UNIVERSIDADE FEDERAL DO CEARÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DA COMPUTAÇÃO

MESTRADO ACADÊMICO EM ENGENHARIA ELÉTRICA E DA COMPUTAÇÃO

VANESSA CARVALHO DO NASCIMENTO

GRAPH ATTENTION NETWORK PARA RECONHECIMENTO DE EMOÇÕES A PARTIR DE ELETROENCEFALOGRAMAS

SOBRAL

2026

VANESSA CARVALHO DO NASCIMENTO

GRAPH ATTENTION NETWORK PARA RECONHECIMENTO DE EMOÇÕES A PARTIR DE ELETROENCEFALOGRAMAS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia Elétrica e da Computação do Programa de Pós-Graduação em Engenharia Elétrica e da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia da Computação. Área de Concentração: Engenharia da Computação.

Orientador: Prof. Dr. Antônio Josefran de Oliveira Bastos.

SOBRAL

2026

VANESSA CARVALHO DO NASCIMENTO

GRAPH ATTENTION NETWORK PARA RECONHECIMENTO DE EMOÇÕES A PARTIR DE ELETROENCEFALOGRAMAS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia Elétrica e da Computação do Programa de Pós-Graduação em Engenharia Elétrica e da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia da Computação. Área de Concentração: Engenharia da Computação.

Aprovada em: 25/05/2026.

BANCA EXAMINADORA

Prof. Dr. Antônio Josefran de Oliveira Bastos (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Carlos Alexandre Rolim Fernandes
Universidade Federal do Ceará (UFC)

Profa. Dra. Jermana Lopes de Moraes
Universidade Federal do Ceará (UFC)

Dedico este trabalho aos meus pais, pelo amor, paciência e esforço que tiveram ao cuidar de mim.

AGRADECIMENTOS

Agradeço aos meus pais, por todo o amor e incentivo.

Ao meu namorado Valfrido, por sempre me apoiar e trazer otimismo à minha vida.

Ao Prof. Dr. A. Josefran de Oliveira Bastos pelas risadas, conversas divertidas e orientações sérias.

À Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP), pelo apoio financeiro por meio da manutenção da bolsa de auxílio.

RESUMO

O reconhecimento automático de estados emocionais tem relevância em diversas áreas, como saúde mental, interfaces cérebro-computador e sistemas de monitoramento afetivo. O Eletroencefalograma (EEG) é uma técnica não invasiva de registro da atividade elétrica cerebral que se destaca nesse contexto por refletir diretamente os estados neurais associados às emoções, sendo mais difícil de mascarar do que expressões faciais ou voz. No entanto, características como a natureza não estacionária dos sinais, a baixa relação sinal-ruído e a variabilidade entre indivíduos tornam essa tarefa desafiadora. Nesse cenário, as *Graph Neural Networks* (GNNs) se destacam por modelarem diretamente as relações entre os canais de EEG. Assim, este trabalho propõe um modelo baseado em *Graph Attention Network* (GAT), uma variante de GNN que pondera adaptativamente a contribuição de cada canal vizinho durante a agregação de informação, para a classificação de três estados emocionais (positivo, neutro e negativo) presentes no conjunto de dados *SJTU Emotion EEG Dataset* (SEED). Cada amostra de EEG é representada como um grafo cujos nós correspondem aos canais de EEG e cujas arestas conectam pares de canais cuja similaridade de cosseno entre os respectivos vetores de características supera o percentil 70 da distribuição de similaridades da amostra. Dentre os 62 canais e 5 bandas convencionalmente utilizadas na literatura, foram selecionados apenas 4 canais (FT7, FT8, T7 e T8) e 2 bandas (delta e theta), resultando em uma configuração compacta com potencial de aplicação em dispositivos vestíveis com número limitado de eletrodos. Os experimentos foram conduzidos sob o protocolo *Leave-One-Subject-Out* (LOSO), que avalia a capacidade de generalização entre indivíduos. O modelo alcança acurácia média de 95,38% ao longo de 10 execuções independentes, superando muitos trabalhos comparáveis que utilizam a configuração completa de canais e bandas, indicando que a modelagem baseada em grafos captura padrões discriminativos relevantes mesmo com uma configuração reduzida.

Palavras-chave: Eletroencefalograma; Reconhecimento de Emoções; SEED; Grafos; Graph Neural Networks; Graph Attention Networks.

ABSTRACT

The automatic recognition of emotional states is relevant in several areas, such as mental health, brain-computer interfaces, and affective monitoring systems. Electroencephalography (EEG) is a non-invasive technique for recording brain electrical activity that stands out in this context for directly reflecting the neural states associated with emotions, being harder to mask than facial expressions or voice. However, characteristics such as the non-stationary nature of the signals, low signal-to-noise ratio, and inter-subject variability make this task challenging. In this scenario, Graph Neural Networks (GNNs) stand out for directly modeling the relationships between EEG channels. Thus, this work proposes a model based on Graph Attention Network (GAT), a GNN variant that adaptively weights the contribution of each neighboring channel during information aggregation, for the classification of three emotional states (positive, neutral, and negative) present in the SJTU Emotion EEG Dataset (SEED). Each EEG sample is represented as a graph whose nodes correspond to EEG channels and whose edges connect pairs of channels whose cosine similarity between their respective feature vectors exceeds the 70th percentile of the sample similarity distribution. Among the 62 channels and 5 frequency bands conventionally used in the literature, only 4 channels (FT7, FT8, T7, and T8) and 2 bands (delta and theta) were selected, resulting in a compact configuration with potential application in wearable devices with a limited number of electrodes. Experiments were conducted under the Leave-One-Subject-Out (LOSO) protocol, which evaluates generalization across subjects. The model achieves a mean accuracy of 95.38% over 10 independent runs, outperforming many comparable works that use the full channel and band configuration, indicating that graph-based modeling captures relevant discriminative patterns even with a reduced configuration.

Keywords: Electroencephalography; Emotion Recognition; SEED; Graph Neural Networks; Graph Attention Networks.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

CNNs	*Convolutional Neural Networks*
DL	*Deep Learning*
EEG	Eletroencefalograma
GATs	*Graph Attention Networks*
GCNs	*Graph Convolutional Networks*
GNNs	*Graph Neural Networks*
GRU	*Gated Recurrent Unit*
LOSO	*Leave-One-Subject-Out*
LSTM	*Long Short-Term Memory*
ML	*Machine Learning*
PSD	*Power Spectral Density*
ReLU	*Rectified Linear Unit*
RNNs	*Recurrent Neural Networks*
RvNNs	*Recursive Neural Networks*
SD	*Subject-Dependent*
SEED	*SJTU Emotion EEG Dataset*
SI	*Subject-Independent*
SVM	*Support Vector Machine*

SUMÁRIO

1 INTRODUÇÃO

O EEG é uma técnica não invasiva de registro da atividade elétrica cerebral por meio de eletrodos posicionados sobre o couro cabeludo, com resolução temporal na ordem de milissegundos, o que permite acompanhar diretamente as variações rápidas da atividade neural (ZHANG *et al.*, 2023). Os sinais captados são frequentemente analisados no domínio da frequência, sendo decompostos em bandas clássicas como delta, theta, alfa, beta e gama, cada uma associada a diferentes estados cognitivos e fisiológicos (CHADDAD *et al.*, 2023).

Por meio de sensores relativamente acessíveis e portáteis, é possível adquirir dados em diferentes contextos experimentais e clínicos (SIMMATIS *et al.*, 2023). Suas aplicações incluem a detecção de crises epilépticas (LI *et al.*, 2022a), o monitoramento de doenças neurodegenerativas como o Alzheimer (AVILES *et al.*, 2024), o apoio ao diagnóstico de transtornos do humor (SIMMATIS *et al.*, 2023) e a análise de estados emocionais (KLEPL *et al.*, 2024).

No entanto, a análise de sinais de EEG envolve desafios relevantes. O sinal captado pelos eletrodos apresenta baixa relação sinal-ruído e é frequentemente contaminado por artefatos fisiológicos, como movimentos oculares e tensão muscular, e por interferências externas. Além disso, sua natureza não estacionária dificulta a modelagem por abordagens convencionais (CHADDAD *et al.*, 2023). Outro aspecto importante é o efeito de *volume conduction*, no qual a atividade de uma fonte neural se propaga por múltiplos eletrodos simultaneamente, reduzindo a resolução espacial do sinal (SIMMATIS *et al.*, 2023).

O reconhecimento automático de estados emocionais tem relevância em diversas áreas. No contexto da saúde mental, pode auxiliar no diagnóstico e monitoramento de transtornos como depressão e ansiedade, cujos sintomas frequentemente se manifestam em alterações nos padrões de atividade cerebral (SIMMATIS *et al.*, 2023). Em interfaces cérebro-computador, permite o desenvolvimento de sistemas adaptativos que respondem ao estado afetivo do usuário em tempo real (CHEN *et al.*, 2025). Em contextos de segurança, como a condução de veículos, a detecção de estados como estresse ou sonolência pode contribuir para a prevenção de acidentes (KLEPL *et al.*, 2024). Diferentemente de expressões faciais ou voz, que podem ser mascaradas conscientemente, os sinais de EEG refletem diretamente os estados neurais associados às emoções, o que os torna uma fonte de informação mais objetiva para essa tarefa (ZHENG; LU, 2015).

No contexto específico de reconhecimento de emoções, parte-se do pressuposto de que diferentes estados emocionais estão associados a padrões distintos de atividade elétrica

cerebral, que podem ser observados em diferentes bandas de frequência (ZHENG; LU, 2015). Para investigar essa hipótese, são frequentemente utilizados protocolos experimentais baseados em estímulos audiovisuais, nos quais emoções específicas são induzidas de forma controlada, permitindo a construção de conjuntos de dados rotulados (ZHENG; LU, 2015).

As primeiras abordagens para classificação de emoções a partir de EEG baseavam-se em métodos tradicionais de *Machine Learning* (ML), como o *Support Vector Machine* (SVM), combinados com características extraídas manualmente, como *Power Spectral Density* (PSD) e entropia diferencial (ZHENG; LU, 2015). Embora eficazes em determinados cenários, esses métodos tendem a modelar cada canal, correspondente a um eletrodo, de forma independente, sem capturar as relações de conectividade entre regiões cerebrais.

Com o avanço do *Deep Learning* (DL), diferentes arquiteturas passaram a ser exploradas para lidar com essas limitações. As *Convolutional Neural Networks* (CNNs) têm sido utilizadas para extrair padrões espaciais, muitas vezes a partir de representações como espectrogramas dos sinais (LI *et al.*, 2022). Já as *Recurrent Neural Networks* (RNNs), incluindo variantes como *Long Short-Term Memory* (LSTM) (HOCHREITER; SCHMIDHUBER, 1997) e *Gated Recurrent Unit* (GRU) (CHO *et al.*, 2014), são empregadas para modelar dependências temporais. Em alguns casos, essas abordagens são combinadas, buscando capturar simultaneamente informações espaciais e temporais (TIAN *et al.*, 2022). No entanto, essas arquiteturas impõem uma estrutura euclidiana regular aos dados, o que não é naturalmente adequado para o EEG, cujos canais estão distribuídos segundo relações funcionais e anatômicas que não são capturadas adequadamente pela proximidade espacial (LI *et al.*, 2022).

Nesse contexto, *Graph Neural Networks* (GNNs) permitem modelar diretamente as relações entre os canais de EEG, operando sobre dados estruturados em grafo sem impor uma grade regular. Nessa abordagem, os canais são representados como nós de um grafo e as arestas codificam relações de conectividade funcional entre eles. A propagação de informação ocorre por meio da agregação de características dos nós vizinhos, permitindo capturar interações entre regiões cerebrais (MOHAMMADI; KARWOWSKI, 2024). Entre as diferentes arquiteturas de GNNs, destacam-se as *Graph Attention Networks* (GATs), que utilizam mecanismos de atenção para atribuir pesos distintos às conexões entre os nós (VELIČKOVIĆ *et al.*, 2017).

Neste trabalho, é proposto um modelo baseado em GATs para a tarefa de classificação de emoções usando o conjunto de dados *SJTU Emotion EEG Dataset* (SEED) (ZHENG; LU, 2015). O SEED é composto por registros de EEG de 15 participantes, coletados em três

sessões distintas, durante a exibição de 15 clipes de filmes projetados para induzir três estados emocionais: positivo, neutro e negativo. Os sinais foram adquiridos com 62 canais segundo o sistema internacional 10-20, a uma taxa de amostragem de 1000 Hz, posteriormente reamostrada para 200 Hz.

O modelo proposto utiliza apenas quatro canais (FT7, FT8, T7 e T8) e duas bandas de frequência (delta e theta), correspondendo a uma configuração significativamente mais compacta do que a maioria dos trabalhos da literatura, que utilizam todos os 62 canais e as cinco bandas convencionais. A avaliação é conduzida em um cenário independente de indivíduo, utilizando o protocolo *Leave-One-Subject-Out* (LOSO), no qual o modelo é avaliado em dados de indivíduos não vistos durante o treinamento. O modelo alcança acurácia média de 95,38% ao longo de 10 execuções independentes, superando trabalhos comparáveis que utilizam a configuração completa de canais e bandas, indicando que a modelagem baseada em grafos é capaz de capturar padrões discriminativos relevantes mesmo com um número reduzido de eletrodos, com potencial de aplicação em dispositivos vestíveis.

Este trabalho está organizado da seguinte forma. O Capítulo 2 apresenta a fundamentação teórica, abordando conceitos de teoria de grafos, redes neurais artificiais, arquiteturas como CNNs, RNNs e GNNs, com ênfase nas GATs e no mecanismo de atenção, além da entropia diferencial. O Capítulo 3 discute os trabalhos relacionados. O Capítulo 4 descreve a metodologia adotada, incluindo a base de dados, a seleção de canais, a extração de características, a construção dos grafos, a arquitetura do modelo e o protocolo de treinamento e avaliação. O Capítulo 5 apresenta os resultados obtidos, incluindo o desempenho geral do modelo e a comparação com trabalhos relacionados. Por fim, o Capítulo 6 apresenta as conclusões e perspectivas para trabalhos futuros.

1.1 Objetivos

O objetivo deste trabalho é propor e avaliar um modelo baseado em GATs para a classificação de três estados emocionais (positivo, neutro e negativo) a partir de sinais de EEG, utilizando uma representação compacta em grafo com quatro canais e duas bandas de frequência. A avaliação é conduzida em cenário independente de indivíduo sob o protocolo LOSO.

De forma mais específica, os objetivos deste trabalho são:

– Apresentar a fundamentação teórica de GNNs, incluindo teoria de grafos e arquiteturas convolucionais e de atenção em grafos, necessária para a compreensão do modelo proposto;

- Selecionar um subconjunto compacto de canais e bandas de frequência do *dataset* SEED;
- Extrair características no domínio da frequência por meio de entropia diferencial, utilizando-as como atributos dos nós do grafo;
- Construir grafos por amostra nos quais os canais selecionados são representados como nós e as arestas são definidas por similaridade de cosseno entre os vetores de características;
- Implementar e treinar um modelo baseado em GATs para a classificação dos três estados emocionais presentes no *dataset* SEED;
- Avaliar o modelo sob o protocolo LOSO e comparar os resultados com trabalhos relacionados, distinguindo entre abordagens com e sem separação estrita entre indivíduos durante o treinamento.

2 FUNDAMENTAÇÃO TEÓRICA

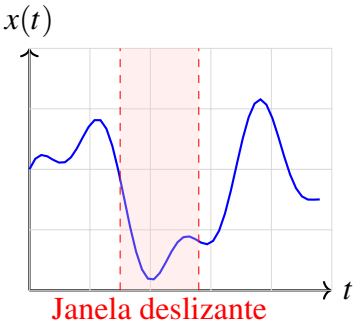
Este Capítulo apresenta os conceitos fundamentais necessários à compreensão deste trabalho. Inicialmente, são discutidas as diferenças entre dados euclidianos e não euclidianos, seguidas dos conceitos de teoria de grafos relevantes para a abordagem adotada. Em seguida, são introduzidos os principais modelos de DL considerados, incluindo redes neurais artificiais, CNNs, RNNs e Recursive Neural Networks, chegando às GNNs e suas variantes, com ênfase nas GATs. São também apresentadas aplicações de GNNs em diferentes domínios. Por fim, são descritas as características gerais dos sinais de EEG e a medida de entropia diferencial utilizada na extração de características.

2.1 Dados Euclidianos e Não Euclidianos

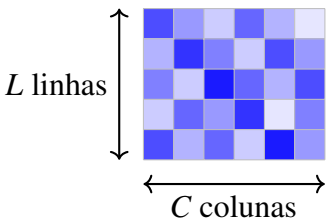
Grande parte dos modelos clássicos de ML e DL foram desenvolvidos para dados organizados em domínios euclidianos, isto é, em espaços vetoriais regulares com estrutura geométrica bem definida (BRONSTEIN *et al.*, 2017). Exemplos comuns incluem séries temporais, representadas como sequências ordenadas, e imagens, organizadas em grades regulares bidimensionais.

Como ilustrado na Figura 1a, séries temporais possuem uma estrutura unidimensional ordenada, em que cada instante t define a posição dos dados ao longo do tempo, possibilitando o uso de janelas deslizantes e arquiteturas que processam os dados sequencialmente. De forma semelhante, na Figura 1b, as imagens apresentam uma estrutura espacial regular na qual cada posição corresponde a um pixel com vizinhos fixos e previsíveis, o que permite a aplicação de convoluções para extrair padrões locais.

Figura 1 – Exemplos de dados euclidianos.



(a) Série temporal (1D).



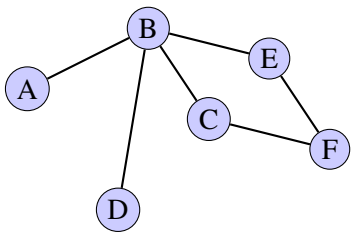
(b) Imagem representada como grade regular de $L \times C$ pixels (2D).

Fonte: Própria autora.

Entretanto, nem todos os dados podem ser representados de forma adequada por estruturas regulares. Em muitos casos, as relações entre os elementos são complexas e não admitem uma organização geométrica uniforme. Dados dessa natureza são denominados não euclidianos. Exemplos incluem redes sociais, moléculas e sistemas de transporte, nos quais as conexões entre entidades são definidas pelas próprias relações entre os elementos (BRONSTEIN *et al.*, 2017).

Como ilustrado na Figura 2, grafos constituem uma forma natural de representar esse tipo de dado, em que entidades correspondem a nós e suas interações a arestas. Diferentemente dos domínios euclidianos, a vizinhança não é determinada pela posição, mas pelas conexões do grafo, como ilustrado na Figura 3. Nesse caso, a informação está associada tanto aos atributos dos nós quanto às relações entre eles, e a ausência de uma ordenação regular dificulta a aplicação direta de operações como convoluções, que pressupõem uma estrutura espacial regular com vizinhanças fixas e predefinidas.

Figura 2 – Exemplo de grafo.



Fonte: Própria autora.

Figura 3 – Comparação entre estruturas euclidianas e não euclidianas, destacando o conceito de vizinhança.

(a) Euclidiano: vizinhança por posição.

(b) Não euclidiano: vizinhança por conexões.

Fonte: Própria autora.

Essa distinção evidencia a necessidade de abordagens capazes de incorporar as relações estruturais presentes nos dados. Esse desafio motivou o surgimento do campo do

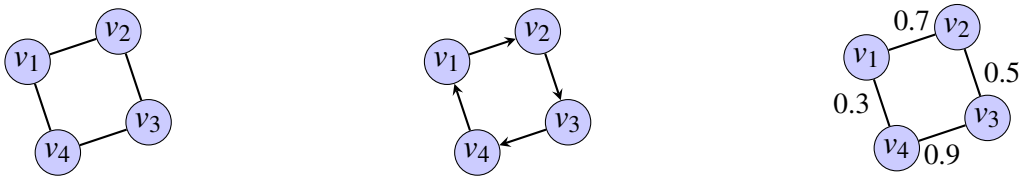
Aprendizado Profundo Geométrico (do inglês, *Geometric Deep Learning*), cujo objetivo é estender arquiteturas de DL para domínios não euclidianos, como grafos (BRONSTEIN *et al.*, 2021).

2.2 Teoria de Grafos

Um grafo G é definido como um par ordenado $G = (V, E)$, em que V é um conjunto finito de vértices (ou nós) e E é um conjunto de arestas. Em grafos não direcionados, cada aresta conecta dois vértices de forma não orientada, indicando uma relação bidirecional, sendo representada por um par não ordenado $\{u, v\}$, com $u, v \in V$, de modo que $E \subseteq \{\{u, v\} \mid u, v \in V\}$. Já em grafos direcionados, também chamados de dígrafos, cada aresta possui orientação e é representada por um par ordenado (u, v), indicando uma conexão de u para v, de modo que $E \subseteq V \times V$.

Além disso, quando existe uma função de peso $w : E \to \mathbb{R}$ que associa valores às arestas, o grafo é denominado ponderado. As principais variações de grafos são ilustradas na Figura 4, que apresenta exemplos de grafos não direcionados, direcionados e ponderados.

Figura 4 – Tipos de grafos: não direcionado, direcionado e ponderado.



(a) Grafo não direcionado (b) Grafo direcionado (c) Grafo ponderado

Fonte: Própria autora.

Para um vértice $u \in V$, define-se seu conjunto de vizinhança, no caso não direcionado, como:

$$\mathcal{N}_u = \{ v \in V : \{u, v\} \in E \}. \tag{2.1}$$

No caso não ponderado, o grau de um vértice é definido como o número de vértices adjacentes:

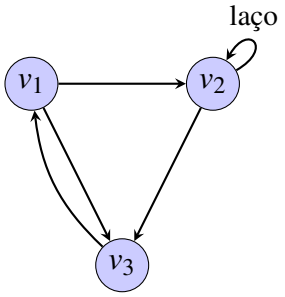
$$\deg(u) = |\mathcal{N}_u|. \tag{2.2}$$

Em grafos ponderados, o grau corresponde à soma dos pesos das arestas incidentes ao vértice:

$$\deg(u) = \sum_{v \in \mathcal{N}_u} w(\{u, v\}). \tag{2.3}$$

Em grafos direcionados, as arestas possuem orientação, o que permite decompor o grau de um vértice em duas componentes. O grau de entrada de um vértice u, denotado por $\deg^-(u)$, corresponde ao número de arestas que chegam a u, enquanto o grau de saída, denotado por $\deg^+(u)$, corresponde ao número de arestas que partem de u. A Figura 5 ilustra esses conceitos, incluindo o caso de laços, que contribuem simultaneamente para o grau de entrada e de saída.

Figura 5 – Grau de entrada ($\deg^-$) e grau de saída ($\deg^+$) em grafos direcionados.



$\deg^-(v_1) = 1$
$\deg^+(v_1) = 2$
$\deg^-(v_2) = 2$
$\deg^+(v_2) = 2$
$\deg^-(v_3) = 2$
$\deg^+(v_3) = 1$

Fonte: Própria autora.

Uma forma importante de representar grafos é por meio de estruturas matriciais. A matriz de adjacência $A \in \mathbb{R}^{|V|\times|V|}$ codifica as conexões entre os vértices, sendo definida como:

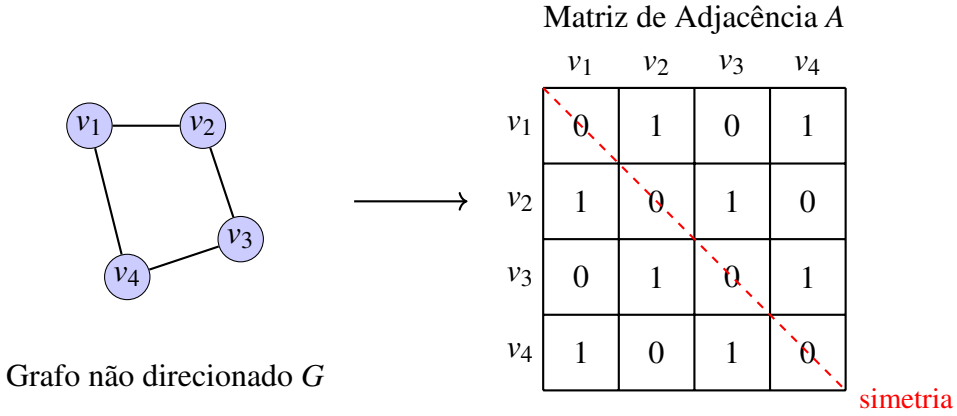
$$A_{ij} = \begin{cases} w(\{v_i, v_j\}), & \text{se } \{v_i, v_j\} \in E \\ 0, & \text{caso contrário,} \end{cases} \tag{2.4}$$

em que $\{v_1, \ldots, v_{|V|}\}$ é uma ordenação arbitrária do conjunto de vértices. No caso não ponderado, assume-se $w(e) = 1$ para toda aresta $e \in E$. A Figura 6 apresenta um exemplo de grafo não direcionado e sua correspondente matriz de adjacência, destacando a propriedade de simetria ($A_{ij} = A_{ji}$).

No caso de grafos direcionados, a matriz de adjacência não é, em geral, simétrica, uma vez que A_{ij} indica a existência de uma aresta que parte de v_i e chega em v_j. A Figura 7 exemplifica essa representação.

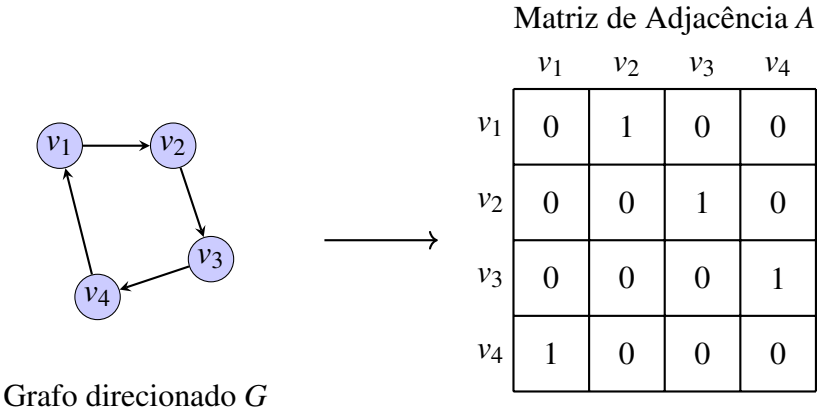
Outra representação importante é a matriz diagonal de graus D, definida como:

Figura 6 – Grafo não direcionado e sua correspondente matriz de adjacência A. A matriz é simétrica ($A_{ij} = A_{ji}$) e os elementos $A_{ij} = 1$ indicam a presença de uma aresta entre v_i e v_j.



Fonte: Própria autora.

Figura 7 – Grafo direcionado e sua matriz de adjacência. Diferente do caso não direcionado, a matriz não é necessariamente simétrica: $A_{ij} = 1$ indica uma aresta que parte de v_i e chega em v_j.



Fonte: Própria autora.

$$D_{ij} = \begin{cases} \deg(v_i), & \text{se } i = j \\ 0, & \text{caso contrário.} \end{cases} \tag{2.5}$$

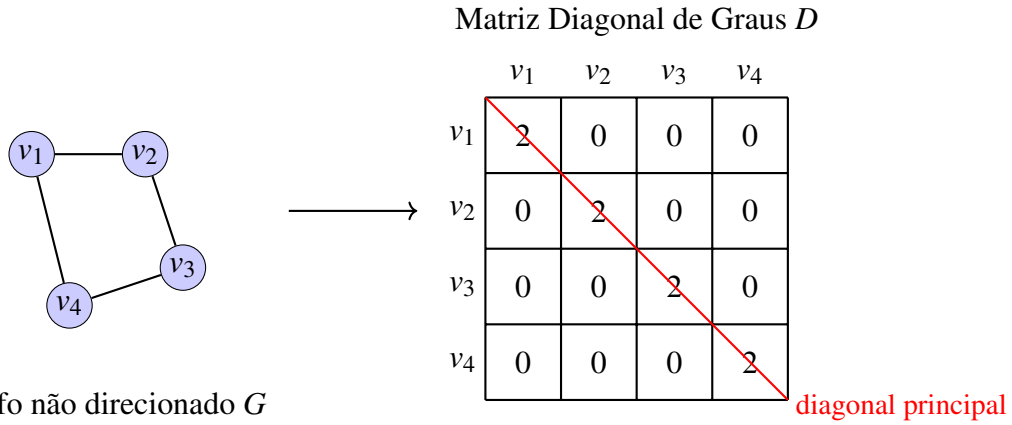
A Figura 8 ilustra a construção dessa matriz, na qual todos os elementos fora da diagonal principal são nulos, e cada elemento diagonal D_{ii} corresponde ao grau do vértice v_i.

2.2.1 Laplaciano

A Teoria Espectral de Grafos estuda as propriedades de um grafo a partir dos autovalores e autovetores de matrizes a ele associadas, em especial o laplaciano. Essa abordagem estabelece uma relação entre a estrutura do grafo e ferramentas da álgebra linear, fundamentando a definição de convoluções espectrais em grafos (DEFFERRARD *et al.*, 2016).

Considere um grafo não direcionado e sem pesos $G = (V, E)$ com $n = |V|$ vértices,

Figura 8 – Grafo e sua correspondente matriz diagonal de graus D, em que $D_{ii} = \deg(v_i)$ e $D_{ij} = 0$ para $i \neq j$.



Fonte: Própria autora.

indexados por $i \in \{1,\ldots,n\}$, cuja matriz de adjacência $A \in \{0,1\}^{n\times n}$ é simétrica por ser o grafo não direcionado. A matriz de graus $D \in \mathbb{R}^{n\times n}$ é diagonal com:

$$D_{ii} = \sum_{j=1}^{n} A_{ij}, \tag{2.6}$$

que corresponde ao número de vizinhos do vértice i.

O laplaciano, ou matriz laplaciana, do grafo é então definido como:

$$L = D - A. \tag{2.7}$$

Como A e D são simétricas, segue que L também é simétrica e, portanto, tem-se:

$$L = L^{\top}. \tag{2.8}$$

Considere um sinal $x \in \mathbb{R}^n$ definido sobre os vértices do grafo, em que $x_i \in \mathbb{R}$ é o valor do sinal no vértice i. O produto Lx produz um vetor cujo i-ésimo elemento é:

$$(Lx)_i \;=\; (Dx - Ax)_i \;=\; D_{ii}x_i - \sum_{j\in\mathcal{N}_i} x_j \;=\; \sum_{j\in\mathcal{N}_i} (x_i - x_j). \tag{2.9}$$

Esse valor acumula as diferenças entre o sinal em i e o sinal em cada um de seus vizinhos, de forma análoga ao operador laplaciano do cálculo diferencial, que mede a curvatura local de uma função contínua. O laplaciano do grafo é, portanto, um operador de diferença, seu valor em i é pequeno quando o sinal varia pouco entre i e seus vizinhos, e grande quando há variação abrupta.

Além disso, L é semidefinida positiva. Uma matriz $M \in \mathbb{R}^{n\times n}$ é dita semidefinida positiva quando, para qualquer vetor $x \in \mathbb{R}^n$, vale $x^{\top}Mx \geq 0$. Essa propriedade implica que todos

os autovalores de M são não negativos, o que é relevante para a análise espectral do grafo. Para verificar que L é semidefinida positiva, desenvolve-se a forma quadrática $x^\top Lx$:

$$\begin{aligned} x^\top Lx &= x^\top (D-A)x \\ &= x^\top Dx - x^\top Ax \\ &= \sum_{i=1}^{n} D_{ii}x_i^2 - \sum_{j=1}^{n}\sum_{i=1}^{n} A_{ji}x_i x_j \\ &= \sum_{i=1}^{n} \deg(i)\, x_i^2 - \sum_{\{i,j\}\in E} 2x_i x_j \\ &= \sum_{\{i,j\}\in E} \left(x_i^2 - 2x_i x_j + x_j^2\right) \\ &= \sum_{\{i,j\}\in E} (x_i - x_j)^2 \;\geq\; 0. \end{aligned} \tag{2.10}$$

O penúltimo passo usa o fato de que $\sum_i \deg(i)\, x_i^2 = \sum_{\{i,j\}\in E}(x_i^2 + x_j^2)$, pois cada aresta $\{i,j\}$ contribui com x_i^2 para o grau de i e com x_j^2 para o grau de j. Como cada parcela $(x_i - x_j)^2$ é não negativa, a soma é necessariamente não negativa, confirmando que L é semidefinida positiva.

A expressão $x^\top Lx = \sum_{\{i,j\}\in E}(x_i - x_j)^2$ mede a variação total do sinal x ao longo das arestas do grafo. Sinais suaves, que variam pouco entre vértices vizinhos, produzem valores pequenos, enquanto sinais que oscilam abruptamente produzem valores grandes.

Na prática, utiliza-se frequentemente o laplaciano normalizado (KIPF; WELLING, 2016):

$$L_{\text{norm}} \;=\; D^{-\frac{1}{2}} L D^{-\frac{1}{2}}. \tag{2.11}$$

Substituindo a definição do laplaciano $L = D - A$, obtém-se:

$$L_{\text{norm}} \;=\; D^{-\frac{1}{2}}(D-A)D^{-\frac{1}{2}} \;=\; D^{-\frac{1}{2}}DD^{-\frac{1}{2}} \;-\; D^{-\frac{1}{2}}AD^{-\frac{1}{2}}. \tag{2.12}$$

Como D é diagonal, tem-se $D^{-\frac{1}{2}}DD^{-\frac{1}{2}} = I$, pois cada elemento diagonal é dado por $D_{ii}^{-1/2} \cdot D_{ii} \cdot D_{ii}^{-1/2} = 1$. Assim, resulta:

$$L_{\text{norm}} \;=\; I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}. \tag{2.13}$$

Aqui, $D^{-1/2}$ denota a matriz diagonal cujos elementos são $D_{ii}^{-1/2}$. Essa operação requer que todos os vértices tenham grau positivo, o que equivale a assumir que o grafo não possui vértices isolados. Quando necessário, isso pode ser garantido adicionando laços ao grafo

(também chamados de autoconexões na literatura de GNNs), o que equivale a somar a matriz identidade à matriz de adjacência: $\tilde{A} = A + I$.

Essa normalização pondera a contribuição de cada aresta $\{i, j\}$ pelo fator $(D_{ii}D_{jj})^{-1/2}$, tornando os autovalores independentes do grau dos vértices e facilitando a comparação entre grafos com estruturas de conectividade distintas. Além disso, L_{norm} herda a simetria e a semidefinição positiva de L, com autovalores restritos ao intervalo $[0, 2]$.

2.2.2 Decomposição Espectral

Como visto na Seção anterior, o laplaciano L é uma matriz real, simétrica e semidefinida positiva. Pelo teorema espectral para matrizes reais simétricas (GOLUB; LOAN, 2013), L admite uma base ortonormal de autovetores. Isso significa que existem vetores $u_1, u_2, \ldots, u_n \in \mathbb{R}^n$, mutuamente ortogonais e de norma unitária, e escalares reais $\lambda_1, \lambda_2, \ldots, \lambda_n$ tais que:

$$Lu_i = \lambda_i u_i, \qquad i = 1, \ldots, n. \tag{2.14}$$

Organizando os autovetores como colunas de uma matriz $U = [u_1 \mid u_2 \mid \cdots \mid u_n]$ e os autovalores na diagonal de $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$, obtém-se a decomposição espectral (DEFFERRARD *et al.*, 2016):

$$L = U\Lambda U^\top, \tag{2.15}$$

em que $U^\top U = I$, em virtude da ortonormalidade dos autovetores.

A propriedade de semidefinição positiva de L, apresentada em (2.10), implica que:

$$x^\top Lx \geq 0, \quad \forall x \in \mathbb{R}^n. \tag{2.16}$$

Para relacionar essa propriedade aos autovalores, considere um autovetor u_i de L, associado ao autovalor λ_i, isto é, $Lu_i = \lambda_i u_i$. Substituindo $x = u_i$ na expressão acima, obtém-se:

$$u_i^\top Lu_i = u_i^\top (\lambda_i u_i) = \lambda_i u_i^\top u_i. \tag{2.17}$$

Como os autovetores são ortonormais, tem-se $u_i^\top u_i = 1$, e, portanto:

$$u_i^\top Lu_i = \lambda_i. \tag{2.18}$$

Como L é semidefinida positiva, segue que $u_i^\top Lu_i \geq 0$, o que implica:

$$\lambda_i \geq 0, \quad \forall i. \tag{2.19}$$

Assim, todos os autovalores de L são não negativos. Por convenção, os autovalores são ordenados de forma crescente:

$$0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n. \tag{2.20}$$

Para identificar o menor autovalor, considere o vetor $\mathbf{1} = (1, 1, \ldots, 1)^\top \in \mathbb{R}^n$. Substituindo esse vetor na expressão de Lx obtida em (2.9), tem-se:

$$(L\mathbf{1})_i = \sum_{j \in \mathcal{N}_i} (1 - 1) = 0, \tag{2.21}$$

para todo vértice i. Logo,

$$L\mathbf{1} = 0, \tag{2.22}$$

o que mostra que $\mathbf{1}$ é um autovetor associado ao autovalor $\lambda = 0$. Esse resultado reflete o fato de que o laplaciano mede variações entre vértices vizinhos, de modo que sinais constantes não apresentam variação ao longo das arestas do grafo. Assim, como todos os autovalores de L são não negativos, conclui-se que o menor deles satisfaz $\lambda_1 = 0$.

A decomposição $L = U\Lambda U^\top$ também simplifica o cálculo de funções de L. Em particular, ao expandir potências de L, os produtos intermediários $U^\top U = I$ se cancelam:

$$L^k = \underbrace{(U\Lambda U^\top)(U\Lambda U^\top)\cdots(U\Lambda U^\top)}_{k \text{ fatores}} = U\Lambda^k U^\top, \tag{2.23}$$

em que $\Lambda^k = \text{diag}(\lambda_1^k, \ldots, \lambda_n^k)$ é simplesmente a matriz diagonal com cada autovalor elevado à potência k.

2.2.3 Transformada de Fourier em Grafos

Os autovetores do laplaciano desempenham, no contexto de grafos, um papel análogo ao das funções senoidais na análise de Fourier clássica (SHUMAN *et al.*, 2013). Na transformada de Fourier discreta, um sinal é decomposto em componentes de diferentes frequências, cada uma associada a uma senoide cuja frequência mede a taxa de variação ao longo do domínio. Em grafos, os autovalores de L assumem esse papel, de modo que o autovetor u_1, associado a $\lambda_1 = 0$, é proporcional a $\mathbf{1}$, sendo constante sobre todos os vértices e representando a componente de frequência zero, enquanto autovetores associados a autovalores maiores apresentam variações progressivamente mais rápidas entre vértices adjacentes.

Seguindo essa analogia, a transformada de Fourier em grafos de um sinal $x \in \mathbb{R}^n$ é definida como sua projeção na base espectral (SHUMAN *et al.*, 2013):

$$\hat{x} = U^\top x. \tag{2.24}$$

Como $U^\top$ dispõe os autovetores como linhas, o i-ésimo elemento do vetor resultante é o produto interno $\hat{x}_i = u_i^\top x$, que mede o quanto o sinal x se alinha com o padrão de variação descrito por u_i. Valores de $|\hat{x}_i|$ grandes indicam que x possui componentes significativas na frequência λ_i. Como $U^\top U = I$, tem-se $UU^\top = I$, e portanto a transformada inversa é dada por:

$$x = U\hat{x} = \sum_{i=1}^{n} \hat{x}_i u_i, \tag{2.25}$$

que recupera x exatamente, mostrando que o sinal se decompõe como combinação linear dos autovetores ponderada pelos coeficientes espectrais.

@articleshuman2013emerging, title=The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains, author=Shuman, David I and Narang, Sunil K and Frossard, Pascal and Ortega, Antonio and Vandergheynst, Pierre, journal=IEEE signal processing magazine, volume=30, number=3, pages=83–98, year=2013, publisher=IEEE

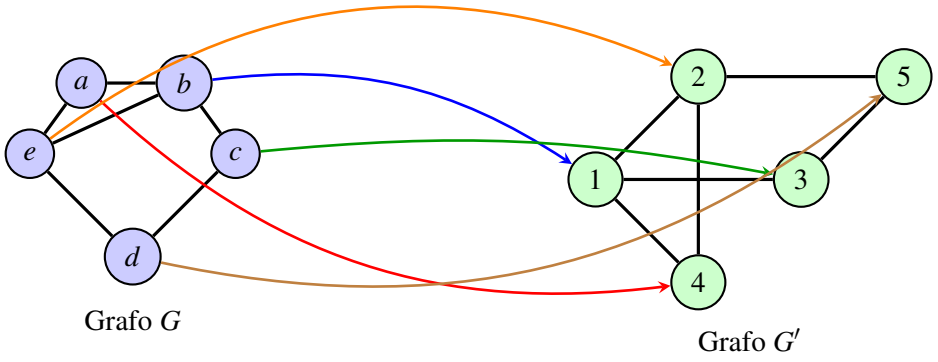
2.2.4 Isomorfismo de Grafos

O conceito de isomorfismo de grafos expressa a ideia de equivalência estrutural entre grafos. Em termos gerais, dois grafos são considerados equivalentes quando apresentam a mesma organização de conexões, ainda que seus vértices estejam rotulados ou dispostos de maneira distinta. Essa noção é ilustrada na Figura 9, na qual dois grafos com representações geométricas bastante diferentes apresentam, contudo, a mesma estrutura topológica.

Dois grafos $G = (V,E)$ e $G' = (V',E')$ são ditos isomorfos se existe uma função bijetiva $\varphi : V \to V'$ que estabelece uma correspondência um para um entre os vértices de G e os de G'. Ou seja, a função φ associa a cada vértice de G exatamente um vértice de G', e cada vértice de G' é imagem de exatamente um vértice de G. Essa função pode ser interpretada como uma renomeação dos vértices de G.

Entretanto, a bijetividade por si só não é suficiente. A função φ deve também preservar a estrutura de adjacência entre os vértices. No caso de grafos não direcionados, essa

Figura 9 – Exemplo de grafos isomorfos com cinco vértices.



Fonte: Própria autora.

condição é dada por:

$$\{u,v\} \in E \iff \{\varphi(u),\varphi(v)\} \in E', \tag{2.26}$$

para todo par $u,v \in V$. Já no caso de grafos direcionados, a preservação deve respeitar a orientação das arestas:

$$(u,v) \in E \iff (\varphi(u),\varphi(v)) \in E'. \tag{2.27}$$

Essa condição garante que G e G' possuem o mesmo padrão de conexões, diferindo apenas na rotulagem dos vértices. Propriedades estruturais como o grau dos vértices, a existência de caminhos e ciclos, e a decomposição em componentes conexas são preservadas por isomorfismos, evidenciando que a noção de equivalência entre grafos é independente da geometria da representação e está associada apenas à estrutura relacional.

A definição de isomorfismo de grafos é importante em diversos problemas teóricos e computacionais (BABAI, 2016). Do ponto de vista computacional, o problema de decidir se dois grafos são isomorfos consiste em verificar a existência de uma bijeção φ que preserve a adjacência. Quando tal bijeção existe, escreve-se:

$$G \cong G'. \tag{2.28}$$

2.3 Redes Neurais Artificiais

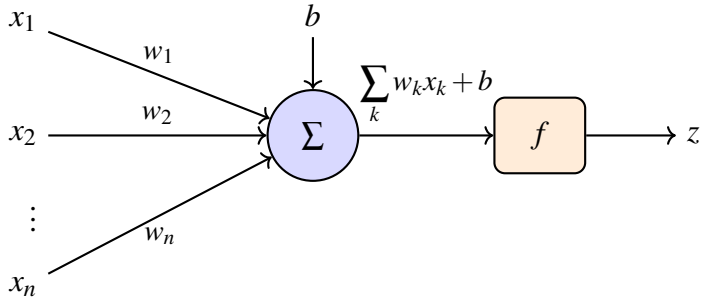
Redes neurais artificiais (do inglês, *Artificial Neural Networks*) são modelos computacionais inspirados no funcionamento do cérebro biológico, compostos por unidades de processamento chamadas neurônios artificiais, organizadas em camadas.

Cada neurônio recebe um conjunto de entradas $x_1, x_2, \ldots, x_n$, multiplica cada uma delas por um peso w_k correspondente, soma os produtos e adiciona um termo de viés b. O resultado dessa combinação linear é então passado por uma função de ativação f, que introduz não linearidade na operação. A saída do neurônio é dada por:

$$z = f\left(\sum_k w_k x_k + b\right). \tag{2.29}$$

Os pesos w_k determinam a importância de cada entrada, enquanto o viés b permite deslocar o limiar de ativação independentemente das entradas. Sem a função de ativação, a rede inteira se reduziria a uma única transformação linear, independentemente do número de camadas. A Figura 10 ilustra a estrutura de um neurônio artificial.

Figura 10 – Estrutura de um neurônio artificial. As entradas x_k são combinadas linearmente pelos pesos w_k, somadas ao viés b, e o resultado é transformado pela função de ativação f.

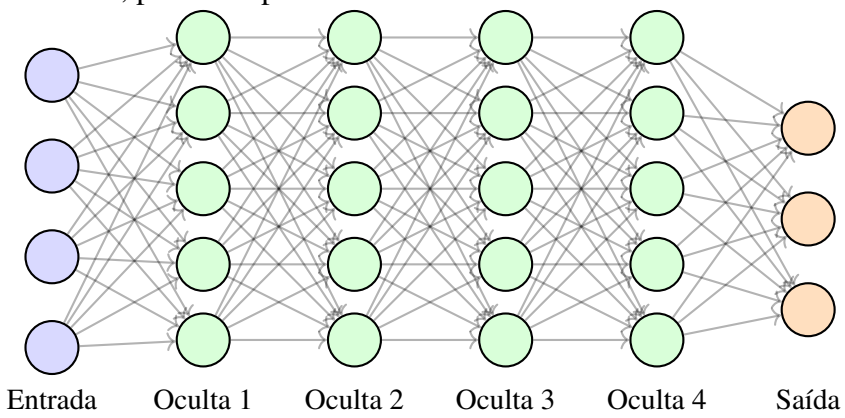


Fonte: Própria autora.

Quando múltiplas camadas de neurônios são empilhadas, a saída de uma camada serve como entrada para a seguinte, formando uma rede neural profunda (do inglês, *Deep Neural Network*), ilustrada na Figura 11. O termo *profundo* refere-se ao número de camadas intermediárias, denominadas camadas ocultas, que separam a camada de entrada da camada de saída. À medida que a informação se propaga pelas camadas, as representações são progressivamente combinadas e refinadas, permitindo que camadas mais profundas capturem estruturas cada vez mais abstratas. Essa capacidade de construir representações hierárquicas a partir dos dados é o que distingue as redes profundas de modelos com uma única camada (GOODFELLOW *et al.*, 2016).

O treinamento de uma rede neural consiste em ajustar os pesos w_k de forma a minimizar o erro do modelo sobre os dados de treinamento. Esse erro é medido por uma função de perda $\mathscr{L}$, que quantifica a discrepância entre as saídas produzidas pelo modelo e os valores esperados. A escolha da função de perda depende da tarefa. Em problemas de classificação

Figura 11 – Rede neural com quatro camadas ocultas. A informação flui da camada de entrada para a camada de saída, passando pelas camadas ocultas.



Fonte: Própria autora.

com duas classes, uma das mais utilizadas é a entropia cruzada binária (do inglês, *binary cross-entropy*), definida para um exemplo com rótulo verdadeiro $y \in \{0,1\}$ e probabilidade prevista $\hat{y} \in (0,1)$ como:

$$\mathscr{L} = -\left[y \log \hat{y} + (1-y)\log(1-\hat{y})\right]. \tag{2.30}$$

Quando $\hat{y}$ está próximo de y, o valor da perda é pequeno. Quando o modelo erra muito, a perda é grande. Em problemas de regressão, utiliza-se o erro quadrático médio (do inglês, *Mean Squared Error* - MSE):

$$\mathscr{L} = \frac{1}{n}\sum_{k=1}^{n}(y_k - \hat{y}_k)^2, \tag{2.31}$$

em que y_k é o valor esperado, $\hat{y}_k$ é o valor previsto pelo modelo para o k-ésimo exemplo e n é o número de exemplos. Quanto maior a diferença entre o valor previsto e o esperado, maior é a perda.

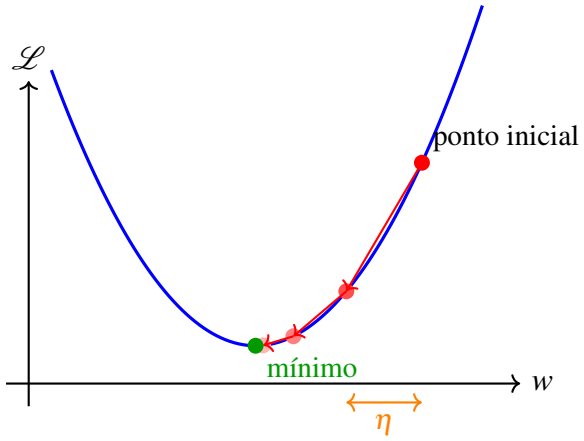
A minimização da função de perda é realizada por algoritmos de otimização baseados em gradiente. A ideia é calcular a derivada parcial da função de perda em relação a cada peso, o que indica a direção em que a perda aumenta, e atualizar os pesos na direção oposta. Esse processo é chamado de gradiente descendente e é ilustrado na Figura 12.

A cada iteração, os pesos são atualizados pela regra:

$$w_k \leftarrow w_k - \eta \frac{\partial \mathscr{L}}{\partial w_k}, \tag{2.32}$$

em que $\eta > 0$ é a taxa de aprendizado, que controla o tamanho do passo de atualização. Valores muito grandes podem fazer o modelo oscilar e não convergir, enquanto valores muito pequenos tornam o treinamento lento.

Figura 12 – Ilustração do gradiente descendente. Os pesos são atualizados iterativamente na direção que reduz a função de perda, convergindo para um mínimo.



Fonte: Própria autora.

Variantes adaptativas do gradiente descendente ajustam a taxa de aprendizado individualmente para cada parâmetro ao longo do treinamento, o que tende a acelerar a convergência e reduzir a sensibilidade à escolha do valor inicial de η. Uma dessas variantes é o Adam, que mantém médias móveis do gradiente e do seu quadrado para adaptar a taxa de aprendizado de cada parâmetro ao longo do treinamento (KINGMA; BA, 2014). O AdamW (LOSHCHILOV; HUTTER, 2017), utilizado neste trabalho, estende o Adam ao modificar a forma como o decaimento de pesos é aplicado. Enquanto o Adam o incorpora indiretamente à função de perda, o AdamW o aplica diretamente sobre os parâmetros, de forma independente do gradiente, o que resulta em regularização mais uniforme e, empiricamente, em melhor generalização.

2.3.1 Funções de Ativação

Funções de ativação introduzem não linearidades na rede, permitindo ao modelo aprender representações complexas dos dados. A *Rectified Linear Unit* (ReLU) é uma das funções mais utilizadas e é definida como:

$$\mathrm{ReLU}(x) = \max(0, x). \tag{2.33}$$

Essa função mantém valores positivos inalterados e atribui valor zero às entradas negativas.

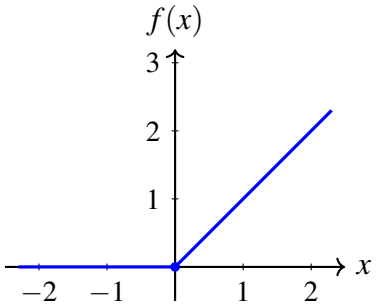
Entretanto, a ReLU apresenta uma limitação conhecida como *dying ReLU*. Esse fenômeno ocorre quando os pesos da rede fazem com que determinadas unidades passem a receber predominantemente entradas negativas, resultando em saídas sempre iguais a zero. Nessas condições, o gradiente também se torna nulo, impedindo que os parâmetros associados a esses neurônios sejam atualizados durante o treinamento.

Uma alternativa proposta para mitigar esse problema é a função *Leaky Rectified Linear Unit* (Leaky ReLU). Diferentemente da ReLU padrão, essa função permite que valores negativos produzam uma pequena ativação linear, em vez de serem completamente anulados. A Leaky ReLU é definida como:

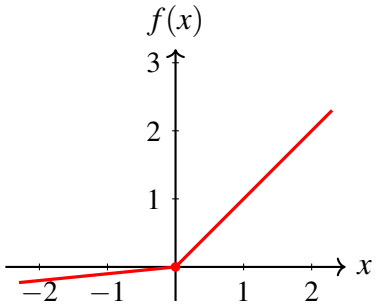
$$\text{LeakyReLU}(x) = \begin{cases} x, & \text{se } x \geq 0 \\ \alpha x, & \text{se } x < 0 \end{cases}, \tag{2.34}$$

em que α é um parâmetro pequeno e positivo, geralmente escolhido como $\alpha \approx 0,01$. Dessa forma, mesmo para entradas negativas, a função mantém um gradiente diferente de zero, o que contribui para reduzir a ocorrência de neurônios permanentemente inativos durante o treinamento. As funções ReLU e Leaky ReLU são ilustradas na Figura 13.

Figura 13 – Funções de ativação ReLU e Leaky ReLU.



(a) ReLU: entradas negativas são anuladas.



(b) Leaky ReLU: entradas negativas produzem pequena ativação com inclinação α.

Fonte: Própria autora.

A escolha da função de ativação afeta o desempenho do modelo em aspectos como velocidade de convergência, estabilidade do treinamento e capacidade de generalização. Em muitos casos, essa escolha é realizada empiricamente, considerando as características específicas do problema e da arquitetura da rede neural.

2.3.2 Regularização

Redes neurais profundas possuem um grande número de parâmetros e podem ajustar-se excessivamente aos dados de treinamento, fenômeno conhecido como *overfitting* (BISHOP; NASRABADI, 2006). Nesse caso, o modelo memoriza os dados de treinamento, incluindo ruídos e variações não representativas da distribuição geral dos dados, o que prejudica sua capacidade

de generalização para dados não vistos. Técnicas de regularização são utilizadas para mitigar esse problema, restringindo a complexidade do modelo durante o treinamento.

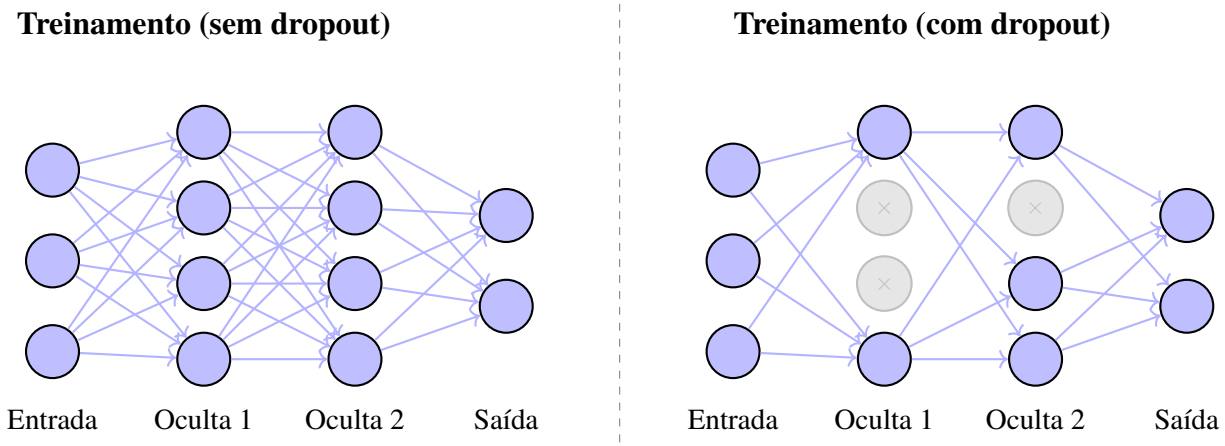
Uma das formas mais comuns de regularização é a penalização L_2, também conhecida como *weight decay* (BISHOP; NASRABADI, 2006). Nesse método, adiciona-se à função de perda um termo proporcional à soma dos quadrados dos pesos do modelo:

$$\mathscr{L}_{\text{reg}} = \mathscr{L} + \frac{\lambda}{2} \sum_k w_k^2, \tag{2.35}$$

em que $\mathscr{L}$ representa a função de perda original, w_k são os parâmetros treináveis do modelo, indexados por k, e $\lambda > 0$ controla a intensidade da penalização. Esse termo penaliza pesos de grande magnitude, pois pesos muito grandes tendem a tornar o modelo sensível a pequenas variações nos dados de entrada. Ao mantê-los pequenos, favorece-se soluções mais suaves e com melhor capacidade de generalização.

Outra técnica de regularização é o *dropout* (SRIVASTAVA *et al.*, 2014). A Figura 14 ilustra seu funcionamento. Durante o treinamento, essa abordagem consiste em desativar aleatoriamente parte dos neurônios da rede a cada iteração, impedindo que unidades específicas se tornem excessivamente dependentes umas das outras para produzir boas predições. Como consequência, o modelo é forçado a aprender representações mais informativas e distribuídas, o que contribui para melhorar sua capacidade de generalização.

Figura 14 – Ilustração do *dropout*. À esquerda, a rede completa durante o treinamento. À direita, neurônios desativados aleatoriamente (em cinza) em uma iteração, forçando o modelo a não depender de unidades específicas.



Fonte: Própria autora.

2.3.3 Normalização

A normalização é uma técnica utilizada em arquiteturas de DL para acelerar o treinamento, melhorar a estabilidade numérica e promover melhor generalização do modelo (SANTURKAR *et al.*, 2018). Seu objetivo é controlar a escala e a distribuição das ativações internas da rede, evitando que valores muito grandes ou muito pequenos dificultem o processo de otimização.

Um dos métodos mais conhecidos é a *Batch Normalization*. Nessa abordagem, as ativações de uma camada são normalizadas durante o treinamento com base nas estatísticas de pequenos subconjuntos dos dados de entrada, denominados mini-lotes (do inglês, *mini-batches*). Um mini-lote corresponde a um grupo de amostras processadas conjuntamente em uma única iteração de treinamento, em vez de utilizar todo o conjunto de dados de uma só vez. Para cada mini-lote, a normalização ajusta as ativações de forma que apresentem média próxima de zero e variância próxima de um. O uso de normalização frequentemente permite o uso de taxas de aprendizado mais elevadas, reduz a sensibilidade à inicialização dos parâmetros e contribui para um treinamento mais rápido e estável (IOFFE; SZEGEDY, 2015).

Outra normalização conhecida é a *Layer Normalization*, que difere da *Batch Normalization* no eixo sobre o qual as estatísticas são calculadas. Em vez de normalizar ao longo das amostras do mini-lote, a *Layer Normalization* calcula média e desvio padrão ao longo das dimensões do próprio vetor de ativação de cada exemplo, de forma independente para cada amostra.

Essa característica a torna independente do tamanho do mini-lote e adequada para cenários em que o número de elementos por amostra é variável, como ocorre em modelos que operam sobre grafos com número variável de nós. Por essa razão, a *Layer Normalization* é adotada na camada de projeção inicial do modelo proposto neste trabalho.

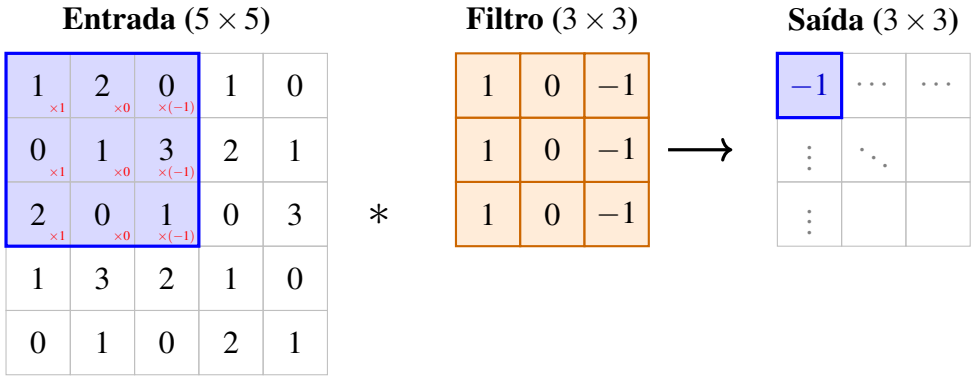
2.4 Convolutional Neural Networks

As *Convolutional Neural Networks* (CNNs) são um tipo de rede neural profunda projetada para processar dados com estrutura espacial regular, como imagens. Ao contrário de arquiteturas totalmente conectadas, as CNNs exploram a estrutura local dos dados por meio de operações de convolução, o que reduz o número de parâmetros e permite capturar padrões espaciais de forma eficiente (GOODFELLOW *et al.*, 2016).

2.4.1 Convolução

A operação de convolução consiste em aplicar um pequeno conjunto de pesos, denominado filtro ou *kernel*, que percorre a entrada em todas as posições possíveis e produz um mapa de características (do inglês, *feature map*) que destaca padrões locais (DUMOULIN; VISIN, 2016). Esse deslocamento do filtro sobre a entrada é controlado por um parâmetro chamado *passo* (do inglês, *stride*), que define quantas posições o filtro avança a cada aplicação, de modo que valores menores de passo produzem mapas de saída com mais posições, enquanto valores maiores resultam em mapas menores, reduzindo o custo computacional das camadas seguintes. Esse processo é ilustrado na Figura 15.

Figura 15 – Operação de convolução com filtro 3×3 e passo 1 aplicado a uma entrada 5×5. A janela destacada em azul é posicionada sobre a entrada, os valores são multiplicados elemento a elemento pelo filtro, e a soma dos produtos gera o valor correspondente no mapa de saída.



Fonte: Própria autora.

O mesmo filtro é reutilizado em todas as posições da entrada, o que é denominado compartilhamento de pesos. Com isso, o modelo aprende um único filtro capaz de detectar um determinado padrão independentemente de sua posição na entrada, reduzindo o número de parâmetros em comparação com arquiteturas totalmente conectadas. Uma CNN empilha múltiplas camadas convolucionais, cada uma aprendendo filtros que capturam padrões de complexidade crescente, desde bordas simples nas primeiras camadas até formas e texturas nas mais profundas, no caso de imagens.

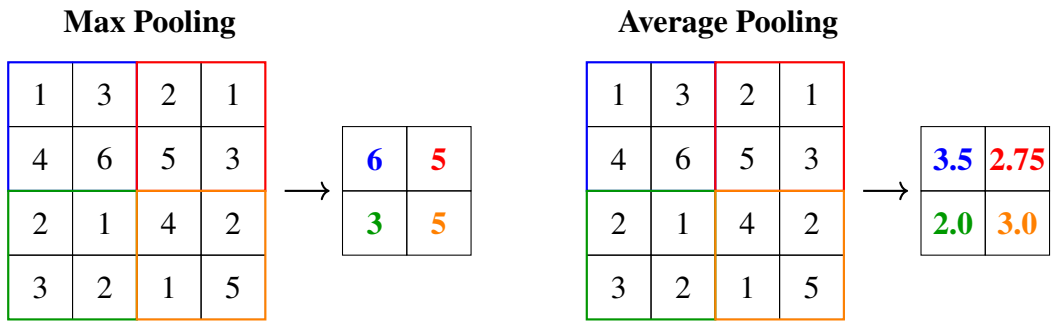
2.4.2 **Pooling**

O *pooling* é uma operação de redução espacial aplicada aos mapas de características após as camadas convolucionais. Essa operação percorre o mapa por meio de uma janela de tamanho fixo, por exemplo 2×2, com um determinado passo, aplicando uma função de

redução em cada posição. Ao reduzir a resolução espacial, o custo computacional das camadas seguintes diminui e o modelo torna-se menos sensível a pequenas translações e distorções na entrada (GOODFELLOW *et al.*, 2016).

As duas operações mais comuns são ilustradas na Figura 16. O *max pooling* seleciona o valor máximo em uma janela local, retendo o valor mais alto de cada janela e favorecendo a detecção de padrões relevantes (KRIZHEVSKY *et al.*, 2012). O *average pooling* calcula a média dos valores na mesma janela, produzindo uma representação mais suave da região.

Figura 16 – Operações de *max pooling* e *average pooling* com janela 2×2 e passo 2.



Fonte: Própria autora.

2.4.3 Camadas Totalmente Conectadas

Após as camadas convolucionais e de *pooling*, os mapas de características resultantes são reorganizados em um único vetor unidimensional por meio de uma operação chamada achatamento (do inglês, *flatten*). Esse vetor concatena todos os valores dos mapas de características e serve como entrada para uma ou mais camadas totalmente conectadas (do inglês, *fully connected layers*), nas quais cada neurônio está conectado a todos os neurônios da camada anterior.

Essas camadas são responsáveis por combinar as características extraídas pelas camadas convolucionais e produzir a saída final da rede, como um vetor de *scores*, em que cada elemento corresponde a uma classe e indica a confiança do modelo naquela predição.

2.5 Recurrent Neural Networks

As RNNs são uma classe de modelos projetados para o processamento de dados sequenciais, nos quais a ordem dos elementos é importante para a interpretação da informação. Diferentemente das redes neurais apresentadas anteriormente, que processam cada entrada de forma independente, as RNNs incorporam conexões recorrentes que permitem que informações

de instantes anteriores influenciem o processamento dos instantes subsequentes, introduzindo um mecanismo de memória interna (GOODFELLOW *et al.*, 2016).

Considere uma sequência de entradas $(x_1, x_2, \ldots, x_T)$. Em uma RNN, o processamento é realizado de forma iterativa, mantendo-se um estado oculto h_t em cada passo temporal t, que acumula informação sobre os elementos processados até aquele instante. A dinâmica desse estado pode ser descrita como:

$$h_t = \sigma(W_x x_t + W_h h_{t-1} + b), \tag{2.36}$$

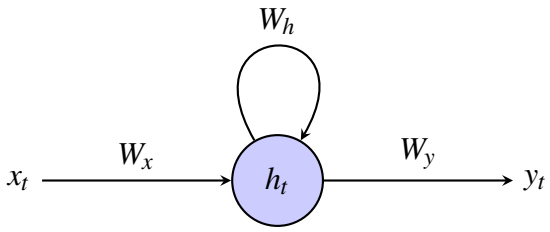
em que W_x e W_h são matrizes de pesos compartilhadas ao longo do tempo, b é um vetor de viés e $\sigma(\cdot)$ é uma função de ativação não linear. A saída y_t é obtida a partir do estado oculto por meio de uma transformação adicional, definida como:

$$y_t = \phi(W_y h_t + c), \tag{2.37}$$

em que W_y representa os pesos associados à camada de saída e $\phi(\cdot)$ é uma função apropriada à tarefa.

A Figura 17 ilustra uma representação compacta desse mecanismo recorrente. Nela, observa-se que o estado oculto h_t recebe tanto a entrada atual x_t quanto sua própria ativação anterior, evidenciando o papel da conexão recorrente na propagação de informação ao longo do tempo.

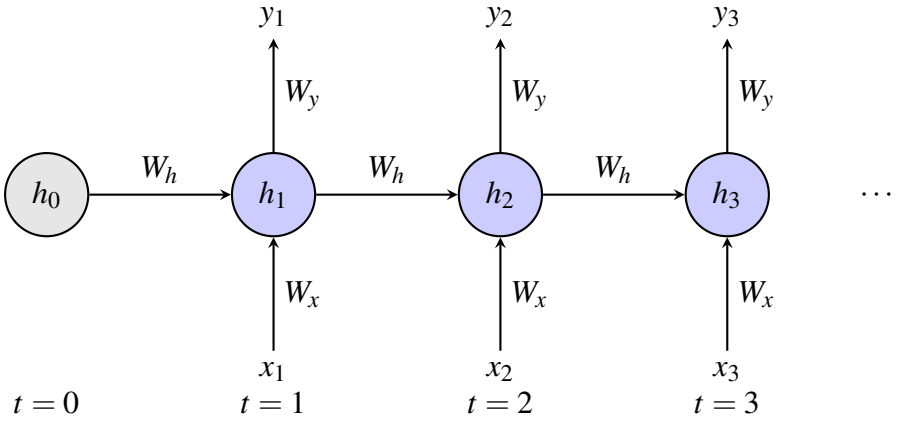
Figura 17 – Representação compacta de uma RNN. O estado oculto h_t recebe a entrada atual x_t e sua própria ativação anterior, modelada pela conexão recorrente.



Fonte: Própria autora.

Embora essa representação compacta seja conveniente, ela oculta a natureza sequencial do processamento. Para tornar mais explícita a dependência temporal, a Figura 18 apresenta a RNN desdobrada no tempo, na qual cada passo temporal é representado como uma cópia da mesma unidade computacional. Nessa visualização, observa-se que os parâmetros são compartilhados entre os diferentes instantes e que o estado oculto estabelece uma cadeia de dependências ao longo da sequência.

Figura 18 – RNN desdobrada no tempo.



Fonte: Própria autora.

O compartilhamento de parâmetros ao longo da dimensão temporal permite que as RNNs processem sequências de comprimento variável e capturem dependências entre elementos distantes da sequência. Essa característica torna esses modelos adequados para aplicações como modelagem de linguagem, reconhecimento de fala e análise de séries temporais (GRAVES *et al.*, 2013).

No entanto, o treinamento de RNNs enfrenta uma limitação relacionada ao algoritmo *Backpropagation Through Time* (BPTT), no qual o gradiente do erro é calculado pela aplicação retroativa da regra da cadeia ao longo dos passos temporais. Quando os pesos recorrentes favorecem o decaimento do sinal, o gradiente torna-se progressivamente nulo ao retroceder no tempo, fenômeno conhecido como desaparecimento do gradiente (do inglês, *vanishing gradient*). No caso oposto, o gradiente cresce de forma descontrolada, levando a instabilidades numéricas durante o treinamento e caracterizando a explosão do gradiente (do inglês, *exploding gradient*) (BENGIO *et al.*, 1994). Em ambos os casos, o modelo perde a capacidade de associar informações separadas por muitos passos temporais, que é o que se espera de uma rede recorrente.

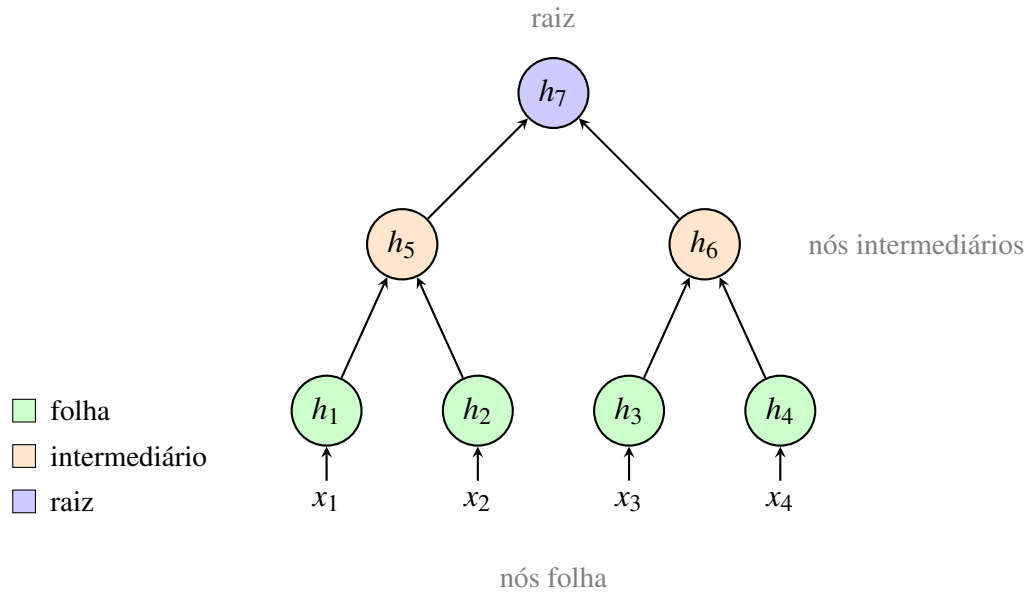
Para mitigar essas limitações, foram propostas variantes recorrentes com mecanismos de controle do fluxo de informação, como as *Long Short-Term Memory* (LSTM) (HOCHREITER; SCHMIDHUBER, 1997) e as *Gated Recurrent Units* (GRU) (CHO *et al.*, 2014). Esses modelos introduzem portas parametrizadas que regulam o que deve ser armazenado, atualizado ou descartado ao longo do tempo, favorecendo a preservação de dependências de longo alcance.

2.6 Recursive Neural Networks

As *Recursive Neural Networks* (RvNNs) generalizam o mecanismo de recorrência das RNNs para dados organizados segundo estruturas hierárquicas, como árvores (SPERDUTI; STARITA, 1997). Enquanto as RNNs operam sobre sequências lineares, nas quais cada elemento depende de um único predecessor, as RvNNs permitem que cada nó seja composto a partir de múltiplos predecessores, conforme definido pela estrutura dos dados.

Nesse contexto, a recorrência deixa de estar associada ao tempo e passa a refletir a organização hierárquica da árvore. Cada nó mantém uma representação latente que é calculada a partir das representações de seus nós filhos. Como ilustrado na Figura 19, esse processo ocorre de forma ascendente: inicialmente são calculadas as representações dos nós folha, que dependem apenas de suas características locais, e, em seguida, essas representações são combinadas progressivamente até a obtenção de uma representação global na raiz, que sintetiza a informação de toda a estrutura.

Figura 19 – Processamento em uma rede neural recursiva. As representações são propagadas de forma ascendente, desde os nós folha até a raiz, que produz uma representação global da estrutura.



Fonte: Própria autora.

Considere uma árvore enraizada na qual cada nó i possui um conjunto de filhos $\mathscr{C}(i)$. A representação latente do nó i é definida como:

$$h_i = \psi\big(x_i, \{\, h_j \mid j \in \mathscr{C}(i) \,\}\big)\,, \tag{2.38}$$

em que x_i representa os atributos do nó i e $\psi(\cdot)$ é uma função de composição compartilhada entre todos os nós. Essa função combina as representações dos filhos e produz uma representação que sintetiza a informação da subárvore enraizada em i.

Uma RNN pode ser vista como um caso particular dessa formulação, em que a estrutura subjacente é uma cadeia linear. Nesse caso, cada nó possui exatamente um predecessor, e a função recursiva reduz-se à atualização sequencial típica das RNNs:

$$h_t = \psi(x_t, h_{t-1}). \tag{2.39}$$

Tanto redes recorrentes quanto recursivas compartilham o princípio de aplicar repetidamente uma mesma função parametrizada ao longo da estrutura dos dados. A diferença reside na topologia sobre a qual essa função opera, que é sequencial nas RNNs e hierárquica nas RvNNs.

As RvNNs têm sido aplicadas em tarefas que envolvem estruturas hierárquicas, como análise sintática em processamento de linguagem natural, em que sentenças são representadas por árvores sintáticas, e classificação sobre estruturas em árvore (SOCHER *et al.*, 2013). Nessas aplicações, a representação da raiz é utilizada como descrição global da estrutura completa.

Apesar de ampliarem o escopo das RNNs, as RvNNs pressupõem uma organização hierárquica bem definida nos dados. Grafos que contêm ciclos ou padrões de conectividade sem estrutura de árvore não podem ser processados diretamente por esse modelo. Essa restrição motivou o desenvolvimento de arquiteturas capazes de operar sobre grafos arbitrários, preservando as relações entre os nós sem depender de uma hierarquia predefinida, o que conduz ao próximo tópico.

2.7 Graph Neural Networks

As Graph Neural Networks (GNNs) são uma classe de redes neurais projetadas para operar sobre dados estruturados na forma de grafos. Essa abordagem é motivada pela natureza não euclidiana de muitos dados do mundo real, como discutido na Seção 2.1, para os quais arquiteturas convencionais como CNNs e RNNs não são diretamente aplicáveis. Os primeiros trabalhos nesse contexto foram propostos por Gori et al. (GORI *et al.*, 2005) e posteriormente desenvolvidos por Scarselli et al. (SCARSELLI *et al.*, 2008), que introduziram modelos capazes de processar grafos por meio de funções iterativas definidas sobre a vizinhança de cada nó.

As GNNs podem ser compreendidas como uma generalização das *recursive neural networks*, nas quais a informação é agregada de nós filhos para seus respectivos pais ao longo de

uma estrutura em árvore. Em um grafo arbitrário, porém, um nó pode estar conectado a múltiplos vizinhos sem uma noção clara de hierarquia, de modo que a atualização de cada nó passa a ser definida a partir de seus vizinhos no grafo, sem depender de relações pai–filho ou de uma ordem fixa de processamento (SCARSELLI *et al.*, 2008).

A cada nó i é associado um vetor h_i, denominado *representação latente*, que codifica as informações relevantes daquele nó a partir de seus atributos e da estrutura do grafo. Essa representação não é observada diretamente nos dados, mas aprendida durante o treinamento de modo a ser útil para a tarefa de interesse. As representações latentes são atualizadas de forma iterativa ao longo das camadas do modelo por meio de operações definidas sobre a vizinhança de cada nó, preservando o princípio de compartilhamento de parâmetros das *recursive neural networks*, mas sem a restrição a estruturas hierárquicas, o que torna as GNNs adequadas para dados com padrões de conectividade arbitrários (BRONSTEIN *et al.*, 2021).

2.7.1 Invariância e Equivariância a Permutações

Em domínios euclidianos, como imagens, as CNNs exploram propriedades estruturais bem definidas. Os padrões relevantes podem ser reconhecidos independentemente de sua posição no espaço, o que caracteriza a invariância à translação, e dependências locais tendem a ser mais informativas do que interações de longa distância, o que caracteriza a localidade espacial. No entanto, essas propriedades não se transferem diretamente para grafos, que não possuem uma estrutura regular nem uma noção canônica de ordenação ou deslocamento.

Em grafos, a ordem em que os nós são indexados é uma escolha de representação e não uma propriedade estrutural do grafo. Dois grafos que diferem apenas nessa indexação são, do ponto de vista estrutural, idênticos, ou seja, isomorfos. A propriedade análoga à invariância à translação é, nesse contexto, a invariância à permutação dos nós. De forma mais geral, modelos de aprendizado em grafos devem satisfazer propriedades de invariância ou equivariância a permutações (BRONSTEIN *et al.*, 2021). Um modelo é invariante quando produz a mesma saída independentemente da ordem dos nós, e equivariante quando as representações de cada nó se reorganizam de acordo com a permutação aplicada à entrada.

Considera-se inicialmente um caso simplificado no qual os dados são representados apenas como um conjunto de nós, sem a presença de arestas. Cada nó i é descrito por um vetor de características $x_i \in \mathbb{R}^d$, e o conjunto de nós pode ser organizado em uma matriz de atributos

$X \in \mathbb{R}^{n \times d}$, na qual cada linha corresponde a um nó:

$$X = \begin{bmatrix} \text{——} \; x_1 \; \text{——} \\ \text{——} \; x_2 \; \text{——} \\ \text{——} \; x_3 \; \text{——} \\ \vdots \\ \text{——} \; x_n \; \text{——} \end{bmatrix}. \tag{2.40}$$

Essa representação matricial impõe uma ordenação aos nós, embora tal ordenação não reflita nenhuma propriedade estrutural do grafo.

Alterações na ordem dos nós podem ser descritas por permutações, que correspondem a rearranjos dos índices do conjunto. Por exemplo, uma permutação $(2,4,1,3)$ de (e_1, e_2, e_3, e_4) indica que o primeiro elemento do vetor resultante (r_1, r_2, r_3, r_4) corresponde ao segundo elemento da entrada, o segundo ao quarto, o terceiro ao primeiro e o quarto ao terceiro, isto é,

$$r_1 \leftarrow e_2, \quad r_2 \leftarrow e_4, \quad r_3 \leftarrow e_1, \quad r_4 \leftarrow e_3.$$

Cada permutação pode ser representada por uma matriz de permutação $P \in \{0,1\}^{n \times n}$, caracterizada por possuir exatamente um elemento igual a 1 em cada linha e em cada coluna, sendo os demais nulos. A multiplicação de P pela matriz X resulta apenas na permutação das linhas de X, sem modificar o conteúdo dos vetores de características:

$$P_{(2,4,1,3)}X = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \text{——} \; x_1 \; \text{——} \\ \text{——} \; x_2 \; \text{——} \\ \text{——} \; x_3 \; \text{——} \\ \text{——} \; x_4 \; \text{——} \end{bmatrix} = \begin{bmatrix} \text{——} \; x_2 \; \text{——} \\ \text{——} \; x_4 \; \text{——} \\ \text{——} \; x_1 \; \text{——} \\ \text{——} \; x_3 \; \text{——} \end{bmatrix}. \tag{2.41}$$

Nesse contexto, uma função f é dita invariante à permutação se satisfaz, para qualquer matriz de permutação P,

$$f(PX) = f(X). \tag{2.42}$$

Um exemplo desse tipo de função é a soma dos vetores de características dos nós, $f(X) = \sum_{i=1}^{n} x_i$, pois o resultado não depende da ordem em que os nós são somados. Essa propriedade é desejável em tarefas cujo objetivo é produzir uma saída associada ao conjunto de nós como um todo.

Por outro lado, em tarefas que requerem uma saída associada individualmente a cada nó, a invariância completa não é adequada, pois elimina a correspondência entre entrada e saída. Nesses casos, busca-se a equivariância à permutação, definida pela condição:

$$f(PX) = Pf(X), \tag{2.43}$$

na qual a aplicação de uma permutação na entrada resulta em uma permutação correspondente na saída, preservando a identidade relativa dos nós. Um exemplo desse tipo de função é a multiplicação por um escalar $f(X) = \alpha X$, pois reordenar as linhas de X e depois escalar produz o mesmo resultado que escalar e depois reordenar.

A análise anterior considerou apenas conjuntos de nós, sem relações entre eles. Ao introduzir arestas, cada par de nós conectados passa a compartilhar uma relação estrutural que também deve ser preservada sob permutações. Assim, as relações entre os nós são representadas por uma matriz de adjacência $A \in \mathbb{R}^{n \times n}$, na qual A_{ij} indica a existência de uma aresta entre os nós i e j, e funções definidas sobre grafos passam a depender tanto das características nodais quanto da conectividade, sendo expressas como $f(X,A)$. Ao permutar os nós por P, torna-se necessário reorganizar de forma correspondente tanto as linhas quanto as colunas de A, o que resulta na transformação $A \mapsto PAP^\top$. As noções de invariância e equivariância estendem-se então para:

$$f(PX, PAP^\top) = f(X,A), \tag{2.44}$$

para funções invariantes, e

$$f(PX, PAP^\top) = Pf(X,A), \tag{2.45}$$

para funções equivariantes. O tipo da saída depende da tarefa, pois funções invariantes produzem uma saída global associada ao grafo como um todo, enquanto funções equivariantes produzem uma saída por nó, de modo que permutar a entrada resulta na mesma permutação na saída.

2.7.2 Localidade

Em grafos, a noção de localidade é definida a partir das relações de vizinhança entre os nós. Para cada nó i, as interações relevantes são determinadas pelos nós pertencentes à sua vizinhança $\mathcal{N}_i$. Um multiconjunto é uma generalização da noção de conjunto que admite a repetição de elementos. A partir dessa estrutura, define-se o multiconjunto de atributos dos nós vizinhos:

$$X_{\mathcal{N}_i} = \{\{x_j \mid j \in \mathcal{N}_i\}\}, \tag{2.46}$$

o qual resume as informações locais disponíveis para o nó i.

Utiliza-se um multiconjunto, e não um conjunto simples, pois diferentes vizinhos podem possuir atributos idênticos, e a multiplicidade desses atributos carrega informação relevante sobre a estrutura local do grafo, como o número de vizinhos com determinadas características. O uso de um conjunto eliminaria essa informação, tornando indistinguíveis vizinhanças com diferentes quantidades de nós, mas com os mesmos valores distintos de atributos.

2.7.3 Formulação Geral de GNNs

Com base nas noções de invariância e equivariância a permutações apresentadas na Seção 2.7.1, é possível estabelecer um princípio geral para a construção de GNNs. A cada nó i é associada uma representação latente $h_i \in \mathbb{R}^{d'}$, que codifica suas informações a partir dos atributos nodais e da estrutura de vizinhança do grafo. O conjunto dessas representações é organizado em uma matriz $H \in \mathbb{R}^{n \times d'}$, na qual a i-ésima linha corresponde a h_i.

Uma GNN pode então ser vista como uma função equivariante $f : (X, A) \mapsto H$ que recebe como entrada a matriz de características nodais $X \in \mathbb{R}^{n \times d}$ e a matriz de adjacência $A \in \mathbb{R}^{n \times n}$, e produz H, em que d' denota a dimensão do espaço de saída, de modo que f pode ser escrita como:

$$f(X,A) = \begin{bmatrix} \text{——} \; g\left(x_1, X_{\mathcal{N}_1}\right) \; \text{——} \\ \text{——} \; g\left(x_2, X_{\mathcal{N}_2}\right) \; \text{——} \\ \vdots \\ \text{——} \; g\left(x_n, X_{\mathcal{N}_n}\right) \; \text{——} \end{bmatrix}, \tag{2.47}$$

em que g é uma função local compartilhada entre todos os nós, aplicada de forma idêntica a cada um deles, e $X_{\mathcal{N}_i}$ é o multiconjunto de atributos dos vizinhos do nó i, introduzido na Seção 2.7.2. Essa representação descreve a primeira camada da rede. De forma geral, ao longo de L camadas, g opera sobre as representações refinadas $h_i^{(l)}$ em vez dos atributos originais x_i. O compartilhamento de g é o que garante a equivariância de f, pois a mesma transformação é aplicada a cada linha de H, de modo que permutar os nós na entrada resulta na mesma permutação nas linhas da saída.

Essa função é implementada como a composição de múltiplas camadas. Denotando por $H^{(l)} \in \mathbb{R}^{n \times d_l}$ a matriz de representações na camada l, tem-se $H^{(0)} = X$ como condição inicial e $H = H^{(L)}$ como saída final após L camadas. A cada camada, a atualização da representação do

nó i pode ser escrita como:

$$h_i^{(l+1)} = g\left(h_i^{(l)}, \{\{h_j^{(l)} \mid j \in \mathcal{N}_i\}\}\right). \tag{2.48}$$

Como a vizinhança $\mathcal{N}_i$ não possui uma ordenação natural, o multiconjunto de representações dos vizinhos deve ser processado por uma função invariante à permutação. Para satisfazer essa propriedade, g pode ser decomposta na forma:

$$g\left(h_i^{(l)}, \{\{h_j^{(l)} \mid j \in \mathcal{N}_i\}\}\right) = \phi\left(h_i^{(l)}, \bigoplus_{j \in \mathcal{N}_i} \psi\left(h_j^{(l)}\right)\right), \tag{2.49}$$

em que ψ transforma individualmente cada representação de vizinho, $\bigoplus$ é um operador de agregação invariante à permutação, como soma, média ou máximo, e ϕ combina o estado atual do nó com a representação agregada da vizinhança, produzindo a nova representação $h_i^{(l+1)}$.

A decomposição de g nas etapas de geração de mensagens, agregação e atualização de estado dá origem ao mecanismo de *message passing*, descrito a seguir.

2.7.4 Mecanismo de Message Passing

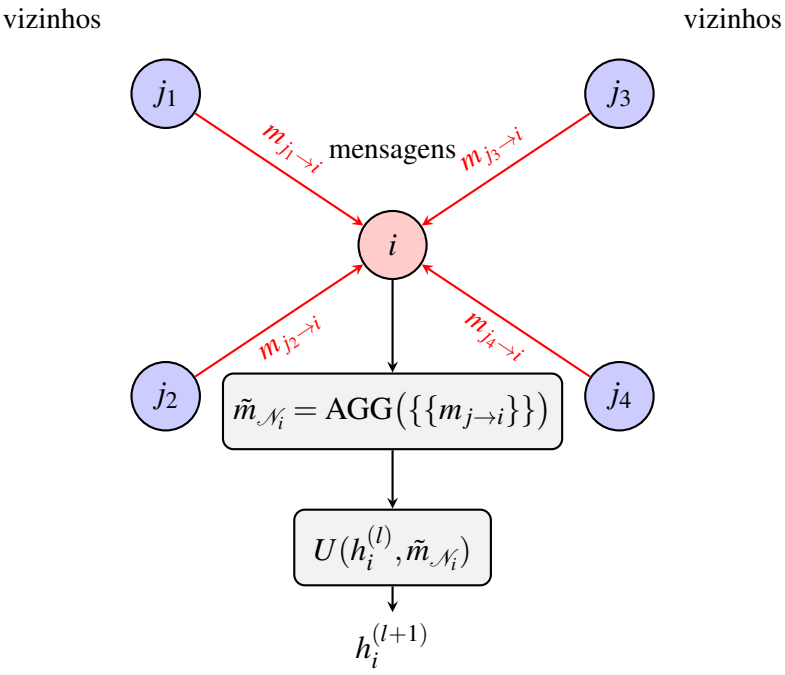
O mecanismo de *message passing* é a forma mais comum de implementar a decomposição de g apresentada na Seção anterior. Nele, cada nó refina sua representação a partir das informações provenientes de sua vizinhança por meio de três etapas: geração de mensagens, agregação e atualização de estado (GILMER *et al.*, 2017). A Figura 20 ilustra esse processo de forma esquemática.

Considere uma camada l e um nó $i \in V$, associado a uma representação latente $h_i^{(l)}$. Na etapa de geração de mensagens, cada vizinho $j \in \mathcal{N}_i$ produz uma mensagem destinada ao nó i por meio de uma função M, que corresponde a ψ na decomposição de g. Essa função pode depender das representações dos nós envolvidos e, quando disponíveis, de atributos associados às arestas e_{ji}:

$$m_{j \to i}^{(l)} = M\left(h_j^{(l)}, h_i^{(l)}, e_{ji}\right). \tag{2.50}$$

Na etapa de agregação, o nó i combina o multiconjunto de mensagens recebidas de seus vizinhos por meio de uma função AGG, que corresponde a $\bigoplus$ na decomposição de g. Para preservar a equivariância a permutações, essa função deve ser invariante à ordem dos elementos do multiconjunto. Operações como soma, média e máximo são escolhas comuns (WU *et al.*,

Figura 20 – Esquema do mecanismo de *message passing*. Cada nó i recebe mensagens $m_{j\to i}$ de seus vizinhos, agrega essas informações por meio de uma função invariante à permutação e atualiza sua representação latente.



Fonte: Própria autora.

2020):

$$\tilde{m}_{\mathcal{N}_i}^{(l)} = \mathrm{AGG}\big(\{\{\, m_{j\to i}^{(l)} \mid j \in \mathcal{N}_i \}\}\big). \tag{2.51}$$

Por fim, na etapa de atualização, a representação latente do nó i é refinada por meio de uma função U, que corresponde a ϕ na decomposição de g e combina o estado anterior $h_i^{(l)}$ com a mensagem agregada $\tilde{m}_{\mathcal{N}_i}^{(l)}$, produzindo a nova representação nodal:

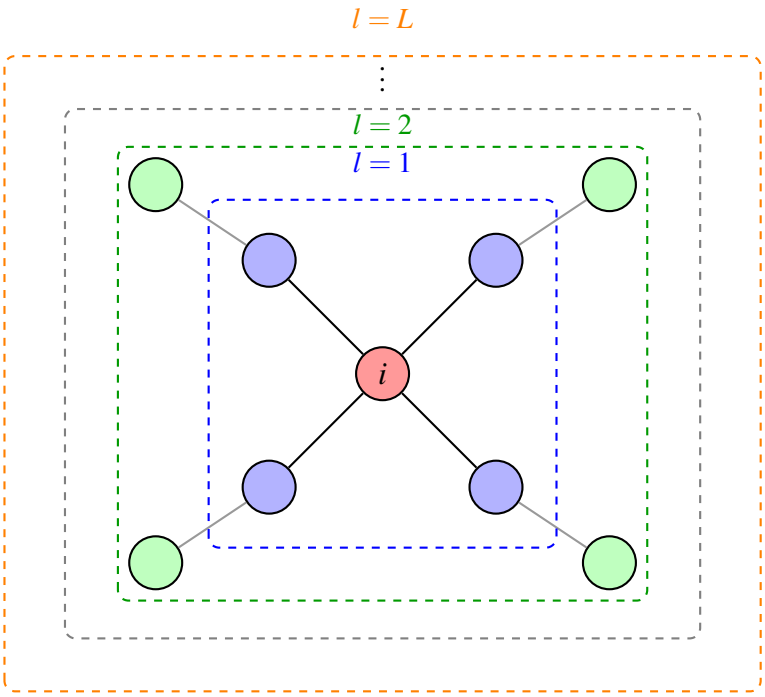
$$h_i^{(l+1)} = U\left(h_i^{(l)}, \tilde{m}_{\mathcal{N}_i}^{(l)}\right). \tag{2.52}$$

Diferentes escolhas para as funções M, AGG e U, como transformações lineares, redes neurais multicamadas ou mecanismos de atenção, resultam em arquiteturas distintas, que serão apresentadas posteriormente neste trabalho. Após a aplicação de L camadas desse processo iterativo, obtêm-se representações nodais finais $h_i^{(L)}$, que incorporam informações provenientes de vizinhanças progressivamente mais amplas no grafo, conforme ilustrado na Figura 21.

2.7.5 Tarefas de GNNs

As representações latentes $h_i^{(L)}$ produzidas ao longo das camadas podem ser utilizadas em diferentes tipos de tarefas, a depender do nível em que se deseja realizar a predição. Em

Figura 21 – Vizinhança de k saltos em GNNs. A cada camada l, a representação do nó i incorpora informações de vizinhos a uma distância máxima de l saltos no grafo. Após L camadas, a representação agrega informações de toda a vizinhança alcançável a até L saltos.



Fonte: Própria autora.

todos os casos, f_{out} denota uma função de saída parametrizada cujos detalhes dependem da tarefa e da arquitetura específica.

Em tarefas de classificação de nós, cada representação nodal final $h_i^{(L)}$ é processada individualmente por f_{out}, produzindo uma predição associada ao nó i:

$$y_i = f_{\text{out}}(h_i^{(L)}), \tag{2.53}$$

em que y_i é geralmente utilizada para inferir a classe ou o valor alvo do nó.

Em tarefas de classificação de grafos, busca-se uma representação global que sintetize as informações de todo o grafo. Para isso, aplica-se uma função de leitura global (do inglês, *readout*), invariante a permutações, que agrega as representações nodais por meio do operador $\bigoplus$, seguida por f_{out}:

$$y_G = f_{\text{out}}\left(\bigoplus_{i \in V} h_i^{(L)}\right), \tag{2.54}$$

em que y_G é a saída associada ao grafo como um todo.

Por fim, em tarefas de predição de ligações, o objetivo é estimar a existência, a probabilidade ou o tipo de relação entre pares de nós. Nesse caso, a saída do modelo é obtida a partir das representações latentes dos nós envolvidos e, quando disponíveis, de atributos

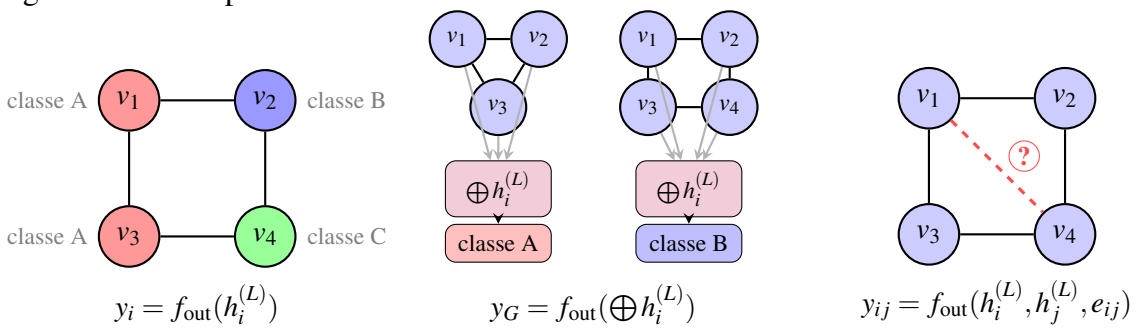
associados às arestas:

$$y_{ij} = f_{\text{out}}(h_i^{(L)}, h_j^{(L)}, e_{ij}), \tag{2.55}$$

em que y_{ij} denota a saída preditiva associada ao par de nós (i, j).

A Figura 22 ilustra as três categorias apresentadas.

Figura 22 – Principais tarefas em GNNs.



(a) Classificação de nós (b) Classificação de grafos (c) Predição de ligações

Fonte: Própria autora.

2.8 Graph Convolutional Networks

As *Graph Convolutional Networks* (GCNs) são uma das principais arquiteturas de GNNs, tendo origem na formulação espectral de convoluções em grafos. Nessa abordagem, a convolução é definida no domínio espectral por meio da Transformada de Fourier em Grafos, introduzida na Seção 2.2.3, aplicando uma transformação sobre os coeficientes espectrais do sinal e retornando ao domínio original.

De modo similar ao processamento de sinais tradicional, em que filtros podem atenuar ou amplificar determinadas frequências, um filtro em grafos corresponde a uma função aplicada aos autovalores do laplaciano. Quando esse filtro é trazido de volta ao domínio dos nós, ele se traduz em uma operação que combina as informações de cada vértice com as de seus vizinhos, produzindo uma nova representação do sinal no grafo.

A formulação original desse tipo de convolução envolve a decomposição espectral do laplaciano, que é computacionalmente custosa para grafos de grande porte. Para contornar essa limitação, Defferrard *et al.* (2016) propuseram aproximar os filtros espectrais por polinômios de Chebyshev, eliminando a necessidade de decomposição explícita. Posteriormente, Kipf e Welling (2016) simplificaram essa construção ao considerar apenas termos de primeira ordem, o

que leva a uma operação local, dependente apenas dos vizinhos imediatos de cada nó, além de introduzir uma normalização que estabiliza o treinamento.

Com essas simplificações, a propagação entre camadas em uma GCN pode ser escrita como:

$$H^{(l+1)} \;=\; \sigma\Big(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\Big), \tag{2.56}$$

em que $\tilde{A} = A + I$ é a matriz de adjacência com auto-conexões e $\tilde{D}$ é a matriz de graus associada, com $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. A inclusão da identidade garante que, a cada camada, a representação de um nó seja atualizada levando em conta não apenas seus vizinhos, mas também seu próprio estado anterior.

A matriz $H^{(l)} \in \mathbb{R}^{n \times d_l}$ reúne as representações dos nós na camada l, sendo que cada linha corresponde ao vetor de atributos de um nó. A multiplicação pela matriz de pesos treináveis $W^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ projeta essas representações para um novo espaço de dimensão d_{l+1}, enquanto o termo $\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ promove a difusão de informação ao longo das arestas do grafo.

Essa operação pode ser interpretada diretamente no domínio dos nós. Expandindo a expressão matricial, a atualização do nó i equivale a uma média ponderada das representações dos nós em sua vizinhança (incluindo ele próprio), na forma:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}_i \cup \{i\}} \frac{1}{\sqrt{\tilde{D}_{ii}\tilde{D}_{jj}}} h_j^{(l)} W^{(l)}\right). \tag{2.57}$$

O termo $\frac{1}{\sqrt{\tilde{D}_{ii}\tilde{D}_{jj}}}$ define o peso da contribuição do nó j na atualização de i, sendo determinado pelos graus dos vértices. Como consequência, esses pesos são fixos para um dado grafo e independem dos atributos dos nós, caracterizando a GCN como um modelo em que a estrutura do grafo determina como a informação é agregada.

Ao empilhar múltiplas camadas, cada nó passa a incorporar informação de vizinhanças progressivamente mais distantes. Após ℓ camadas, a representação de um nó depende de vértices que estão a até ℓ saltos de distância no grafo. Esse mecanismo permite integrar informações estruturais em diferentes escalas, mas também impõe um efeito de suavização, já que a atualização envolve médias sucessivas, de modo que as representações tendem a se tornar mais semelhantes à medida que a profundidade da rede aumenta.

2.9 Mecanismo de Atenção

Mecanismos de atenção (do inglês, *attention*) foram introduzidos originalmente no contexto de modelos sequenciais para tradução automática (BAHDANAU *et al.*, 2014). Em modelos codificador-decodificador baseados em redes recorrentes, toda a informação da sequência de entrada era comprimida em um único vetor de contexto de dimensão fixa, o que limitava a capacidade do modelo à medida que as sequências cresciam. A solução proposta foi permitir que o decodificador, a cada passo de geração, consultasse diretamente todos os estados ocultos do codificador, atribuindo a cada um deles um peso que reflete sua relevância para aquele passo específico. O vetor de contexto passou então a ser uma média ponderada dos estados ocultos, com pesos calculados dinamicamente em função da compatibilidade entre o estado atual do decodificador e cada estado do codificador.

Esse princípio foi posteriormente generalizado no mecanismo de autoatenção (do inglês, self-attention) (VASWANI *et al.*, 2017), no qual as relações de dependência são inferidas diretamente entre todos os pares de elementos do próprio conjunto de entrada. Cada elemento consulta os demais e pondera suas contribuições em função do conteúdo, sem depender de qualquer ordem ou estrutura predefinida.

Assim, a atenção pode ser interpretada como um operador que, dado um conjunto de vetores de entrada, produz para cada elemento uma combinação ponderada dos demais, cujos pesos são determinados dinamicamente pelo conteúdo dos vetores. Diferentemente de esquemas de agregação com pesos fixos, essa operação permite que o modelo selecione, de maneira adaptativa, quais elementos são mais relevantes para a construção de cada representação.

No contexto de grafos, mecanismos de atenção constituem uma extensão natural do paradigma de *message passing*. Como discutido anteriormente, uma camada de GNN atualiza a representação de cada nó i agregando informações de sua vizinhança $\mathcal{N}_i$. Nas GCNs, os pesos dessa agregação derivam exclusivamente da estrutura topológica do grafo, como a normalização pelos graus dos vértices em (2.56). Esses pesos são fixos e independem do conteúdo das representações nodais. Os mecanismos de atenção substituem esses pesos fixos por coeficientes aprendidos a partir das próprias representações, tornando a agregação sensível ao conteúdo e não apenas à topologia.

Considere um nó i com representação $h_i \in \mathbb{R}^d$ e seus vizinhos $j \in \mathcal{N}_i$. Define-se uma função de pontuação $e(h_i, h_j) \in \mathbb{R}$ que mede a compatibilidade entre as representações de i e j, indicando o quanto a informação de j é relevante para atualizar a representação de i. A

forma específica de $e(\cdot,\cdot)$ varia entre arquiteturas, mas é comum que os valores $e(h_i, h_j)$ sejam normalizados sobre a vizinhança de i por meio de uma função *softmax*:

$$\alpha_{ij} = \frac{\exp\big(e(h_i, h_j)\big)}{\sum_{k \in \mathcal{N}_i} \exp\big(e(h_i, h_k)\big)}, \tag{2.58}$$

produzindo coeficientes $\alpha_{ij} \in (0,1)$ que satisfazem:

$$\sum_{j \in \mathcal{N}_i} \alpha_{ij} = 1. \tag{2.59}$$

A função *softmax* torna os coeficientes comparáveis entre si, independentemente da magnitude dos valores de pontuação, e garante que a representação atualizada de i seja uma média convexa das contribuições dos vizinhos. Além disso, a normalização é invariante à ordem em que os vizinhos são listados, pois reordenar os termos no denominador não altera os valores dos coeficientes, preservando assim a equivariância a permutações exigida em GNNs.

Com os coeficientes α_{ij} definidos, a representação atualizada de i é obtida pela soma ponderada das representações dos vizinhos, cada uma previamente transformada por uma projeção linear aprendida:

$$h_i' = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W h_j\right), \tag{2.60}$$

em que $W \in \mathbb{R}^{d' \times d}$ é uma matriz de pesos treinável que projeta as representações de dimensão d para um novo espaço de dimensão d', e $\sigma(\cdot)$ é uma função de ativação não linear. A projeção por W é aplicada antes da agregação, de modo que o modelo aprende simultaneamente como transformar as representações e como ponderá-las.

Comparando (2.60) com (2.56), percebe-se que nas GCNs, o peso atribuído à aresta (i, j) é $(\tilde{D}_{ii}\tilde{D}_{jj})^{-1/2}$, determinado unicamente pelos graus de i e j e portanto fixo para um dado grafo. Já nos mecanismos de atenção, o coeficiente α_{ij} é recalculado a cada camada em função das representações correntes de i e j, podendo variar ao longo do treinamento e se adaptar à tarefa. Essa dependência do conteúdo permite ao modelo distinguir vizinhos mais informativos de conexões menos relevantes, o que é particularmente útil em grafos nos quais nem todas as arestas contribuem igualmente para a tarefa. Por outro lado, há um aumento no custo computacional, pois os coeficientes de atenção precisam ser calculados individualmente para cada aresta a cada passagem pela rede.

2.10 Graph Attention Networks

As *Graph Attention Networks* (GATs), propostas por Veličković *et al.* (2017), incorporam o mecanismo de atenção ao contexto de grafos ao definir coeficientes adaptativos para ponderar a contribuição de cada vizinho durante a etapa de agregação. Diferentemente das GCNs, nas quais os pesos de agregação dependem apenas da estrutura do grafo e permanecem fixos para um dado conjunto de adjacências, as GATs aprendem esses pesos a partir das próprias representações dos nós, permitindo atribuir maior importância aos vizinhos mais relevantes para cada contexto local.

Considere um nó i com representação $h_i \in \mathbb{R}^d$ e um de seus vizinhos $j \in \mathcal{N}_i$. Antes de calcular os coeficientes de atenção, as representações de ambos os nós são projetadas para um espaço de dimensão d' por uma transformação linear compartilhada $W \in \mathbb{R}^{d' \times d}$:

$$h_i \mapsto Wh_i, \quad h_j \mapsto Wh_j. \tag{2.61}$$

Com base nessas representações projetadas, calcula-se um peso de atenção entre os nós i e j dado por:

$$e_{ij} \;=\; \mathrm{LeakyReLU}\Big(a^\top \left[Wh_i \,\|\, Wh_j\right]\Big), \tag{2.62}$$

em que $a \in \mathbb{R}^{2d'}$ é um vetor de parâmetros treináveis, $\|$ denota a concatenação de Wh_i e Wh_j em um único vetor de dimensão $2d'$, e LeakyReLU é uma função de ativação que admite gradientes pequenos mas não nulos para entradas negativas, estabilizando o treinamento. Os coeficientes são normalizados sobre a vizinhança de i via *softmax*:

$$\alpha_{ij} \;=\; \frac{\exp(e_{ij})}{\displaystyle\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}, \tag{2.63}$$

com $\alpha_{ij} \in (0,1)$ e $\sum_{j \in \mathcal{N}_i} \alpha_{ij} = 1$, de modo que os coeficientes refletem a contribuição relativa de cada vizinho j para a atualização de i. Para que o próprio nó participe de sua atualização, inclui-se i em sua vizinhança por meio de autoconexões. A nova representação de i é então obtida por uma soma ponderada das representações de seus vizinhos:

$$h_i' \;=\; \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W h_j\right). \tag{2.64}$$

Sob a perspectiva do *message passing*, a mensagem transmitida de j para i corresponde à representação transformada Wh_j, enquanto a agregação consiste em uma soma

ponderada pelos coeficientes de atenção. Como essa operação independe da ordem dos vizinhos, preserva-se a invariância a permutações.

Para ampliar a capacidade representacional do modelo e reduzir a instabilidade do processo de aprendizado, as GATs empregam atenção multi-cabeça (do inglês, *multi-head attention*), na qual K mecanismos de atenção independentes são executados em paralelo, cada um com parâmetros próprios $W^{(k)}$ e $a^{(k)}$. Nas camadas intermediárias, as saídas produzidas por cada cabeça são concatenadas:

$$h_i' = \|_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(k)} W^{(k)} h_j \right), \tag{2.65}$$

resultando em uma representação de dimensão Kd'. Já na camada final, utiliza-se a média entre as cabeças para evitar crescimento excessivo da dimensionalidade:

$$h_i' = \sigma\left(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(k)} W^{(k)} h_j \right). \tag{2.66}$$

Embora ofereçam maior flexibilidade que arquiteturas baseadas em pesos fixos de agregação, as GATs apresentam maior custo computacional que as GCNs, uma vez que os coeficientes de atenção devem ser recalculados para cada aresta e para cada cabeça em toda passagem pela rede.

Além disso, assim como outras arquiteturas de GNNs, as GATs também enfrentam dificuldades na modelagem de dependências de longo alcance. Como cada camada propaga informação apenas entre vizinhos imediatos, capturar relações distantes requer o empilhamento de múltiplas camadas. Entretanto, redes muito profundas tendem a sofrer com o fenômeno de *oversmoothing*, no qual as representações nodais se tornam progressivamente mais homogêneas e menos discriminativas (LI *et al.*, 2018). Esse comportamento decorre da aplicação repetida de operações de agregação, que suavizam os sinais sobre o grafo e podem degradar o desempenho do modelo em arquiteturas mais profundas. Estudos recentes mostram que esse fenômeno também afeta redes baseadas em atenção, incluindo GATs (WU *et al.*, 2023).

Apesar dessas limitações, as GATs são particularmente adequadas para cenários em que diferentes conexões possuem relevâncias distintas. No contexto deste trabalho, essa característica é especialmente interessante, pois os grafos construídos a partir de medidas de conectividade entre eletrodos de EEG podem conter relações cuja importância para a classificação de emoções varia entre diferentes regiões cerebrais, o que motiva a adoção de GATs no modelo proposto.

2.11 Aplicações de GNNs

A capacidade das GNNs de operar sobre estruturas em grafo tem motivado sua adoção em muitas áreas. Esta Seção apresenta algumas dessas aplicações.

2.11.1 Sistemas de Informação

Em ciência da computação, as GNNs têm sido aplicadas em problemas nos quais entidades e suas relações podem ser naturalmente modeladas como grafos, como redes sociais, sistemas de recomendação e grafos de conhecimento.

Em redes sociais, os usuários correspondem aos nós do grafo, enquanto as interações entre eles, como relações de amizade ou trocas de mensagens, são modeladas como arestas. Nesse contexto, a estrutura do grafo carrega informação relevante por si só, pois nós conectados tendem a compartilhar características semelhantes, fenômeno conhecido como *homofilia*. As GNNs exploram essa propriedade ao combinar atributos individuais com informações estruturais da rede, sendo aplicadas em tarefas como predição de vínculos e classificação de usuários (WU *et al.*, 2020).

Em sistemas de recomendação, as interações entre usuários e itens formam naturalmente grafos bipartidos, nos quais dois conjuntos distintos de nós, usuários e itens, se conectam apenas entre si. Ao propagar informações ao longo dessas conexões, as GNNs conseguem capturar padrões de preferência que métodos tradicionais de filtragem colaborativa dificilmente exploram, pois estes dependem quase exclusivamente da matriz de interações sem considerar a estrutura relacional entre os elementos. O modelo PinSage ilustra a viabilidade dessa abordagem em escala industrial, tendo sido aplicado com resultados expressivos em tarefas de recomendação de conteúdo (YING *et al.*, 2018).

Outro domínio relevante é o de grafos de conhecimento, que são estruturas que organizam entidades, como pessoas, organizações ou conceitos, e as relações entre elas. Por reunirem múltiplos tipos de relações, esses grafos exigem arquiteturas capazes de distinguir o tipo de cada conexão durante a agregação. As Relational Graph Convolutional Networks (SCHLICHTKRULL *et al.*, 2018) foram propostas especificamente para esse cenário, introduzindo mecanismos de agregação sensíveis ao tipo de aresta. Essas abordagens têm sido aplicadas em tarefas como previsão de relações, inferência em grafos de conhecimento e sistemas de pergunta e resposta.

2.11.2 Ciências Biomoleculares

A biologia computacional e a química computacional estão entre os primeiros domínios nos quais as GNNs passaram a ser exploradas. Nesses contextos, moléculas podem ser naturalmente representadas como grafos, em que átomos correspondem aos nós e ligações químicas às arestas, preservando a estrutura relacional do composto.

Tradicionalmente, tarefas em química computacional baseavam-se em *molecular fingerprints*, que consistem em vetores de características construídos a partir de padrões estruturais pré-definidos, como subestruturas químicas recorrentes. Embora eficazes em diversos cenários, esses descritores dependem de regras manuais e tendem a não capturar interações mais complexas entre diferentes partes da molécula. Com GNNs, tornou-se possível aprender representações diretamente da estrutura molecular, de forma orientada pelos dados (DUVENAUD *et al.*, 2015).

Essas abordagens têm sido utilizadas na predição de propriedades moleculares, estimativa de atividade biológica e triagem virtual de compostos, que consiste na avaliação computacional de um grande número de moléculas candidatas a fármacos. Além disso, modelos baseados em grafos vêm sendo utilizados para analisar interações entre proteínas e relações entre fármacos e efeitos colaterais. Nesse sentido, o trabalho de Zitnik et al. propôs o uso de convoluções em grafos para modelar efeitos de polifarmácia, isto é, situações em que múltiplos medicamentos são utilizados simultaneamente e podem interagir entre si (ZITNIK *et al.*, 2018). Em diversos cenários, esses modelos apresentam desempenho superior a métodos baseados em descritores fixos, evidenciando sua capacidade de capturar relações estruturais não lineares (DUVENAUD *et al.*, 2015).

2.11.3 Sistemas de Transporte

Redes de transporte urbano também podem ser modeladas como grafos, nos quais nós representam interseções ou estações, e arestas correspondem às conexões entre elas. Além da estrutura espacial, esses sistemas apresentam dinâmica temporal importante, já que variáveis como fluxo e velocidade variam ao longo do tempo.

Para lidar com essas duas dimensões, foram propostas arquiteturas espaço-temporais que combinam GNNs com mecanismos de modelagem temporal. O Diffusion Convolutional Recurrent Neural Network integra convoluções baseadas em processos de difusão, que modelam como a informação se propaga ao longo do grafo, com unidades recorrentes para modelar a

evolução temporal do tráfego (LI *et al.*, 2017). Já o Spatio-Temporal Graph Convolutional Network substitui as unidades recorrentes por convoluções aplicadas diretamente ao longo do eixo temporal, obtendo desempenho competitivo em tarefas de previsão de tráfego com menor custo computacional (YU *et al.*, 2017).

A combinação entre estrutura espacial e dinâmica temporal nesses modelos reflete uma característica importante das redes de transporte de que o fluxo em um ponto da rede não depende apenas de seu histórico local, mas também do que ocorre nas regiões vizinhas, tornando a modelagem conjunta dessas duas dimensões indispensável para previsões confiáveis.

2.12 Eletroencefalogramas

O eletroencefalograma (EEG) é uma técnica que registra a atividade elétrica do cérebro por meio de eletrodos posicionados sobre o couro cabeludo, sendo utilizada tanto em pesquisas em neurociência quanto em aplicações clínicas (ZHANG *et al.*, 2023), como ilustrado na Figura 23. Cada eletrodo é um sensor metálico que capta variações de potencial elétrico na região do couro cabeludo onde está posicionado. O sinal registrado por cada eletrodo ao longo do tempo é denominado canal, de modo que o número de canais corresponde ao número de eletrodos utilizados na aquisição.

Figura 23 – Touca de EEG com eletrodos posicionados sobre o couro cabeludo.

Fonte: BCMI Lab, Shanghai Jiao Tong University (2021).

Os sinais captados refletem variações de potencial elétrico geradas predominantemente por correntes pós-sinápticas em populações de neurônios piramidais do córtex cerebral. Quando a atividade dessas populações neuronais ocorre de forma sincronizada, os campos

elétricos resultantes atingem magnitude suficiente para serem detectados externamente pelos eletrodos (NUNEZ; SRINIVASAN, 2006).

Uma das principais vantagens do EEG é o fato de ser um método não invasivo, não exigindo procedimentos cirúrgicos nem a inserção de dispositivos no interior do crânio. Isso permite a aquisição de dados de forma relativamente rápida, segura e com baixo custo, tornando o EEG particularmente adequado para estudos que exigem monitoramento prolongado ou a coleta de grandes volumes de dados.

Entre as principais técnicas utilizadas para mensurar a atividade cerebral, destacam-se também a Ressonância Magnética Funcional (do inglês, *Functional Magnetic Resonance Imaging* - fMRI) e a Eletrocorticografia (do inglês, *Electrocorticography* - ECoG). A fMRI baseia-se na medição indireta da atividade neural por meio de variações no fluxo sanguíneo cerebral associadas ao consumo de oxigênio pelas regiões ativas do cérebro (POLDRACK *et al.*, 2024). Essa técnica oferece alta resolução espacial, da ordem de milímetros, permitindo identificar regiões cerebrais ativadas durante tarefas cognitivas ou sensoriais. No entanto, sua resolução temporal é limitada, geralmente na escala de segundos, além de exigir equipamentos de alto custo e ambientes experimentais altamente controlados (LOGOTHETIS, 2008).

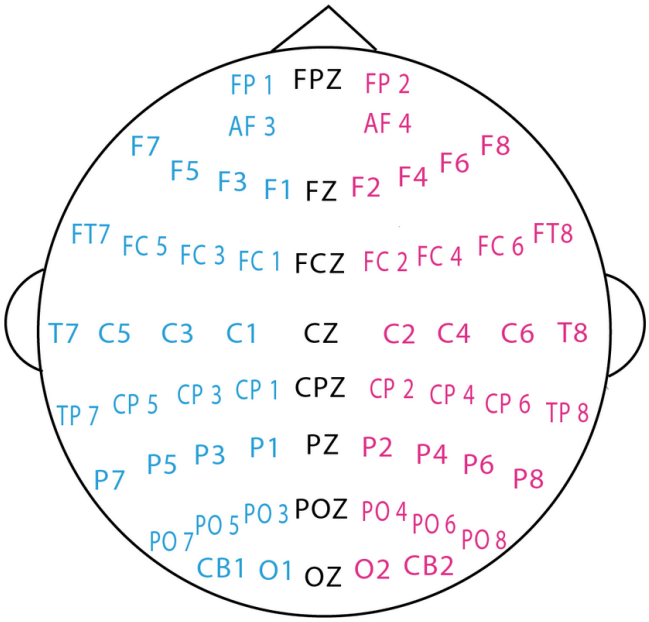
Por outro lado, a ECoG registra diretamente a atividade elétrica cortical por meio de eletrodos implantados sobre a superfície do córtex cerebral, posicionados abaixo do crânio, o que resulta em sinais com elevada resolução temporal e melhor resolução espacial em comparação ao EEG, pois sofrem menor atenuação ao atravessar os tecidos cranianos (VOLKOVA *et al.*, 2019). Entretanto, trata-se de um método invasivo, geralmente utilizado apenas em contextos clínicos específicos, como no planejamento cirúrgico para tratamento de epilepsia.

O EEG, por sua vez, apresenta um equilíbrio entre essas características. Sua resolução temporal na ordem de milissegundos permite observar diretamente a dinâmica rápida das oscilações neurais e dos processos de comunicação entre regiões cerebrais. No entanto, sua resolução espacial é relativamente limitada devido ao fenômeno de condução de volume, no qual os potenciais elétricos se difundem através dos tecidos cranianos antes de serem detectados pelos eletrodos (NUNEZ; SRINIVASAN, 2006).

A aquisição de sinais de EEG depende da correta disposição dos eletrodos sobre o couro cabeludo. Para garantir reprodutibilidade e comparabilidade entre experimentos, são utilizados sistemas padronizados de posicionamento. Um dos mais usados é o sistema internacional 10-20, que define posições de eletrodos com base em proporções relativas entre marcos

anatômicos da cabeça, como o násio, o ínio e os pontos pré-auriculares. Nesse sistema, os eletrodos são distribuídos de modo a cobrir regiões frontais (F), centrais (C), temporais (T), parietais (P) e occipitais (O) do cérebro. Nesse sistema, os eletrodos são distribuídos de modo a cobrir regiões frontais (F), centrais (C), temporais (T), parietais (P) e occipitais (O) do cérebro, conforme ilustrado na Figura 24.

Figura 24 – Distribuição dos eletrodos do sistema internacional 10-20 utilizado no conjunto de dados SEED. Eletrodos do hemisfério esquerdo em azul, direito em rosa e linha central em preto.



Fonte: BCMI Lab, Shanghai Jiao Tong University (2019).

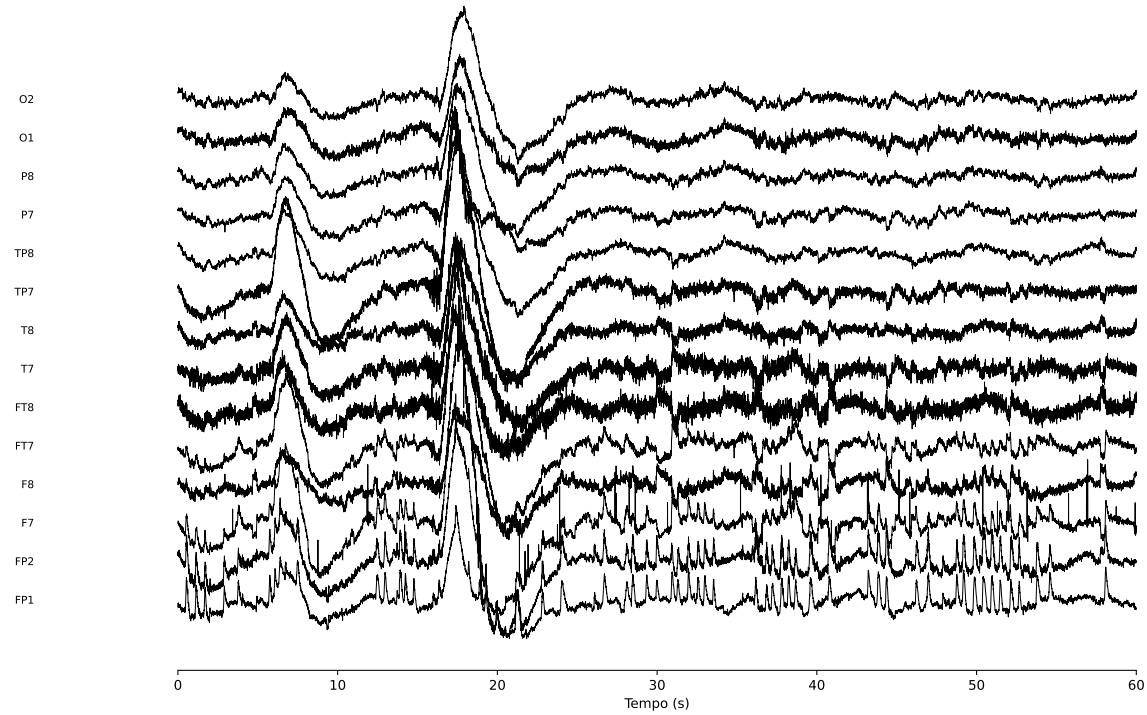
Para aplicações que exigem maior resolução espacial, foram desenvolvidas extensões desse sistema, como o sistema 10-10 e o sistema 10-5. Essas variações aumentaram a densidade de eletrodos disponível, permitindo aquisições com dezenas ou centenas de canais e, consequentemente, maior detalhamento espacial (JURCAK *et al.*, 2007).

Os sinais de EEG são frequentemente analisados no domínio da frequência, uma vez que a atividade neural apresenta padrões oscilatórios característicos. Essas oscilações são tradicionalmente agrupadas em bandas de frequência associadas a diferentes estados fisiológicos e cognitivos. Entre as principais bandas destacam-se: delta (0,5-4 Hz), frequentemente associada ao sono profundo, mas também observada durante tarefas cognitivas que demandam concentração; theta (4-8 Hz), associada a processos de memória e aprendizagem; alfa (8-13 Hz), geralmente observada em estados de relaxamento e repouso com olhos fechados; beta (13-30 Hz), associada à atenção, atividade motora e processamento cognitivo ativo; e gama (acima de 30 Hz), frequentemente vinculada a processos de integração sensorial e funções cognitivas de

alto nível (CHADDAD *et al.*, 2023).

A Figura 25 ilustra um exemplo de traçado de EEG multicanal obtido a partir do conjunto de dados SEED, mostrando a atividade elétrica registrada simultaneamente em diferentes regiões do couro cabeludo ao longo do tempo.

Figura 25 – Traçado de EEG multicanal obtido a partir do conjunto de dados SEED, exibindo os sinais de 14 canais distribuídos simetricamente entre os hemisférios esquerdo e direito.



Fonte: Elaborada pela autora.

No contexto clínico, o EEG é utilizado no diagnóstico e monitoramento de epilepsia, na avaliação de distúrbios do sono e na investigação de alterações cognitivas associadas a doenças neurológicas (GKINTONI *et al.*, 2025). Além disso, estudos recentes têm explorado o uso de EEG na identificação de biomarcadores para transtornos psiquiátricos, como depressão, esquizofrenia e transtorno bipolar (YUN, 2024), bem como em doenças neurodegenerativas, incluindo Alzheimer e Parkinson (AVILES *et al.*, 2024).

No campo da engenharia biomédica, o EEG também é utilizado no desenvolvimento de sistemas de Interface Cérebro-Computador (do inglês, *Brain-Computer Interface* - BCI). Esses sistemas permitem a comunicação direta entre o cérebro e dispositivos externos, possibilitando aplicações como controle de próteses robóticas, cadeiras de rodas motorizadas e sistemas de comunicação assistiva para pacientes com graves limitações motoras (CHEN *et al.*, 2025).

2.13 Entropia Diferencial

A entropia é uma medida originária da teoria da informação, utilizada para quantificar o grau de incerteza ou variabilidade associado a uma variável aleatória. Para variáveis aleatórias discretas, a entropia de Shannon mede a quantidade média de informação produzida por um processo estocástico. No entanto, em muitos contextos, incluindo a análise de sinais biológicos como o EEG, as variáveis de interesse assumem valores contínuos. Nesses casos, utiliza-se a chamada entropia diferencial, que estende esse conceito para o domínio contínuo.

Considere uma variável aleatória contínua X com função densidade de probabilidade $f(x)$. A entropia diferencial de X é definida como:

$$h(X) = -\int f(x) \log f(x)\, dx, \tag{2.67}$$

assumindo que a integral converge. Essa quantidade reflete o grau de dispersão ou imprevisibilidade associado à distribuição de probabilidade da variável. Distribuições mais concentradas tendem a apresentar menor entropia diferencial, enquanto distribuições mais espalhadas geralmente apresentam valores maiores de entropia.

Diferentemente da entropia discreta, a entropia diferencial pode assumir valores negativos e não é invariante sob transformações contínuas de escala. Apesar disso, ela mantém um papel conceitual semelhante ao da entropia de Shannon ao caracterizar a quantidade de incerteza associada a uma variável contínua.

Um exemplo importante ocorre quando X segue uma distribuição normal $\mathcal{N}(\mu, \sigma^2)$. Nesse caso, a entropia diferencial assume a forma:

$$h(X) = \frac{1}{2} \log \left(2\pi e \sigma^2\right). \tag{2.68}$$

Essa expressão mostra que a entropia diferencial depende apenas da variância da distribuição, o que reflete o fato de que distribuições gaussianas com maior dispersão concentram menos informação e apresentam maior incerteza.

Na análise de sinais de EEG, a entropia diferencial tem sido utilizada como medida de complexidade e variabilidade estatística, permitindo identificar mudanças nos padrões de oscilação neural associadas a diferentes estados cognitivos ou fisiológicos. Na prática, como os sinais são amostrados e segmentados em janelas temporais de duração finita, a integral contínua não pode ser calculada diretamente. Uma abordagem comum consiste em aproximar a distribuição de probabilidade por um histograma normalizado e calcular uma versão discreta da entropia diferencial, conforme descrito na Seção 4.4.

3 TRABALHOS RELACIONADOS

O reconhecimento de emoções a partir de sinais de EEG tem sido bastante investigado nos últimos anos, impulsionado pelo avanço de interfaces cérebro-computador e pelo desenvolvimento de técnicas de ML. Entre os conjuntos de dados disponíveis, o SEED destaca-se como uma das principais referências, sendo frequentemente utilizado na classificação de três estados emocionais (positivo, neutro e negativo) (ZHENG; LU, 2015).

Nos estudos dessa área, dois cenários experimentais são predominantemente considerados: *Subject-Dependent* (SD) e *Subject-Independent* (SI). No cenário SD, os modelos são treinados e avaliados com dados do mesmo indivíduo, geralmente por meio de divisões entre sessões distintas. Esse protocolo tende a produzir acurácias mais altas, pois o modelo captura padrões específicos de cada indivíduo sem precisar generalizar para outros. Já no cenário SI, o modelo é avaliado em indivíduos não vistos durante o treinamento, exigindo que capture padrões emocionais comuns entre pessoas distintas. Esse cenário é mais próximo de uma aplicação real, mas consideravelmente mais difícil. Um dos protocolos mais importantes nesse contexto é o Leave-One-Subject-Out (LOSO), no qual, a cada iteração, os dados de um indivíduo são reservados exclusivamente para teste enquanto os dos demais são utilizados para treinamento, repetindo esse procedimento para todos os indivíduos.

Os primeiros trabalhos na área baseavam-se em métodos tradicionais de aprendizado de máquina com extração manual de características. Zheng e Lu (2015) compararam diferentes representações de características, como entropia diferencial e PSD, avaliando seu poder discriminativo para o reconhecimento de emoções. Os resultados apresentaram superioridade das características baseadas em entropia diferencial, com acurácia de 86,08% no cenário SD utilizando todos os 62 canais. Adicionalmente, os autores mostraram que a seleção de subconjuntos reduzidos de eletrodos pode manter ou melhorar o desempenho, com configurações de 4, 6, 9 e 12 canais atingindo respectivamente 82,88%, 85,03%, 84,02% e 86,65%, sendo que esta última superou o modelo treinado com o conjunto completo.

Em uma linha semelhante, Li *et al.* (2018) exploraram a assimetria hemisférica cerebral, ou seja, as diferenças na atividade elétrica entre os hemisférios esquerdo e direito, como característica discriminativa, alcançando 83,28% em cenário SI. Dewangan *et al.* (2023) utilizaram 32 características estatísticas combinadas com classificadores SVM no SEED-IV, uma extensão do SEED com quatro classes emocionais, obtendo 95,73% em SD e 83,7% em SI com um classificador de kernel gaussiano. Esses resultados evidenciam a queda de desempenho ao

considerar a generalização entre indivíduos, um desafio recorrente na área.

Com o avanço do DL, arquiteturas baseadas em CNNs passaram a ser bastante adotadas, permitindo a extração automática de padrões espaciais e temporais dos sinais EEG. Wang *et al.* (2022) propuseram uma CNN aplicada diretamente aos dados brutos sem extração manual de características, com kernels distintos para as dimensões espacial e temporal do sinal, alcançando 86,10% em SD com validação cruzada de 10 partições e divisão 90/10 entre treino e teste. Pusarla *et al.* (2022), por sua vez, utilizaram espectrogramas como representação de entrada para uma arquitetura baseada em DenseNet, alcançando 97,91% em LOSO.

Abordagens mais recentes incorporam mecanismos de atenção e Transformers, arquiteturas originalmente propostas para o processamento de linguagem natural que se destacam pela capacidade de modelar dependências de longo alcance em sequências. Wei e Zhou (2024) propuseram um modelo com dois encoders Transformer paralelos, um voltado para dependências temporais dentro de cada canal e outro para relações espaciais entre os 62 canais de EEG, obtendo 95,73% em SD e 87,38% em um cenário com mistura de indivíduos no conjunto de dados SEED.

As GNNs têm se destacado por sua capacidade de modelar diretamente as relações entre canais. Zhong *et al.* (2020) propuseram a *Regularized Graph Neural Network* (RGNN), que constrói a matriz de adjacência do grafo com base na topologia biológica conhecida entre regiões cerebrais e incorpora regularizadores para lidar com a variabilidade entre indivíduos e com rótulos ruidosos durante o treinamento, alcançando 85,30% em LOSO no SEED.

Li *et al.* (2021) introduziram a *Self-Organized Graph Neural Network* (SOGNN), na qual a estrutura do grafo é aprendida durante o treinamento a partir dos próprios dados, em vez de ser definida previamente com base na posição dos eletrodos ou em critérios neuroanatômicos, atingindo 86,81% no SEED e 75,27% no SEED-IV, ambos em LOSO. Kong *et al.* (2022) propuseram a *Causal Graph Convolutional Neural Network* (CGCNN), que constrói uma matriz de adjacência assimétrica por meio da causalidade de Granger, uma medida estatística que quantifica o quanto a série temporal de um canal ajuda a prever a de outro, capturando assim relações direcionais entre eletrodos e alcançando 93,36% no SEED e 75,48% no SEED-IV em SD.

Uma linha promissora dentro das abordagens baseadas em grafos consiste em estimar fontes neurais e utilizá-las como nós, em vez dos eletrodos diretamente. Essa estratégia contorna o problema da condução de volume, no qual o sinal de uma fonte neural se espalha por múltiplos eletrodos, introduzindo correlações artificiais entre canais adjacentes. Asadzadeh *et al.* (2022)

propuseram estimar fontes corticais por meio de um modelo Bayesiano, inicializado com o sLORETA, uma técnica que estima a distribuição de atividade neural no córtex a partir dos potenciais medidos no couro cabeludo, e utilizá-las como nós de uma GNN com matriz de adjacência atualizada dinamicamente, alcançando 98,51% em SI e 99,25% em SD. Asadzadeh *et al.* (2023) adotaram o sLORETA diretamente para estimar as fontes corticais e utilizá-las como nós de uma GNN, alcançando 97,50% em SI e 98,75% em SD. Ambos os trabalhos, no entanto, avaliam apenas classificação binária, o que limita a comparabilidade com abordagens de três classes emocionais.

A análise dos trabalhos apresentados permite identificar algumas lacunas relevantes. Grande parte dos modelos avaliados em cenário SI com protocolo LOSO utiliza todos os 62 canais disponíveis no SEED e as cinco bandas de frequência convencionais, resultando em configurações de alta dimensionalidade. Trabalhos que exploram redução de canais, como Zheng e Lu (2015), fazem isso no cenário SD, sem avaliar o impacto dessa redução na generalização entre indivíduos. Abordagens que alcançam alto desempenho em SI com LOSO, como Ning *et al.* (2021), utilizam amostras rotuladas do indivíduo de teste durante o treinamento, caracterizando um cenário *few-shot* que difere do cenário estritamente *zero-shot* avaliado neste trabalho. Adicionalmente, poucos trabalhos investigam sistematicamente a contribuição de diferentes combinações de canais e bandas na tarefa de classificação entre indivíduos.

Diante dessas lacunas, este trabalho propõe um modelo baseado em GATs avaliado em cenário SI estritamente *zero-shot* com protocolo LOSO, sem acesso a dados do indivíduo de teste durante o treinamento. A principal contribuição é demonstrar que uma configuração compacta de apenas 4 canais fronto-temporais e 2 bandas de frequência (delta e theta), identificada por busca sistemática entre pares simétricos de canais, é suficiente para alcançar desempenho competitivo, superando trabalhos que utilizam a configuração completa de 62 canais e 5 bandas sob as mesmas condições de avaliação. Essa configuração reduzida aproxima o modelo de cenários práticos com dispositivos vestíveis de baixo custo e número limitado de eletrodos.

4 METODOLOGIA

Este Capítulo descreve os procedimentos adotados no desenvolvimento deste trabalho. Inicialmente, são apresentadas as configurações do ambiente experimental. Em seguida, é descrito o *dataset* utilizado nos experimentos. Posteriormente, são apresentados os processos de pré-processamento, extração de características e construção dos grafos, incluindo as estratégias empregadas para definição das representações dos nós e das arestas. Por fim, é descrita a arquitetura do modelo proposto, baseado em GAT, assim como as configurações adotadas para o treinamento e avaliação do modelo.

4.1 Configuração Experimental

Os experimentos foram implementados em linguagem *Python*, utilizando o ambiente interativo *Jupyter Notebook*. O modelo foi desenvolvido com o framework *PyTorch* (PASZKE *et al.*, 2019), e as camadas GAT e operações de *readout* foram implementadas com o auxílio da biblioteca *PyTorch Geometric* (FEY; LENSSEN, 2019), uma extensão do *PyTorch* voltada para aprendizado em grafos. Os experimentos foram executados em um computador com processador AMD Ryzen 5 5600X e GPU NVIDIA GeForce RTX 3060 com 12 GB de memória dedicada.

4.2 Base de Dados

Os experimentos foram conduzidos utilizando o *dataset* SEED (*SJTU Emotion EEG Dataset*), bastante empregado em tarefas de reconhecimento de emoções a partir de sinais de EEG (ZHENG; LU, 2015). O *dataset* é composto por registros de EEG obtidos de 15 participantes, cada um realizando três sessões experimentais em dias distintos, nas quais assistiu a 15 estímulos audiovisuais por sessão. Esses estímulos foram selecionados para induzir três classes emocionais distintas: positiva, neutra e negativa.

Os sinais foram originalmente registrados com uma taxa de amostragem de 1 000 Hz, utilizando 62 eletrodos posicionados de acordo com o sistema internacional 10-20. Para este trabalho, os sinais disponibilizados já foram reamostrados para 200 Hz e filtrados na faixa de 0,5 a 75 Hz pelos autores do *dataset*, removendo artefatos de alta frequência. Todos os sinais foram padronizados para uma duração fixa de 4 minutos, de modo que cada canal passou a conter exatamente 48 000 pontos. No total, foram utilizadas 675 exames de EEG, correspondentes às 45 instâncias de cada um dos 15 indivíduos.

4.3 Pré-processamento

Os sinais de EEG disponibilizados no SEED já foram submetidos a etapas iniciais de pré-processamento pelos autores do conjunto de dados, incluindo reamostragem de 1000 Hz para 200 Hz e filtragem passa-banda entre 0,5 e 75 Hz (ZHENG; LU, 2015). A partir desses sinais, foram realizadas as seguintes etapas adicionais.

Os sinais foram segmentados em janelas temporais de 1 segundo sem sobreposição, resultando em 240 janelas por amostra. Em seguida, cada janela foi decomposta nas cinco bandas de frequência clássicas do EEG: delta (0,5-4 Hz), theta (4-8 Hz), alpha (8-13 Hz), beta (13-30 Hz) e gamma (30-45 Hz). Para isso, foi aplicado um filtro passa-banda do tipo *Butterworth* de quarta ordem com filtragem de fase zero.

4.4 Extração de Características

A etapa de extração de características tem como objetivo transformar os segmentos de EEG pré-processados em representações vetoriais adequadas para a tarefa de classificação. Para cada canal, janela temporal e banda de frequência, foi calculada a entropia diferencial. A entropia diferencial foi adotada como característica por ter demonstrado superioridade em relação a outras representações espectrais, como a PSD, em tarefas de reconhecimento de emoções com EEG (ZHENG; LU, 2015).

Diferentemente de abordagens que assumem distribuição gaussiana do sinal, neste trabalho a entropia diferencial foi estimada de forma não paramétrica por meio de histogramas. Essa escolha é motivada pelo fato de que sinais de EEG podem apresentar distribuições não gaussianas, especialmente após filtragem passa-banda em bandas estreitas, e a estimação por histograma não impõe restrições sobre a forma da distribuição subjacente. A distribuição de probabilidade do sinal em cada janela foi aproximada por um histograma normalizado com 30 intervalos, e a estimativa da entropia diferencial calculada como:

$$\hat{h}(X) = -\sum_{i} p(x_i) \log p(x_i) + \log(\Delta x), \tag{4.1}$$

em que $p(x_i)$ é a probabilidade associada ao i-ésimo intervalo do histograma, Δx é a largura dos intervalos e o termo $\log(\Delta x)$ corrige a discretização da integral contínua.

Ao final dessa etapa, cada canal é representado por um vetor de dimensão $W \times B$, em que $W = 240$ é o número de janelas temporais e B é o número de bandas selecionadas.

4.5 Construção dos Grafos

Após a extração das características, cada amostra de EEG é representada como um grafo $G = (V, E, X)$, em que os nós correspondem aos canais selecionados, as arestas codificam relações de similaridade entre eles, e X é a matriz de atributos nodais. Essa representação em grafo constitui a entrada do modelo.

Cada nó i é descrito por um vetor de atributos $x_i \in \mathbb{R}^{W \cdot B}$, obtido pela concatenação dos valores de entropia diferencial calculados para aquele canal ao longo das $W = 240$ janelas temporais e das B bandas de frequência consideradas. Por exemplo, com $B = 2$ bandas (delta e theta), cada nó é representado por um vetor de dimensão $240 \times 2 = 480$. O conjunto de vetores de todos os nós forma a matriz de atributos $X \in \mathbb{R}^{C \times (W \cdot B)}$, em que C é o número de canais selecionados.

Antes de atribuir as características aos nós, foi realizada uma normalização *z-score* independente por canal, subtraindo-se a média e dividindo-se pelo desvio padrão. Essa etapa tem como objetivo garantir que cada canal contribua de forma equilibrada durante o processamento pelo modelo, independentemente da magnitude absoluta dos valores de entropia diferencial.

Os pesos das arestas foram definidos pela similaridade de cosseno entre os vetores de características originais, antes da normalização:

$$s_{ij} = \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\| \, \|\mathbf{x}_j\|} \tag{4.2}$$

A matriz de adjacência foi construída de forma adaptativa para cada amostra, retendo apenas as arestas cujo peso supera o percentil 70 da distribuição de similaridades, sendo as autoconexões removidas. O limiar do percentil 70 foi determinado empiricamente por meio de experimentos preliminares, avaliando diferentes valores. Esse percentil produziu o melhor equilíbrio entre conectividade do grafo e desempenho de classificação, retendo as conexões mais relevantes sem tornar o grafo excessivamente esparso ou denso.

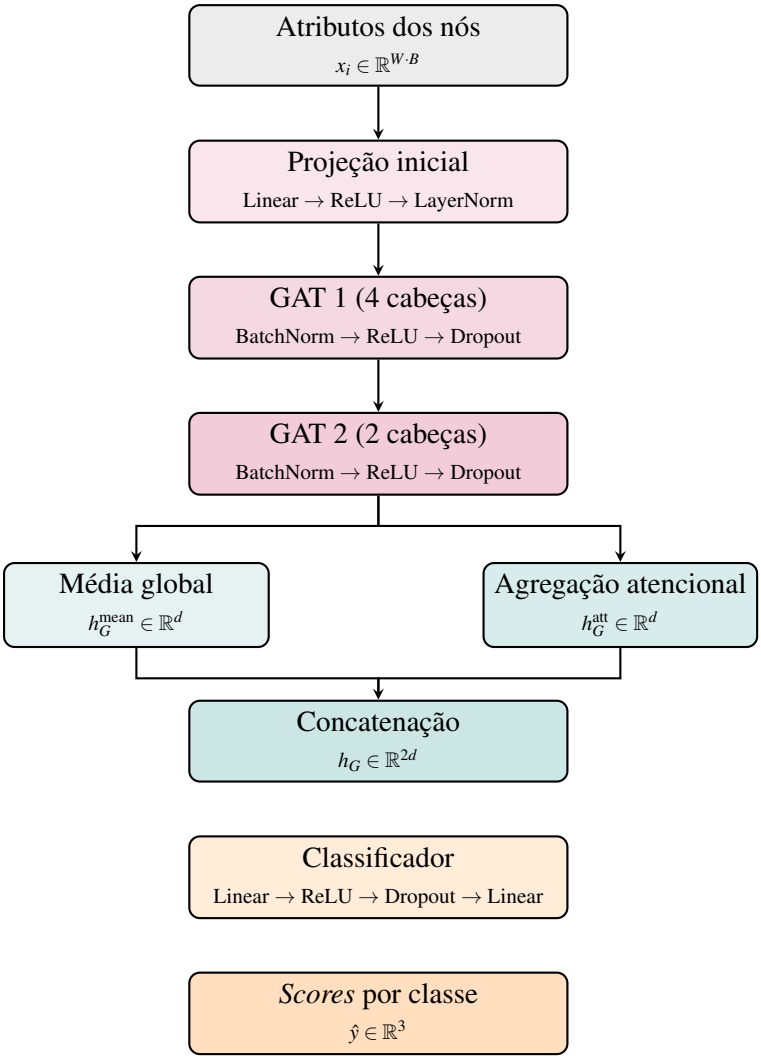
4.6 Arquitetura do Modelo

A escolha de uma arquitetura baseada em GATs é motivada por três razões principais. Primeira, grafos são a representação natural para modelar relações funcionais entre canais de EEG, capturando dependências que arquiteturas euclidianas como CNNs e RNNs não conseguem representar diretamente. Segunda, o mecanismo de atenção da GAT permite que o modelo aprenda adaptativamente quais canais são mais relevantes para cada estado emocional, em vez

de agregar todos os vizinhos com pesos fixos como nas GCNs. Terceira, por não depender da estrutura global do grafo para calcular os coeficientes de atenção, o GAT se aplica naturalmente ao cenário sujeito-independente, no qual os grafos de teste não foram vistos durante o treinamento.

O modelo proposto é composto por três estágios sequenciais: uma camada de projeção inicial, duas camadas de atenção em grafos e um módulo de leitura global seguido de uma camada de classificação. Ele está esquematizado na Figura 26.

Figura 26 – Arquitetura do modelo proposto.



Fonte: Própria autora.

Na primeira etapa, os atributos de cada canal são projetados para um espaço de dimensão oculta por uma camada linear, seguida de ativação ReLU e *Layer Normalization*. Em seguida, duas camadas GAT atualizam as representações por meio de agregação ponderada por coeficientes de atenção aprendidos. Cada coeficiente α_{ij} quantifica a relevância do canal j para a atualização do canal i, sendo calculado a partir das representações de ambos e normalizado via

softmax sobre a vizinhança de i, que inclui o próprio canal, garantindo que sua representação anterior seja preservada.

A primeira camada utiliza quatro cabeças de atenção e a segunda utiliza duas, ambas com agregação por média, de modo que a dimensão de saída permanece igual à dimensão oculta independentemente do número de cabeças. O uso de múltiplas cabeças permite capturar simultaneamente diferentes padrões de relação entre os canais, de forma análoga a múltiplos filtros em redes convolucionais. Após cada camada, são aplicadas *Batch Normalization*, seguida de ativação ReLU e *dropout*, que desativa aleatoriamente uma fração das ativações durante o treinamento, reduzindo o risco de *overfitting*.

As representações dos canais são então agregadas em uma única representação do grafo por duas estratégias de *readout* aplicadas em paralelo: média global, que pondera todos os canais igualmente, e agregação atencional, na qual cada canal recebe um peso aprendido por uma rede com duas camadas lineares e ativação ReLU. Enquanto a média oferece uma visão global e uniforme, a agregação atencional aprende quais canais são mais relevantes para a classificação. As duas representações são concatenadas, resultando em um vetor de dimensão $2 \times d_{\text{oculta}}$. Por fim, esse vetor é processado por duas camadas lineares com ativação ReLU e *dropout* entre elas. A camada final produz uma pontuação por classe emocional.

4.7 Treinamento e Avaliação

O treinamento foi conduzido sob o protocolo LOSO, descrito anteriormente, no qual um indivíduo do conjunto de treinamento é designado como conjunto de validação, garantindo que os dados de teste permaneçam completamente independentes ao longo de todo o processo.

Antes do treinamento, é realizada uma etapa adicional de normalização sobre os atributos dos nós, distinta das normalizações aplicadas nas etapas anteriores. Enquanto a normalização *z-score* descrita na Seção 4.5 opera por canal dentro de cada amostra individualmente, esta etapa calcula a média e o desvio padrão de cada atributo sobre todos os grafos do conjunto de treinamento e aplica esses valores aos conjuntos de treino, validação e teste. Ao utilizar exclusivamente estatísticas do conjunto de treinamento para normalizar todos os conjuntos, garante-se que nenhuma informação dos conjuntos de validação ou teste influencie o treinamento, evitando vazamento de informação.

O modelo é otimizado com o AdamW (LOSHCHILOV; HUTTER, 2017), utilizando entropia cruzada com suavização de rótulos (do inglês, *label smoothing*), que distribui uma pe-

quena fração da probabilidade entre as classes incorretas, penalizando predições excessivamente confiantes e favorecendo a generalização. Durante o cálculo dos gradientes, é aplicado *gradient clipping* (PASCANU *et al.*, 2013), técnica em que a norma euclidiana do vetor de gradientes é monitorada a cada atualização e, quando ultrapassa um limiar predefinido, os gradientes são reescalados proporcionalmente. Esse procedimento previne atualizações de grande magnitude, o que é relevante em arquiteturas com mecanismos de atenção, nos quais os coeficientes α_{ij} podem gerar gradientes elevados nas primeiras épocas.

A taxa de aprendizado é reduzida dinamicamente quando não há melhora no conjunto de validação por um determinado número de épocas. O treinamento é interrompido antecipadamente (*early stopping*) quando nenhuma melhora é observada por um número de épocas igual ao parâmetro de paciência, evitando sobreajuste. Ao final, o modelo com melhor desempenho na validação é avaliado no conjunto de teste correspondente.

A métrica adotada é a acurácia de classificação, definida como a proporção entre o número de amostras corretamente classificadas e o total avaliado. Para obter estimativas mais confiáveis, o protocolo LOSO completo foi repetido 10 vezes e os resultados são reportados com base na média dessas repetições.

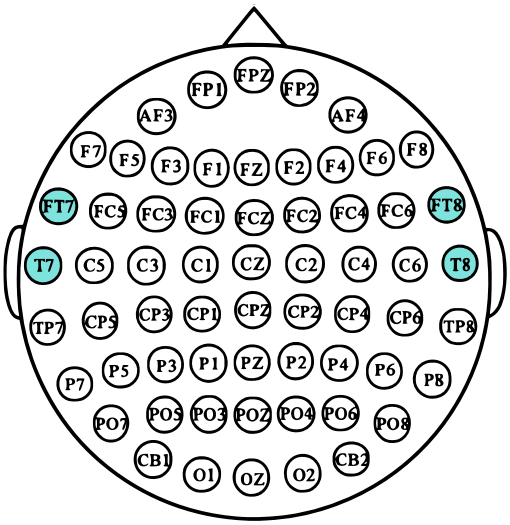
4.8 Seleção de Canais

A seleção de canais foi realizada com o objetivo de reduzir a dimensionalidade do problema preservando as regiões cerebrais mais relevantes para o reconhecimento de emoções. Foi adotada uma abordagem orientada por desempenho, na qual pares simétricos de eletrodos foram avaliados em combinação com diferentes bandas de frequência sob o protocolo LOSO. Os resultados dessa busca, apresentados na Seção 5.1, indicaram que a configuração com maior acurácia foi obtida com dois pares de canais localizados nas regiões frontotemporal e temporal central: FT7, FT8, T7 e T8, ilustrados na Figura 27.

A inclusão de pares adicionais não resultou em melhora de desempenho, indicando a presença de redundância informacional entre os demais canais. Esse resultado sugere que um subconjunto reduzido de eletrodos é suficiente para capturar padrões discriminativos relevantes para a tarefa, em consonância com achados da literatura (ZHENG; LU, 2015).

A escolha de pares simétricos entre os hemisférios esquerdo e direito também permite capturar possíveis padrões de assimetria hemisférica, frequentemente explorados em estudos de reconhecimento de emoções (LI *et al.*, 2018). No total, a configuração adotada utiliza apenas

Figura 27 – Distribuição dos 62 eletrodos do sistema 10-20 utilizado no conjunto de dados SEED. Os eletrodos destacados correspondem aos quatro canais selecionados: FT7, FT8, T7 e T8, localizados nas regiões frontotemporal e temporal central do escalpo.



Fonte: Adaptada de Zhdanov *et al.* (2022).

quatro canais, o que reduz o custo computacional do modelo e favorece a viabilidade de aplicação em dispositivos de aquisição com número reduzido de eletrodos.

5 RESULTADOS

Este capítulo apresenta os resultados obtidos com o modelo proposto para a tarefa de reconhecimento de emoções a partir de sinais de EEG. O código-fonte da implementação está disponível em: repositório público no GitHub.

5.1 Análise de Canais e Bandas

A Figura 28 apresenta a acurácia LOSO média obtida para cada combinação de par de canais simétricos e banda de frequência. Cada combinação foi avaliada em 5 execuções independentes do protocolo LOSO, sendo a média dos resultados utilizada como estimativa de desempenho.

Em relação às bandas de frequência, delta e theta destacam-se como as mais informativas para o reconhecimento de emoções neste contexto. A banda delta concentra os maiores valores de acurácia para a maioria dos pares de canais, enquanto a banda theta apresenta desempenho igualmente elevado. As bandas alfa, beta e gama, por sua vez, tendem a produzir acurácias inferiores independentemente do par de canais considerado, sugerindo que as oscilações de baixa frequência carregam informação emocional mais discriminativa neste conjunto de dados. Essa observação é consistente com estudos que associam as bandas delta e theta a processos de regulação emocional e memória de trabalho (NIEDERMEYER; SILVA, 2005).

Em relação aos pares de canais, os maiores valores de acurácia concentram-se nas regiões frontal inferior e temporal central, com destaque para FT7/FT8, T7/T8 e C5/C6, que atingem acurácias médias entre 0,87 e 0,92 nas bandas delta e theta. Essas regiões correspondem ao córtex frontal inferior e às áreas temporais, reconhecidos na literatura de neurociência afetiva como regiões centrais no processamento emocional (ZHENG; LU, 2015). Em contraste, pares de canais nas regiões centrais e parietais, como CP1/CP2 e C1/C2, apresentam desempenho inferior independentemente da banda de frequência considerada.

Com base nessa análise, os pares FT7/FT8 e T7/T8 em conjunto com as bandas delta e theta foram selecionados como configuração de entrada do modelo proposto. Essa combinação apresenta alto poder discriminativo e reduz significativamente a dimensionalidade do problema em relação ao uso dos 62 canais convencionais, tornando a abordagem mais adequada para aplicações práticas com dispositivos de aquisição de poucos canais.

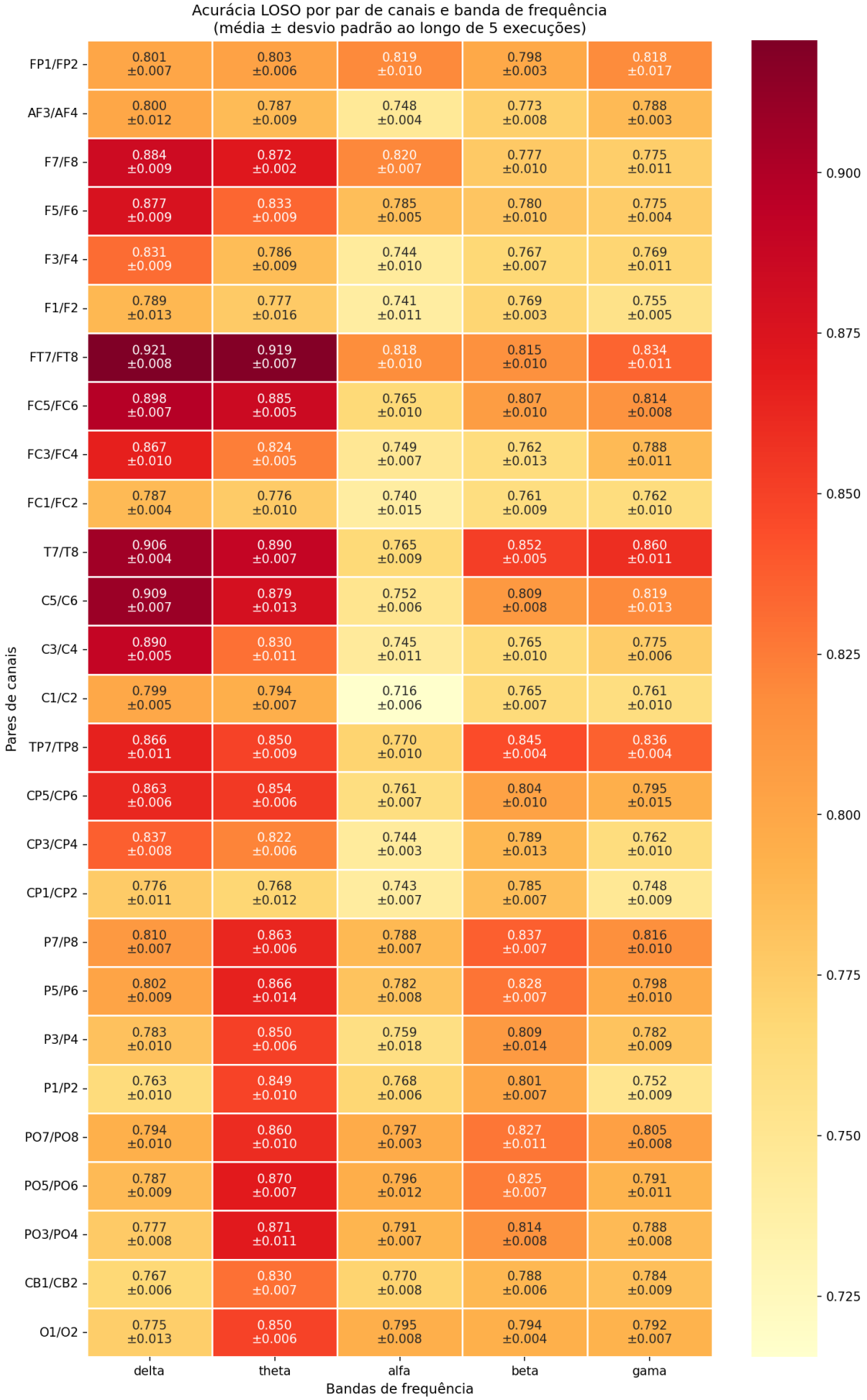
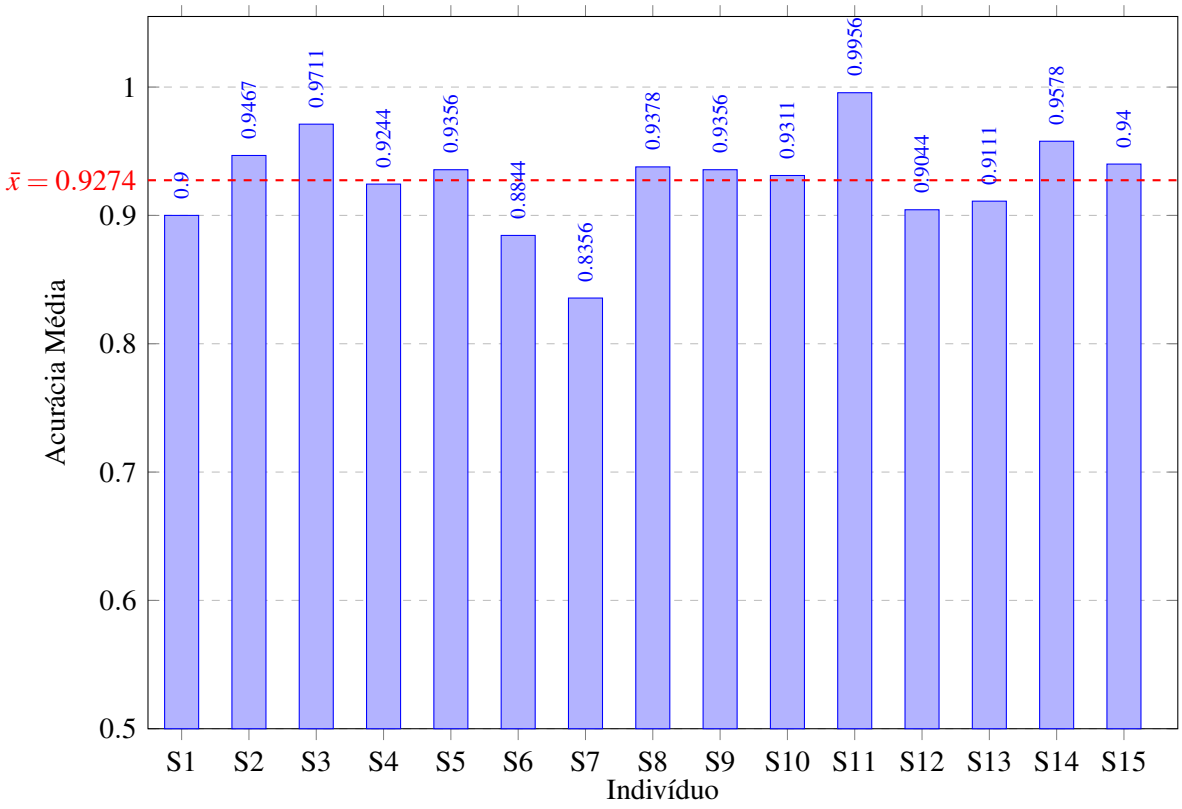


Figura 28 – Acurácia LOSO média por par de canais simétricos e banda de frequência, estimada sobre 5 execuções independentes. Valores mais altos indicam maior poder discriminativo da combinação para o reconhecimento de emoções.

5.2 Desempenho

A Figura 29 apresenta a acurácia média por indivíduo ao longo de 10 execuções completas do protocolo LOSO utilizando todos os 62 canais e as cinco bandas de frequência, servindo como referência para avaliar o impacto da seleção de canais e bandas. O modelo alcança acurácia média global de 92,74% nessa configuração.

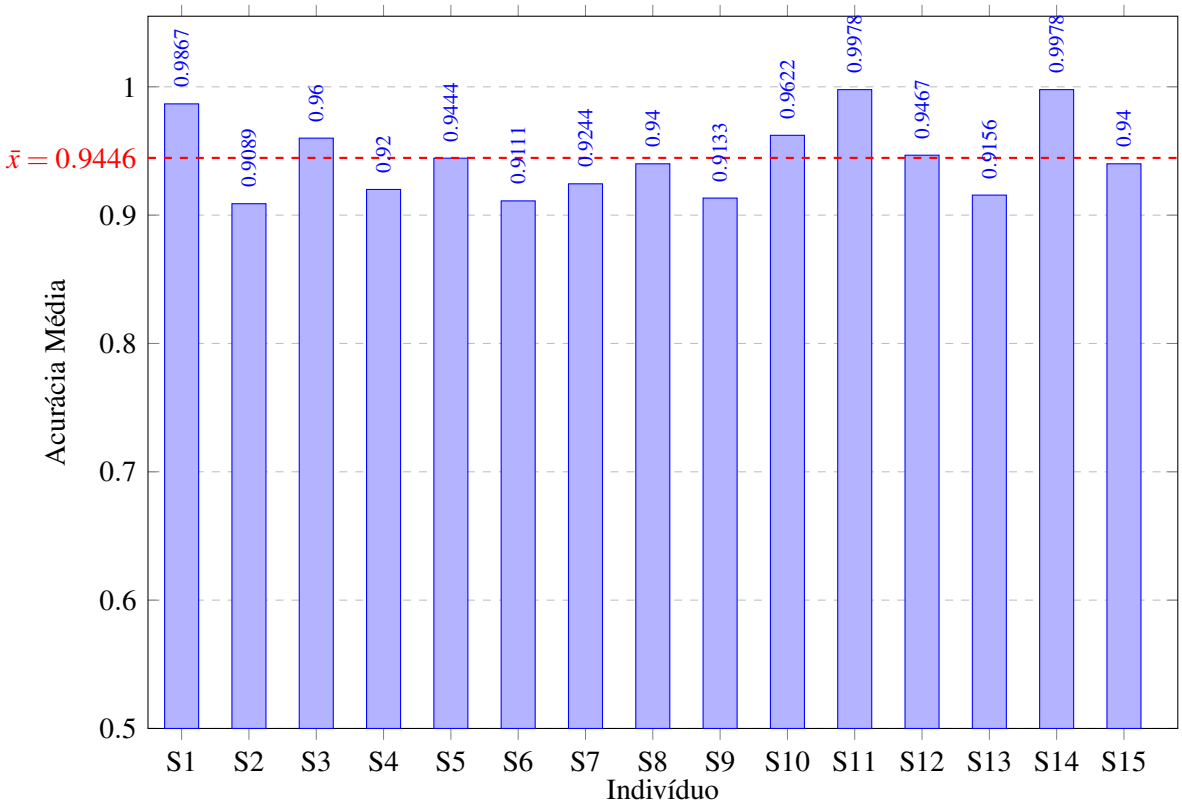
Figura 29 – Acurácia média por indivíduo ao longo das 10 execuções LOSO utilizando todos os 62 canais e as cinco bandas de frequência. A linha vermelha tracejada indica a acurácia média geral de 92,74%.



A Figura 30 apresenta os resultados obtidos com a configuração restrita aos quatro canais selecionados (FT7, FT8, T7 e T8) e as cinco bandas de frequência. A redução de 62 para 4 canais resulta em ganho de acurácia média de 92,74% para 94,46%, indicando que os canais fronto-temporais e temporais centrais concentram a maior parte da informação discriminativa para a tarefa.

A Figura 31 apresenta os resultados da configuração final adotada neste trabalho, utilizando os quatro canais selecionados e restringindo as bandas a delta e theta. O modelo proposto alcança acurácia média global de 95,38%, representando um ganho adicional em relação à configuração com cinco bandas e refletindo a variabilidade natural entre indivíduos esperada

Figura 30 – Acurácia média por indivíduo ao longo das 10 execuções LOSO utilizando os quatro canais selecionados (FT7, FT8, T7 e T8) e as cinco bandas de frequência. A linha vermelha tracejada indica a acurácia média geral de 94,46%.



em cenários SI.

A acurácia média por indivíduo varia entre 91,11% e 99,78% . Os indivíduos S6 e S13 destacam-se como os mais difíceis, com acurácias médias de 88,67% e 85,78%, respectivamente, evidenciando a variabilidade entre indivíduos característica de cenários SI.

A Figura 32 apresenta a matriz de confusão obtida em uma única execução do protocolo LOSO. Das 675 amostras avaliadas, 648 foram classificadas corretamente, resultando em acurácia de 96,00%. A classe positiva apresentou o maior número de acertos (216), seguida da classe neutra (217) e da classe negativa (215). As confusões mais frequentes ocorreram entre as classes negativa e neutra, com 6 amostras neutras classificadas como negativas e 5 amostras negativas classificadas como neutras.

5.3 Comparação com trabalhos relacionados

A Tabela 1 apresenta a comparação entre o modelo proposto e alguns trabalhos da literatura que utilizam a base SEED para reconhecimento de emoções. Uma tendência identificada é que resultados obtidos em cenários SD são, em geral, superiores aos obtidos

Figura 31 – Acurácia média por indivíduo ao longo das 10 execuções LOSO (bandas delta e theta). A linha vermelha tracejada indica a acurácia média geral de 95,38%.

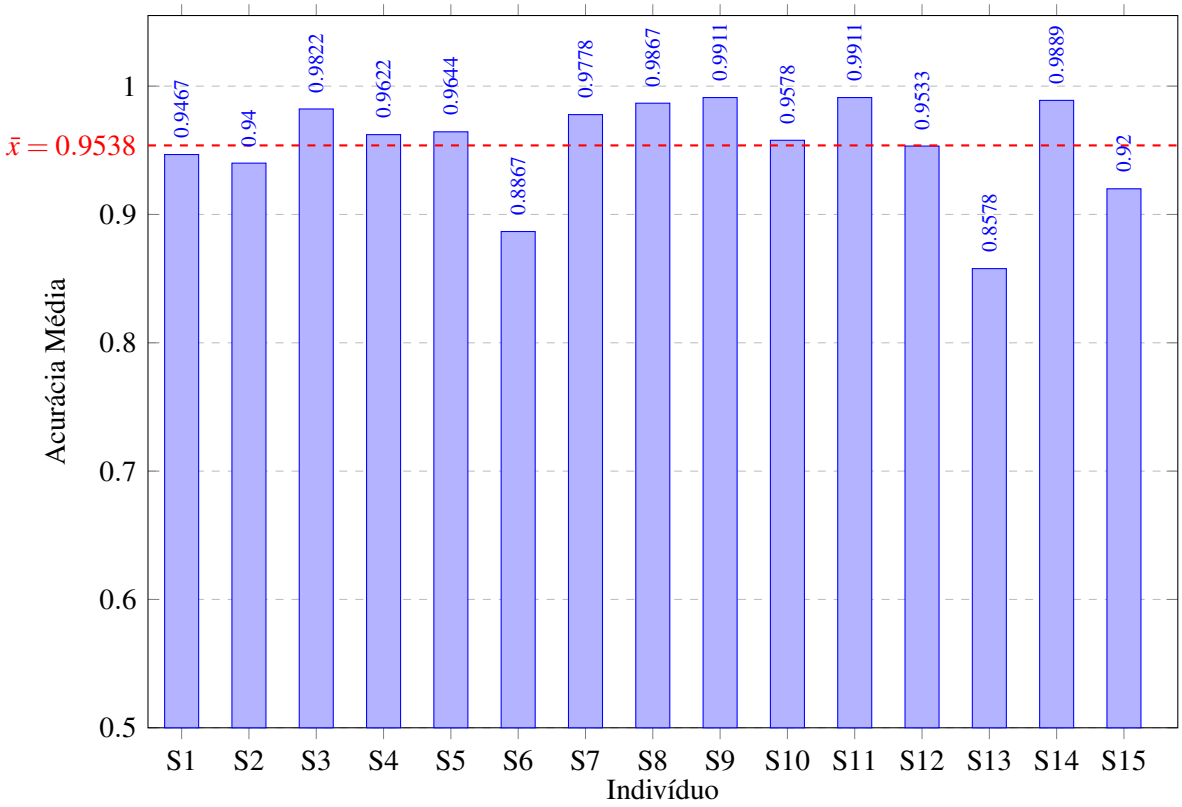


Figura 32 – Matriz de confusão obtida em uma única execução do protocolo LOSO.

em cenários SI. Isso ocorre porque, no cenário SD, o modelo tem acesso a dados do mesmo indivíduo tanto no treinamento quanto no teste, permitindo capturar padrões específicos daquele indivíduo. Em contraste, no cenário SI sob protocolo LOSO, o modelo precisa generalizar para indivíduos nunca vistos durante o treinamento, tornando a tarefa mais desafiadora. Por essa

razão, comparações diretas entre cenários devem ser feitas com cautela.

Tabela 1 – Comparação de trabalhos que utilizam a base SEED para reconhecimento de emoções.

Referência	**Classes**	**SD (%)**	**SI (%)**	**LOSO (SI)**
(ZHENG; LU, 2015)	3	86.08	-	-
(LI *et al.*, 2018)	3	92.38	83.28	Sim
(LIU *et al.*, 2020)	2	-	97.56	Sim
(CIZMECI *et al.*, 2022)	3	99.83	-	-
(WANG *et al.*, 2022)	3	86.10	-	-
(PUSARLA *et al.*, 2022)	3	-	97.91	Sim
(WEI; ZHOU, 2024)	3	95.73	87.38	Não
(WANG *et al.*, 2025)	3	92.10	85.68	Sim
(CAO *et al.*, 2025)	3	95.20	-	-
(ZHONG *et al.*, 2020)	3	94.24	85.30	Sim
(LI *et al.*, 2021)	3	-	86.81	Sim
(KONG *et al.*, 2022)	3	93.36	-	-
(LIN *et al.*, 2023)	3	90.22	81.85	Não
(FENG *et al.*, 2025)	3	96.38	-	-
(ASADZADEH *et al.*, 2022)	2	99.25	98.51	Não
(ASADZADEH *et al.*, 2023)	2	98.75	97.50	Não
(TIAN *et al.*, 2022)	3	-	97.40	Sim
(NING *et al.*, 2021)	3	-	97.66	Sim
(QUAN *et al.*, 2023)	3	-	92.83	Sim
(LI *et al.*, 2022b)	3	92.75	-	-
(XU *et al.*, 2023)	3	-	92.59	Sim
(TIAN *et al.*, 2023)	3	-	97.21	Não
(VALDERRAMA; SHEORAN, 2025)	3	-	79.30	Sim
(JIN; KIM, 2020)	3	99.63	-	-
Proposto	**3**	**-**	**95.38**	**Sim**

Nem todos os trabalhos com alto desempenho reportado operam sob as mesmas condições experimentais. Por exemplo, Jin e Kim (2020) reporta 99,63% utilizando validação cruzada intra-indivíduo no cenário SD e Cizmeci *et al.* (2022) reporta 99,83% a partir de uma divisão sobre o conjunto agregado de todos os indivíduos, sem separação por indivíduo entre treino e teste, o que favorece artificialmente o desempenho reportado. Ambos os trabalhos não avaliam generalização entre indivíduos.

Outros trabalhos apresentam alto desempenho em cenário SI, mas sob condições que diferem do protocolo adotado neste trabalho. Liu *et al.* (2020) reporta 97,56% em LOSO, porém avalia apenas classificação binária com duas classes emocionais, reduzindo a complexidade da tarefa em relação ao cenário de três classes adotado aqui. Tian *et al.* (2022) reporta 97,40% em

LOSO com três classes, porém o modelo foi desenvolvido originalmente para indivíduos com deficiência auditiva e os hiperparâmetros não foram ajustados especificamente para o SEED, o que dificulta a interpretação isolada desse resultado. Ning *et al.* (2021) alcança 97,66% em LOSO por meio de aprendizado *few-shot* com adaptação de domínio. Nessa abordagem, além dos dados dos demais indivíduos, o modelo recebe durante o treinamento um pequeno número de amostras rotuladas do próprio indivíduo que será testado, sendo que no melhor resultado reportado foram utilizadas 20 amostras por classe. Isso difere do cenário estritamente *zero-shot* adotado neste trabalho, no qual nenhuma amostra do indivíduo de teste é acessada durante o treinamento. Da mesma forma, Quan *et al.* (2023) utiliza uma pequena quantidade de dados rotulados do indivíduo alvo para melhorar a capacidade de adaptação do modelo, caracterizando um cenário semi-supervisionado que também difere do adotado aqui.

Dentre os trabalhos avaliados sob protocolo LOSO com três classes em cenário estritamente SI, Pusarla *et al.* (2022) reporta a maior acurácia, 97,91%, utilizando espectrogramas bidimensionais como representação de entrada em conjunto com uma arquitetura DenseNet profunda que incorpora informações temporais e de frequência de forma abrangente a partir dos 62 canais disponíveis. Tian *et al.* (2023) reporta 97,21% em cenário SI sem LOSO, utilizando um modelo VAE-GAN para geração de amostras sintéticas que ampliam o conjunto de treinamento.

Em contraste, o modelo proposto utiliza apenas quatro canais e duas bandas de frequência (delta e theta), reduzindo significativamente a dimensionalidade do problema em relação a todos os trabalhos citados. Essa configuração compacta torna a tarefa inerentemente mais desafiadora no cenário entre indivíduos, pois dispõe de substancialmente menos informação espectral e espacial. A título de comparação, Cao *et al.* (2025) observa que ao reduzir o número de canais de 62 para aproximadamente 32 em cenário SD, a acurácia cai de 95,20% para 93,0%, evidenciando o impacto da redução de canais mesmo em condições mais favoráveis.

Ainda assim, o modelo proposto atinge 95,38% sob protocolo LOSO sem acesso a EEGs do indivíduo de teste durante o treinamento, superando trabalhos que utilizam todos os 62 canais e as cinco bandas de frequência convencionais, como Zhong *et al.* (2020) (85,30%), Li *et al.* (2021) (86,81%), Wang *et al.* (2025) (85,68%) e Valderrama e Sheoran (2025) (79,30%). Supera também Xu *et al.* (2023) (92,59%), que emprega adaptação adversarial de domínio com todos os 62 canais, e Lin *et al.* (2023) (81,85%), que utiliza seleção adaptativa de canais por mecanismo de atenção. Esses resultados evidenciam que a abordagem proposta oferece um equilíbrio favorável entre desempenho e viabilidade prática: ao operar com poucos canais

e bandas limitadas, o modelo se aproxima de configurações compatíveis com dispositivos de aquisição portáteis e de baixo custo, sem abrir mão de acurácia competitiva no cenário SI com três classes emocionais.

6 CONCLUSÃO

Este trabalho apresentou uma abordagem baseada em GNNs para a classificação de estados emocionais a partir de sinais de EEG, na qual os canais são representados como nós de um grafo, permitindo explorar diretamente as relações espaciais e funcionais entre diferentes regiões cerebrais.

Uma das contribuições do trabalho foi a adoção de uma estratégia de seleção de canais orientada por desempenho, na qual pares simétricos de eletrodos foram avaliados em combinação com diferentes bandas de frequência. Os resultados indicaram que, dentre as configurações avaliadas, um subconjunto de quatro canais frontotemporais e temporais centrais (FT7, FT8, T7 e T8) combinado com as bandas delta e theta apresentou o melhor desempenho, utilizando apenas 4 dos 62 canais disponíveis e 2 das 5 bandas de frequência convencionalmente adotadas na literatura. Esse resultado reforça a viabilidade de sistemas de reconhecimento de emoções baseados em dispositivos de aquisição com número reduzido de eletrodos.

Os experimentos foram conduzidos sob o protocolo LOSO, que avalia a capacidade de generalização entre indivíduos. O modelo alcançou acurácia média de 95,38% ao longo de 10 execuções independentes, apresentando desempenho competitivo em relação a trabalhos da literatura que utilizam configurações de entrada com maior número de canais e bandas de frequência.

Além do desempenho obtido, os resultados contribuem para uma melhor compreensão da importância relativa de diferentes regiões cerebrais no processamento emocional, destacando o papel das áreas frontotemporais e temporais centrais, em consonância com achados da literatura de neurociência afetiva.

Dessa forma, os resultados indicam que a combinação de representações em grafos com uma seleção compacta de canais e bandas de frequência constitui uma abordagem promissora para o reconhecimento de emoções a partir de sinais de EEG, conciliando desempenho competitivo e potencial de aplicação em dispositivos vestíveis com número reduzido de eletrodos.

REFERÊNCIAS

ASADZADEH, S.; REZAII, T. Y.; BEHESHTI, S.; MESHGINI, S. Accurate emotion recognition using bayesian model based eeg sources as dynamic graph convolutional neural network nodes. **Scientific Reports**, Nature Publishing Group UK London, v. 12, n. 1, p. 10282, 2022.

ASADZADEH, S.; REZAII, T. Y.; BEHESHTI, S.; MESHGINI, S. Accurate emotion recognition utilizing extracted eeg sources as graph neural network nodes. **Cognitive Computation**, Springer, v. 15, n. 1, p. 176–189, 2023.

AVILES, M.; SÁNCHEZ-REYES, L. M.; ÁLVAREZ-ALVARADO, J. M.; RODRÍGUEZ-RESÉNDIZ, J. Machine and deep learning trends in eeg-based detection and diagnosis of alzheimer's disease: A systematic review. **Eng**, MDPI, v. 5, n. 3, p. 1464–1484, 2024.

BABAI, L. Graph isomorphism in quasipolynomial time. In: **Proceedings of the forty-eighth annual ACM symposium on Theory of Computing**. [*S. l.*: *s. n.*], 2016. p. 684–697.

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

BCMI Lab, Shanghai Jiao Tong University. **SEED Dataset**. 2019. Acesso em: maio 2026. Disponível em: https://bcmi.sjtu.edu.cn/home/seed/seed.html.

BCMI Lab, Shanghai Jiao Tong University. **SEED-SD Dataset**. 2021. Acesso em: maio 2026. Disponível em: https://bcmi.sjtu.edu.cn/home/seed/seed-SD.html.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. **IEEE transactions on neural networks**, IEEE, v. 5, n. 2, p. 157–166, 1994.

BISHOP, C. M.; NASRABADI, N. M. **Pattern recognition and machine learning**. [*S. l.*]: Springer, 2006. v. 4.

BRONSTEIN, M. M.; BRUNA, J.; COHEN, T.; VELIČKOVIĆ, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. **arXiv preprint arXiv:2104.13478**, 2021.

BRONSTEIN, M. M.; BRUNA, J.; LECUN, Y.; SZLAM, A.; VANDERGHEYNST, P. Geometric deep learning: going beyond euclidean data. **IEEE Signal Processing Magazine**, IEEE, v. 34, n. 4, p. 18–42, 2017.

CAO, L.; ZHAO, W.; SUN, B. Emotion recognition using multi-scale eeg features through graph convolutional attention network. **Neural Networks**, Elsevier, v. 184, p. 107060, 2025.

CHADDAD, A.; WU, Y.; KATEB, R.; BOURIDANE, A. Electroencephalography signal processing: A comprehensive review and analysis of methods and techniques. **Sensors**, MDPI, v. 23, n. 14, p. 6434, 2023.

CHEN, S.; CHEN, M.; WANG, X.; LIU, X.; LIU, B.; MING, D. Brain–computer interfaces in 2023–2024. **Brain-x**, Wiley Online Library, v. 3, n. 1, p. e70024, 2025.

CHO, K.; MERRIËNBOER, B. V.; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014.

CIZMECI, H.; OZCAN, C.; DURGUT, R. Channel selection and feature extraction on deep eeg classification using metaheuristic and welch psd. **Soft Computing**, Springer, v. 26, n. 19, p. 10115–10125, 2022.

DEFFERRARD, M.; BRESSON, X.; VANDERGHEYNST, P. Convolutional neural networks on graphs with fast localized spectral filtering. **Advances in neural information processing systems**, v. 29, 2016.

DEWANGAN, N.; THAKUR, K.; SINGH, B.; SONI, A.; MANDAL, S. Subject dependent and subject independent analysis for emotion recognition using electroencephalogram (eeg) signal. In: IOP PUBLISHING. **Journal of Physics: Conference Series**. [*S. l.*], 2023. v. 2576, n. 1, p. 012001.

DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. **arXiv preprint arXiv:1603.07285**, 2016.

DUVENAUD, D. K.; MACLAURIN, D.; IPARRAGUIRRE, J.; BOMBARELL, R.; HIRZEL, T.; ASPURU-GUZIK, A.; ADAMS, R. P. Convolutional networks on graphs for learning molecular fingerprints. **Advances in neural information processing systems**, v. 28, 2015.

FENG, T.; WU, C.; NIU, Y.; LI, F.; LI, Y.; FU, B.; ZHAO, Z.; WANG, X. Adaptive progressive attention graph neural network for eeg emotion recognition. **arXiv preprint arXiv:2501.14246**, 2025.

FEY, M.; LENSSEN, J. E. Fast graph representation learning with pytorch geometric. **arXiv preprint arXiv:1903.02428**, 2019.

GILMER, J.; SCHOENHOLZ, S. S.; RILEY, P. F.; VINYALS, O.; DAHL, G. E. Neural message passing for quantum chemistry. In: PMLR. **International conference on machine learning**. [*S. l.*], 2017. p. 1263–1272.

GKINTONI, E.; VANTARAKIS, A.; GOURZIS, P. Neuroimaging insights into the public health burden of neuropsychiatric disorders: a systematic review of electroencephalography-based cognitive biomarkers. **Medicina**, MDPI, v. 61, n. 6, p. 1003, 2025.

GOLUB, G. H.; LOAN, C. F. V. **Matrix computations**. [*S. l.*]: JHU press, 2013.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016. (Adaptive Computation and Machine Learning series). ISBN 9780262035613. Disponível em: https://books.google.com.br/books?id=Np9SDQAAQBAJ.

GORI, M.; MONFARDINI, G.; SCARSELLI, F. A new model for learning in graph domains. In: IEEE. **Proceedings. 2005 IEEE international joint conference on neural networks, 2005.** [*S. l.*], 2005. v. 2, p. 729–734.

GRAVES, A.; MOHAMED, A.-r.; HINTON, G. Speech recognition with deep recurrent neural networks. In: IEEE. **2013 IEEE international conference on acoustics, speech and signal processing**. [*S. l.*], 2013. p. 6645–6649.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT press, v. 9, n. 8, p. 1735–1780, 1997.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: PMLR. **International conference on machine learning**. [*S. l.*], 2015. p. 448–456.

JIN, L.; KIM, E. Y. Interpretable cross-subject eeg-based emotion recognition using channel-wise features. **Sensors**, MDPI, v. 20, n. 23, p. 6719, 2020.

JURCAK, V.; TSUZUKI, D.; DAN, I. 10/20, 10/10, and 10/5 systems revisited: their validity as relative head-surface-based positioning systems. **Neuroimage**, Elsevier, v. 34, n. 4, p. 1600–1611, 2007.

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

KIPF, T. N.; WELLING, M. Semi-supervised classification with graph convolutional networks. **arXiv preprint arXiv:1609.02907**, 2016.

KLEPL, D.; WU, M.; HE, F. Graph neural network-based eeg classification: A survey. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, IEEE, v. 32, p. 493–503, 2024.

KONG, W.; QIU, M.; LI, M.; JIN, X.; ZHU, L. Causal graph convolutional neural network for emotion recognition. **IEEE Transactions on Cognitive and Developmental Systems**, IEEE, v. 15, n. 4, p. 1686–1693, 2022.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, 2012.

LI, J.; LI, S.; PAN, J.; WANG, F. Cross-subject eeg emotion recognition with self-organized graph neural network. **Frontiers in Neuroscience**, Frontiers Media SA, v. 15, p. 611653, 2021.

LI, Q.; HAN, Z.; WU, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. In: **Proceedings of the AAAI conference on artificial intelligence**. [*S. l.*: *s. n.*], 2018. v. 32, n. 1.

LI, X.; ZHANG, Y.; TIWARI, P.; SONG, D.; HU, B.; YANG, M.; ZHAO, Z.; KUMAR, N.; MARTTINEN, P. Eeg based emotion recognition: A tutorial and review. **ACM Computing Surveys**, ACM New York, NY, v. 55, n. 4, p. 1–57, 2022.

LI, Y.; YU, R.; SHAHABI, C.; LIU, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. **arXiv preprint arXiv:1707.01926**, 2017.

LI, Y.; ZHENG, W.; CUI, Z.; ZHANG, T.; ZONG, Y. A novel neural network model based on cerebral hemispheric asymmetry for eeg emotion recognition. In: **IJCAI**. [*S. l.*: *s. n.*], 2018. p. 1561–1567.

LI, Z.; HWANG, K.; LI, K.; WU, J.; JI, T. Graph-generative neural network for eeg-based epileptic seizure detection via discovery of dynamic brain functional connectivity. **Scientific reports**, Nature Publishing Group UK London, v. 12, n. 1, p. 18998, 2022.

LI, Z.; ZHAO, X.; YANG, Y.; GAO, Q.; SONG, Y. Hvfm: an emotion classification model based on horizontal and vertical flow domain-adaptive. In: IEEE. **2022 IEEE International conference on mechatronics and automation (ICMA)**. [*S. l.*], 2022. p. 455–460.

LIN, X.; CHEN, J.; MA, W.; TANG, W.; WANG, Y. Eeg emotion recognition using improved graph neural network with channel selection. **Computer Methods and Programs in Biomedicine**, Elsevier, v. 231, p. 107380, 2023.

LIU, S.; WANG, X.; ZHAO, L.; ZHAO, J.; XIN, Q.; WANG, S.-H. Subject-independent emotion recognition of eeg signals based on dynamic empirical convolutional neural network. **IEEE/ACM transactions on computational biology and bioinformatics**, IEEE, v. 18, n. 5, p. 1710–1721, 2020.

LOGOTHETIS, N. K. What we can do and what we cannot do with fmri. **Nature**, Nature Publishing Group UK London, v. 453, n. 7197, p. 869–878, 2008.

LOSHCHILOV, I.; HUTTER, F. Decoupled weight decay regularization. **arXiv preprint arXiv:1711.05101**, 2017.

MOHAMMADI, H.; KARWOWSKI, W. Graph neural networks in brain connectivity studies: Methods, challenges, and future directions. **Brain Sciences**, MDPI, v. 15, n. 1, p. 17, 2024.

NIEDERMEYER, E.; SILVA, F. L. da. **Electroencephalography: basic principles, clinical applications, and related fields**. [*S. l.*]: Lippincott Williams & Wilkins, 2005.

NING, R.; CHEN, C. P.; ZHANG, T. Cross-subject eeg emotion recognition using domain adaptive few-shot learning networks. In: IEEE. **2021 IEEE international conference on bioinformatics and biomedicine (BIBM)**. [*S. l.*], 2021. p. 1468–1472.

NUNEZ, P. L.; SRINIVASAN, R. **Electric fields of the brain: the neurophysics of EEG**. [*S. l.*]: Oxford university press, 2006.

PASCANU, R.; MIKOLOV, T.; BENGIO, Y. On the difficulty of training recurrent neural networks. In: PMLR. **International conference on machine learning**. [*S. l.*], 2013. p. 1310–1318.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L. *et al.* Pytorch: An imperative style, high-performance deep learning library. **Advances in neural information processing systems**, v. 32, 2019.

POLDRACK, R. A.; MUMFORD, J. A.; NICHOLS, T. E. **Handbook of functional MRI data analysis**. [*S. l.*]: Cambridge University Press, 2024.

PUSARLA, N.; SINGH, A.; TRIPATHI, S. Learning densenet features from eeg based spectrograms for subject independent emotion recognition. **Biomedical signal processing and control**, Elsevier, v. 74, p. 103485, 2022.

QUAN, J.; LI, Y.; WANG, L.; HE, R.; YANG, S.; GUO, L. Eeg-based cross-subject emotion recognition using multi-source domain transfer learning. **Biomedical Signal Processing and Control**, Elsevier, v. 84, p. 104741, 2023.

SANTURKAR, S.; TSIPRAS, D.; ILYAS, A.; MADRY, A. How does batch normalization help optimization? **Advances in neural information processing systems**, v. 31, 2018.

SCARSELLI, F.; GORI, M.; TSOI, A. C.; HAGENBUCHNER, M.; MONFARDINI, G. The graph neural network model. **IEEE transactions on neural networks**, IEEE, v. 20, n. 1, p. 61–80, 2008.

SCHLICHTKRULL, M.; KIPF, T. N.; BLOEM, P.; BERG, R. V. D.; TITOV, I.; WELLING, M. Modeling relational data with graph convolutional networks. In: SPRINGER. **European semantic web conference**. [*S. l.*], 2018. p. 593–607.

SHUMAN, D. I.; NARANG, S. K.; FROSSARD, P.; ORTEGA, A.; VANDERGHEYNST, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. **IEEE signal processing magazine**, IEEE, v. 30, n. 3, p. 83–98, 2013.

SIMMATIS, L.; RUSSO, E. E.; GERACI, J.; HARMSEN, I. E.; SAMUEL, N. Technical and clinical considerations for electroencephalography-based biomarkers for major depressive disorder. **Npj Mental Health Research**, Nature Publishing Group UK London, v. 2, n. 1, p. 18, 2023.

SOCHER, R.; PERELYGIN, A.; WU, J.; CHUANG, J.; MANNING, C. D.; NG, A. Y.; POTTS, C. Recursive deep models for semantic compositionality over a sentiment treebank. In: **Proceedings of the 2013 conference on empirical methods in natural language processing**. [*S. l.*: *s. n.*], 2013. p. 1631–1642.

SPERDUTI, A.; STARITA, A. Supervised neural networks for the classification of structures. **IEEE transactions on neural networks**, IEEE, v. 8, n. 3, p. 714–735, 1997.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

TIAN, C.; MA, Y.; CAMMON, J.; FANG, F.; ZHANG, Y.; MENG, M. Dual-encoder vae-gan with spatiotemporal features for emotional eeg data augmentation. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, IEEE, v. 31, p. 2018–2027, 2023.

TIAN, Z.; LI, D.; YANG, Y.; HOU, F.; YANG, Z.; SONG, Y.; GAO, Q. A novel domain adversarial networks based on 3d-lstm and local domain discriminator for hearing-impaired emotion recognition. **IEEE Journal of Biomedical and Health Informatics**, IEEE, v. 27, n. 1, p. 363–373, 2022.

VALDERRAMA, C. E.; SHEORAN, A. Identifying relevant eeg channels for subject-independent emotion recognition using attention network layers. **Frontiers in psychiatry**, Frontiers Media SA, v. 16, p. 1494369, 2025.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.

VELIČKOVIĆ, P.; CUCURULL, G.; CASANOVA, A.; ROMERO, A.; LIO, P.; BENGIO, Y. Graph attention networks. **arXiv preprint arXiv:1710.10903**, 2017.

VOLKOVA, K.; LEBEDEV, M. A.; KAPLAN, A.; OSSADTCHI, A. Decoding movement from electrocorticographic activity: a review. **Frontiers in neuroinformatics**, Frontiers Media SA, v. 13, p. 74, 2019.

WANG, J.; HUANG, Y.; SONG, S.; WANG, B.; SU, J.; DING, J. A novel fourier adjacency transformer for advanced eeg emotion recognition. In: SPRINGER. **International Conference on Medical Image Computing and Computer-Assisted Intervention**. [*S. l.*], 2025. p. 13–23.

WANG, J.-G.; SHAO, H.-M.; YAO, Y.; LIU, J.-L.; SUN, H.-P.; MA, S.-W. Electroencephalograph-based emotion recognition using convolutional neural network without manual feature extraction. **Applied Soft Computing**, Elsevier, v. 128, p. 109534, 2022.

WEI, C.; ZHOU, G. Eeg emotion recognition based on attention mechanism fusion transformer network. In: **Proceedings of the 2024 11th International Conference on Biomedical and Bioinformatics Engineering**. [*S. l.*: *s. n.*], 2024. p. 146–150.

WU, X.; AJORLOU, A.; WU, Z.; JADBABAIE, A. Demystifying oversmoothing in attention-based graph neural networks. **Advances in Neural Information Processing Systems**, v. 36, p. 35084–35106, 2023.

WU, Z.; PAN, S.; CHEN, F.; LONG, G.; ZHANG, C.; YU, P. S. A comprehensive survey on graph neural networks. **IEEE transactions on neural networks and learning systems**, IEEE, v. 32, n. 1, p. 4–24, 2020.

XU, T.; DANG, W.; WANG, J.; ZHOU, Y. Dagam: a domain adversarial graph attention model for subject-independent eeg-based emotion recognition. **Journal of Neural Engineering**, IOP Publishing, v. 20, n. 1, p. 016022, 2023.

YING, R.; HE, R.; CHEN, K.; EKSOMBATCHAI, P.; HAMILTON, W. L.; LESKOVEC, J. Graph convolutional neural networks for web-scale recommender systems. In: **Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining**. [*S. l.*: *s. n.*], 2018. p. 974–983.

YU, B.; YIN, H.; ZHU, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. **arXiv preprint arXiv:1709.04875**, 2017.

YUN, S. Advances, challenges, and prospects of electroencephalography-based biomarkers for psychiatric disorders: a narrative review. **Journal of Yeungnam Medical Science**, Journal of Yeungnam Medical Science, v. 41, n. 4, p. 261–268, 2024.

ZHANG, H.; ZHOU, Q.-Q.; CHEN, H.; HU, X.-Q.; LI, W.-G.; BAI, Y.; HAN, J.-X.; WANG, Y.; LIANG, Z.-H.; CHEN, D. *et al.* The applied principles of eeg analysis methods in neuroscience and clinical neurology. **Military Medical Research**, Springer, v. 10, n. 1, p. 67, 2023.

ZHDANOV, M.; STEINMANN, S.; HOFFMANN, N. Investigating brain connectivity with graph neural networks and gnnexplainer. In: IEEE. **2022 26th International Conference on Pattern Recognition (ICPR)**. [*S. l.*], 2022. p. 5155–5161.

ZHENG, W.-L.; LU, B.-L. Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks. **IEEE Transactions on autonomous mental development**, IEEE, v. 7, n. 3, p. 162–175, 2015.

ZHONG, P.; WANG, D.; MIAO, C. Eeg-based emotion recognition using regularized graph neural networks. **IEEE Transactions on Affective Computing**, IEEE, v. 13, n. 3, p. 1290–1301, 2020.

ZITNIK, M.; AGRAWAL, M.; LESKOVEC, J. Modeling polypharmacy side effects with graph convolutional networks. **Bioinformatics**, Oxford University Press, v. 34, n. 13, p. i457–i466, 2018.