

Instruções: Responda às questões de múltipla escolha assinalando apenas uma alternativa. Nas questões dissertativas, apresente o raciocínio passo a passo, incluindo fórmulas e cálculos intermediários quando solicitado. Mantenha a notação utilizada em aula: $G = (V, E)$ para o grafo, A para a *adjacency matrix*, X para os atributos de nó, $\mathcal{N}(v)$ para a vizinhança de v , $h_v^{(k)}$ para o embedding de v na camada k , $\tilde{A} = A + I$ e $\tilde{D} = \text{diag}(\tilde{A}\mathbf{1})$ para a normalização do GCN, e α_{vu} para os coeficientes de atenção do GAT. Em valores numéricos use a vírgula como separador decimal.

Questões de Múltipla Escolha.

- I) Considere o grafo não-dirigido G com $V = \{1, 2, 3, 4\}$ e arestas $E = \{(1, 2), (2, 3), (2, 4), (3, 4)\}$. A partir dessa descrição, qual é o vetor de graus (d_1, d_2, d_3, d_4) dos quatro nós?
- (a) (1, 3, 2, 2)
 - (b) (2, 4, 3, 3)
 - (c) (1, 2, 2, 1)
 - (d) (2, 3, 3, 2)
- II) Uma camada de *message passing* agrega as mensagens dos vizinhos por meio de uma função AGG. Que propriedade essa função precisa satisfazer para que a camada seja equivariante à permutação dos nós do grafo?
- (a) Ser uma transformação linear dos embeddings dos vizinhos.
 - (b) Ser indiferente à ordem em que os vizinhos são apresentados.
 - (c) Ser uma função monotônica crescente do grau do nó central.
 - (d) Ser injetiva sobre vetores individuais de atributos de nó.
- III) No grafo da questão I, considere uma agregação por média que inclui o próprio nó,

$$\frac{1}{|\mathcal{N}(v) \cup \{v\}|} \sum_{u \in \mathcal{N}(v) \cup \{v\}} h_u^{(0)},$$

com atributos escalares iniciais $h^{(0)} = (1, 2, 3, 4)^\top$. Qual é o valor agregado para o nó 2?

- (a) 2,5
 - (b) 3,0
 - (c) 4,5
 - (d) 2,0
- IV) Na forma matricial do GCN, $H^{(k)} = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(k-1)} W^{(k)})$, o fator $\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$ é calculado uma única vez e reaproveitado a cada camada e a cada época. Qual característica do problema justifica esse reaproveitamento?
- (a) Esse fator não contém parâmetros aprendíveis e depende somente da estrutura fixa do grafo.
 - (b) Esse fator é recalculado por *backpropagation* mas converge nas primeiras épocas.
 - (c) Esse fator depende de $W^{(k)}$, que permanece congelado durante todo o treino.
 - (d) Esse fator muda a cada camada porque incorpora a não-linearidade σ .

- V) Considere a aresta entre os nós 2 e 3 do grafo da questão I. No GCN, o peso aplicado à mensagem que viaja por essa aresta é $1/\sqrt{\tilde{d}_v\tilde{d}_u}$, onde $\tilde{d}$ é o grau no grafo com auto-laços. Qual é, aproximadamente, esse peso?
- (a) 0,41
 - (b) 0,29
 - (c) 0,50
 - (d) 0,33
- VI) No GCN clássico, dois nós com a mesma estrutura local e os mesmos atributos recebem embeddings idênticos após a propagação. No grafo da questão I, com atributos escalares iniciais iguais para todos os nós, qual par de nós necessariamente termina com o mesmo embedding?
- (a) Os nós 1 e 2.
 - (b) Os nós 2 e 3.
 - (c) Os nós 3 e 4.
 - (d) Os nós 1 e 4.
- VII) Um modelo treinado em um conjunto de grafos de moléculas precisa ser aplicado a moléculas inéditas no momento do teste. Entre GCN na forma matricial e GraphSAGE com *neighbor sampling*, qual escolha é mais adequada e por qual razão estrutural?
- (a) GCN, pois a matriz $\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ generaliza diretamente para grafos não vistos.
 - (b) GraphSAGE, pois aprende uma função de agregação aplicável a vizinhanças de qualquer grafo.
 - (c) GCN, pois o reaproveitamento do fator de propagação reduz o custo em grafos novos.
 - (d) GraphSAGE, pois a amostragem de vizinhos elimina a necessidade de atributos de nó.
- VIII) No GAT, um nó v tem três vizinhos cujos *scores* de atenção não normalizados são $e_{v,u_1} = 2,0$, $e_{v,u_2} = 1,0$ e $e_{v,u_3} = 1,5$. Após a normalização por softmax sobre $\mathcal{N}(v)$, qual é, aproximadamente, o coeficiente α_{v,u_1} ?
- (a) 0,33
 - (b) 0,19
 - (c) 0,51
 - (d) 0,67
- IX) Compare como GCN e GAT atribuem peso à mensagem de cada vizinho durante a agregação. Qual afirmação descreve corretamente a diferença entre os dois mecanismos?
- (a) No GCN o peso é fixado pelos graus dos nós, enquanto no GAT o peso é função dos embeddings dos nós envolvidos.
 - (b) No GCN o peso é aprendido por descida de gradiente, enquanto no GAT o peso é fixado pela estrutura do grafo.
 - (c) Ambos aprendem os pesos, mas o GAT restringe a soma dos pesos de cada nó a um.

- (d) Ambos usam pesos fixos, e a diferença está apenas na função de ativação aplicada após a agregação.
- X) Uma equipe empilha 12 camadas de GCN para classificação de nós em um grafo de citações de diâmetro pequeno e observa que a acurácia de validação cai em relação a um modelo de 3 camadas. Qual fenômeno explica essa queda e qual intervenção o ataca diretamente?
- (a) *Overfitting* de parâmetros, mitigado por aumentar a taxa de aprendizado.
 - (b) *Oversmoothing* dos embeddings, mitigado por *skip connections* entre camadas.
 - (c) *Vanishing gradient* na entrada, mitigado por remover a normalização de grau.
 - (d) Desbalanceamento de classes, mitigado por reamostrar os nós de treino.
- XI)  Considere dois grafos não isomorfos, ambos com todos os nós de grau 2 e atributos iniciais idênticos: G_1 é um ciclo de seis nós e G_2 são dois triângulos disjuntos. Sobre a capacidade de uma GNN de *message passing* de distinguir esses dois grafos, qual afirmação é correta?
- (a) Distingue após uma camada, pois os graus diferem entre os grafos.
 - (b) Distingue se a agregação usar soma em vez de média, pois a soma é injetiva.
 - (c) Não distingue, pois em toda iteração os *multisets* de rótulos coincidem entre os grafos.
 - (d) Não distingue, pois ambos os grafos têm o mesmo número de arestas.

Questões Dissertativas.

- I) (**Propagação do GCN à mão**) Considere o caminho G com $V = \{1, 2, 3\}$ e arestas $E = \{(1, 2), (2, 3)\}$, atributos escalares iniciais $h^{(0)} = (2, 4, 6)^\top$, $W^{(1)} = 1$ e $\sigma = \text{Id}$.
- (a) Escreva $\tilde{A} = A + I$ e o vetor de graus $\tilde{d}$ no grafo com auto-laços.
 - (b) Calcule a matriz normalizada $\hat{A} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$, entrada a entrada.
 - (c) Calcule $h^{(1)} = \hat{A} h^{(0)}$ e interprete por que o valor do nó central difere dos valores das pontas.
- II) (**Tarefas e readout**) Para cada uma das três tarefas canônicas em grafos, descreva o que se prevê e qual estratégia de *readout* é adequada.
- (a) *Node classification*: o que é previsto e como o embedding por nó é usado na saída.
 - (b) *Link prediction*: o que é previsto e como combinar os embeddings dos dois nós da aresta candidata.
 - (c) *Graph classification*: o que é previsto e por que o *readout* precisa ser invariante à permutação dos nós.
- III)  (**Soma, média e expressividade**) Considere um nó v com dois vizinhos e um nó v' com um único vizinho, todos os vizinhos carregando o mesmo atributo escalar a . Os *multisets* de vizinhos são $\{a, a\}$ para v e $\{a\}$ para v' .
- (a) Calcule o resultado de agregar por média e por soma em cada caso e indique qual operação distingue v de v' .
 - (b) Explique por que essa propriedade é descrita como injetividade sobre *multisets* e relacione-a à camada do GIN, $h_v^{(k)} = \text{MLP}^{(k)}((1 + \epsilon^{(k)})h_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(k-1)})$.

- (c) Dê um exemplo de tarefa em que essa distinção entre vizinhanças importa na prática e justifique.

IV) (**Profundidade e *oversmoothing***) Uma equipe quer aumentar o *receptive field* de uma GCN para capturar dependências de longo alcance e cogita empilhar muitas camadas.

- (a) Explique o que acontece com os embeddings dos nós à medida que o número de camadas cresce em um grafo de diâmetro pequeno, e por que isso degrada o desempenho.
- (b) Descreva duas mitigações vistas em aula e explique, para cada uma, como ela ataca o problema.
- (c) Comente o *trade-off* entre profundidade, *receptive field* e diversidade dos embeddings, indicando por que não é possível maximizar os três ao mesmo tempo.

Gabarito

Múltipla Escolha.

- I) **(a)** Contando as arestas incidentes a cada nó: o nó 1 aparece só em (1, 2), logo $d_1 = 1$; o nó 2 aparece em (1, 2), (2, 3), (2, 4), logo $d_2 = 3$; o nó 3 aparece em (2, 3), (3, 4), logo $d_3 = 2$; o nó 4 aparece em (2, 4), (3, 4), logo $d_4 = 2$. O vetor (2, 4, 3, 3) corresponde aos graus no grafo com auto-laços $\tilde{A} = A + I$, não ao grafo original. As outras opções não batem com a lista de arestas.
- II) **(b)** A equivariância à permutação exige que reordenar os vizinhos não altere o resultado da agregação, ou seja, que AGG seja invariante à ordem de seus argumentos (soma, média, máximo e atenção satisfazem isso). Linearidade não é necessária nem suficiente. Dependendo do grau ou ser injetiva sobre vetores individuais são propriedades distintas que não garantem invariância à ordem.
- III) **(a)** A vizinhança de 2 é $\{1, 3, 4\}$, então $\mathcal{N}(2) \cup \{2\} = \{1, 2, 3, 4\}$ com os quatro valores (1, 2, 3, 4). A média é $(1 + 2 + 3 + 4)/4 = 10/4 = 2,5$. A opção 3,0 ignora o próprio nó na contagem, 4,5 usa apenas alguns vizinhos e 2,0 não corresponde a nenhuma agregação coerente.
- IV) **(a)** O fator $\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$ depende apenas de $\tilde{A}$ e $\tilde{D}$, que são determinados pela estrutura do grafo e não contêm parâmetros treináveis, por isso é constante durante o treino. Ele não é produzido por *backpropagation*, não depende de $W^{(k)}$ e não incorpora σ , que é aplicada fora dele.
- V) **(b)** Com auto-laços, os graus são $\tilde{d}_2 = 4$ e $\tilde{d}_3 = 3$, então o peso é $1/\sqrt{4 \cdot 3} = 1/\sqrt{12} \approx 0,29$. O valor $0,41 = 1/\sqrt{6}$ usaria graus 2 e 3, 0,50 ignoraria a normalização cruzada e 0,33 corresponderia a uma média simples por grau.
- VI) **(c)** Os nós 3 e 4 são estruturalmente simétricos: ambos têm grau 2 e estão ligados ao nó 2 e entre si, formando vizinhanças intercambiáveis. Com atributos iniciais iguais, o GCN não consegue diferenciá-los e produz o mesmo embedding. O nó 1 tem grau 1 e o nó 2 tem grau 3, então nenhum dos outros pares é simétrico.
- VII) **(b)** GraphSAGE aprende uma função de agregação que age sobre vizinhanças locais, então pode ser aplicada a grafos novos sem reprocessar o grafo de treino: é indutivo por construção. O GCN na forma matricial precisa do grafo inteiro para montar o fator de propagação, o que o torna naturalmente transdutivo. A amostragem de vizinhos não dispensa os atributos de nó.
- VIII) **(c)** A softmax dos *scores* (2,0, 1,0, 1,5) dá $\exp(2,0) \approx 7,39$, $\exp(1,0) \approx 2,72$ e $\exp(1,5) \approx 4,48$, com soma $\approx 14,59$. Então $\alpha_{v,u_1} \approx 7,39/14,59 \approx 0,51$. Os valores 0,33, 0,19 e 0,67 correspondem a outras normalizações ou aos demais coeficientes.
- IX) **(a)** No GCN o peso de cada vizinho é $1/\sqrt{\tilde{d}_v \tilde{d}_u}$, fixado pelos graus e idêntico para qualquer tarefa. No GAT o coeficiente α_{vu} é obtido de $e_{vu} = \text{LeakyReLU}(\mathbf{a}^\top [W\mathbf{h}_v \| W\mathbf{h}_u])$, ou seja, depende do conteúdo dos embeddings e se ajusta à tarefa. As demais opções trocam os papéis ou descrevem ambos como fixos.
- X) **(b)** Empilhar muitas camadas em um grafo de diâmetro pequeno leva ao *oversmoothing*: após poucas camadas cada nó já vê quase todo o grafo e os embeddings convergem para vetores quase idênticos, perdendo poder discriminativo. *Skip connections* atacam isso preservando a informação da camada anterior em cada passo. As outras opções diagnosticam fenômenos que não explicam a queda observada com mais camadas.

- XI) (c) Ambos os grafos têm todos os nós de grau 2, então cada nó sempre computa o mesmo $\text{HASH}(c, \{\{c, c\}\})$ e os *multisets* de rótulos coincidem em toda iteração do 1-WL. Como toda MPNN é no máximo tão poderosa quanto o 1-WL, nenhuma camada de *message passing* separa G_1 de G_2 . A soma não ajuda aqui, pois os *multisets* de entrada já são idênticos; ter o mesmo número de arestas é consequência, não a causa.

Dissertativas.

- I) (a) Para o caminho 1, 2, 3, a matriz de adjacência é

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \tilde{A} = A + I = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Os graus com auto-laços são as somas das linhas: $\tilde{d} = (2, 3, 2)$.

- (b) Cada entrada é $\hat{A}_{ij} = \tilde{A}_{ij} / \sqrt{\tilde{d}_i \tilde{d}_j}$. Com $\tilde{d} = (2, 3, 2)$:

$$\hat{A} = \begin{bmatrix} 1/2 & 1/\sqrt{6} & 0 \\ 1/\sqrt{6} & 1/3 & 1/\sqrt{6} \\ 0 & 1/\sqrt{6} & 1/2 \end{bmatrix} \approx \begin{bmatrix} 0,50 & 0,41 & 0 \\ 0,41 & 0,33 & 0,41 \\ 0 & 0,41 & 0,50 \end{bmatrix}.$$

- (c) Aplicando a $h^{(0)} = (2, 4, 6)^\top$:

$$h_1^{(1)} \approx 0,50 \cdot 2 + 0,41 \cdot 4 = 2,63,$$

$$h_2^{(1)} \approx 0,41 \cdot 2 + 0,33 \cdot 4 + 0,41 \cdot 6 = 4,60,$$

$$h_3^{(1)} \approx 0,41 \cdot 4 + 0,50 \cdot 6 = 4,63,$$

ou seja, $h^{(1)} \approx (2,63, 4,60, 4,63)^\top$. O nó central 2 recebe contribuições dos três valores (1, 3 e ele próprio), com peso menor sobre si mesmo por ter grau maior, enquanto as pontas 1 e 3 agregam apenas dois valores com peso 0,50 sobre si, o que explica a diferença entre o valor central e os das pontas.

- II) (a) *Node classification*: prevê-se o rótulo de cada nó (por exemplo, a subárea de um artigo em uma rede de citações). O *readout* usa diretamente o embedding h_v do nó, passado por um classificador.
- (b) *Link prediction*: prevê-se se uma aresta (u, v) existe ou existirá (por exemplo, recomendar um item a um usuário). O *readout* combina os embeddings h_u e h_v , por concatenação, produto interno ou um MLP sobre o par, para produzir um *score* de existência.
- (c) *Graph classification*: prevê-se um rótulo do grafo inteiro (por exemplo, se uma molécula é tóxica). O *readout* agrega todos os h_v por *pooling* (soma, média, máximo ou atenção) para obter h_G . Esse *pooling* precisa ser invariante à permutação porque o grafo não tem ordem canônica de nós: reordenar os nós não pode mudar a predição sobre o grafo.
- III) (a) Média: para v , $\frac{a+a}{2} = a$; para v' , $\frac{a}{1} = a$. Os dois resultados coincidem, então a média não distingue. Soma: para v , $a + a = 2a$; para v' , a . Como $2a \neq a$ (para $a \neq 0$), a soma distingue v de v' .

- (b) A soma é injetiva sobre *multisets* porque preserva a multiplicidade dos elementos: *multisets* diferentes (como $\{a, a\}$ e $\{a\}$) produzem somas diferentes, ao passo que média e máximo descartam quantos elementos havia. A camada do GIN usa exatamente uma soma sobre os vizinhos seguida de um MLP, justamente para manter essa injetividade e atingir o teto de expressividade dado pelo teste 1-WL.
- (c) Em classificação de moléculas, por exemplo, um átomo de carbono ligado a dois hidrogênios é quimicamente diferente de um ligado a um único hidrogênio. Se a agregação usar média ou máximo e os atributos dos vizinhos forem iguais, o modelo não vê a diferença de multiplicidade; com soma, vê. Por isso GIN costuma vencer GCN e GAT em tarefas de classificação de grafo que dependem de subestruturas.
- IV) (a) Cada camada faz o nó agregar informação de mais um salto, então após K camadas o embedding de v reúne informação a até K -hops. Em um grafo de diâmetro pequeno, depois de poucas camadas todos os nós já enxergam praticamente o grafo inteiro e suas representações passam a misturar as mesmas mensagens repetidamente, convergindo para embeddings quase idênticos. Esse é o *oversmoothing*: os nós deixam de ser distinguíveis e a acurácia cai.
- (b) Duas mitigações vistas em aula: (i) *skip connections* no estilo $h_v^{(k)} = h_v^{(k-1)} + \text{GNN}^{(k)}(\dots)$, que preservam a informação da camada anterior e impedem que o embedding seja totalmente substituído pela média dos vizinhos; (ii) *Jumping Knowledge* (JK-Net), que concatena ou faz *max-pool* das ativações de todas as camadas no final, recuperando representações menos suavizadas das camadas iniciais. (Outras respostas válidas: *DropEdge*, normalizações como *PairNorm*, ou atenção global de Graph Transformers.)
- (c) Aumentar a profundidade amplia o *receptive field*, mas, passado o diâmetro do grafo, esse ganho vem acompanhado de *oversmoothing*, que reduz a diversidade dos embeddings. Manter embeddings diversos favorece camadas rasas, que por sua vez limitam o *receptive field*. Como aumentar a profundidade simultaneamente eleva o *receptive field* e derruba a diversidade, não há como maximizar os três ao mesmo tempo: é preciso escolher um ponto de equilíbrio, em geral poucas camadas combinadas com mitigações.