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
2025

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
2025

Resumo

Este estudo investiga a sustentabilidade de aplicações em aprendizado profundo, com ênfase na eficiência computacional e especialização de modelos, explorando a arquitetura Transformer e técnicas de otimização para aprimorar sua eficiência e adaptabilidade, tornando essas tecnologias mais acessíveis e alinhadas às necessidades humanas e ambientais. Destaca-se a importância de um ecossistema flexível, no qual modelos podem ser treinados, otimizados e compartilhados de forma sustentável. Para isso, realizou-se um estudo de caso sobre a triagem de artigos relevantes, utilizando apenas oito exemplos por classe em 22 bases de dados de revisões sistemáticas da literatura. Foram avaliadas técnicas de otimização, como poda, quantização e composição eficiente de habilidades ajustadas, aplicadas ao ajuste fino do modelo Transformer SPECTER, empregado para a representação semântica de trechos de textos científicos. O desempenho foi mensurado pela métrica quantitativa de trabalho salvo. Os desafios identificados ressaltam a necessidade de explorar novos regimes de treinamento para aprimorar a adaptabilidade dos modelos e desenvolver estratégias para mensurar o impacto ambiental do uso contínuo dessas tecnologias. Além disso, discute-se o papel da eficiência computacional na promoção de avanços científicos, otimizando pipelines de uso contínuo para prototipagem de soluções e integração de conhecimento interdisciplinar de soluções baseadas em revisões sistemáticas da literatura.

Palavras-chave: aprendizado profundo. sustentabilidade. eficiência computacional. Transformer. otimização de modelos. representação semântica. revisão sistemática.

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. 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 têm demonstrado que estes modelos têm fortes habilidades em adaptação para a inicialização em diversos domínios, dentre os quais 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, essas 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 das representações da linguagem humana por esses modelos emerge como um fator essencial para impulsionar o desenvolvimento de sistemas de IA mais avançados. Nesse contexto, o GPT-4 da OpenAI (OPENAI, 2023) demonstra habilidades impressionantes, comparáveis às humanas em jogos interpretativos, incluindo a teoria da mente. Essa teoria avalia a capacidade do ser humano de atribuir representações independentes a si mesmo e aos outros, o que implica na habilidade de compreender e atribuir estados mentais, como crenças, emoções e intenções. Essa capacidade, por sua vez, contribui 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, é essencial garantir que os avanços dessa tecnologia sejam utilizados de forma responsável. A implementação de regulamentações adequadas e a adoção de uma governança ética são fundamentais para assegurar que os benefícios da IA sejam acessíveis a todos. Este trabalho propõe uma abordagem mais sustentável para o desenvolvimento de sistemas de IA avançados, aplicando métodos de otimização em redes neurais profundas para reduzir o consumo de recursos computacionais e mitigar os impactos ambientais, promovendo a disseminação responsável dessa tecnologia emergente e gerando benefícios para a sociedade. Dessa forma, busca-se maximizar as capacidades dos modelos de IA disponíveis, equilibrando inovação com preocupações ambientais e sociais.

1.3 Objetivos do Trabalho

Este trabalho tem como objetivo central investigar e avaliar técnicas de otimização e aprimoramento de modelos de aprendizado profundo para a sustentabilidade de ecossistemas de aplicações energeticamente eficientes, com ênfase no estudo de caso da triagem automatizada de documentos científicos relevantes em revisões sistemáticas da literatura. O intuito é fornecer suporte a avanços tecnológicos e científicos que atendam às necessidades da sociedade de maneira eficaz e responsável.

1.3.1 Objetivos Específicos

- **Estudar técnicas de aprendizado profundo e modelos Transformers**: Explorar os fundamentos do aprendizado profundo, redes neurais artificiais e o funcionamento do mecanismo de atenção, incluindo modelos baseados na arquitetura Transformer, como o modelo SPECTER na representação semântica de textos de científicos.
- **Estudar as principais técnicas de otimização**: Explorar os fundamentos das técnicas de otimização aplicadas a modelos de aprendizado profundo.
- **Investigar estratégias de treinamento eficientes para modelos de base**: Examinar abordagens de pré-treinamento e ajuste fino, buscando um desempenho sustentável e eficaz no treinamento de modelos de base.
- **Avaliar o desempenho e a eficiência computacional do modelo de base com técnicas de otimização na triagem eficiente de documentos científicos**: Avaliar o impacto das técnicas de otimização na eficiência computacional e na redução do esforço necessário para a triagem automatizada de documentos científicos relevantes em revisões sistemáticas da literatura. Aplicar a análise a 22 bancos de dados para verificar a viabilidade dessas otimizações, considerando um número limitado de exemplos de treino.
- **Discutir resultados, sustentabilidade e direções futuras para pesquisas baseadas em aprendizado profundo**: Identificar desafios, limitações e possíveis melhorias na abordagem de otimização utilizada. Além disso, discutir a sustentabilidade de ecossistemas de aplicações baseadas em aprendizado profundo e seu impacto no avanço científico.

2 Fundamentação Teórica

2.1 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:** Neste tipo de aprendizagem, o modelo é treinado com dados não rotulados e busca descobrir padrões, estruturas ou agrupamentos por conta própria. Técnicas como *t-SNE* são usadas para reduzir a dimensionalidade dos dados e visualizar como diferentes amostras se organizam no espaço, revelando agrupamentos naturais mesmo sem categorias definidas previamente (MAATEN; HINTON, 2008).

- **Aprendizagem Auto-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.

- **Aprendizado Contrastivo:** Técnica utilizada no treinamento de modelos que visa aproximar representações semelhantes e afastar as distintas no espaço vetorial. Modelos como o **SPECTER** (COHAN et al., 2020) aplicam essa abordagem ao treinar com pares de artigos científicos que se citam (positivos) e que não se citam (negativos), gerando embeddings mais informativos para tarefas como recomendação, predição de citações e classificação de textos.

Na fase inicial de pré-treinamento, o aprendizado auto-supervisionado permite que o modelo desenvolva uma compreensão autônoma das estruturas e padrões dos dados. Esse conhecimento pode ser transferido para tarefas específicas por meio de ajuste fino (*fine-tuning*), em um processo conhecido como Transferência de Aprendizado. Essa abordagem tem se mostrado fundamental na construção de modelos de base, ao permitir que arquiteturas mais complexas aproveitem conhecimento prévio e sejam facilmente adaptadas a aplicações com poucos dados. Técnicas como o aprendizado contrastivo e por reforço, aplicadas sobre modelos base, têm sido essenciais para gerar representações mais discriminativas e informativas, impulsionando o avanço dos sistemas de inteligência artificial (ROMBACH et al., 2021; LI et al., 2022; LI et al., 2023; SHEN et al., 2023; OPENAI, 2023).

2.2 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, à medida que estes algoritmos evoluem, o conceito de "inteligência artificial" torna-se cada vez mais subjetivo ă 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 anălise (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 ă implementação física real do sistema.

Nesta seção, abordaremos os princípios băsicos 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).

2.2.1 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 viăvel aproximar qualquer função, desde que os pesos adequados sejam aplicados ă 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).

O modelo Perceptron, proposto por Rosenblatt (1958), é uma versão aprimorada do primeiro modelo apresentado por McCulloch e Pitts (1943). No entanto, uma de suas características mais marcantes é a limitação na resolução de problemas mais complexos, como classificações que não podem ser separadas linearmente. Esse desafio foi evidenciado pelo famoso problema XOR, proposto por Minsky e Papert (1969) em seu livro *Perceptrons: An Introduction to Computational Geometry.* A dificuldade em resolver esse problema levou a uma desmotivação na pesquisa de redes neurais por cerca de 20 anos.

Assim, considerando $x \in \mathbb{R}^d$ como o dado de entrada e $y \in \mathbb{R}$ como o dado de saída, o Perceptron pode ser representado por:

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

Esse modelo consiste em uma unidade de processamento básica que recebe um conjunto de entradas ponderadas por pesos w e aplica uma função de ativação, denotada por $\phi(.)$, para produzir uma saída.

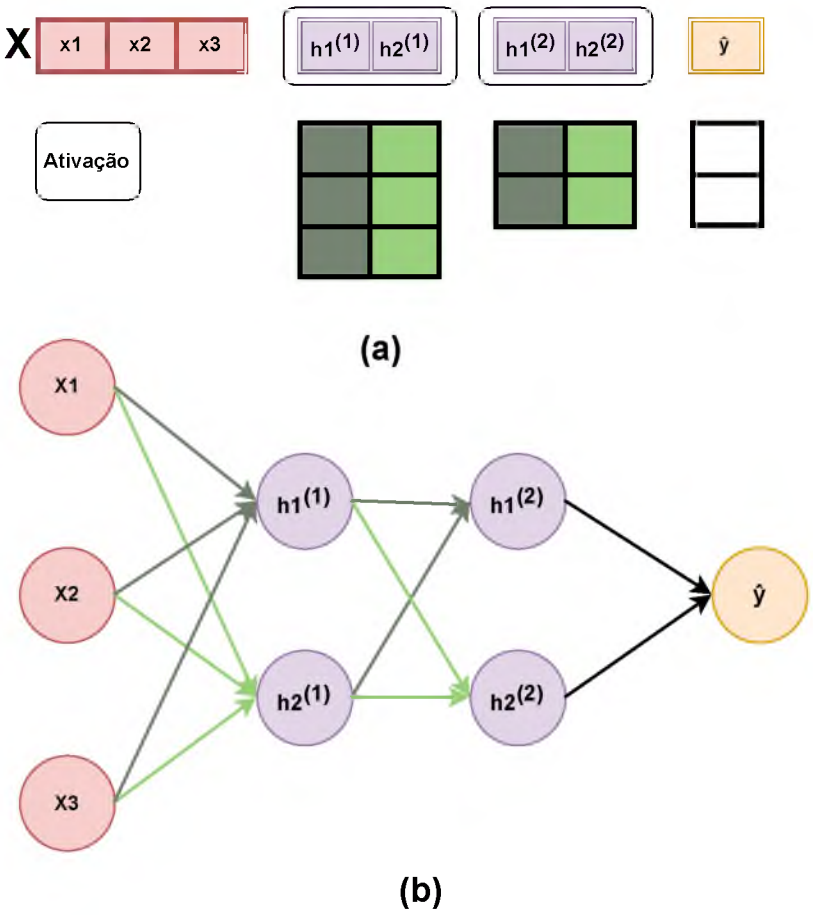

Figura 1: *Multilayer Perceptron.*

O *Multilayer Perceptron* (MLP), proposto por Rumelhart, Hinton e Williams (1986), foi desenvolvido para superar essas limitações e expandir as capacidades do Perceptron tradicional, impulsionando significativamente a pesquisa em redes neurais nas últimas três décadas. Essa arquitetura é composta por várias camadas de perceptrons interconectados, onde cada camada recebe as saídas dos perceptrons da camada anterior como entrada, aplicando funções de ativação para transformar essas informações.

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:

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

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 fundamental na pesquisa em redes neurais. Essas funções, geralmente não lineares, têm o objetivo de proporcionar representações mais complexas ao modelo entre as camadas. Na literatura, algumas das funções mais comuns são monotonicamente crescentes, conforme ilustrado na Figura 2.

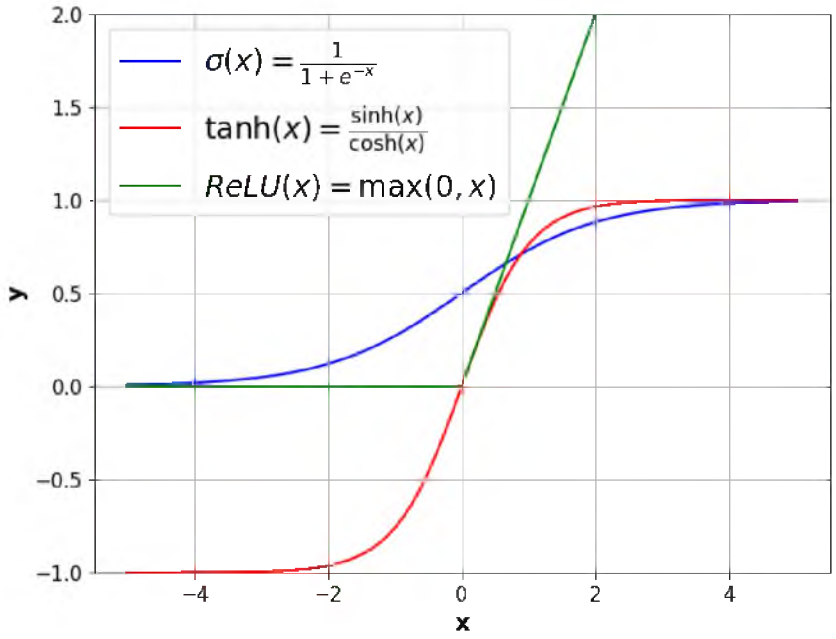

Figura 2: Funções de Ativação.

No exemplo apresentado, $\phi_3(\cdot)$ corresponde a uma função identidade ($f(x) = x$), utilizada na última camada. Dessa forma, o uso das funções não lineares $\phi_1(.)$ e $\phi_2(.)$ nas camadas anteriores permite modelar uma regressão não linear. A escolha dessas funções de ativação depende do nível de representação desejado pelo pesquisador e da otimização

do treinamento.

2.2.2 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:

$$\begin{aligned} \nabla W_t &= \nabla_W E_{X_k}[L_W] \\ W_{t+1} &:= W_t - \eta \nabla W_t \end{aligned} \tag{2}$$

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.

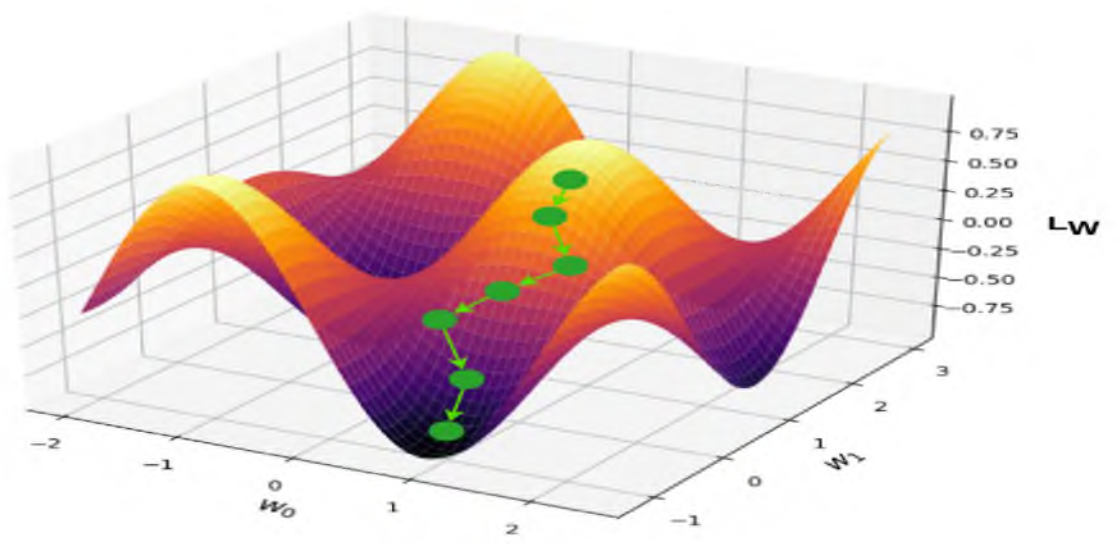

Figura 3: Descida do gradiente.

2.2.3 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).

2.2.4 Embedding

Os embeddings são amplamente utilizados devido à sua capacidade de representar sequências de entrada como vetores que preservam relações espaciais. Sua formulação baseia-se na representação de um vocabulário de símbolos, onde, ao passar por uma função de dicionário, cada símbolo é mapeado para um vetor correspondente.

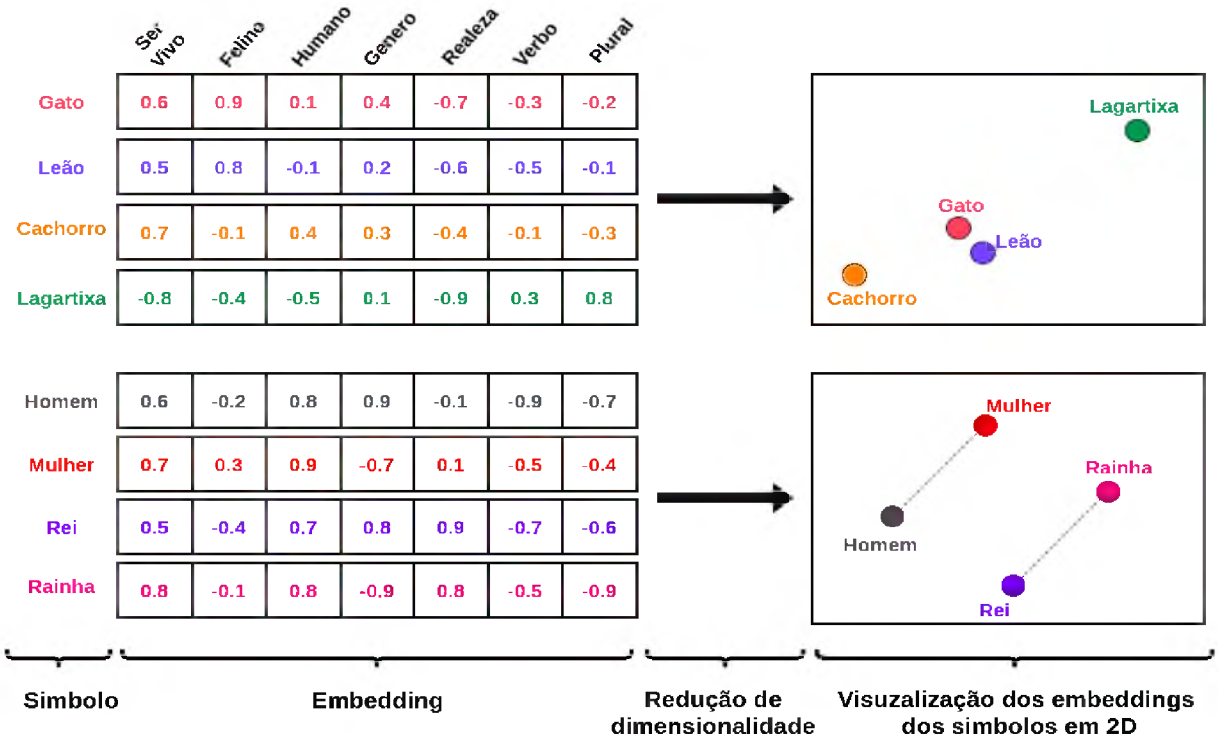

Figura 4: Representação dos Embeddings.

Após o treinamento, esses embeddings adquirem a habilidade de codificar relações simbólicas espaciais. As dimensões de seus vetores podem ser interpretadas como características representacionais dos símbolos do dicionário, possibilitando operações lógicas e simbólicas modeladas por um algoritmo de aprendizado.

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.

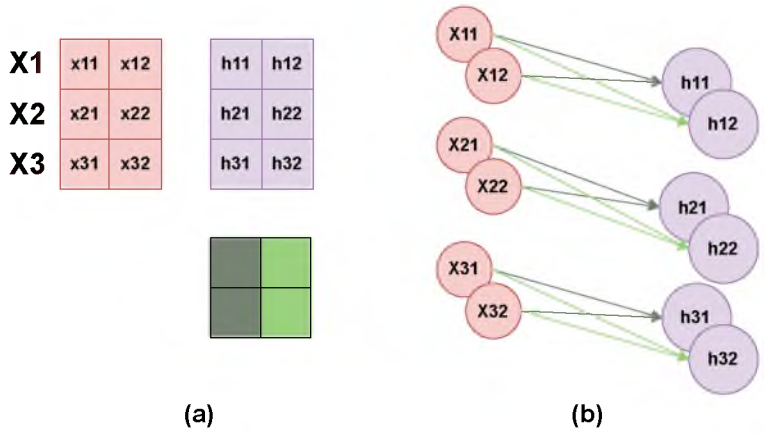

Figura 5: Computação Neural de uma sequência.

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.

2.2.5 Mecanismo de Atenção

O mecanismo de atenção, proposto por Vaswani et al. (2017), tem se mostrado um componente fundamental nas arquiteturas de aprendizado profundo. Esse mecanismo permite que o modelo atribua maior importância a determinadas partes do dado de entrada, destacando informações relevantes para a tarefa em questão.

Ao representar um vetor de entrada como $x \in \mathbb{R}^{1\times d}$, o mecanismo de atenção projeta essa entrada em três representações principais: *Query* (Q), *Key* (K) e *Value* (V). Essas representações são obtidas por meio de projeções lineares, parametrizadas pelos pesos aprendidos durante o treinamento, utilizando $W_Q, W_K, W_V \in \mathbb{R}^{d\times k}$, onde d é a dimensão original do vetor de entrada e k a dimensão latente (ou do espaço de atenção). Generalizando para uma sequência de entrada, seja, $X^T = [x_1 \ \ldots \ x_n] \in \mathbb{R}^{d\times n}$, em

que $X \subset E$ representa vetores no espaço representacional E (por exemplo, embeddings), obtemos as matrizes $Q, K, V \in \mathbb{R}^{n \times k}$, que contèm as projeções latentes dos n elementos da sequència. Essas representações compactas viabilizam o cálculo das similaridades internas entre os vetores por meio do mecanismo de atenção, definido como:

$$\begin{gathered} Q = XW_Q, \quad K = XW_K, \quad V = XW_V, \\ \alpha = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right), \quad Attention_X(Q, K, V) = \alpha V, \end{gathered} \tag{3}$$

Aplicando a função $\text{softmax}(x_i) = \exp(x_i)/\sum_{j=1}^{n} \exp(x_j)$ aos produtos escalares resultantes de QK^T, obtém-se a matriz estocástica $\alpha \in \mathbb{R}^{n \times n}$. Esses produtos são normalizados pelo termo $d_k = k$, que limita a variância dos valores e garante maior estabilidade numérica e dos gradientes durante o treinamento. Em seguida, o produto $\alpha V \in \mathbb{R}^{n \times k}$ representa uma soma ponderada das projeções em V, onde os pesos são determinados pelas similaridades entre as representações em Q e K, conforme expresso em α. Esse resultado modela a influència de cada vetor latente da sequència na formação da representação final de cada elemento pelo mecanismo de atenção, denotado por $Attention_X(Q, K, V) \in \mathbb{R}^{n \times k}$, oferecendo uma contextualização global.

Em suma, a função primordial do mecanismo de atenção é utilizar as representações abstratas dos símbolos de entrada, dadas por Q, K e V, e, por meio das similaridades entre eles (representadas por α), ponderar essas representações para gerar novos símbolos latentes no espaço representacional E. Essa abordagem possibilita a criação de novas representações do conteúdo de X, capturando as relações contextuais entre os elementos da sequència e produzindo uma nova sequència de símbolos abstratos, que podem ou não pertencer ao vocabulário original (embeddings).

Assim, o mecanismo de atenção oferece uma maneira eficaz de capturar e gerar informações relevantes, ponderando a importância dos diferentes elementos de entrada. Entretanto, uma desvantagem computacional reside no cálculo da multiplicação entre as matrizes Q e K, que pode se tornar custoso em função do tamanho da sequència de entrada. Estudos recentes, porém, demonstram que esse mecanismo pode armazenar padrões contextuais de forma exponencialmente eficiente, utilizando métodos simples de aprendizado interativo, como as chamadas *Hopfield Networks* (RAMSAUER et al., 2020).

2.2.6 Transformers

Os Transformers, introduzidos por Vaswani et al. (2017) em 2017, revolucionaram o Aprendizado de Máquina ao empregar o mecanismo de atenção para capturar relações complexas e processar sequências em paralelo. Treinados de forma auto-supervisionada, tornam-se modelos base robustos, impulsionando avanços por meio do aprendizado por transferência.

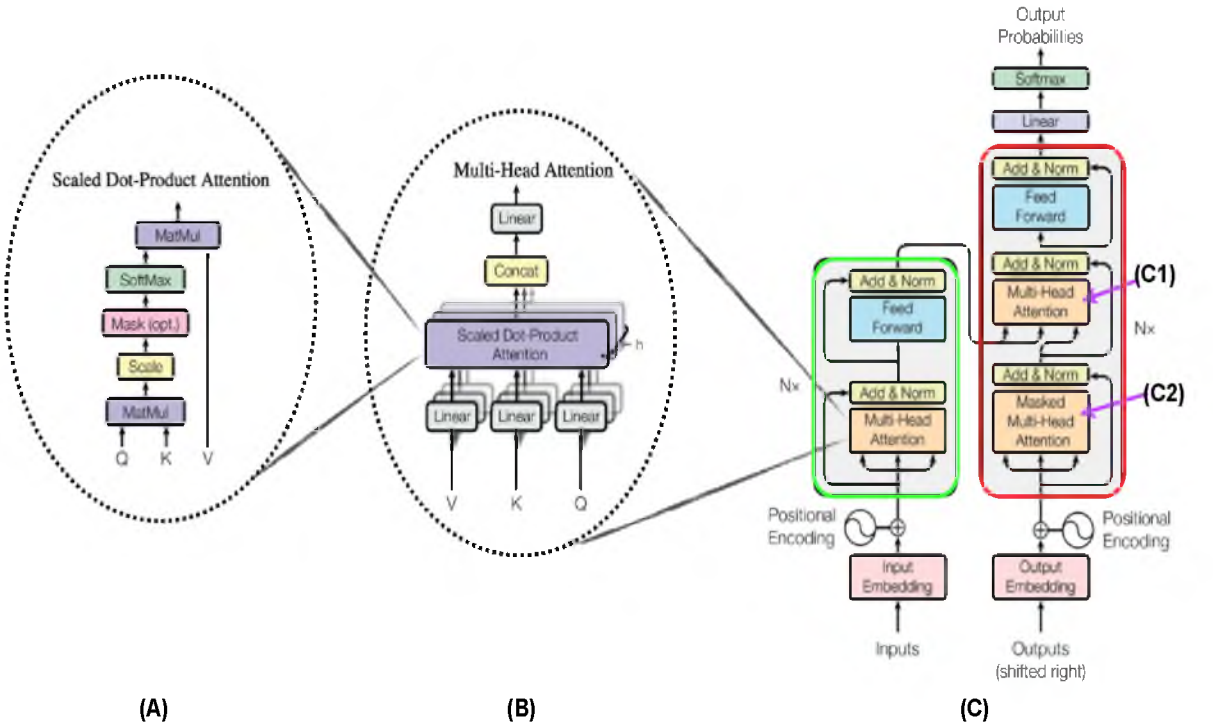

Figura 6: Estrutura do Transformer (VASWANI et al., 2017), incluindo Encoder, Decoder e Cross Attention.

A arquitetura Transformer, como demonstrado na Figura 6, generaliza o mecanismo de Atenção e é estruturada em três variações principais: **Encoder**, **Decoder** e uma combinação de ambos. O **Encoder** é projetado para aprendizado bidirecional, permitindo que o modelo capte relações contextuais em toda a sequência de entrada, como no BERT (DEVLIN et al., 2018). O **Decoder**, por outro lado, opera de forma auto-regressiva, gerando sequências com base nas entradas anteriores, característica fundamental do GPT (RADFORD; NARASIMHAN, 2018). Quando combinados por meio do *Cross Attention*, resultam na arquitetura Transformer original proposta por Vaswani et al. (2017).

O **Multi-Head Attention** ($MHA(Q, K, V)$) é o núcleo do Transformer, consistindo na aplicação simultânea de múltiplos mecanismos de atenção para capturar diferentes aspectos da sequência de entrada. Na Figura 7, é ilustrado o cálculo do MHA, no qual

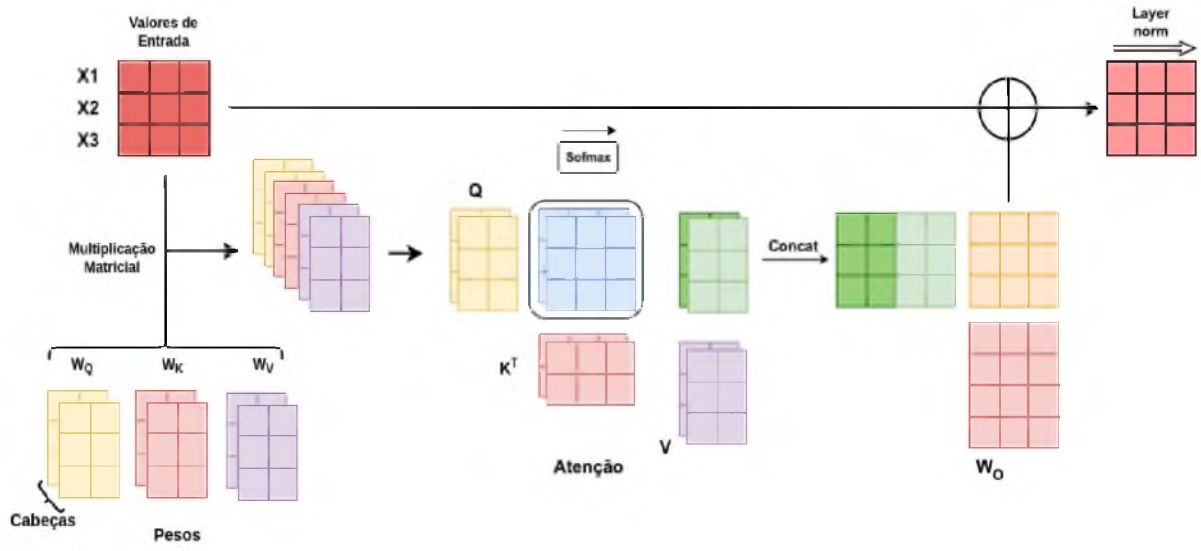

Figura 7: Representação computacional do *Multi-Head Attention*, com uma sequência de 3 entradas (X_1, X_2, X_3), 2 cabeças e 2 dimensões para Q, K e V.

cada cabeça gera suas próprias representações Q, K e V, computa os mapas de correlação α e os pondera de forma paralela. Em seguida, os resultados de todas as cabeças são concatenados e passam por uma projeção linear com $W_O \in \mathbb{R}^{n\times(k\cdot h)}$, onde h é o número de cabeças; essa projeção retorna as representações ao espaço original dos embeddings E. O processo é completado por uma **conexão residual** (HE et al., 2015), que soma os resultados à entrada original, e por uma **Layer Normalization** (BA; KIROS; HINTON, 2016), contribuindo para a estabilidade dos gradientes e a eficiência do treinamento.

Além do MHA, os Transformers incluem um **Feed Forward Network (FFN)** com ativação ReLU entre duas camadas densas, introduzindo a única não linearidade no modelo. Esse bloco Transformer é empilhado N vezes para proporcionar representações mais profundas, conforme ilustrado na Figura 6.

Os Transformers podem assumir diferentes configurações dependendo da tarefa. No **Encoder-Only**, a saída do Transformer passa por uma camada linear e normalização softmax, como no BERT, utilizado para modelagem de linguagem e compreensão contextual. No **Decoder-Only**, o modelo remove a conexão entre Encoder e Decoder e incorpora um **Masked Multi-Head Attention (Masked MHA)**, garantindo que a sequência de entrada atenda apenas aos tokens anteriores. Essa estrutura, amplamente adotada pelo GPT, pode ser interpretada como um processo auto-regressivo formulado como uma densidade de probabilidade condicional:

$$p_W(X_k|X_{k-1}, ..., X_1)$$

Na versão original do Transformer, representada na Figura 6 (C), o **Cross Attention** (C1) interliga os modelos Encoder e Decoder. Nessa configuração, o Encoder fornece as projeções de K e V, enquanto o Decoder utiliza a projeção Q, permitindo a geração de novas sequências a partir da entrada.

Os modelos Transformers baseiam-se em embeddings para representar palavras, adicionando *Positional Embeddings* para codificar a posição sequencial dos tokens. Na formulação original, esses embeddings são definidos como:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d})$$

onde pos é a posição na sequência, d é a dimensão do vetor e i é o índice da dimensão correspondente (VASWANI et al., 2017).

Embora o *Multi-Head Attention* seja altamente eficaz na captura de relações semânticas, sua implementação é computacionalmente custosa devido ao grande número de parâmetros e aos cálculos intensivos das matrizes de covariância α. O pré-treinamento de Transformers requer vastas quantidades de dados e o uso de hardwares especializados, como GPUs e TPUs, que empregam processamento massivamente paralelo otimizado para operações matriciais.

À medida que esses modelos crescem em escala e são amplamente adotados, a demanda computacional aumenta exponencialmente, gerando desafios críticos em consumo energético e escalabilidade. Para viabilizar o uso de Transformers em larga escala, é fundamental adotar técnicas de otimização que reduzam os custos operacionais, mitiguem impactos ambientais e promovam um uso mais sustentável da infraestrutura computacional.

2.2.7 Modelo SPECTER: Representação Semântica de textos Científicos

A classificação de documentos científicos exige modelos que capturem não apenas informações contextuais do texto, mas também relações entre publicações, como citações e relevância semântica. Modelos pré-treinados baseados na arquitetura **Transformer**, como o **BERT**, têm sido amplamente utilizados em tarefas de Processamento de Linguagem Natural (PLN), pois oferecem representações vetoriais eficientes (DEVLIN et al., 2018). Entretanto, esses modelos são predominantemente treinados em textos generalistas, como os da Wikipedia e do BookCorpus, e não são otimizados para as particularidades da literatura científica.

Para suprir essa limitação, variantes especializadas foram desenvolvidas, como o **SciBERT**, que continua o pré-treinamento do BERT utilizando um corpus composto por 1,14 milhão de artigos científicos do **Semantic Scholar**, totalizando 3,17 bilhões de tokens (BELTAGY; LO; COHAN, 2019). Esse ajuste possibilita uma melhor adaptação ao domínio acadêmico, aprimorando sua capacidade de representação para tarefas como reconhecimento de entidades nomeadas (NER), extração de informações científicas (PICO Extraction) e classificação de textos científicos.

Apesar da especialização do SciBERT, esse modelo ainda trata os artigos de maneira isolada, sem considerar a estrutura de citações que os interliga. Para superar essa limitação, foi desenvolvido o **SPECTER** (*Scientific Paper Embeddings using Citation-informed Transformers*), que aprimora a representação semântica ao incorporar informações de citações no treinamento (COHAN et al., 2020).

O SPECTER é baseado no treinamento do modelo SciBERT, mas se diferencia por empregar um treinamento supervisionado com um objetivo contrastivo, explorando a rede de citações científicas. O treinamento foi realizado com um conjunto de **146 mil artigos científicos do Semantic Scholar**, totalizando **26,7 milhões de tokens**. Diferentemente dos modelos anteriores, o SPECTER aprende a projetar textos extraidos de artigos em um espaço vetorial, no qual publicações relacionadas, isto é, aquelas que se citam mutuamente, possuem embeddings mais próximos, enquanto documentos não relacionados são afastados.

No treinamento do SPECTER, utiliza-se o **aprendizado contrastivo**, no qual pares de textos de artigos são organizados da seguinte maneira:

- **Positivos**: pares de artigos em que um cita o outro;
- **Negativos**: pares de artigos sem relação de citação.

A representação vetorial final de cada artigo é obtida a partir do **token [CLS]**, que sintetiza as informações contextuais do título e do resumo. Esse modelo tem demonstrado alto desempenho em tarefas como:

- **Classificação de documentos**, utilizando a similaridade entre artigos para categorização;
- **Predição de citações**, identificando quais artigos são relevantes para determinada pesquisa;

- **Recomendação de artigos**, auxiliando na descoberta de publicações sem depender exclusivamente de palavras-chave.

2.3 Otimização de Redes Neurais Profundas

A otimização de Redes Neurais Profundas desempenha um papel essencial no avanço da inteligência artificial, permitindo aprimoramentos significativos no treinamento e na inferência de modelos. Embora sejam técnicas eficientes para reduzir custos computacionais, este é um campo de pesquisa ativo, buscando otimizar arquiteturas e tarefas específicas sem comprometer a performance do modelo. Um dos principais motores desse progresso é o movimento *open-source*, que promove um ecossistema sustentável ao incentivar a colaboração e a disseminação de algoritmos de otimização e modelos de IA. Iniciativas disruptivas, como o **DeepSeek**[1], desafiam o domínio dos modelos fechados adotados por empresas como a **OpenAI**[2], disponibilizando arquiteturas avançadas, modelos de base pré-treinados e estratégias de otimização acessíveis à comunidade.

Além de reduzir custos computacionais e minimizar o impacto ambiental, essas otimizações tornam a IA mais acessível, eficiente e escalável, ampliando suas aplicações em diferentes domínios. O acesso aberto a essas tecnologias não apenas acelera a inovação, mas também reduz barreiras técnicas e operacionais, permitindo que pesquisadores e desenvolvedores aprimorem, adaptem e criem modelos de base de forma descentralizada. Esse movimento fortalece um ambiente de pesquisa mais inclusivo e sustentável, criando um ciclo virtuoso de avanço tecnológico.

Nesta seção, serão exploradas as principais estratégias de otimização para Redes Neurais Profundas, desde algoritmos fundamentais até abordagens avançadas investigadas neste estudo, com o objetivo de aprimorar a eficiência e promover a democratização do uso da IA.

2.3.1 Quantização

A **quantização** é uma técnica que transforma valores contínuos em discretos, permitindo a representação numérica com menos bits e reduzindo a demanda por recursos computacionais. Essa técnica é geralmente classificada em duas abordagens principais:

- **Quantização Dinâmica**: Apenas os pesos são quantizados, enquanto as ativações

[1] ⟨https://www.deepseek.com/⟩
[2] ⟨https://openai.com/⟩

permanecem em ponto flutuante. Não requer calibração prévia, pois a desquantização ocorre durante a execução, reduzindo o consumo de memória e melhorando a eficiência computacional em relação ao uso integral de ponto flutuante. No entanto, pode introduzir latência adicional devido às operações de desquantização em tempo real.

- **Quantização Estática**: Tanto pesos quanto ativações são quantizados, permitindo cálculos diretamente em baixa precisão (por exemplo, *int8*). Essa abordagem reduz significativamente a carga computacional e o uso de memória, mas exige uma etapa de calibração para determinar faixas de quantização adequadas. Além disso, modelos que dependem de alta precisão numérica podem sofrer degradação no desempenho.

Além disso, a quantização pode ser **assimétrica**, deslocando os pesos por uma constante, ou **simétrica**, onde os valores podem incluir ou não números negativos. A escolha entre essas abordagens depende do tipo de entrada, das ativações do modelo e da precisão desejada.

Na quantização em 8 bits, por exemplo, os intervalos abrangem $256 = 2^8$ valores inteiros distintos, convertendo valores contínuos em representações discretas. No entanto, a precisão da quantização afeta o desempenho do modelo, pois o arredondamento e as restrições de valores no intervalo podem comprometer a exatidão (NAGEL et al., 2021; FOURNARAKIS, 2021).

2.3.2 Poda

A **poda** (*pruning*) é uma técnica para reduzir a complexidade de redes neurais profundas, removendo parâmetros de baixa relevância, promovendo esparsidade e melhorando a latência do modelo.

A importância dos parâmetros pode ser determinada por métricas como magnitude dos pesos, sensibilidade dos gradientes ou impacto na função de perda (KURTIC et al., 2022; LIEBENWEIN, 2021). As principais abordagens incluem:

- **Poda estruturada**: Remove blocos inteiros, como camadas, filtros ou conexões específicas, mantendo a coerência da arquitetura e otimizando a execução do modelo.
- **Poda de compressão**: Aplicada a modelos pré-treinados, pode ocorrer durante o treinamento via compressão ascendente (no pré-treinamento) ou descendente (para

uma tarefa específica), aumentando a esparsidade.

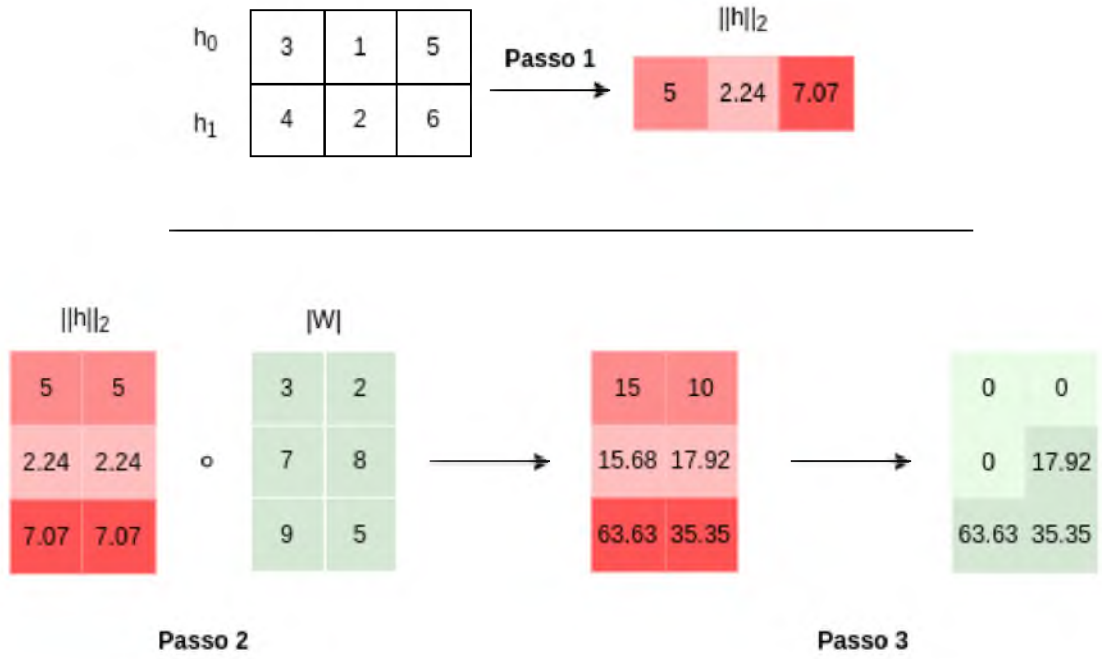

Figura 8: Poda Wanda com 50% dos valores zerados. Adaptado de Sun et al. (2023).

Entre as técnicas de poda, a **Wanda** destaca-se pela simplicidade e eficiência (SUN et al., 2023). Como ilustrado na Figura 8, a técnica aplica a norma L2 sobre os vetores de pré-ativação $\{h_0, ..., h_n\}$ (passo 1), multiplica elemento a elemento a matriz de pesos absolutos (passo 2) e zera os valores mais próximos de zero (passo 3), promovendo esparsidade não estruturada.

2.3.3 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 9, utiliza do professor para orientar o aluno transferindo seu conhecimento, mas não se limitando, através da predição final.

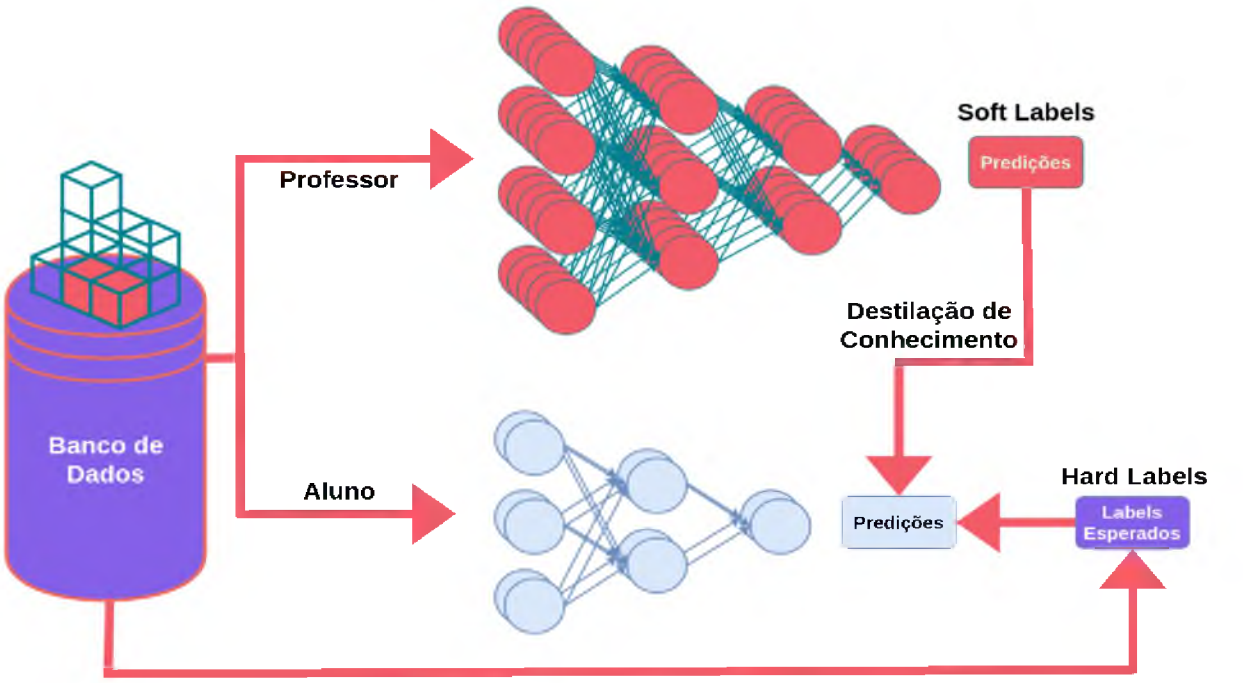

Figura 9: Destilação de conhecimento.

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:

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

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.

2.3.4 Treinamento Eficiente de Modelos de 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.

2.3.4.1 Pré-Treinamento

Como mencionado anteriormente, o pré-treinamento de modelos de 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 modelos (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 adoção de softwares eficientes, como processamento assíncrono de dados em CPUs, caching, pré-carregamento de dados

e otimizações no formato de armazenamento de arquivos, reduz significativamente o tempo de preparação antes do processamento (DAO et al., 2022; RAJBHANDARI et al., 2020; LECLERC et al., 2023). O uso de hardwares especializados para computação intensiva, como GPUs, também desempenha um papel crucial. Além disso, a integração de fontes de energia renovável, como solar e eólica, ou iniciativas como a Green AI Cloud[3], que visa reduzir as emissões de carbono e alcançar uma taxa de CO negativa, promove a sustentabilidade e contribui para a mitigação das mudanças climáticas, sem comprometer a performance no treinamento ou uso dos 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

[3]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.

2.3.4.2 Ajuste Fino

O ajuste fino de modelos de 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).
- **SetFit (*Sentence Transformer Fine-Tuning*):** O SetFit é uma técnica que combina aprendizado contrastivo e ajuste fino eficiente para melhorar a classificação de textos com poucos exemplos rotulados (TUNSTALL et al., 2022). Diferentemente de abordagens convencionais, ele realiza um treinamento em duas etapas: primeiro, aplica aprendizado contrastivo para refinar as representações dos textos e, em seguida, ajusta um classificador linear sobre essas representações. Essa metodologia permite que modelos baseados em Transformers obtenham alto desempenho em tarefas de classificação sem a necessidade de grandes volumes de dados rotulados, tornando o processo de ajuste fino mais rápido e eficiente.

A técnica **LoRa** (Low Rank Adaptation), utilizada neste trabalho, permite adicionar novas funcionalidades a um modelo sem aumentar o número de parâmetros após o

ajuste fino. Em vez de atualizar diretamente os pesos do modelo base $W_0 \in \mathbb{R}^{n_1 \times n_2}$ via gradiente ∇W_0, onde a atualização tradicional seria dada por $W := W_0 + \nabla W_0$, a técnica introduz uma matriz de decomposição de baixo posto, definida como $W_l := AB$, onde $A \in \mathbb{R}^{n_1 \times r}$ e $B \in \mathbb{R}^{r \times n_2}$, com r representando o valor do posto reduzido (HU et al., 2022). Dessa forma, o treinamento ocorre de maneira indireta, no qual apenas A e B são atualizados pela descida do gradiente, enquanto W_0 permanece fixo. O ajuste fino, portanto, é realizado na forma $W := W_0 + (W_l + \nabla W_l) = W_0 + L$, de modo que, após o treinamento, as atualizações capturadas por L são incorporadas a W_0. Esse processo reduz a complexidade computacional sem aumentar a latência do modelo com novos parâmetros. Similar aos métodos quasi-Newton, que aproximam a matriz Hessiana por uma de baixa ordem, o LoRa melhora a eficiência do ajuste fino e facilita a reutilização dos módulos A e B como compressores de conhecimento, permitindo sua transferência para outros sistemas de IA baseados no mesmo modelo de referência.

Essa técnica pode ter extensões, como no caso da **QLoRa** (DETTMERS et al., 2023), que combina essa decomposição com quantização, reduzindo ainda mais o consumo de recursos. A QLoRa é amplamente aplicada nos pesos dos blocos de atenção em modelos Transformers, ajustando correlações com base nos valores de entrada e diminuindo os recursos computacionais necessários para adaptar modelos com grandes quantidades de parâmetros (LIAO; TAN; MONZ, 2023).

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.

2.3.4.3 Composição Eficiente de Modelos de Base Adaptados

A medida que surgem mais modelos especializados em um ecossistema baseado em um modelo de base de referência, 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.

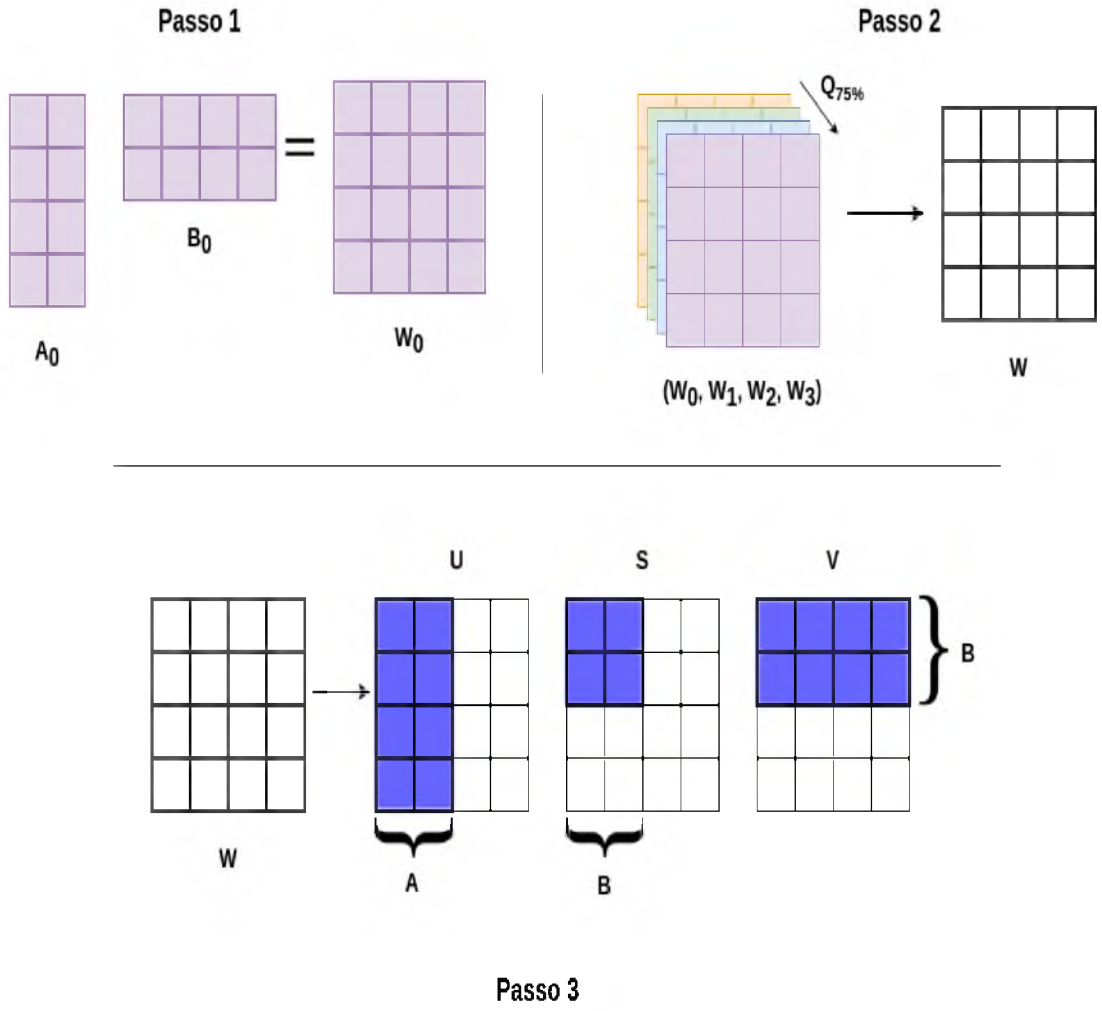

Figura 10: Método de Hibridação dos módulos LoRa.

Conforme ilustrado na Figura 10, 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 $W_0 = A_0 B_0$, 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 10, 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.

3 Metodologia

3.1 Metodologia de Estudo

Esta pesquisa tem como objetivo contribuir para a investigação do aprendizado profundo 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, serão abordados o ajuste fino eficiente para modelos de grande porte (HE et al., 2021), a composição de habilidades e considerações sobre hardware, software, arquiteturas e treinamentos eficientes para 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 avanços 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

revisão sistemática pode ser dividida em várias etapas até sua conclusão. Na Tabela 1, apresentamos um resumo dessas etapas, destacando que, neste estudo de aplicação, nos concentramos exclusivamente na automação do passo de triagem de citações (SLR6).

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).

Esse passo, em particular, é reconhecido como o mais demorado, pois exige que um ou mais especialistas reduzam a quantidade de citações em um banco de dados de referência gerado a partir dos resultados da busca, utilizando os termos de pesquisa da etapa anterior (SLR5). Essa redução é realizada por meio da classificação dos exemplos como relevantes ou não, de acordo com o critério de inclusão definido 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 20 bancos 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 de citações relevantes em Revisões Sistemáticas da Literatura (SLRs), métricas convencionais podem não refletir com precisão seu desempenho, pois esses bancos de dados são altamente desbalanceados, con-

tendo mais exemplos irrelevantes do que relevantes. Assim, a métrica WSS (*Work Saved over Sampling*) (KUSA et al., 2023), introduzida por Cohen et al. (2006), quantifica a economia de trabalho ao estimar a porcentagem de exemplos irrelevantes corretamente descartados pelo classificador, relativa apenas à quantidade total de exemplos irrelevantes presentes no banco de dados específico. Esse aspecto pode dificultar a comparação entre diferentes conjuntos de dados, pois a proporção de exemplos irrelevantes pode variar significativamente entre bases distintas. Para evitar a perda excessiva de exemplos relevantes, adota-se um critério que garante uma retenção mínima de 95%.

Entretanto, em conjuntos de dados com distribuições de classes distintas, os valores extremos da WSS podem variar significativamente, dificultando a comparação entre experimentos (MELO et al., 2022; FARIA et al., 2022). Para mitigar essa limitação, Melo et al. (2022) propôs a métrica AWSS (*Adjusted Work Saved over Sampling*), uma versão normalizada da WSS com valores entre $[-1, 1]$. A AWSS é definida como:

$$AWSS@TP\% = TN\% - (1 - TP\%) \tag{3.2.1}$$

onde $TN\%$ e $TP\%$ representam, respectivamente, a taxa de verdadeiros negativos e a taxa de verdadeiros positivos. Sua interpretação é a seguinte:

- **AWSS entre 0 e 1:** O modelo economiza trabalho em relação à amostragem aleatória. Por exemplo, se uma base contém 1000 citações (800 irrelevantes e 200 relevantes) e o modelo exclui corretamente 80% dos irrelevantes ($TN\% = 0.80$) enquanto retém 95% dos relevantes ($TP\% = 0.95$), temos:

$$AWSS = 0.80 - (1 - 0.95) = 0.75. \tag{3.2.2}$$

  Isso significa que 75% dos exemplos podem ser descartados sem comprometer a retenção dos 95% dos relevantes .

- **AWSS = 0:** O desempenho do modelo equivale ao de uma amostragem aleatória.

- **AWSS entre -1 e 0:** O modelo falha em excluir exemplos irrelevantes. Quanto mais próximo de -1, maior a porcentagem de inclusão dos exemplos irrelevantes.

Dessa forma, a métrica AWSS torna a comparação entre diferentes conjuntos de dados mais consistente, permitindo uma avaliação mais precisa do trabalho salvo pelo mesmo modelo de referência.

3.2.2 Configuração do Experimento

Para a modelagem da classificação de citações relevantes utilizando a arquitetura **Transformers**, adotou-se a metodologia **SETFIT** (TUNSTALL et al., 2022), conforme descrito na Seção 2.3.4.2, em conjunto com o modelo **SPECTER**, apresentado na Seção 2.2.7. O modelo **SPECTER**, desenvolvido por Cohan et al. (2020), foi escolhido devido ao seu treinamento especializado em textos científicos e à sua capacidade de vetorização semântica utilizando o token "[CLS]", tornando-o particularmente adequado para o domínio desta pesquisa. Esse modelo representa sentenças em um espaço vetorial, no qual textos cientificamente similares possuem embeddings mais próximos entre si.

O ajuste fino utilizando a abordagem **SETFIT** ocorre em dois passos. No primeiro, o modelo é ajustado com **aprendizado contrastivo**, aprimorando as representações semânticas dos textos de entrada. No segundo, realiza-se a **classificação baseada em embeddings**, onde um classificador é treinado sobre essas novas representações para a tarefa de classificação final.

3.2.2.1 Fase 1: Aprendizado Contrastivo

Inicialmente, os exemplos para o ajuste fino são organizados em pares aleatórios. Pares pertencentes à mesma classe recebem o rótulo P (**pares similares**), enquanto pares de classes distintas recebem o rótulo N (**pares não similares**). Cada par representa uma amostra utilizada no treinamento contrastivo, cujo objetivo é aproximar exemplos da mesma classe e afastar aqueles pertencentes a classes diferentes, com base em suas representações vetoriais.

Para esse treinamento, foram selecionados aleatoriamente **8 exemplos por classe**, totalizando **16 exemplos**. A partir desses exemplos, foram gerados **40 pares positivos** e **40 pares negativos**, resultando em um conjunto de treino com **80 pares contrastivos**.

A vetorização dos exemplos é realizada utilizando o modelo **SPECTER**, no qual cada entrada na amostra de treino é representada pela concatenação do **título** e do **resumo** do respectivo artigo científico. O truncamento é aplicado para um máximo de **512 tokens**, respeitando a limitação de entrada do modelo. Essas frases são então convertidas em representações vetoriais por meio do vetor resultante do token de entrada **[CLS]**, que captura a representação semântica do texto de entrada na saída do modelo.

Neste treinamento contrastivo, foi aplicado um ajuste fino eficiente com **LoRa**, treinando apenas os **módulos de baixo posto**, configurados com dimensionalidade igual

a 4, conforme descrito na Seção 2.3.4.2. Esses módulos são aplicados às matrizes V e K do mecanismo de atenção, conforme ilustrado na Figura 7, nas **três últimas camadas** do modelo SPECTER. O treinamento foi conduzido por **duas épocas**, atualizando esses módulos com **lotes de 16 pares contrastivos** e uma taxa de aprendizado (*learning rate*) igual a **0,007**.

A função de perda adotada é uma adaptação da **função de perda contrastiva supervisionada** de Khosla et al. (2021), definida como:

$$Loss(x,y) = \frac{-1}{|P|} \sum_{p \in P} \log \frac{\exp\{sim(x_p, y_p)/\tau\}}{\sum_{n \in N} \exp\{sim(x_n, y_n)/\tau\}}, \quad sim(x,y) = \frac{x \cdot y}{||x||_2 ||y||_2} \tag{3.2.3}$$

onde $p \in P$ representa os exemplos contrastivos da mesma classe e $n \in N$ pertence a classes distintas. A métrica de similaridade $sim(x, y)$ utilizada foi a **similaridade do cosseno**, com o hiperparâmetro τ fixado em **0,2**.

Duas modificações foram incorporadas em relação à formulação original: (i) os pares foram selecionados aleatoriamente, em vez de considerar todas as combinações possíveis dentro da amostra; e (ii) o denominador da função de perda inclui apenas exemplos negativos, aplicando a técnica de **Hard Negative Sampling**. Esse método proporciona maior estabilidade ao treinamento, enfatizando a separação adequada entre exemplos contrastivos, conforme discutido por Wang e Liu (2021).

3.2.2.2 Fase 2: Classificação Baseada em Embeddings

Após o ajuste fino contrastivo, os **16 exemplos** utilizados na Fase 1 são empregados para treinar o modelo de classificação baseado na **regressão logística**, associando-os às suas respectivas classes. Esse treinamento utiliza as novas representações vetoriais (**embeddings**) extraídas do token **[CLS]** do modelo Transformer ajustado pela fase 1. Essa abordagem possibilita a classificação dos textos de entrada como **relevantes** ou **não relevantes**, além de permitir a avaliação da performance do modelo adaptado por meio da métrica **AWSS@95%**, que mensura a economia de trabalho na seleção de artigos relevantes em uma revisão sistemática da literatura.

3.2.2.3 Otimização

Para aprimorar a eficiência computacional do modelo, algumas técnicas de otimização foram aplicadas. Este estudo foca em abordagens simples que permitem tornar os modelos de base mais sustentáveis por meio dos seguintes métodos:

- **Quantização dinâmica em 4 bits**.
- **Poda de parâmetros** utilizando o método **Wanda**, proposto por Sun et al. (2023) e ilustrado na Figura 8. Esse método é aplicado em todas as camadas do modelo, exceto nos módulos LoRa, que são utilizados paralelamente aos pesos, conforme apresentado na Seção 2.3.4.2.

Além disso, foram realizadas análises sobre o consumo de recursos computacionais e a latência durante o treinamento e a inferência do modelo Transformer, considerando diferentes configurações de precisão e hardware:

Hardware	**Especificação**	**TDP (W)**
CPU	Intel Core i7-11800H (11ª geração)	45 - 109
GPU	NVIDIA GeForce RTX 3060 Mobile	60 - 115

Tabela 2: Especificações de hardware utilizadas nos experimentos .

A avaliação da latência de inferência foi realizada sobre uma amostra de **32 exemplos** de textos, enquanto a avaliação do treinamento seguiu as especificações da Fase 1. Todos os exemplos contendo **512 tokens**.

3.2.2.4 Composição de Habilidades

Com o objetivo de consolidar o conhecimento das novas representações semânticas adquiridas após o treinamento contrastivo da Fase 1 nos 20 bancos de dados, selecionamos os cinco melhores resultados após o ajuste fino, com base na métrica de trabalho salvo **AWSS@95%**. Essa métrica foi utilizada para avaliar a performance da separação semântica entre classes **locais** (dentro de um mesmo banco de dados) e **globais** (entre diferentes bancos de dados), na composição de um único modelo com habilidades combinadas. Para isso, realizamos a **composição de módulos LoRa**, conforme descrito na Seção 2.3.4.3, configurando-a para gerar os novos módulos A e B com **rank 8**.

4 Resultados

4.1 Análise quantitativa dos bancos de dados

Os bancos de dados utilizados em revisões sistemáticas da literatura consistem em extensas coleções de referências textuais, submetidas a um rigoroso processo de triagem para a seleção de estudos relevantes. O objetivo desse procedimento é identificar artigos que respondam a uma questão específica ou fundamentem uma investigação científica. Em bancos de dados de revisões sistemáticas da literatura (**SLRs**), observa-se frequentemente um **desequilíbrio acentuado** entre o número de artigos relevantes (**Incluídos**) e não relevantes (**Excluídos**). Esse padrão é evidente nos bancos de dados analisados neste estudo, conforme apresentado na Tabela 3, onde as classificações **Excluído (-1)** e **Incluído (1)** foram atribuídas pelos autores com base no título e no resumo de cada artigo. Em alguns casos, essa desproporção é expressiva, com menos de **10% dos artigos sendo incluídos** na 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 3: Distribuição de rótulos para cada respectivo banco de dados.

A análise da distribuição da contagem de tokens foi realizada considerando a limitação de **512 tokens** como entrada para o modelo base **SPECTER** (COHAN et al., 2020), que recebe como entrada a concatenação do título e do resumo de cada artigo. O maior banco de dados analisado, **Transgeracional**, apresenta um número significativo de *outliers*, especialmente entre os textos classificados como **excluídos (-1)**, que frequentemente ultrapassam esse limite, conforme indicado pelo *boxplot* vermelho na Figura 11. De modo geral, ao analisar a distribuição dos tokens nos diferentes bancos de dados e suas respectivas classes, observa-se que **75% dos textos de entrada permanecem dentro do limite do modelo**, com apenas alguns exemplos necessitando de truncamento para viabilizar sua utilização.

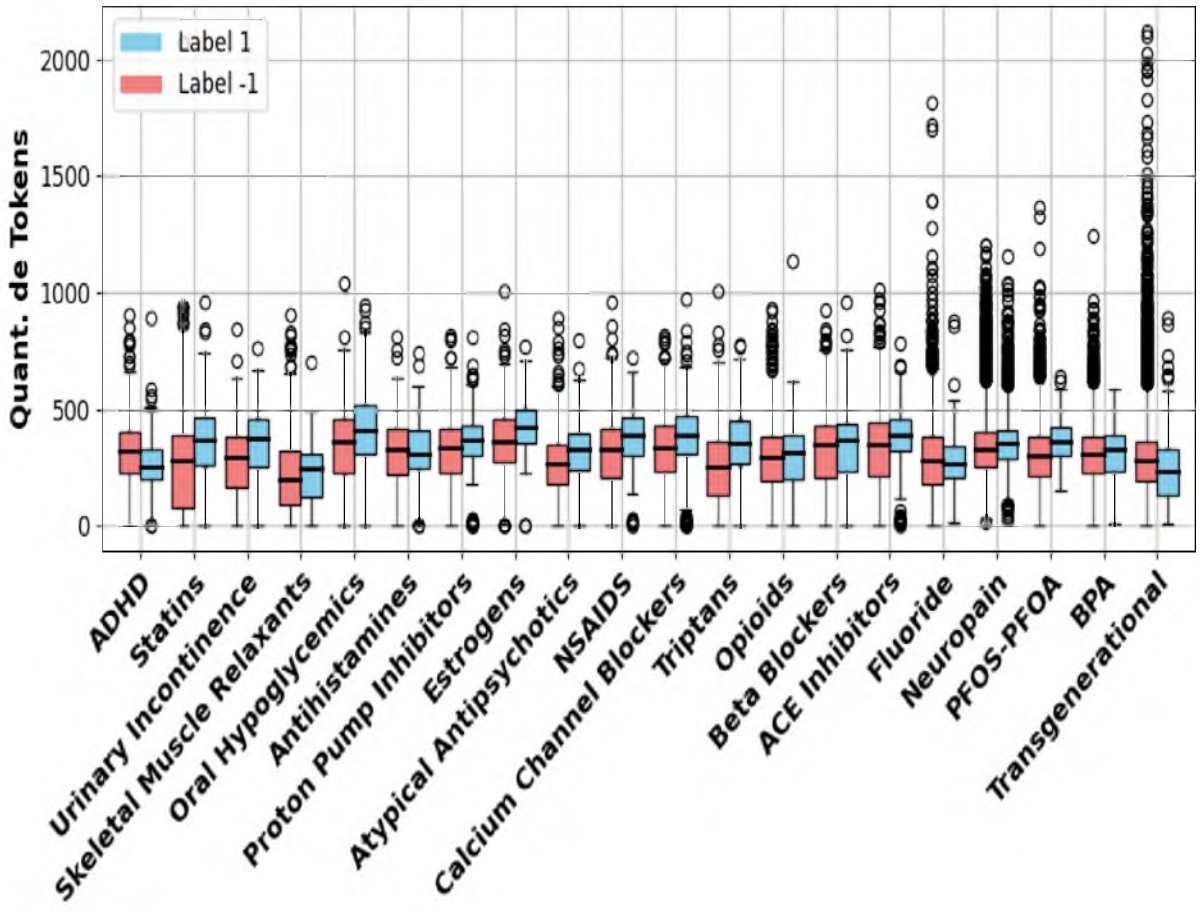

Figura 11: Distribuição de tokens para cada banco de dados.

Embora os bancos de dados contenham terminologia específica de suas respectivas áreas de pesquisa, a segmentação realizada pelo tokenizador permite que o modelo interprete os textos de forma adequada, sem a necessidade de um vocabulário especializado para cada domínio. Dessa forma, a capacidade máxima de **512 tokens** dos modelos **Transformers** é utilizada para gerar representações vetoriais contextualizadas do título

e do resumo concatenados. Essas representações são extraídas a partir do token especial **[CLS]** e posteriormente empregadas na tarefa de classificação, utilizando um modelo de **regressão logística** em conjunto com aprendizado contrastivo.

4.2 Análise exploratória do modelo e treinamento contrastivo

O modelo Transformer **SPECTER** (COHAN et al., 2020), utilizado neste estudo, foi inicialmente pré-treinado por meio de um treinamento contrastivo. Esse processo teve como objetivo aproximar vetores semanticamente semelhantes e afastar aqueles sem correlação, utilizando a representação vetorial de **768 dimensões**, gerada como saída do modelo a partir do token de entrada **[CLS]**. Dessa forma, o modelo estabelece um espaço semântico inicial capaz de contextualizar a estrutura dos textos de entrada.

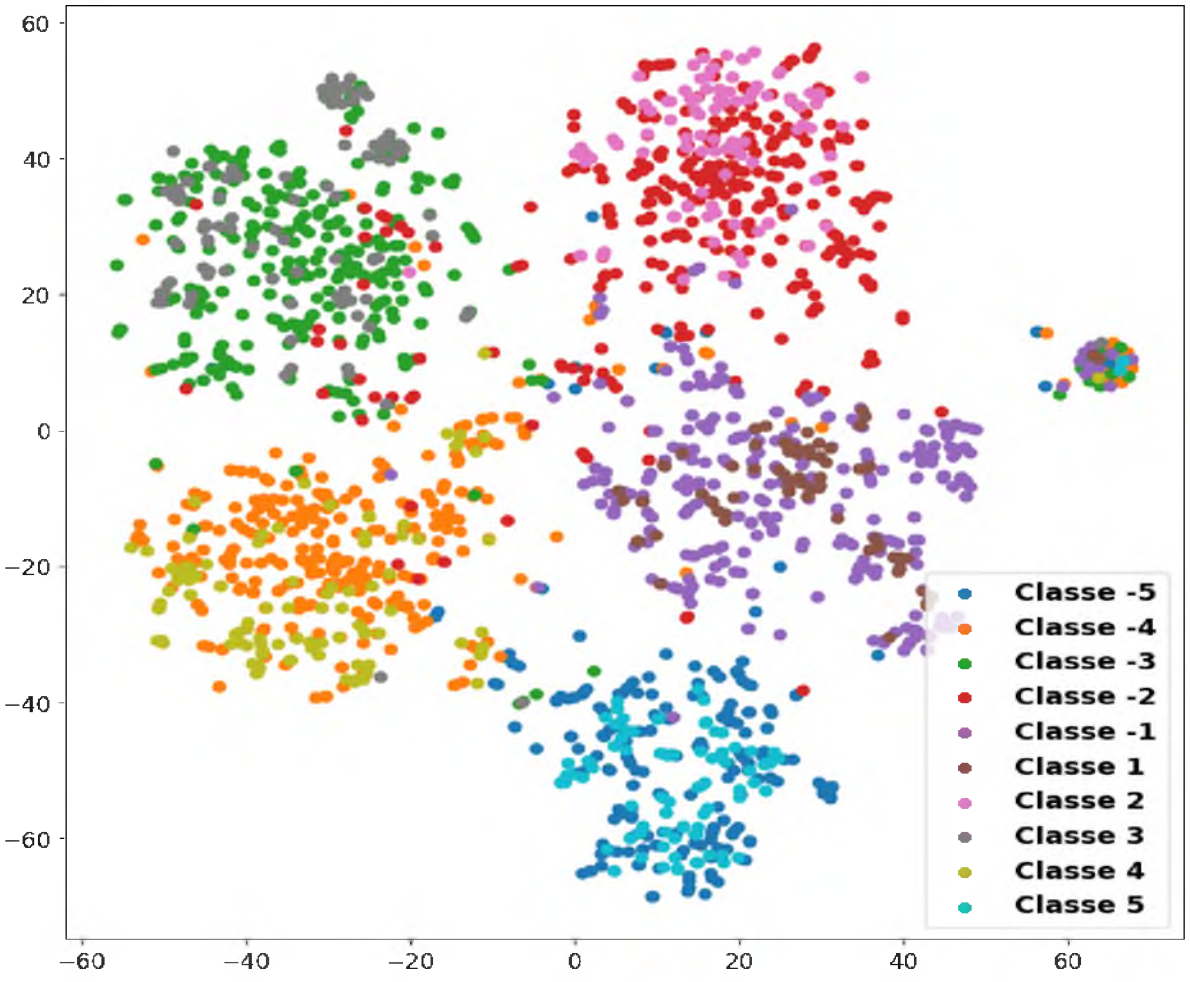

Figura 12: Visualização T-SNE dos vetores [CLS] de cinco bancos de dados utilizando o modelo SPECTER: (1) NSAIDS, (2) Neuropain, (3) Oral Hypoglycemics, (4) Statins e (5) Antihistamines.

Para avaliar a capacidade inicial do espaço semântico criado pelo modelo de base, foram selecionados 300 exemplos por classe em cinco bancos de dados escolhidos aleatoriamente. Cada exemplo foi representado pela concatenação do título com o resumo do respectivo artigo, respeitando o limite de 512 tokens.

Os rótulos foram atribuídos conforme a origem dos exemplos em cada banco de dados. No caso do banco Antihistamines, por exemplo, os índices 5 e -5 foram usados para representar, respectivamente, os exemplos incluídos (1) e excluídos (-1). Essa categorização foi mantida para os demais bancos de dados, permitindo a comparação entre classes e entre bancos distintos.

Para visualizar as representações geradas pelo modelo, utilizou-se o método de redução de dimensionalidade t-SNE (MAATEN; HINTON, 2008) para projetar os vetores de 768 dimensões, obtidos a partir do token [CLS], em um espaço bidimensional. A Figura 12 apresenta o resultado dessa projeção, onde observa-se que os pontos exibem agrupamentos semânticos coerentes entre os diferentes bancos de dados.

Entretanto, ao analisar a separação entre as classes dentro de cada banco, nota-se que o modelo não consegue distingui-las de forma eficiente. Esse problema é particularmente evidente no banco Antihistamines, onde exemplos incluídos e excluídos não apresentam separação clara no espaço vetorial. Esse comportamento sugere que, embora o modelo capture a estrutura semântica geral dos textos, ele não diferencia adequadamente os exemplos pertencentes a classes opostas dentro de um mesmo banco de dados.

Para uma análise mais detalhada da capacidade da representação semântica resultante do token **[CLS]** do modelo de base e sua adaptabilidade na separação das classes, utilizou-se o banco de dados **NSAIDS**, com **16 exemplos de treino** (8 por classe) e uma amostra de validação composta por **360 exemplos**, sendo **41 incluídos (1)** e **319 excluídos (-1)**, organizados conforme essa ordem.

A Figura 13 ilustra o desempenho da representação gerada pelo modelo antes do treinamento contrastivo nas imagens (a) e (c). A imagem (a), obtida por meio da redução de dimensionalidade **t-SNE**, indica que o modelo não conseguiu capturar visualmente a separação semântica entre as classes dentro do banco de dados. Já a imagem (c), que representa a matriz de similaridade do cosseno dos índices organizados conforme especificado, revela uma alta correlação entre exemplos de diferentes classes, evidenciando a dificuldade do modelo em distinguir padrões semânticos antes do ajuste fino.

Por outro lado, após o ajuste fino da **Fase 1**, observa-se na imagem (b) que o modelo passa a agrupar os exemplos de acordo com suas classes rotuladas. Esse com-

portamento é reforçado na imagem (d), onde a similaridade entre exemplos da mesma classe atinge valores superiores a **0,75**, especialmente para a classe **(-1)**, nos índices de **42 a 360**. No entanto, para os índices de **1 a 41**, que representam a classe **(1)**, algumas amostras foram erroneamente classificadas como negativas, apresentando valores de similaridade próximos a **-0,75**, sugerindo um agrupamento indevido com a classe **-1**.

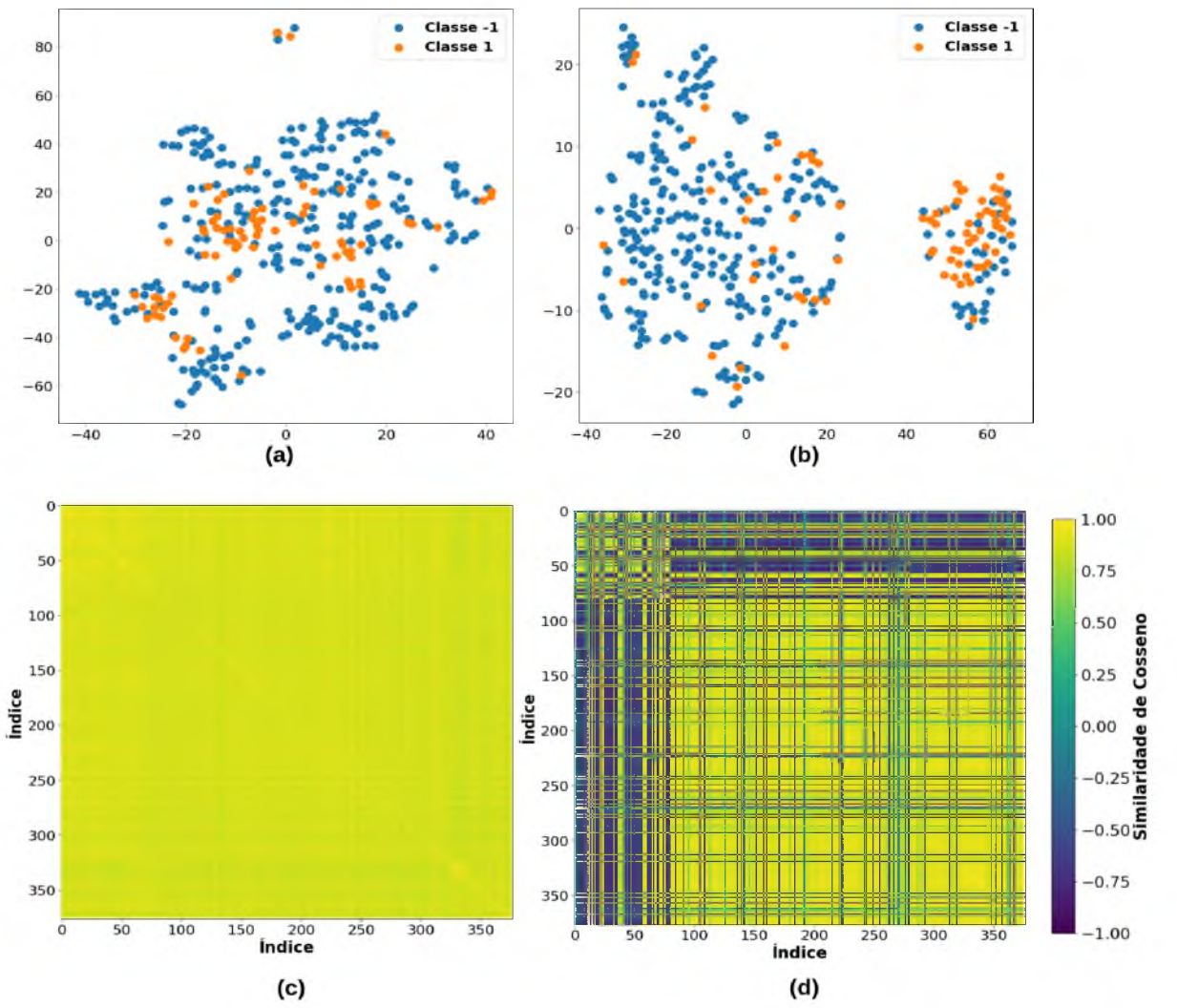

Figura 13: Visualização T-SNE dos vetores [CLS] do banco de dados NSAIDS antes e depois do ajuste fino. As imagens (a) e (b) representam, respectivamente, a distribuição dos vetores antes e após o treinamento contrastivo. As imagens (c) e (d) exibem a matriz de similaridade do cosseno entre os exemplos antes e depois do ajuste fino.

Para avaliar a eficácia do modelo em termos de economia de trabalho, utilizou-se um **modelo de regressão logística** treinado com a representação vetorial do token **[CLS]** gerada pelo modelo Transformer antes e depois do ajuste fino da **Fase 1**, mantendo os mesmos **16 exemplos de referência**. Na Figura 14, a imagem (a) apresenta a distribuição das classes antes do treinamento contrastivo no espaço entre **0 e 1**, projetado pelo modelo de regressão logística. Nota-se que o modelo de base, sem ajuste, já consegue

criar uma representação semântica com uma distinção inicial entre as densidades das classes **1** e **-1**, conforme indicado pelas curvas **"Densidade 1"** e **"Densidade -1"**.

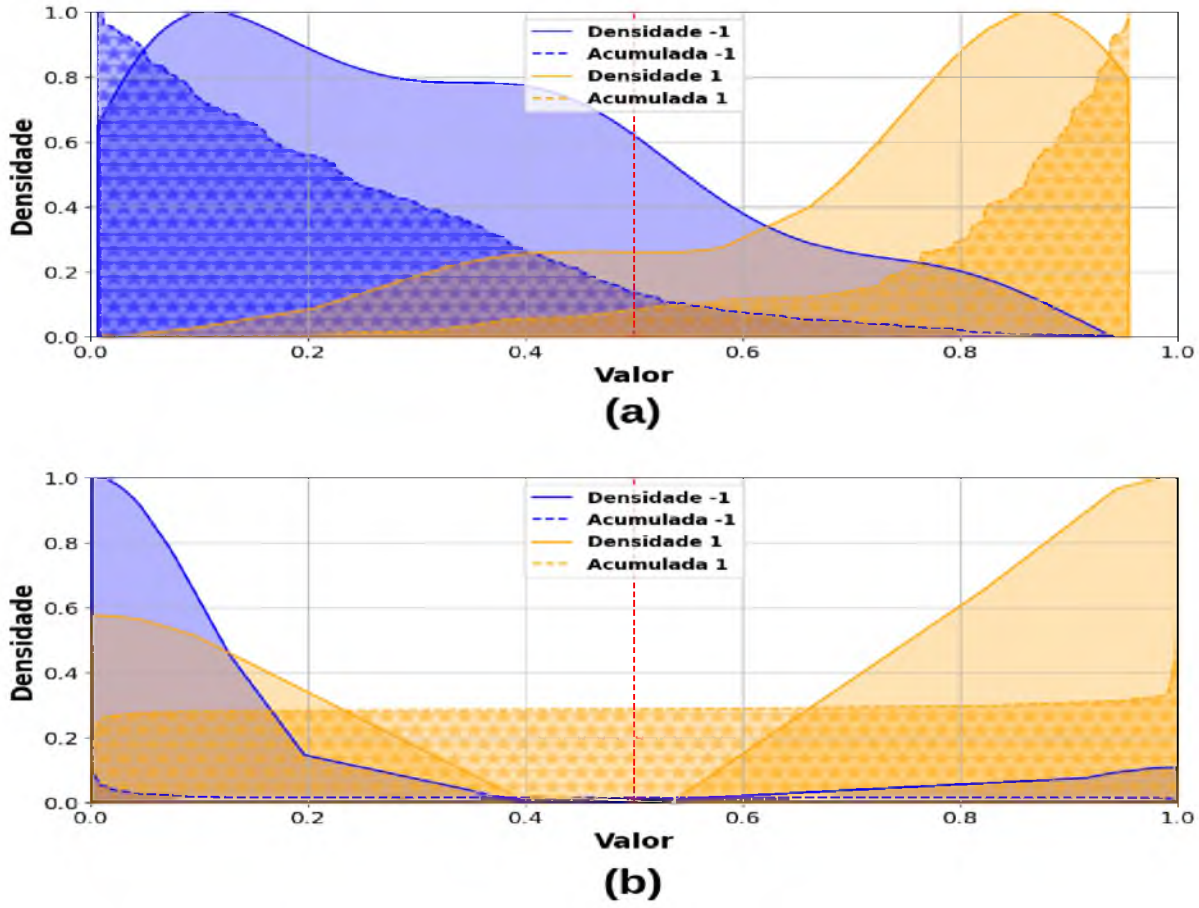

Figura 14: Curvas normalizadas da densidade de probabilidade e densidade acumulada (estrelada) das representações vetoriais do token [CLS] pelo modelo Transformer para o banco de dados NSAIDS antes (a) e depois (b) do ajuste fino, utilizando a regressão logística ajustada no conjunto de treino.

Comparando com o modelo ajustado conforme a especificação da **Fase 1**, a imagem (b) da Figura 14 mostra que a distribuição das densidades indica uma separação mais definida, posicionando os exemplos de validação mais próximos dos extremos **0 e 1** da regressão logística. Além disso, ao analisar a **função de sobrevivência** (definida como **1 menos a função acumulada**), percebe-se que a separação entre as classes é mais evidente da imagem (a) para a (b). Esse comportamento sugere que o modelo, após o treinamento contrastivo, apresenta maior confiança na classificação dos exemplos, corroborando as análises apresentadas na Figura 13, onde as imagens (a) e (c) ilustram o estado antes do treinamento, enquanto as imagens (b) e (d) representam o comportamento após o ajuste fino da **Fase 1**.

Entretanto, ao avaliar o desempenho do modelo pela métrica **AWSS@95%**, mantendo **95% dos exemplos relevantes**, observa-se que o modelo treinado conse-

guiu excluir **21,3%** dos exemplos irrelevantes, enquanto o modelo sem ajuste contrastivo apresentou um desempenho superior, excluindo **28,9%** dos exemplos irrelevantes.

Embora o ajuste fino tenha melhorado a separação semântica, a regressão logística treinada com os **16 exemplos de referência** sobre as representações vetoriais do modelo **SPECTER** sem ajuste apresentou melhor desempenho no banco **NSAIDs**. Esse resultado indica que, apesar de o modelo apresentar maior confiança na clusterização dos exemplos, a métrica **AWSS@95%** revela que, ao ajustar o limiar para atender a esse critério, mais exemplos irrelevantes acabam sendo erroneamente incluídos após o ajuste fino, reduzindo a eficiência do modelo na economia de trabalho.

4.3 Anãlise de Otimização

A avaliação do desempenho do modelo é essencial para garantir sua escalabilidade e viabilidade em aplicações práticas. A escolha de técnicas de otimização impacta diretamente a eficiência computacional e a qualidade dos resultados. Para mensurar esses efeitos, foram conduzidos experimentos comparando diferentes precisões numéricas durante o treinamento e a inferência do modelo.

Precisão	Treinamento	Inferência GPU	Inferência CPU
FLOAT32	$132\text{s} \pm 0.10$ (1x)	$0.72\text{s} \pm 0.10$ (1x)	$10.92\text{s} \pm 0.21$ (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	-	$0.15\text{s} \pm 0.04$ (4.8x)	$6.38\text{s} \pm 0.31$ (1.7x)

Tabela 4: Latência em diferentes tipos de precisão para inferência e configuração do treinamento contrastivo especificado na Fase 1. Para o treinamento contrastivo, foram utilizados 1280 pares de exemplos contrastivos, cada um com 512 tokens, treinados por 2 épocas com lotes de 16 exemplos. Para a inferência, utilizou-se um lote de 32 exemplos, também com 512 tokens.

A Tabela 4 apresenta as estatísticas de desempenho do treinamento na **Fase 1**. Ao empregar a precisão **BFLOAT16**, observa-se uma **redução significativa no tempo de treinamento**, resultando em um desempenho **3,3 vezes superior** em comparação à precisão **FLOAT32**. Com um nível de confiança de **95%**, não foi identificada diferença estatisticamente significativa no valor final da função de perda, conforme verificado por um **teste-t**, em relação à inicialização padrão dos modelos em **FLOAT32**.

No que se refere à inferência, utilizando um lote de **32 exemplos**, cada um contendo **512 tokens**, a execução na **GPU** com precisão **FLOAT16** apresentou uma **redução de 2,4 vezes na latência** em relação à **FLOAT32**. No entanto, ao realizar a inferência na **CPU**, a precisão **FLOAT16** resultou em uma latência aproximadamente

duas vezes superior à obtida com **FLOAT32**. Esse comportamento pode ser atribuído à **otimização da arquitetura da CPU**, que processa de forma mais eficiente precisões específicas e pré-determinadas.

Para valores de menor precisão na **CPU**, como **INT8**, utilizando **quantização dinâmica** — na qual apenas os pesos são quantizados, enquanto os valores de ativação permanecem em **FLOAT32** —, obteve-se uma **redução significativa da latência**, com um desempenho até **70% superior** em relação à **FLOAT32**. Na **GPU**, essa melhoria foi ainda mais expressiva, resultando em uma inferência **4,8 vezes mais rápida**, com uma latência de apenas **0,15 segundos**. Considerando o **TDP** como uma métrica aproximada do consumo energético, conforme apresentado na Tabela 2, verifica-se que a **GPU é mais eficiente para inferência**, uma vez que sua **arquitetura otimizada para operações matriciais** proporciona menor latência e maior economia de energia em comparação à CPU.

Precisão	**SPECTER 110M (100%)**	**LoRa 36.864 (0,03%)**
32 bits	420 MB	144 KB (0,14 MB)
16 bits	209 MB	72 KB (0,07 MB)
8 bits	104 MB	36 KB (0,035 MB)
4 bits	52 MB	18 KB (0,018 MB)

Tabela 5: Consumo de recursos computacionais em diferentes precisões para o modelo SPECTER e os módulos LoRa.

A alocação de recursos computacionais para o modelo de base utilizado neste estudo é diretamente influenciada pela precisão dos pesos, impactando significativamente a demanda por armazenamento e processamento. Como ilustrado na Tabela 5, a precisão original do modelo **SPECTER**, que contém **110 milhões de parâmetros**, consome aproximadamente **420MB**. Esse valor pode ser considerado elevado para dispositivos com restrições de memória, como **dispositivos móveis**. Entretanto, ao reduzir a precisão dos pesos, é possível obter reduções de até **8 vezes** no consumo de memória, como no caso do modelo quantizado em **4 bits**.

Por outro lado, ao analisar os módulos **LoRa**, conforme a configuração da **Fase 1 do treinamento contrastivo**, observa-se uma redução expressiva no número de parâmetros treináveis necessários para o ajuste fino. Esses módulos representam menos de **0,1%** do tamanho do modelo base, proporcionando um armazenamento altamente eficiente para o acervo de habilidades adaptadas, atingindo um tamanho reduzido de apenas **18 KB** (0,018 MB). Essa característica possibilita a transmissão eficiente dos módulos

por redes de comunicação, além de viabilizar seu armazenamento tanto em dispositivos com baixa capacidade computacional quanto em grandes bancos de dados de habilidades, sem impactar significativamente o consumo de recursos em datacenters.

Além da análise da **latência** e do consumo de **recursos computacionais**, avaliou-se a capacidade do modelo base e o impacto da otimização especificada na Seção 3.2.2.3. Para isso, analisaram-se as configurações da **Fase 1 do treinamento contrastivo**, utilizando os mesmos 16 exemplos de referência para treinar o modelo de regressão logística sobre as representações vetoriais do token [CLS] desses exemplos, conforme determinado na Fase 2, em cada um dos **20 bancos de dados**. A **performance** dessas representações foi medida antes e depois do ajuste fino com **LoRa**, por meio da **métrica AWSS@95%**, com o objetivo de avaliar a capacidade do modelo base de **expandir suas habilidades** em diferentes tópicos de revisões sistemáticas da literatura. Os resultados dessa comparação são apresentados na Figura 15.

Conforme ilustrado na Figura 15, ao manter 95% dos exemplos relevantes na classificação, conforme a métrica adotada, o modelo **SPECTER**, identificado como **"Base"**, apresentou um desempenho satisfatório no conjunto de bancos de dados do **SWIFT**, descrito na Tabela 3, reduzindo em mais de **70%** a quantidade de documentos irrelevantes. No entanto, em alguns cenários, o ajuste fino não proporcionou melhorias significativas na métrica, como observado no caso do **NSAIDS**, descrito anteriormente na Seção 2.3.4.2. Além disso, em situações extremas, como no banco de dados **Estrogens**, a métrica de trabalho salvo **diminuiu de 35% para 12%**, evidenciando limitações na adaptação do modelo a determinadas bases de dados.

Após o ajuste fino do treinamento contrastivo utilizando **LoRa**, aplicou-se, paralelamente, a **quantização dinâmica em 4 bits** e a **poda Wanda**, sendo esta última responsável por zerar **50% dos pesos**. Esses métodos são referenciados, respectivamente, como **"Quant. 4Bit"** e **"Prune"**. Utilizando o modelo de regressão logística gerado pelo ajuste fino, os resultados indicam que, em diversos bancos de dados, há uma degradação na métrica **AWSS@95%** em comparação com os valores obtidos pelo ajuste **LoRa** e pelo modelo base. No entanto, de forma surpreendente, em alguns casos, a métrica apresenta recuperação de desempenho ou até mesmo uma melhoria em relação ao ajuste fino. Esse fenômeno é observado, por exemplo, nos bancos de dados **"Urinary Incontinence"** e **"Opioids"**, ao empregar a técnica de poda. No banco **"Opioids"**, a métrica de trabalho salvo atinge aproximadamente **50%**, superando os **40%** do ajuste fino. Comportamento semelhante é identificado em outros bancos ao utilizar a quantização, como nos casos de **"PFOS-PFOA"** e **"Calcium Channel Blockers"**. De maneira geral, o modelo base

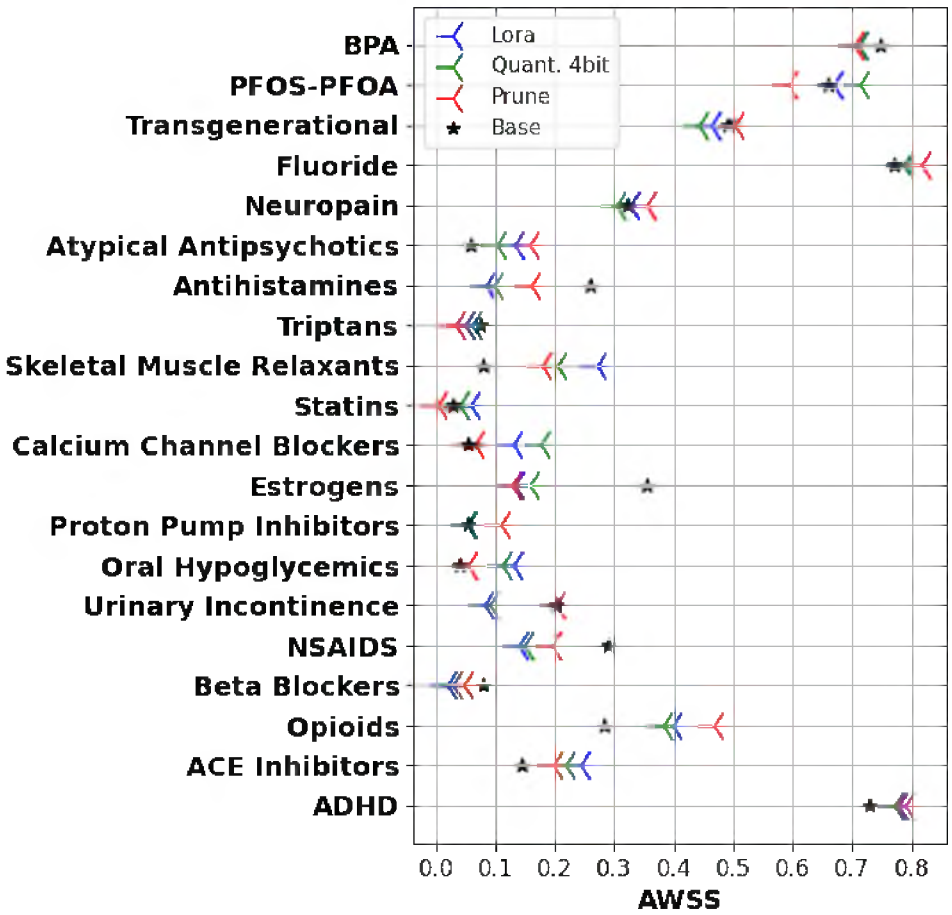

Figura 15: Métrica AWSS@95% para diferentes técnicas de otimização por banco de dados.

apresenta um desempenho inicial mais robusto quando comparado aos resultados obtidos das técnicas de ajuste fino e dos métodos de otimização propostos. Em 10 dos bancos de dados analisados, essas técnicas resultaram em melhorias de desempenho; entretanto, nos outros 10 casos, verificou-se ao menos uma ocorrência de degradação em relação ao desempenho na métrica pelo modelo de base.

Por fim, considerando os cinco bancos de dados com os melhores resultados na métrica **AWSS@95%**, conforme apresentado na Figura 15, realizou-se uma análise exploratória para avaliar o impacto da combinação de múltiplas habilidades ao longo do tempo. Partiu-se da hipótese de que, caso o modelo base fosse ajustado para aprender e acumular diferentes habilidades por meio do ajuste fino, seria possível consolidar essas representações em um único modelo híbrido, utilizando a composição dos módulos **LoRa** em uma estrutura unificada, preservando a performance específica de cada tarefa.

Para esse experimento, adotou-se a metodologia descrita na Seção 2.3.4.3, que viabiliza a combinação dos módulos. A configuração do **rank** foi definida como 8 para os novos módulos **A** e **B** do LoRa híbrido, permitindo a fusão das representações vetoriais

adquiridas pelo modelo ajustado em diferentes bancos de dados.

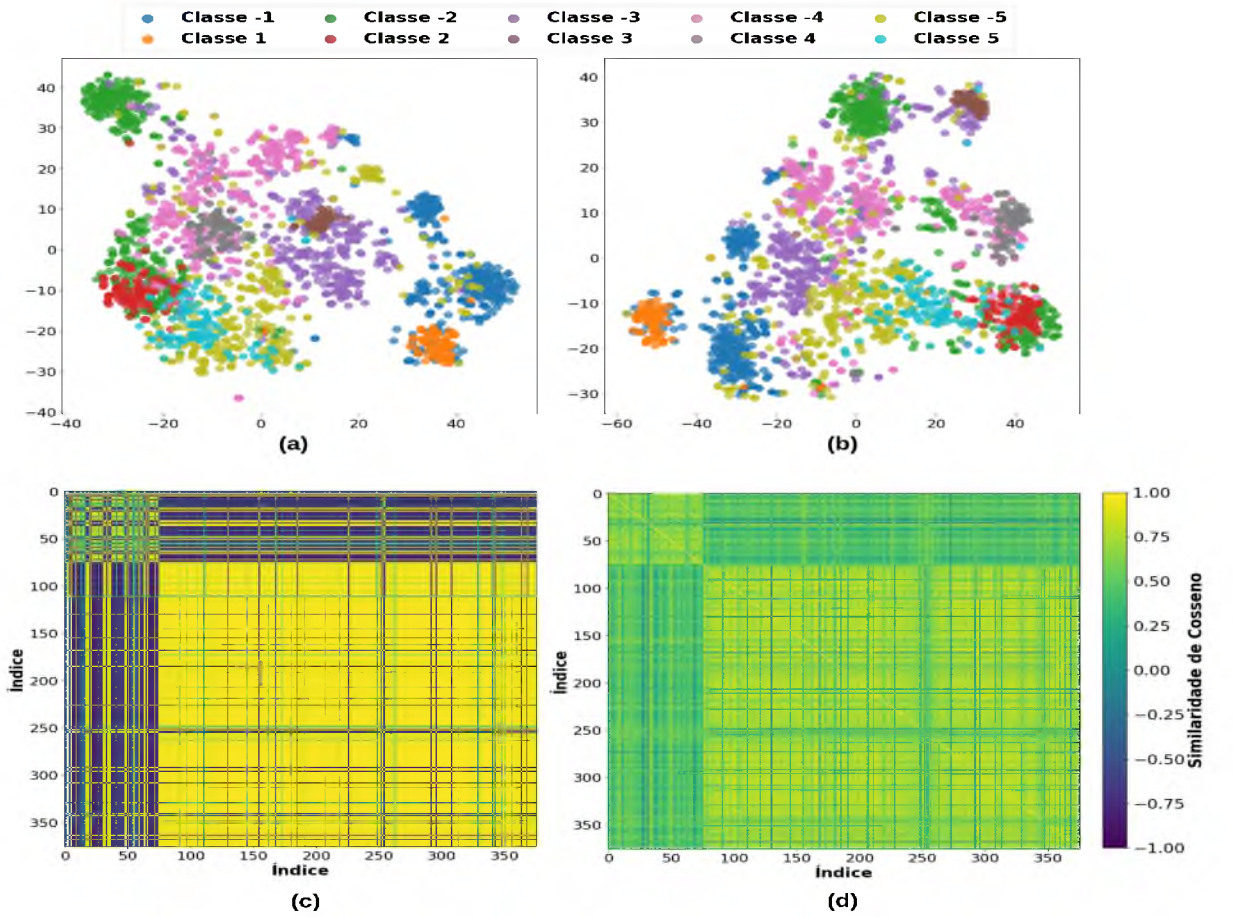

Figura 16: Visualização T-SNE dos vetores [CLS] de cinco bancos de dados utilizando o modelo SPECTER. Os bancos considerados são: (1) ADHD, (2) BPA, (3) Fluoride, (4) PFOS-PFOA e (5) Transgenerational. A subfigura (a) apresenta o modelo sem ajuste fino, enquanto a subfigura (b) exibe o modelo ajustado por meio da abordagem híbrida. As matrizes de similaridade do cosseno para o banco de dados PFOS-PFOA são apresentadas nas subfiguras (c) e (d), representando, respectivamente, os resultados após o ajuste fino convencional e a composição híbrida.

Na Figura 16, onde amostras de validação composta por **360 exemplos** para cada banco de dados, a subfigura **(a)** apresenta o espaço vetorial do modelo base antes de qualquer ajuste, considerando os cinco bancos de dados analisados. Em contrapartida, a subfigura **(b)** exibe a projeção resultante da composição híbrida dos módulos de habilidades, utilizando a técnica *t-SNE*. Embora a visualização ocorra em um espaço bidimensional, observa-se que, na subfigura **(b)**, o modelo passa a promover uma melhor separação das classes dentro de cada banco de dados, mantendo, ao mesmo tempo, uma distinção global entre os diferentes bancos. Esse efeito pode ser notado, por exemplo, no banco **"PFOS-PFOA"**, onde as classes **4** e **-4**, correspondentes aos rótulos **incluído (1)** e **excluído (-1)**, tornam-se mais claramente diferenciadas após a composição híbrida.

Na imagem (c), apresenta-se a **matriz de similaridade cosseno** para o banco

de dados **PFOS-PFOA**, utilizando o resultado do **ajuste fino** correspondente à Fase 1. Para os índices de **1 a 80**, que representam instâncias dos rótulos **incluído (1)** e **excluído (-1)**, a similaridade entre os vetores é aproximadamente **1**, indicando **alta semelhança** entre instâncias da mesma classe. Em contrapartida, ao comparar instâncias de **rótulos distintos**, essa relação se aproxima de **-1**, refletindo a separação entre os grupos.

Ao comparar esse resultado com a imagem (d), que representa os mesmos exemplos utilizando a representação vetorial do modelo resultante da composição híbrida, observa-se que a **matriz de similaridade** mantém uma estrutura semelhante à da matriz (c). No entanto, os valores de similaridade não permanecem **perfeitamente próximos** de **1** e **-1**, como anteriormente. Em vez disso, os valores para **instâncias da mesma classe** situam-se em torno de **0,8**, enquanto para **instâncias de classes distintas** variam entre **-0,25 e 0,25**, aproximando-se de uma **configuração ortogonal**.

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

Já na **Tabela 6**, é apresentada uma **comparação de desempenho** entre os modelos utilizando a métrica de trabalho salvo (AWSS@95%). A avaliação foi realizada considerando os mesmos modelos de classificação resultantes do ajuste fino da Fase 1 com LoRa, específicos para cada base, aplicados para rotular os vetores gerados pelo modelo **"Híbrido"** e pelo modelo **LoRa**, comparando-os com a rotulagem do modelo de classificação treinado sobre as representações sem ajuste do **modelo base**.

Os resultados indicam que, com exceção do banco de dados **BPA**, todos os demais apresentam **pequena melhora na métrica AWSS@95%**. Esse efeito pode ser atribuído ao **processo de regularização** decorrente da **proximidade da ortogonalidade entre as classes**, como analisado na Figura 16 imagem (d).

5 Discussão e Trabalhos Futuros

Conforme apresentado na Seção 4.2, o modelo base demonstrou boa capacidade de representação vetorial dos textos entre diferentes bancos de dados, como ilustrado na Figura 12. No entanto, essa representação não capturou agrupamentos locais entre classes dentro de um mesmo banco de dados. Com a aplicação do ajuste fino baseado em aprendizado contrastivo, o modelo aprimorou a separação dessas classes, conforme evidenciado na Figura 13. Esse efeito também foi observado na regressão logística, onde as densidades dos exemplos convergiram para valores extremos, conforme demonstrado na Figura 14. Entretanto, ao aplicar a mesma metodologia a diferentes bancos de dados, não houve melhora significativa na métrica de trabalho salvo, conforme indicado na Figura 15.

Esses resultados sugerem que o treinamento contrastivo reforça a similaridade entre exemplos de uma mesma base, promovendo agrupamentos mais coesos (13d). Contudo, ao manter 95% dos artigos relevantes na métrica de trabalho salvo, podem ser adicionados exemplos irrelevantes próximos ao limiar de classificação na regressão logística.

Por outro lado, as técnicas de otimização aplicadas ao modelo, como quantização e poda de parâmetros, não comprometeram significativamente o desempenho da métrica após o ajuste contrastivo. Além de reduzir a carga computacional, permitiram compactar o modelo em até oito vezes, reduzindo seu tamanho de 420MB para apenas 52MB, o que é essencial para hardware com restrições de capacidade. Essa compactação viabiliza a implementação do modelo em dispositivos especializados, servidores remotos e acervos de modelos base, sem prejuízo significativo no desempenho. Além disso, os módulos LoRa utilizados neste estudo representam menos de 0,1% do total de parâmetros, exigindo apenas 144KB de armazenamento, podendo ser reduzidos para 18KB com 4 bits de precisão. Isso possibilita uma transferência eficiente de conhecimento entre servidores e usuários, permitindo a adaptação de um modelo base para diferentes contextos com baixo custo computacional. Essa flexibilidade fomenta um ecossistema modular, onde habilidades específicas podem ser adicionadas e distribuídas de maneira eficiente.

A aplicação da metodologia descrita na Seção 2.3.4.3 nos cinco melhores modelos, segundo a métrica AWSS@95%, demonstrou que é possível preservar habilidades do modelo base enquanto se incorporam novas especializações. Como ilustrado na Figura 16, essa abordagem resulta na formação de agrupamentos mais estruturados e melhor separação de classes dentro de cada banco de dados. Esse efeito também é observado na matriz de similaridade, que se mantém estável na transição do modelo ajustado para o

híbrido (16c e 16d). Embora o modelo híbrido perca parte da confiança absoluta na classificação, ele mantém representações semânticas bem definidas. Essa abordagem viabiliza um ecossistema de habilidades específicas, onde módulos podem ser ativados conforme necessário, aumentando a eficiência computacional, especialmente em hardwares especializados para computação paralela, como GPUs. Dessa forma, otimiza-se o uso dos recursos computacionais, reduzindo o consumo energético e promovendo maior sustentabilidade.

Vale destacar que novas representações vetoriais foram adquiridas com apenas oito exemplos por classe e tempo reduzido de processamento, tornando essa abordagem sustentável para a especialização de modelos. Assim, o custo computacional dessas adaptações é insignificante para servidores especializados, permitindo sua aplicação em contextos específicos e facilitando a democratização do conhecimento científico de forma acessível e eficiente.

Diante dos achados deste estudo, a aplicabilidade de modelos base transcende a simples otimização de desempenho. Futuras pesquisas podem explorar o uso desses modelos para otimizar revisões sistemáticas, reduzindo custos e carga de trabalho na seleção de referências relevantes para pesquisas científicas. A otimização do trabalho salvo pode ser aprimorada combinando abordagens de aprendizado contínuo e modelos compactos especializados para extração automatizada e análise de padrões em artigos científicos. Essa estratégia permite a construção de um repositório dinâmico de conhecimento, no qual módulos especializados podem ser atualizados continuamente sem necessidade de re-treinamento completo, viabilizando um ecossistema evolutivo de IA aplicada à ciência.

Embora os resultados obtidos apresentem diversas perspectivas promissoras, algumas limitações foram identificadas na configuração de treinamento utilizada. Para aprimoramentos futuros, sugere-se a exploração de novos regimes de ajuste fino, como a destilação de conhecimento em modelos Transformer, conforme proposto por Wang et al. (2020b). Essa técnica pode permitir uma compactação ainda mais eficiente do modelo, mantendo sua eficácia em tarefas específicas.

Por fim, considerando o crescente interesse na sustentabilidade e na redução da pegada de carbono, futuras pesquisas devem incorporar métricas de impacto ambiental no treinamento e uso de modelos de IA. Estudos recentes Patterson et al. (2021), Strubell, Ganesh e McCallum (2019), Dodge et al. (2022) ressaltam a importância de avaliar as emissões de carbono, promovendo práticas mais sustentáveis na área. A exploração de novos regimes de ajuste fino e a avaliação direta das técnicas de otimização sobre o modelo base, sem adaptações adicionais, representam caminhos promissores para aprimorar a aplicabilidade da inteligência artificial em benefício da sociedade e do meio ambiente.

6 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. *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⟩.

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.

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.

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⟩.

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⟩.

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⟩.