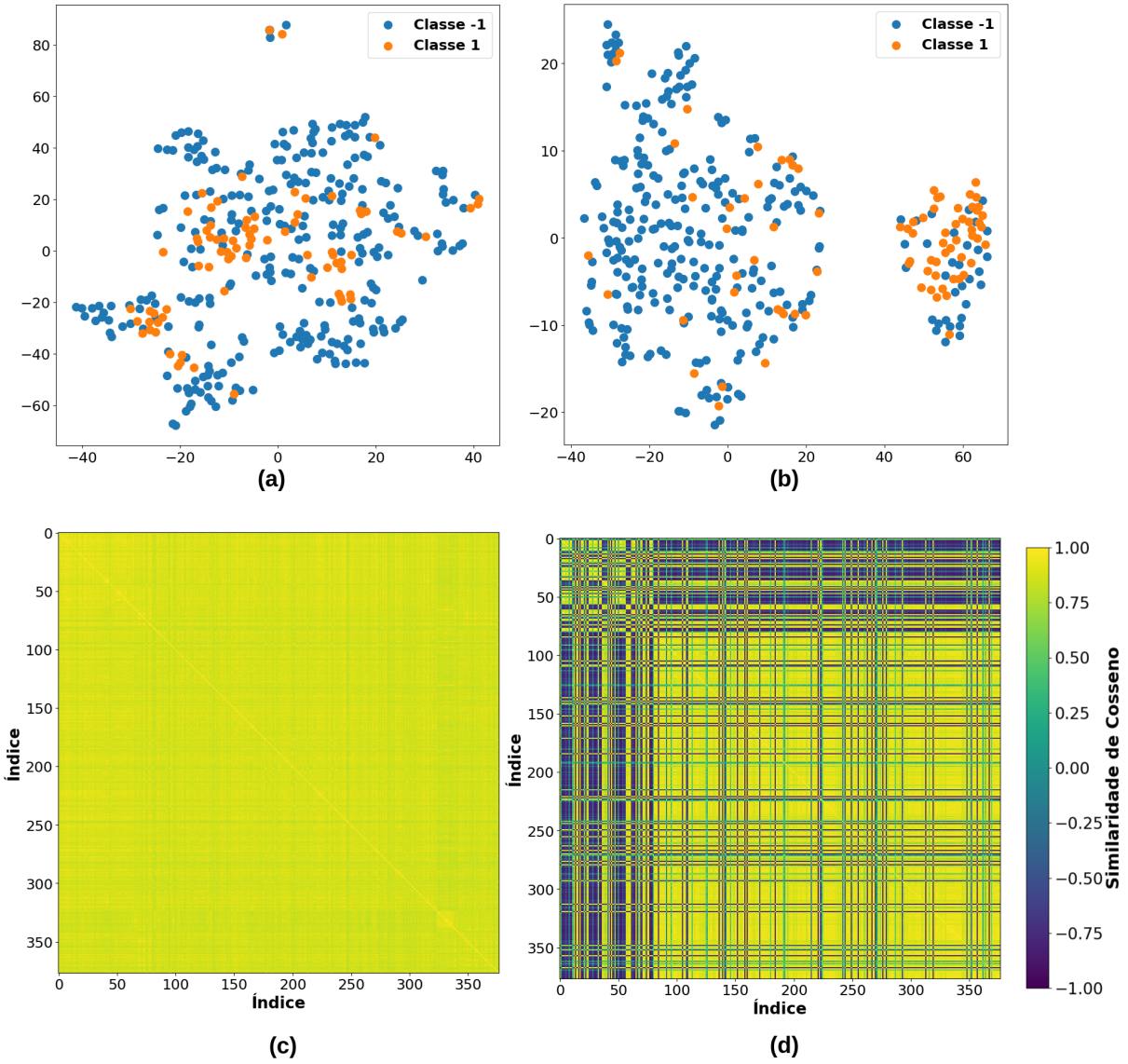

Figura 15: Visualização T-SNE dos vetores [CLS] do banco de dados NSAIDS, antes e depois do ajuste fino. Representados em (a) e (b) respectivamente. Visualização da matriz de similaridade de cosseno entre os exemplos, antes e depois do ajuste fino, ilustradas em (c) e (d) respectivamente.

No entanto, para uma análise mais detalhada das capacidades do modelo em se-

parar essas classes locais de forma eficiente, foi efetuado o treinamento de poucos exemplos utilizando o aprendizado contrastivo supervisionado (KHOSLA et al., 2021) que possui a capacidade de separar as classes no espaço de similaridade semântica (WANG; LIU, 2021; CHEN et al., 2020). Para a configuração do treinamento da arquitetura Transformers, foi aplicado o mecanismo de adaptação LoRa (HU et al., 2022) com rank igual a 4 para as matrizes rotacionais V e K do mecanismo de atenção dos últimos 3 blocos do Transfomers. Já para os exemplos de treino contrastivo, utilizou-se apenas 8 exemplos por classe, 16 exemplos no total, e foi construído 40 pares aleatórios de tal forma que, para os pares de classes iguais o valor esperado é igual a 1, e para os pares diferentes o valor esperado é -1, resultando em 640 exemplos de treino contrastivo. É importante ressaltar que pelo fato da arquitetura Transformers possuir Dropout (SRIVASTAVA et al., 2014) nas suas camadas lineares (DEVLIN et al., 2018), mesmo que haja pares de exemplo iguais, o modelo não é determinístico na representação final da vetorização do texto na fase de treino.

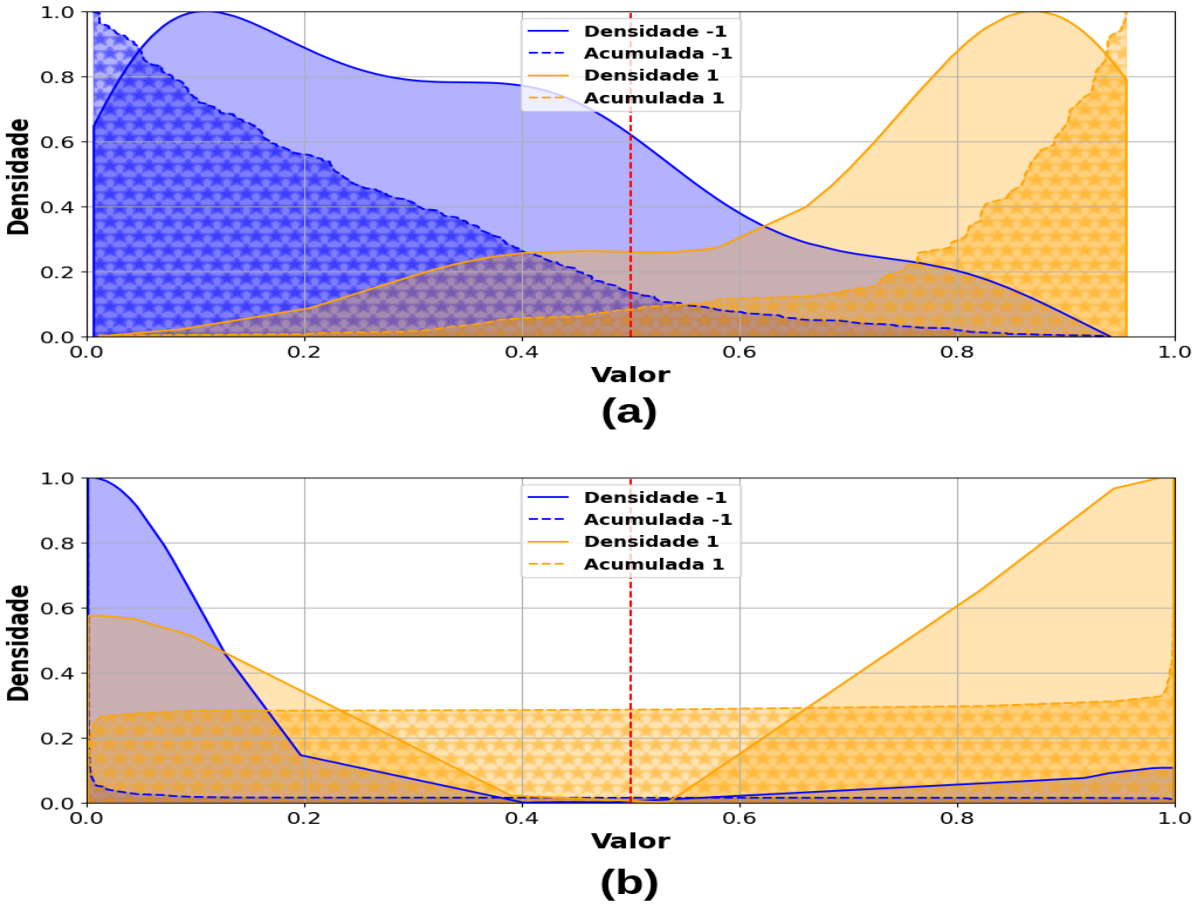

Figura 16: Curvas normalizadas da densidade de probabilidade e densidade acumulada (estrelada) das representações vetoriais do token [CLS] pelo modelo Transformers para o banco de dados NSAIDS antes (a) e depois (b) do ajuste fino utilizando a regressão logística ajustada no banco de treino.

Assim, tomando como referência o banco de dados NSAIDS e uma amostra de 350 exemplos para avaliação, o treinamento foi realizado por duas épocas. Como resultado, antes do treinamento, na Figura 15 (a), nota-se que pela redução de dimensionalidade, não foi possível capturar a separação semântica. Na imagem (c), a matriz de similaridade demonstra alta correlação entre todos os exemplos, o que reflete a visualização em (a). Por outro lado, após o treinamento, na imagem (b), é possível observar que o modelo tenta agrupar os exemplos de acordo com a similaridade entre as classes rotuladas. Na imagem (d), é notável a similaridade negativa entre os exemplos de classes diferentes, onde os índices de 0 a 80 representam os exemplos com rótulo positivo.

Ao treinar a regressão logística nos 16 exemplos de treino vetorizados e avaliado na mesma amostra de 350, utilizando o modelo não treinado, observa-se na Figura 16 (a) que as densidades se sobrepõem perto do valor 0.5, indicando que a regressão logística tem maior dificuldade na classificação. No entanto, para o modelo treinado, na imagem (b), as densidades estão bem separadas e a acumulada possui maior concentração nos extremos 0 e 1, indicando que o modelo separa de forma mais clara as representações vetoriais e possui menos indecisão perto do valor 0.5.

4.3 Análise de Otimização

A análise do desempenho de um modelo é fundamental para garantir a escalabilidade e a sustentabilidade da infraestrutura de uma aplicação. Avaliar o desempenho do modelo em relação aos métodos de otimização é uma tarefa complexa, que envolve uma série de etapas rigorosas com o objetivo de identificar a combinação ideal de métricas de desempenho e técnicas de otimização. Com este propósito, todos os experimentos deste estudo foram conduzidos utilizando as seguintes especificações de hardware: para CPU, um processador Intel Core i7-11800H da 11ª geração; para GPU, uma placa NVIDIA GeForce RTX 3060 Mobile.

Precisão	**Treinamento**	**Inferência GPU**	**Inferência CPU**
FLOAT32	$132\text{s} \pm 0.10$ (1x)	$0.72\text{s} \pm 0.10$	$10.92\text{s} \pm 0.21$ (1x)
BFLOAT16	$40\text{s} \pm 0.10$ (3.3x)	-	-
FLOAT16	-	$0.32\text{s} \pm 0.07$ (2.4x)	$19.86\text{s} \pm 0.22$ (0.55x)
INT8	-	-	$6.38\text{s} \pm 0.31$ (1.7x)

Tabela 3: Tempo de execução para treinar e inferir com o modelo em diferentes tipos de precisão. Para o treino, utilizando lotes de tamanho 16 e 512 tokens, com 640 exemplos por duas épocas. Para a inferência, um lote de tamanho 32 com 512 tokens.

A Tabela 3 apresenta as estatísticas de treinamento utilizando as mesmas confi-

gurações do Capítulo 4.2. Ao realizar o teste de treinamento, constatamos que ao utilizar a precisão BFLOAT16, observa-se uma superioridade significativa em relação ao tempo de treinamento, alcançando uma velocidade 3.3 vezes maior e com 95% de confiança de que não há diferença significativa no valor final da função de perda, conforme verificado pelo teste-t em comparação com a inicialização padrão de modelos em FLOAT32.

Em relação à inferência, foram conduzidos testes de desempenho utilizando uma amostra de 32 exemplos, cada um com 512 tokens. Utilizando a GPU, observamos uma melhoria significativa de 2.4 vezes na velocidade ao empregar a precisão FLOAT16 em comparação com FLOAT32. No entanto, ao realizar a inferência utilizando a CPU, notamos que o uso de FLOAT16 resulta em uma desaceleração de quase duas vezes. Isso se deve à otimização da configuração da CPU para lidar com precisões específicas pré-determinadas.

Para alcançar precisões menores, como a precisão INT8, foi implementada a técnica de quantização dinâmica. Nesse método, apenas os pesos são quantizados, enquanto os valores de ativação permanecem na precisão original, que neste caso é FLOAT32. Como resultado, constatamos uma melhoria significativa no desempenho do modelo em relação a CPU, com uma velocidade de até 70% mais rápida, o mesmo teste não foi possível ser feito na GPU, pois este não possuía a configuração para esta precisão.

	SPECTER	**LoRa**
Total de Parâmetros	110M (100%)	36864 (0.03%)
32 bits	420MB	0.14MB (ou 144KB)
16 bits	209MB	0.07MB (ou 72KB)
8 bits	104MB	0.035MB (ou 36KB)
4 bits	52MB	0.018MB (ou 18KB)

Tabela 4: Resumo do tamanho dos parâmetros dos modelos SPECTER e LoRa.

Em relação à carga computacional necessária para utilizar o modelo de referência neste estudo, a precisão dos pesos exerce um impacto significativo na memória computacional, como evidenciado na Tabela 4. A precisão original consome 420MB, dada a presença de 110M de parâmetros no modelo, o que pode ser considerável para dispositivos móveis. No entanto, ao adotar outras precisões, podemos alcançar reduções de até 8 vezes, como é o caso dos modelos de 4 bits. Por outro lado, ao aplicar a configuração do LoRa conforme definido no treinamento do capítulo 4.2, notamos uma redução significativa na quantidade de parâmetros necessários para o treinamento, representando menos de 0.5% do tamanho do modelo base. Além disso, o consumo de memória é extremamente reduzido, chegando a apenas 18KB, e demonstrando ser eficaz para disponibilizar novas funcionalidades entre

as aplicações que utilizam o mesmo modelo base.

Além da análise da velocidade do modelo e recursos computacionais, também foi feita uma análise de métodos de otimização de redes neurais pós-treino e comparadas com o modelo base sem treinamento em FLOAT32 utilizando a métrica AWSS (MELO et al., 2022). Primeiramente, os modelos foram treinados utilizando a precisão BFLOAT16 com as mesmas configurações do capitulo 4.2 utilizando o método de treino eficiente LoRa (HU et al., 2022). Após este ajuste fino, foi realizado os método de poda proposto por Sun et al. (2023) e a quantização em 4bits dos pesos agrupados por neurônio. Assim, para computar a métrica de trabalho salvo AWSS pela classificação do modelo, utilizou-se a regressão logística no qual a decisão das classes é efetuada no valor de referência 0.5 e treinado com os 16 exemplos de treino seguindo a proposta do estudo de Tunstall et al. (2022) utilizando a vetorização do modelo Transformers após o ajuste fino, contextualizado pelo token [CLS].

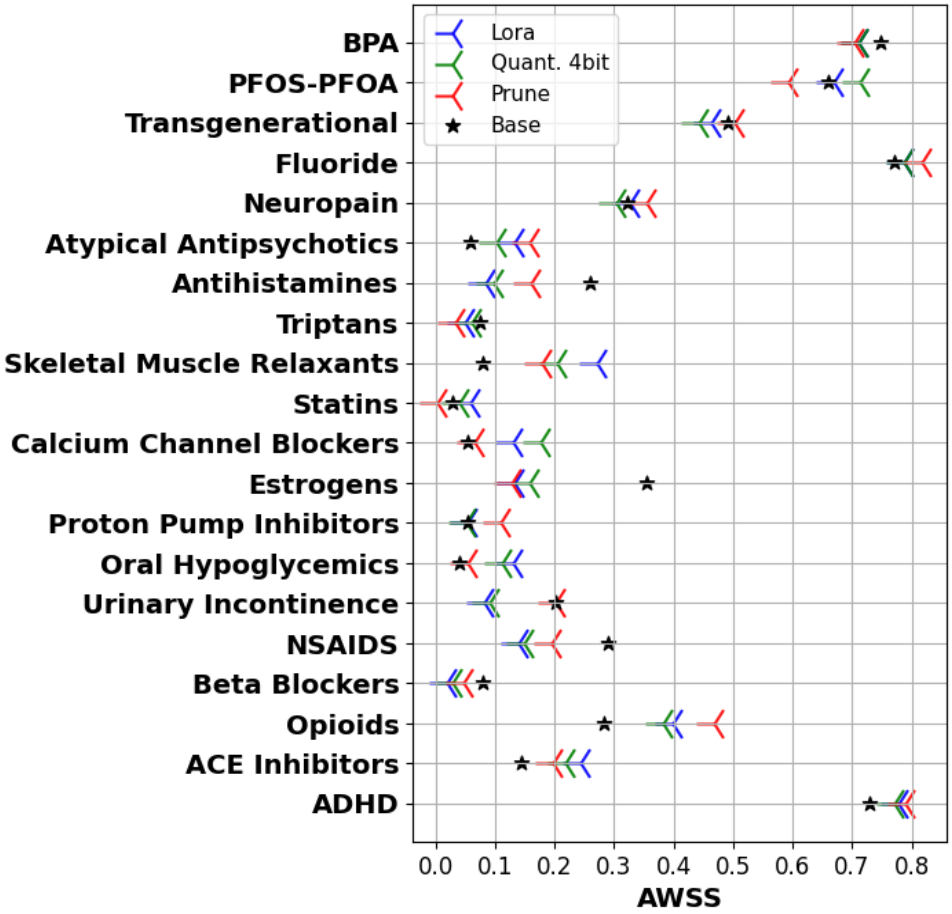

Figura 17: AWSS@95% para diferentes tipos de otimização por banco de dados.

Na Figura 17, é apresentado a comparação entre todos os dos bancos de dados sob a métrica AWSS a 95% de recall para a análise de otimização pós-treino. É interessante

observar que o modelo SPECTER (COHAN et al., 2020) demonstra um bom desempenho nos bancos de dados do estudo SWIFT (HOWARD et al., 2016), demonstrando a capacidade de redução de mais de 70% de documentos irrelevantes na base de dados. Por outro lado, em alguns casos, o ajuste fino não resulta em melhorias na métrica AWSS, como evidenciado no caso extremo do banco Estrogens. Ao analisar o modelo quantizado em 4 bits, observa-se uma pequena degradação na maioria dos casos em comparação com o ajuste fino do LoRa. Quanto à poda, eliminando 50% dos parâmetros do modelo, em alguns casos observamos uma melhoria na métrica, assim como na quantização, indicando que o modelo está sobre-parametrizado em determinados bancos de dados, como no caso dos Opioides. No geral, o modelo base apresenta um bom desempenho, e o ajuste fino com poucos exemplos serve principalmente para aumentar a confiança do modelo na reestruturação da similaridade semântica no espaço de representação, como discutido no Capítulo 4.2.

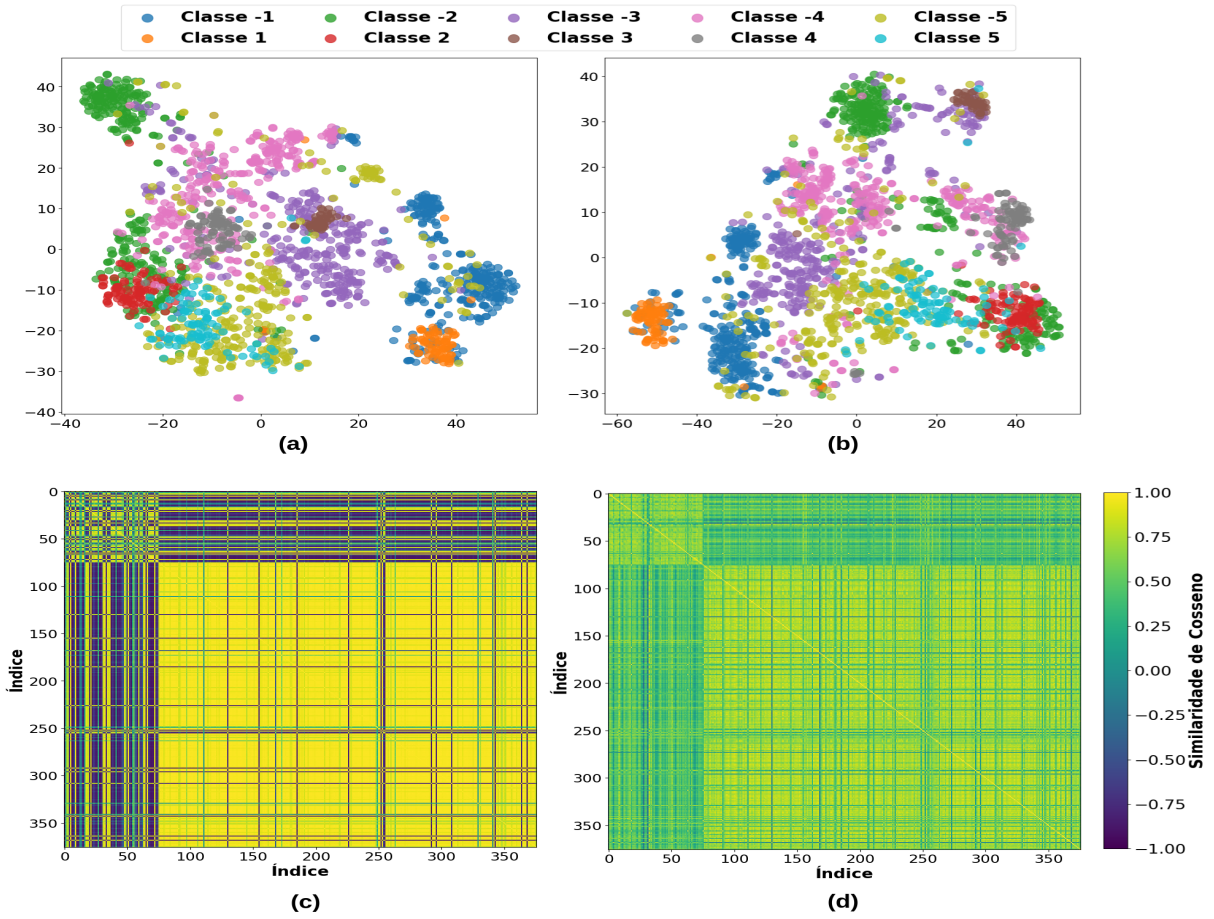

Figura 18: Visualização T-SNE dos vetores [CLS] de cinco bancos de dados usando o modelo SPECTER para os exemplos: 1 - ADHD, 2 - BPA, 3 - Fluoride, 4 - PFOS-PFOA, 5 - Transgenerational. Na subfigura (a), é apresentado o modelo sem ajuste. Na subfigura (b), é exibido o modelo com ajuste híbrido. As matrizes de similaridade de cosseno para o banco de dados ADHD são mostradas em (c) para as previsões após o ajuste fino e em (d) para o ajuste híbrido.

Por fim, foi feita uma análise de aplicação com a hipótese de que se o modelo base fosse utilizado para obter múltiplas habilidades ao longo do tempo, qual seria o impacto ao juntar os módulos LoRa em apenas um e se haveria degradação nas habilidades do modelo adaptado sem a adição de novos treinamentos. Logo, para esta análise, utilizando a metodologia proposta no Capítulo 3.3.3, no qual para achar os novos módulos do LoRa A e B foi estabelecido o rank igual a 8. Este experimento foi realizado nos 5 banco de dados que obtiveram os melhores resultados na métrica AWSS conforme a Figura 17.

Na Figura 18, a imagem (a) representa o espaço definido pela vetorização do modelo base em relação aos cinco bancos de dados considerados. Em contraste, a imagem (b) mostra o resultado da abordagem de combinação das habilidades do modelo. É interessante notar que, embora esta seja uma projeção em duas dimensões, o modelo busca distribuir as classes localmente, ao mesmo tempo em que mantém uma perspectiva global.

Na imagem (c), é apresentada a matriz de similaridade para o banco de dados ADHD, utilizando o módulo LoRa correspondente. Para os índices de 0 a 70, que representam as instâncias positivas, a similaridade entre os vetores é aproximadamente 1, indicando uma alta semelhança entre instâncias da mesma classe. Por outro lado, para instâncias de classes diferentes, essa relação é próxima de -1. Entretanto, na imagem (d), que representa os mesmos exemplos sob a representação vetorial do modelo híbrido, a matriz de similaridade permanece quase idêntica à matriz (c). No entanto, as similaridades não são perfeitamente próximas de 1 e -1; em vez disso, os valores para instâncias da mesma classe são aproximadamente 0.8, enquanto para instâncias de classes distintas estão próximos do intervalo entre -0.25 e 0.25, buscando aproximar uma independência entre elas.

	AWSS@95%		
Banco de Dados	Híbrido	Lora	base
ADHD	0.78	0.77	0.71
BPA	0.71	0.73	0.71
Fluoride	0.82	0.80	0.77
PFOS-PFOA	0.70	0.67	0.65
Transgenerational	0.50	0.47	0.49

Tabela 5: Comparação entre abordagens e os valores do AWSS@95% para os 5 melhores desempenhos do modelo.

Para uma análise mais detalhada, treinou-se a regressão logística nos dados de treino vetorizados pelos respectivos modelos e classificou-os nos respectivos bancos de

dados. Os resultados dessa análise são apresentados na Tabela 5, utilizando a métrica AWSS com 95% de recall. Observou-se que em alguns casos o modelo apresenta uma degradação na métrica, enquanto em outros casos há uma melhoria. Por exemplo, no banco de dados ADHD, foi observada uma melhoria significativa, possivelmente indicando um processo de regularização, conforme ilustrado na Figura 18, imagem (c), especialmente em relação aos exemplos positivos conforme o canto superior esquerdo.

4.4 Discussão e Trabalhos Futuros

Conforme os resultados na Seção 4.2, o modelo base possui boa representação vetorial dos textos entre os bancos de dados como ilustrado na Figura 14, mas não possui os agrupamentos locais entre as classes do respectivo banco de dados. Ao aplicar o ajuste fino utilizando o aprendizado contrastivo, o modelo ganha confiança para para separar estas classes como ilustrado na Figura 15, e ao aplicar a regressão logística, nota-se as densidades convergindo para os valores extremos, conforme a Figura 16.

No entanto, ao aplicar a mesma metodologia de treino para diferentes bancos de dados, o ajuste fino não desempenha melhora significativa na métrica de trabalho salvo, Figura 17. Isso sugere que os contextos utilizados para a vetorização através do modelo base podem conter informações conflitantes, dificultando a melhoria significativa do agrupamento de classes e até mesmo levando à degradação das habilidades originais do modelo, como observado no caso do banco de dados Estrogens.

Por outro lado, ao analisar os métodos de otimização aplicados ao modelo, como a quantização e a poda, observamos que, na maioria dos casos, não há uma degradação significativa na métrica de desempenho. Isso possibilita reduzir consideravelmente a carga computacional necessária para a utilização desses modelos, chegando a uma redução de até 8 vezes no tamanho original do modelo, ou seja, cerca de 52MB. Isso viabiliza a implementação desses modelos em hardwares especializados ou até mesmo a transferência para outros servidores ou dispositivos para download.

Seguindo a mesma linha de discussão, embora as habilidades específicas adquiridas durante o treinamento do modelo não tenham gerado melhorias significativas para alguns bancos de dados em questão, é notável que os módulos de adaptação do modelo representam menos de 0.03% do total de parâmetros do modelo. Isso viabiliza uma transferência eficiente de habilidades entre servidores e usuários para o mesmo modelo base, com um custo computacional mínimo, ocupando apenas 144KB, ou em casos mais extremos utilizando precisões em 4 bits, ocupando apenas 18KB. Essa capacidade abre caminho

para um ecossistema rico em novas habilidades, construídas sobre um mesmo modelo.

Ao construir um ecossistema de habilidades específicas, também é importante analisar a capacidade do modelo de integrar essas habilidades e torná-las híbridas. Ao aplicar a metodologia descrita na seção 3.3.3 nos 5 melhores resultados da métrica AWSS@95%, observamos que é possível preservar algumas das habilidades originais do modelo, bem como incorporar novas habilidades locais. Em alguns casos, essas novas habilidades emergem como agrupamentos locais, conforme destacado na Figura 18 (b), em contraste com a representação original na Figura 18 (a). Isso também é evidente na Figura 18 (c), onde a similaridade das representações do modelo ajustado é quase mantida intacta em comparação com o modelo hibrido conforme a Figura 18 (d), perdendo apenas em confiança, mas mantendo as representações agrupadas.

Dessa forma, à medida que a sociedade avança e novos hardwares e softwares são desenvolvidos, os modelos base disponíveis permitem a construção sustentável de novos ecossistemas. É relevante ressaltar neste trabalho que novas representações para o modelo base foram adquiridas com apenas 8 exemplos por classe em poucos segundos, tornando o custo computacional dessas novas habilidades praticamente insignificante. Isso abre possibilidades para diversas aplicações em casos mais específicos e contribui para a democratização do conhecimento.

Embora as configurações de treinamento para alguns bancos de dados não tenham lidado adequadamente com a complexidade dos respectivos dados, os métodos estudados aqui oferecem uma ampla gama de possibilidades para melhorias em trabalhos futuros. Eles permitem explorar desde diferentes regimes de treinamento até estratégias para agrupar as habilidades do modelo.

No entanto, mesmo diante de desafios em alguns bancos de dados, os métodos de otimização são aplicáveis independentemente do ajuste fino do modelo. Um dos trabalhos futuros a serem realizados é avaliar os métodos de otimização diretamente no modelo base sem adaptações. Outra área de pesquisa interessante seria realizar a destilação de conhecimento das camadas do Transformers, conforme proposto por Wang et al. (2020b), especialmente nas camadas que não foram adaptadas durante o processo de ajuste fino.

Além disso, com o crescente interesse na sustentabilidade e na redução da pegada de carbono, entender o impacto ambiental das atividades de treinamento e uso de modelos de inteligência artificial tornou-se uma preocupação importante. Incorporar medidas para estimar as emissões de carbono associadas ao uso desses modelos pode ser um desafio, como demonstrado em estudos como Patterson et al. (2021), Strubell, Ganesh

e McCallum (2019), Dodge et al. (2022), mas contribui significativamente para práticas mais ecológicas na área de IA. Uma investigação mais aprofundada posteriormente sobre técnicas e metodologias para inferir essas emissões é essencial para futuras pesquisas.

Por fim, este estudo demonstra diversas possibilidades de aprimoramento, focando na análise qualitativa e quantitativa dos métodos de otimização, o que apresenta um novo paradigma para as aplicações desenvolvidas para a sociedade. Este trabalho incentiva a pesquisa e a construção de uma nova metodologia para o uso da IA, fundamentada em um ecossistema rico de habilidades e aplicações sustentáveis, visando o benefício tanto da sociedade quanto da natureza como um todo.

5 Conclusão

Este estudo explorou diversas técnicas e metodologias para aprimorar modelos de aprendizado profundo, visando melhorar sua eficiência e adaptabilidade em diferentes contextos. Ao longo da pesquisa, foi investigado desdes os componentes básicos de modelos de redes neurais artificiais, até métodos de ajuste fino eficientes, otimização e hibridação de modelos, além de avaliar sua aplicabilidade em uma variedade de conjuntos de dados de revisão sintemática da literatura e estruturar o conhecimento de modelos base pela arquitetura Transformers.

Observamos que, embora que as configurações de treino empregadas para a abordagem de ajuste fino do modelo não tenham obtido sucesso na métrica de trabalho salvo em todos os cenários avaliados, este abriu portas para uma série de possibilidades de melhoria e inovação. A análise qualitativa e quantitativa dos métodos de otimização revelou insights valiosos sobre o desempenho e a robustez dos modelos, fornecendo um novo paradigma para o desenvolvimento de aplicações baseadas em IA, ressaltando a importância de um ecossistema rico de habilidades específicas para os modelos e sustentáveis, trazendo benefícios tanto para a sociedade quanto para a preservação da natureza.

Dessa forma, este trabalho oferece uma contribuição significativa para o campo da inteligência artificial, incentivando a continuidade da pesquisa e desenvolvimento de soluções que atendam às necessidades da sociedade de forma ética, eficiente e sustentável.

Referências

AINSLIE, J. et al. *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints*. 2023.

BA, J.; KIROS, J. R.; HINTON, G. E. Layer normalization. *ArXiv*, abs/1607.06450, 2016. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:8236317⟩.

BANNACH-BROWN, A. et al. Machine learning algorithms for systematic review: reducing workload in a preclinical review of animal studies and reducing human screening error. *Systematic reviews*, BioMed Central, v. 8, n. 1, p. 1–12, 2019.

BELTAGY, I.; LO, K.; COHAN, A. Scibert: A pretrained language model for scientific text. In: *Conference on Empirical Methods in Natural Language Processing*. [s.n.], 2019. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:202558505⟩.

BROHAN, A. et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: *arXiv preprint arXiv:2307.15818*. [S.l.: s.n.], 2023.

BROWN, T. B. et al. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. Disponível em: ⟨https://arxiv.org/abs/2005.14165⟩.

BUBECK, S. et al. *Sparks of Artificial General Intelligence: Early experiments with GPT-4*. 2023.

CHEN, T. et al. Big self-supervised models are strong semi-supervised learners. *ArXiv*, abs/2006.10029, 2020. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:219721239⟩.

CHOROMANSKI, K. et al. *Rethinking Attention with Performers*. 2022.

COHAN, A. et al. Specter: Document-level representation learning using citation-informed transformers. *ArXiv*, abs/2004.07180, 2020. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:215768677⟩.

COHEN, A. M. et al. Reducing workload in systematic review preparation using automated citation classification. *Journal of the American Medical Informatics Association*, BMJ Group BMA House, Tavistock Square, London, WC1H 9JR, v. 13, n. 2, p. 206–219, 2006.

CSÁJI, B. C. et al. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, Citeseer, v. 24, n. 48, p. 7, 2001.

DAO, T. et al. *FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness*. 2022.

DETTMERS, T. et al. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *ArXiv*, abs/2208.07339, 2022. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:251564521⟩.

DETTMERS, T. et al. *QLoRA: Efficient Finetuning of Quantized LLMs*. 2023.

DEVLIN, J. et al. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. Disponível em: ⟨http://arxiv.org/abs/1810.04805⟩.

DING, Y. et al. Task and motion planning with large language models for object rearrangement. *ArXiv*, abs/2303.06247, 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:257496672⟩.

DINTER, R. van; CATAL, C.; TEKINERDOGAN, B. A multi-channel convolutional neural network approach to automate the citation screening process. *Applied Soft Computing*, Elsevier, v. 112, p. 107765, 2021.

DODGE, J. et al. Measuring the carbon intensity of ai in cloud instances. *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.

DOSOVITSKIY, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. Disponível em: ⟨https://arxiv.org/abs/2010.11929⟩.

FARIA, A. V. A. et al. Automated slr with a few labeled papers and a fair workload metric. In: *International Conference on Web Information Systems and Technologies*. [s.n.], 2022. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:261683391⟩.

FAWZI, A. et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, v. 610, p. 47 – 53, 2022. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:252717185⟩.

FOURNARAKIS, M. *A Practical Guide to Neural Network Quantization*. 2021. Apresentação de Slide. Disponível em: ⟨https://cms.tinyml.org/wp-content/uploads/industry-news/tinyML_Talks-_Marios_Fournarakis_210929.pdf⟩.

FRANTAR, E. et al. Gptq: Accurate post-training quantization for generative pre-trained transformers. *ArXiv*, abs/2210.17323, 2022. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:253237200⟩.

GANDHI, K. et al. Understanding social reasoning in language models with language models. *ArXiv*, abs/2306.15448, 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:259262573⟩.

GUNASEKAR, S. et al. *Textbooks Are All You Need*. 2023.

HE, J. et al. Towards a unified view of parameter-efficient transfer learning. *CoRR*, abs/2110.04366, 2021. Disponível em: ⟨https://arxiv.org/abs/2110.04366⟩.

HE, K. et al. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 770–778, 2015. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:206594692⟩.

HINTON, G.; VINYALS, O.; DEAN, J. *Distilling the Knowledge in a Neural Network*. 2015.

HOWARD, B. E. et al. Swift-review: a text-mining workbench for systematic review. *Systematic reviews*, Springer, v. 5, n. 1, p. 1–16, 2016.

HU, E. J. et al. LoRA: Low-rank adaptation of large language models. In: *International Conference on Learning Representations*. [s.n.], 2022. Disponível em: ⟨https://openreview.net/forum?id=nZeVKeeFYf9⟩.

HU, Z. et al. *LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models*. 2023.

JAEGLE, A. et al. *Perceiver: General Perception with Iterative Attention*. 2021.

JUMPER, J. et al. Highly accurate protein structure prediction with alphafold. *Nature*, Nature Publishing Group, v. 596, n. 7873, p. 583–589, 2021.

KALAMKAR, D. D. et al. A study of bfloat16 for deep learning training. *ArXiv*, abs/1905.12322, 2019. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:168170136⟩.

KEELE, S. et al. *Guidelines for performing systematic literature reviews in software engineering*. [S.l.]: Technical report, ver. 2.3 ebse technical report. ebse, 2007.

KHOSLA, P. et al. *Supervised Contrastive Learning*. 2021.

KIELA, D. et al. *Dynabench: Rethinking Benchmarking in NLP*. 2021.

KURTIC, E. et al. *The Optimal BERT Surgeon: Scalable and Accurate Second-Order Pruning for Large Language Models*. 2022.

KUSA, W. et al. An analysis of work saved over sampling in the evaluation of automated citation screening in systematic literature reviews. *Intelligent Systems with Applications*, 2023.

LECLERC, G. et al. *FFCV: Accelerating Training by Removing Data Bottlenecks*. 2023.

LEE-THORP, J. et al. *FNet: Mixing Tokens with Fourier Transforms*. 2022.

LI, J. et al. *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. 2023.

LI, J. et al. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. *CoRR*, abs/2201.12086, 2022. Disponível em: ⟨https://arxiv.org/abs/2201.12086⟩.

LIALIN, V. et al. Stack more layers differently: High-rank training through low-rank updates. *ArXiv*, abs/2307.05695, 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:259836974⟩.

LIAO, B.; TAN, S.; MONZ, C. *Make Your Pre-trained Model Reversible: From Parameter to Memory Efficient Fine-Tuning*. 2023.

LIEBENWEIN, L. *Efficient Deep Learning: From Theory to Practice*. Tese (Doutorado) — Massachusetts Institute of Technology, 2021.

LIN, J. et al. Awq: Activation-aware weight quantization for llm compression and acceleration. *ArXiv*, abs/2306.00978, 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:258999941⟩.

LIU, H. et al. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *ArXiv*, abs/2305.14342, 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:258841030⟩.

LIU, H. et al. *Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning*. 2022.

LUCCIONI, A. S.; VIGUIER, S.; LIGOZAT, A.-L. Estimating the carbon footprint of bloom, a 176b parameter language model. *ArXiv*, abs/2211.02001, 2022.

MAATEN, L. van der; HINTON, G. E. Visualizing data using t-sne. *Journal of Machine Learning Research*, v. 9, p. 2579–2605, 2008. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:5855042⟩.

MARR, D. *Vision: A computational investigation into the human representation and processing of visual information*. [S.l.]: MIT press, 2010.

MARTINS, P. H.; MARINHO, Z.; MARTINS, A. F. T. *∞-former: Infinite Memory Transformer*. 2022.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, p. 115–133, 1943.

MELO, M. K. de et al. Few-shot approach for systematic literature review classifications. In: INSTICC. *Proceedings of the 18th International Conference on Web Information Systems and Technologies - Volume 1: WEBIST,*. [S.l.]: SciTePress, 2022. p. 33–44. ISBN 978-989-758-613-2.

MINSKY, M.; PAPERT, S. Perceptrons - an introduction to computational geometry. In: . [S.l.: s.n.], 1969.

NAGEL, M. et al. A white paper on neural network quantization. *ArXiv*, abs/2106.08295, 2021. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:235435934⟩.

OPENAI. *GPT-4 Technical Report*. 2023.

PACKER, C. et al. MemGPT: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

PATTERSON, D. A. et al. Carbon emissions and large neural network training. *CoRR*, abs/2104.10350, 2021. Disponível em: ⟨https://arxiv.org/abs/2104.10350⟩.

PESTE, A. et al. AC/DC: alternating compressed/decompressed training of deep neural networks. *CoRR*, abs/2106.12379, 2021. Disponível em: ⟨https://arxiv.org/abs/2106.12379⟩.

RADFORD, A.; NARASIMHAN, K. Improving language understanding by generative pre-training. In: . [s.n.], 2018. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:49313245⟩.

RADFORD, A. et al. Improving language understanding by generative pre-training. OpenAI, 2018.

RAJBHANDARI, S. et al. *ZeRO: Memory Optimizations Toward Training Trillion Parameter Models*. 2020.

RAMSAUER, H. et al. Hopfield networks is all you need. *CoRR*, abs/2008.02217, 2020. Disponível em: ⟨https://arxiv.org/abs/2008.02217⟩.

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019. Disponível em: ⟨http://arxiv.org/abs/1908.10084⟩.

ROMBACH, R. et al. High-resolution image synthesis with latent diffusion models. *CoRR*, abs/2112.10752, 2021. Disponível em: ⟨https://arxiv.org/abs/2112.10752⟩.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, v. 65 6, p. 386–408, 1958.

ROSER, M.; RITCHIE, H.; MATHIEU, E. Technological change. *Our World in Data*, 2023. Https://ourworldindata.org/technological-change.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representations by error propagation. In: . [S.l.: s.n.], 1986.

SAXENA, S. et al. *Sparse Iso-FLOP Transformations for Maximizing Training Efficiency*. 2023.

SELLAK, H.; OUHBI, B.; FRIKH, B. Using rule-based classifiers in systematic reviews: a semantic class association rules approach. In: *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*. [S.l.: s.n.], 2015. p. 1–5.

SEVIM, N. et al. *Fast-FNet: Accelerating Transformer Encoder Models via Efficient Fourier Layers*. 2023.

SHAZEER, N. *Fast Transformer Decoding: One Write-Head is All You Need*. 2019.

SHEN, Y. et al. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *ArXiv*, abs/2303.17580, 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:257833781⟩.

SINGHAL, K. et al. *Towards Expert-Level Medical Question Answering with Large Language Models*. 2023.

SONG, Y. et al. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, v. 55, p. 1 – 40, 2022. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:248798765⟩.

SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, v. 15, p. 1929–1958, 2014. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:6844431⟩.

STRUBELL, E.; GANESH, A.; MCCALLUM, A. Energy and policy considerations for deep learning in nlp. *ArXiv*, abs/1906.02243, 2019.

SUN, M. et al. *A Simple and Effective Pruning Approach for Large Language Models*. 2023.

THANGARASA, V. et al. *SPDF: Sparse Pre-training and Dense Fine-tuning for Large Language Models*. 2023.

TOMLINSON, B. et al. The carbon emissions of writing and illustrating are lower for ai than for humans. *ArXiv*, abs/2303.06219, 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:257496530⟩.

TOMLINSON, B.; TORRANCE, A.; RIPPLE, W. J. Scientists' warning on technology. *ArXiv*, abs/2304.11271, 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:258298351⟩.

TSAFNAT, G. et al. Automated screening of research studies for systematic reviews using study characteristics. *Systematic reviews*, Springer, v. 7, n. 1, p. 1–9, 2018.

TUNSTALL, L. et al. *Efficient Few-Shot Learning Without Prompts*. 2022.

VASWANI, A. et al. Attention is all you need. In: *NIPS*. [S.l.: s.n.], 2017.

WANG, F.; LIU, H. *Understanding the Behaviour of Contrastive Loss*. 2021.

WANG, S. et al. *Linformer: Self-Attention with Linear Complexity*. 2020.

WANG, W. et al. *MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers*. 2020.

WORKSHOP, B. et al. *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model*. 2023.

WU, C. et al. *Fastformer: Additive Attention Can Be All You Need*. 2021.

XI, H. et al. *Training Transformers with 4-bit Integers*. 2023.

XI, Z. et al. The rise and potential of large language model based agents: A survey. *ArXiv*, abs/2309.07864, 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:261817592⟩.

XIA, M. et al. Sheared llama: Accelerating language model pre-training via structured pruning. *ArXiv*, abs/2310.06694, 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:263830786⟩.

XIE, S. M. et al. *DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining*. 2023.

XIE, T. et al. Openagents: An open platform for language agents in the wild. *ArXiv*, abs/2310.10634, 2023. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:264172893⟩.

YANG, G. et al. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *ArXiv*, abs/2203.03466, 2022. Disponível em: ⟨https://api.semanticscholar.org/CorpusID:247292726⟩.

ZHANG, J.; DENG, N.; CHEN, L. New quasi-newton equation and related methods for unconstrained optimization. *Journal of Optimization Theory and Applications*, Springer, v. 102, n. 1, p. 147–167, 1999.