

TP558 - Tópicos avançados em Machine Learning: *Graph Neural Networks*



Inatel

Adicione aqui seu nome
Adicione aqui seu email

Introdução

- As Graph Neural Networks (GNNs) surgiram como uma poderosa ferramenta para lidar com dados em forma de grafos, como redes sociais, sistemas de recomendação, bioinformática, entre outros.
- São uma classe de modelos de aprendizado profundo projetados para operar em estruturas de dados de grafos que consistem em nós e arestas que os conectam.

Fundamentação teórica

GRAFICOS

- O **Gráfico** é o tipo de estrutura de dados que contém nós e arestas.

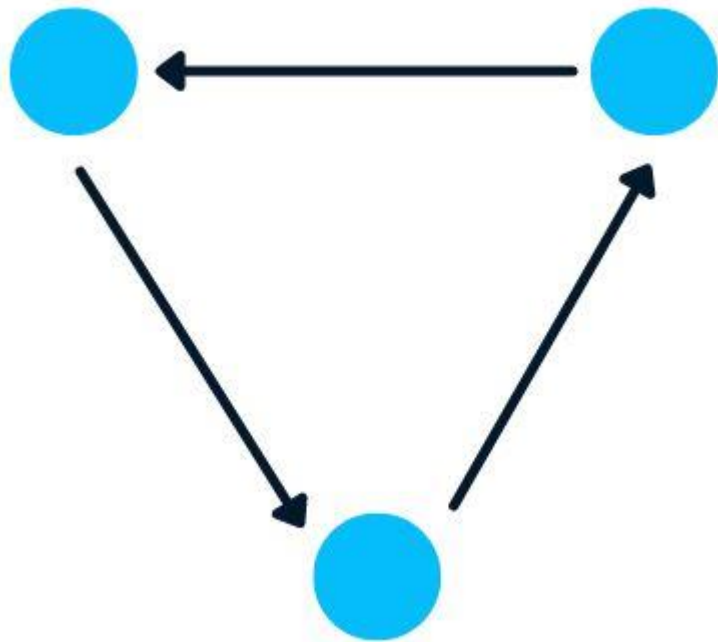
NÓS

- Pessoas
- Lugar

ARESTAS

- Direcionadas
- Não Direcionadas

Fundamentação teórica



- Os círculos azuis são nós.
- As setas são arestas.
 - Direção das arestas define dependências entre dois nós.

Fundamentação teórica

- Os dados em forma de grafos possuem propriedades únicas que os diferenciam de outras estruturas de dados, tais como:



Fundamentação teórica

Relações Complexas

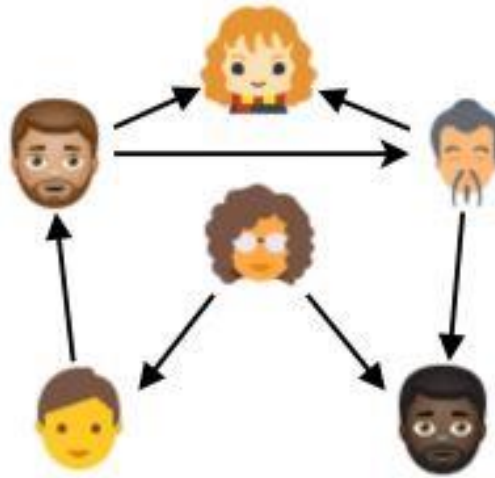
Os grafos são especialmente adequados para representar relações complexas entre entidades. Cada nó no grafo pode representar uma entidade, e as arestas entre os nós podem representar as relações entre essas entidades. Essas relações podem ser direcionadas ou não direcionadas, ponderadas ou não ponderadas, o que permite representar uma grande variedade de interações complexas.

Fundamentação teórica

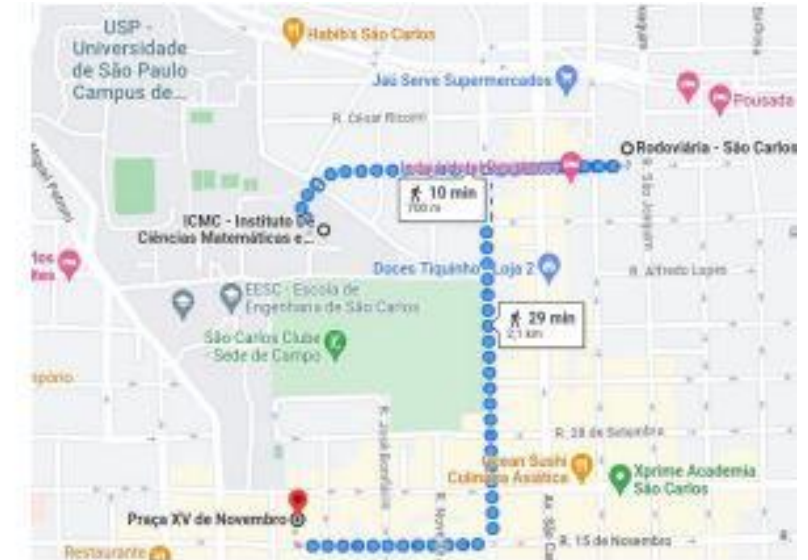
Relações Complexas



Grafo não direcionado



Grafo direcionado



Grafo ponderado

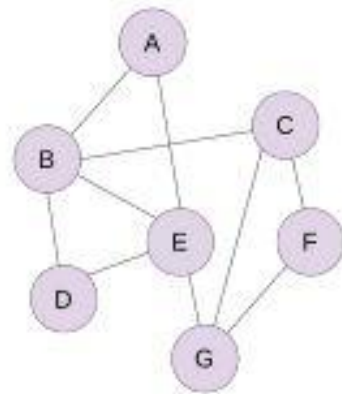
Fundamentação teórica

Topologias Variadas

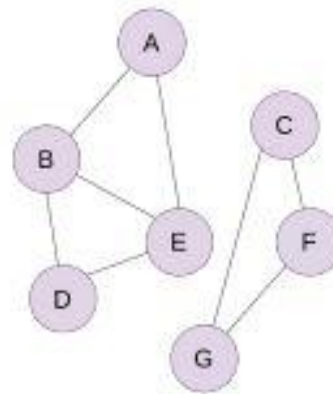
Os grafos podem ter uma ampla variedade de topologias. Eles podem ser densos, com muitas conexões entre os nós, ou esparsos, com apenas algumas conexões. Eles também podem ter topologias específicas, como árvores, grafos bipartidos, grafos cíclicos ou grafos direcionados acíclicos (DAGs). Essa flexibilidade permite modelar uma variedade de problemas do mundo real de forma eficaz.

Fundamentação teórica

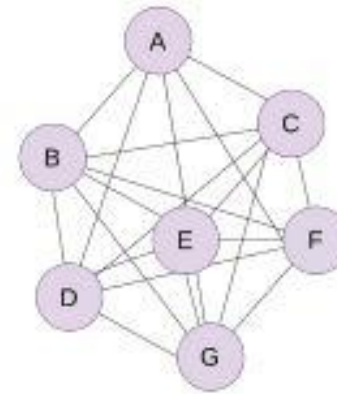
Topologias Variadas



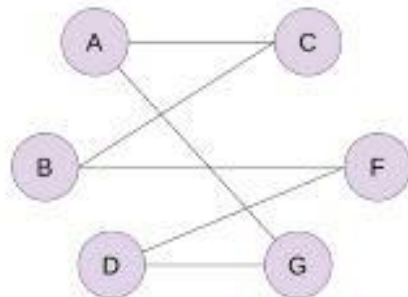
Conexo



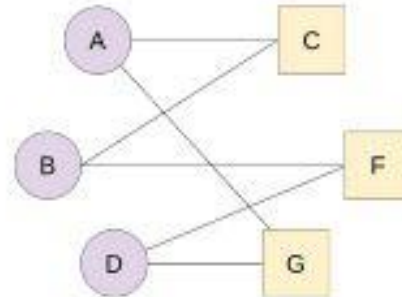
Desconexo



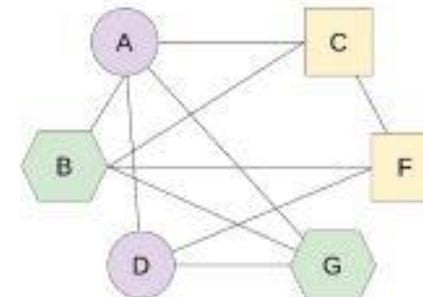
Completo



Homogênea



Heterogêneo e bipartido



Heterogêneo

Fundamentação teórica

Escalabilidade

Os grafos são escaláveis e podem ser usados para representar conjuntos de dados de diferentes tamanhos. Eles podem ser pequenos, com apenas alguns nós e arestas, ou enormes, com milhões ou até bilhões de nós e arestas. Isso os torna adequados para uma ampla gama de aplicações, desde redes sociais e sistemas de recomendação até modelagem de infraestrutura de TI e biologia computacional.

Fundamentação teórica

Algoritmos Específicos

Existem algoritmos específicos para grafos que exploram suas propriedades únicas. Esses algoritmos podem ser usados para realizar uma variedade de tarefas, como encontrar caminhos mais curtos, calcular a centralidade dos nós, identificar comunidades na rede e muito mais. Esses algoritmos são fundamentais para análise e processamento eficiente de dados em grafos.

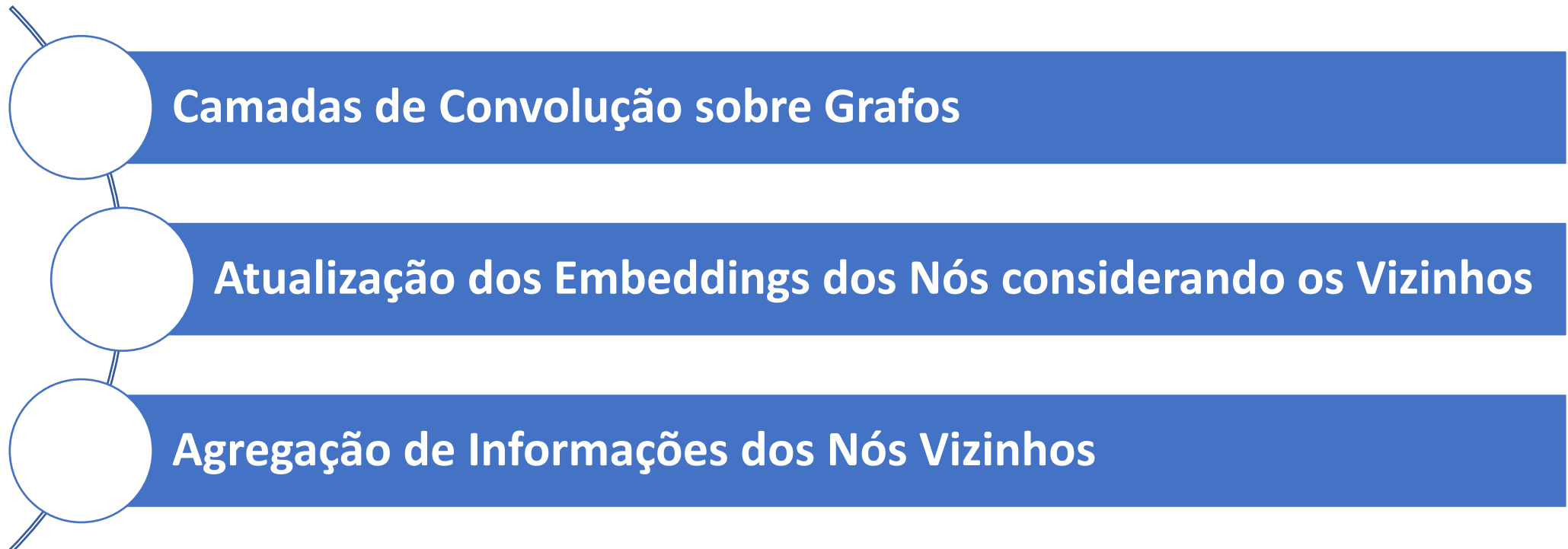
Fundamentação teórica

O **objetivo** das Redes Neurais de Grafos (GNNs - Graph Neural Networks) é aprender representações eficazes dos nós e arestas em um grafo. Isso é alcançado através da incorporação de informações locais e globais presentes na estrutura do grafo.

As GNNs são uma classe de modelos de aprendizado de máquina projetados especificamente para lidar com dados estruturados na forma de grafos.

Arquitetura e funcionamento

Uma Rede Neural de Grafos (GNN) é uma arquitetura de rede neural projetada para trabalhar com dados representados na forma de grafos. Conceito básica da arquitetura de uma GNN:



Arquitetura e funcionamento

Camadas de Convolução sobre Grafos

Assim como em redes neurais convolucionais (CNNs) para dados de imagem, as GNNs também possuem camadas de convolução. No entanto, em vez de operar em grades de pixels, as convoluções em GNNs são aplicadas diretamente aos nós e arestas do grafo. Cada nó no grafo tem um vetor de características associado (ou seja, um embedding), e as operações de convolução são aplicadas a esses embeddings para aprender representações mais ricas e informativas.

Arquitetura e funcionamento

Atualização dos Embeddings dos Nós considerando os Vizinhos

Uma característica central das GNNs é a maneira como elas atualizam os embeddings dos nós levando em consideração as informações dos nós vizinhos. Em cada camada da GNN, os embeddings dos nós são atualizados agregando as informações dos embeddings de seus vizinhos. Essa agregação pode ser realizada de várias maneiras, como média, soma ou concatenação dos embeddings dos vizinhos.

Arquitetura e funcionamento

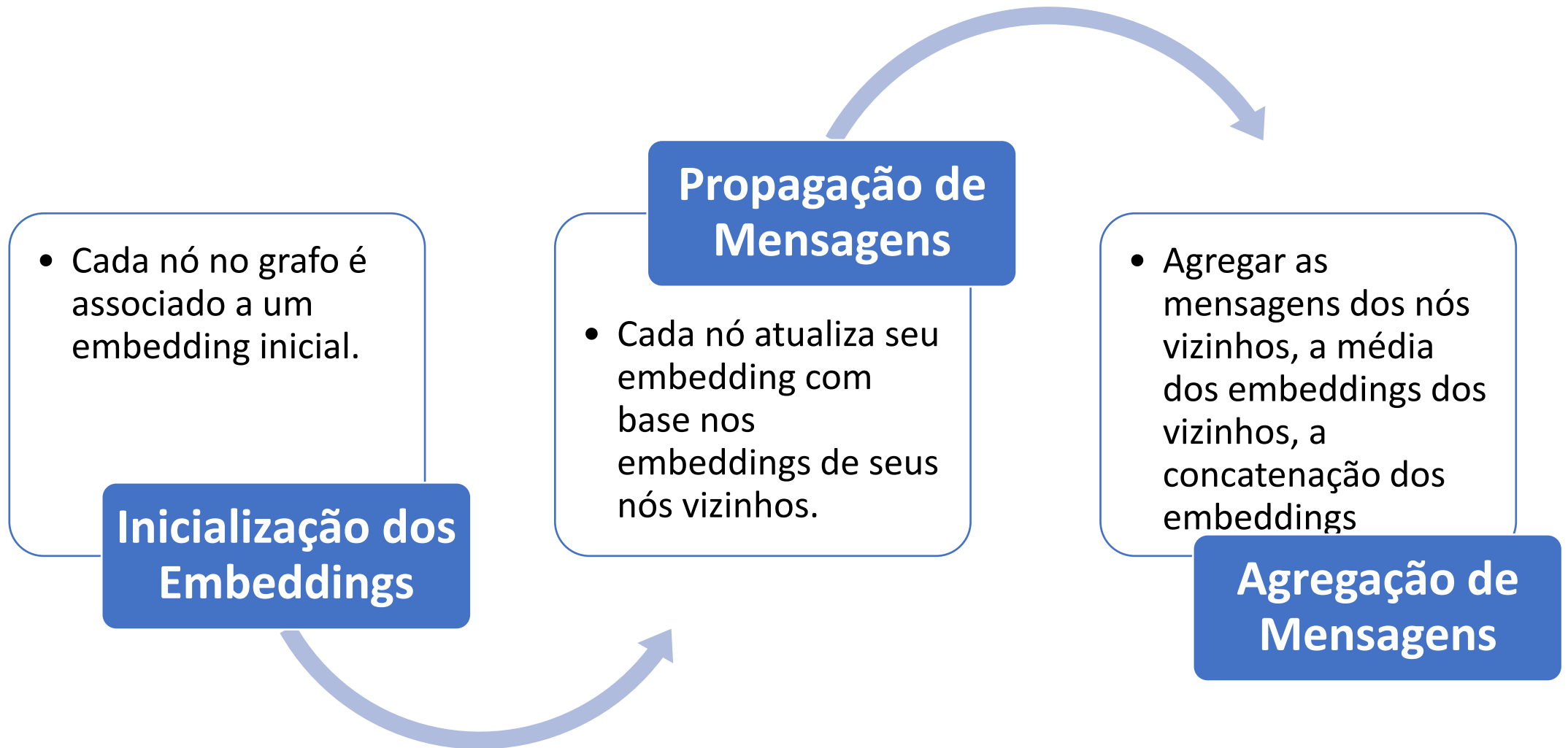
Agregação de Informações dos Nós Vizinhos

Esta etapa é crucial para permitir que os nós do grafo troquem informações entre si. Depois que as informações dos vizinhos são agregadas, o resultado é combinado com o embedding do nó atual para produzir uma nova representação do nó. Essa nova representação contém informações tanto locais quanto globais, capturando assim o contexto do nó dentro do grafo.

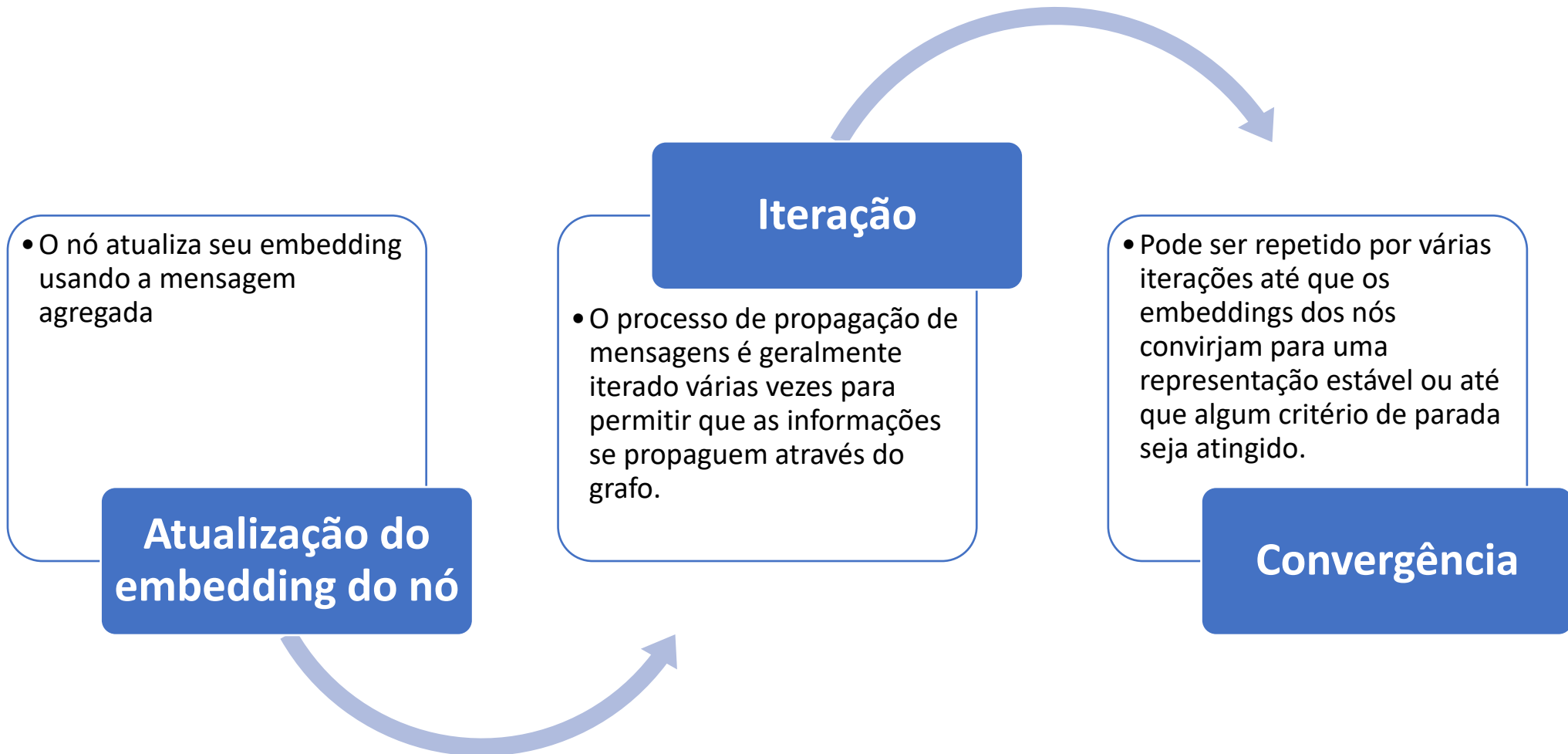
Arquitetura e funcionamento

O **processo** de propagação de mensagens, também conhecido como passagem de mensagens ou message passing, é uma técnica fundamental em diversos campos da ciência da computação e da inteligência artificial, como aprendizado de máquina em grafos, processamento de linguagem natural e visão computacional. Ele é usado para atualizar os embeddings (representações vetoriais) dos nós em um grafo, com base nas informações dos nós vizinhos.

Arquitetura e funcionamento



Arquitetura e funcionamento



Treinamento e otimização

O **treinamento** de Graph Neural Networks (GNNs) pode ser realizado tanto por métodos supervisionados quanto por métodos não supervisionados.

Treinamento e otimização

- **Supervisionado:** No treinamento supervisionado de GNNs, os nós ou arestas do grafo têm rótulos associados que representam as classes ou as informações que se deseja prever. O objetivo é aprender uma representação dos nós ou das arestas que seja útil para prever esses rótulos. O treinamento supervisionado geralmente envolve a minimização de alguma função de perda que compara as previsões do modelo com os rótulos verdadeiros.

Treinamento e otimização

- **Não supervisionado:** No treinamento não supervisionado de GNNs, não há rótulos associados aos nós ou arestas do grafo. Em vez disso, o objetivo é aprender representações úteis dos nós ou das arestas do grafo sem a necessidade de rótulos explícitos. Isso pode ser alcançado através de técnicas como autoencoders, onde o modelo é treinado para reconstruir o próprio grafo de entrada, ou através de métodos de agrupamento (clustering), onde o objetivo é agrupar nós semelhantes.

Treinamento e otimização

Utilização de algoritmos de **otimização**, como gradiente descendente estocástico, para ajustar os parâmetros do modelo.

Treinamento e otimização

- O uso de algoritmos de otimização, como o gradiente descendente estocástico (SGD - Stochastic Gradient Descent). Esses algoritmos são usados para ajustar os parâmetros de um modelo de modo a minimizar uma função de perda, que representa a discrepância entre as previsões do modelo e os valores reais dos dados.
- O gradiente descendente estocástico é uma versão estocástica do gradiente descendente clássico.

Treinamento e otimização

1. Inicialização: Os parâmetros do modelo são inicializados aleatoriamente ou com valores pré-definidos.

2. Loop de Iteração:

- 1. Seleção de Mini-batch:** Uma pequena amostra (mini-batch) de dados é selecionada aleatoriamente do conjunto de dados completo.
- 2. Cálculo do Gradiente:** O gradiente da função de perda em relação aos parâmetros é calculado usando os dados do mini-batch.
- 3. Atualização dos Parâmetros:** Os parâmetros do modelo são atualizados na direção oposta do gradiente, multiplicado por uma taxa de aprendizado (learning rate).

3. Critério de Parada: O processo de iteração continua até que um critério de parada seja atendido, como um número fixo de iterações, convergência da função de perda, ou outra condição definida pelo usuário.

Treinamento e otimização

- As Redes Neurais com Grafos (GNNs) apresentam **desafios** únicos no treinamento devido à estrutura dos dados em forma de grafos e à natureza das tarefas que elas abordam.

Treinamento e otimização

- **Propagação eficiente de informações:** Em grafos grandes e complexos, é fundamental garantir que as informações se propaguem eficientemente por meio dos nós e arestas. Isso pode ser desafiador devido à natureza dispersa e irregular dos grafos. Estratégias como agregação de vizinhos, amostragem de vizinhos ou técnicas de pooling podem ser usadas para lidar com esse desafio.

Treinamento e otimização

- **Prevenção de problemas de overfitting:** Assim como em outras redes neurais, as GNNs podem sofrer de overfitting, especialmente em conjuntos de dados pequenos ou com grafos irregulares. Estratégias como regularização, dropout, aumento de dados e validação cruzada são comumente usadas para mitigar esse problema.

Treinamento e otimização

- **Captura de informações de longo alcance:** Em grafos grandes, é importante que as GNNs possam capturar informações de longo alcance para realizar tarefas complexas, como classificação de nós ou grafos. No entanto, a propagação eficiente de informações muitas vezes favorece conexões de curto alcance. Arquiteturas de GNNs mais avançadas, como redes recorrentes ou mecanismos de atenção, podem ser usadas para lidar com esse desafio.

Treinamento e otimização

- **Agrupamento e agregação de informações:** A maneira como as informações são agrupadas e agregadas de vizinhos em um grafo pode afetar significativamente o desempenho da GNN. Estratégias como agregação média, soma, máximo ou mecanismos de atenção podem ser usadas, e a escolha correta depende da natureza da tarefa e da estrutura do grafo.

Treinamento e otimização

- **Generalização para grafos diferentes:** Uma GNN bem treinada deve ser capaz de generalizar para grafos diferentes dos observados durante o treinamento. Isso pode ser desafiador devido à variabilidade na estrutura e nas características dos grafos. O uso de técnicas de regularização e arquiteturas robustas pode ajudar a melhorar a capacidade de generalização da GNN.

Vantagens e desvantagens

Vantagens

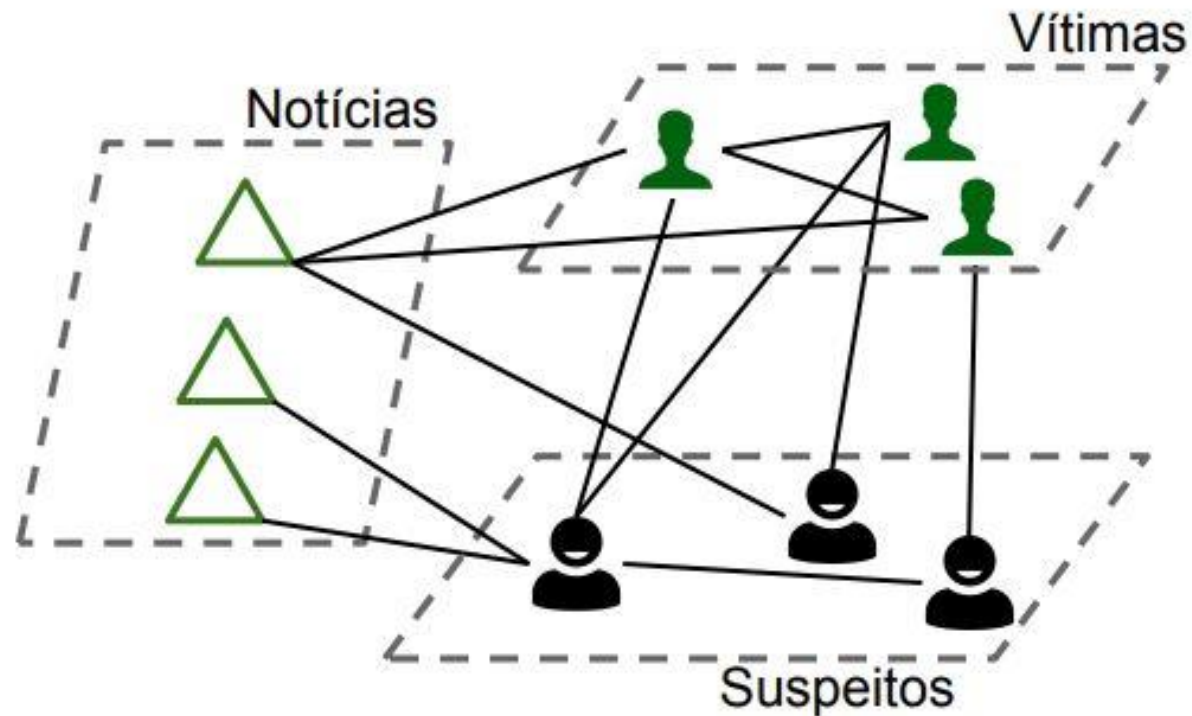
- Capazes de lidar com dados estruturados em forma de grafos.
- Capturam informações locais e globais de maneira eficiente.
- Aplicáveis em uma variedade de domínios, incluindo redes sociais, química computacional e sistemas de recomendação.

Vantagens e desvantagens

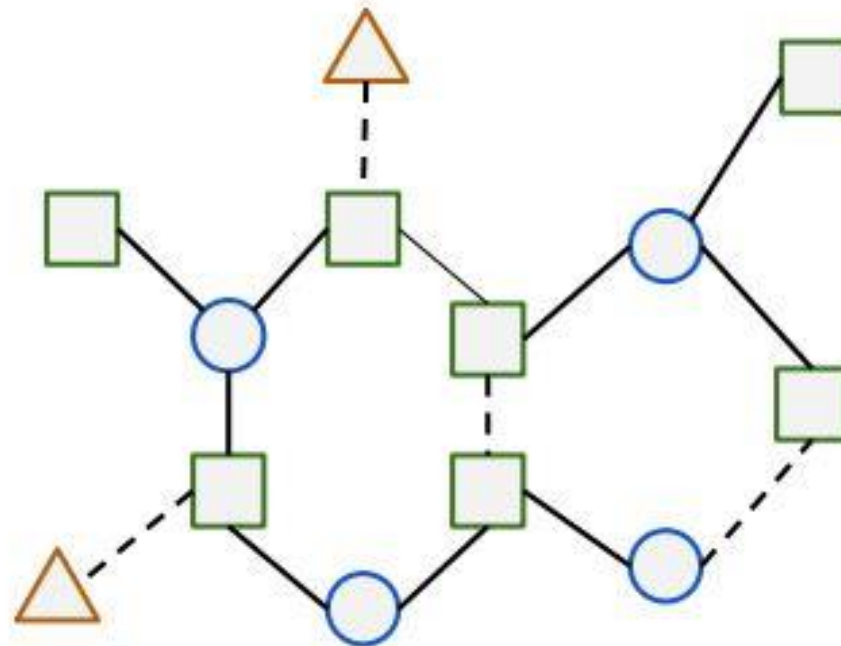
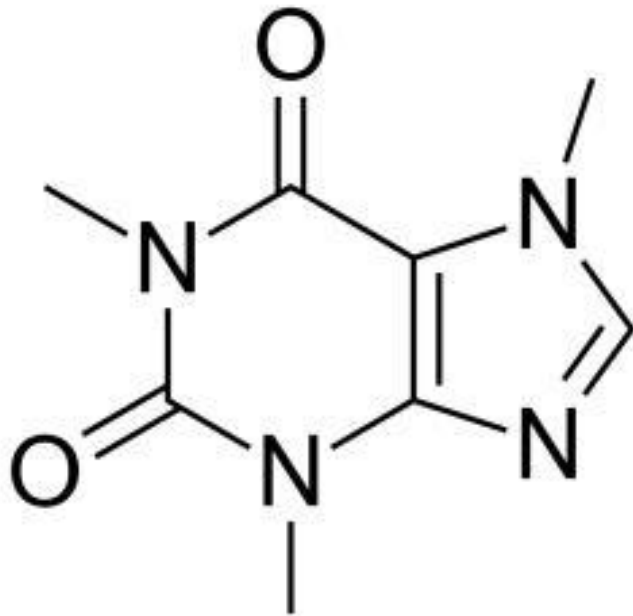
Desvantagens

- Requerem grande poder computacional, especialmente para grafos grandes.
- Dificuldade na interpretação dos resultados devido à complexidade dos modelos.
- Sensíveis à qualidade e à representação dos dados de entrada.

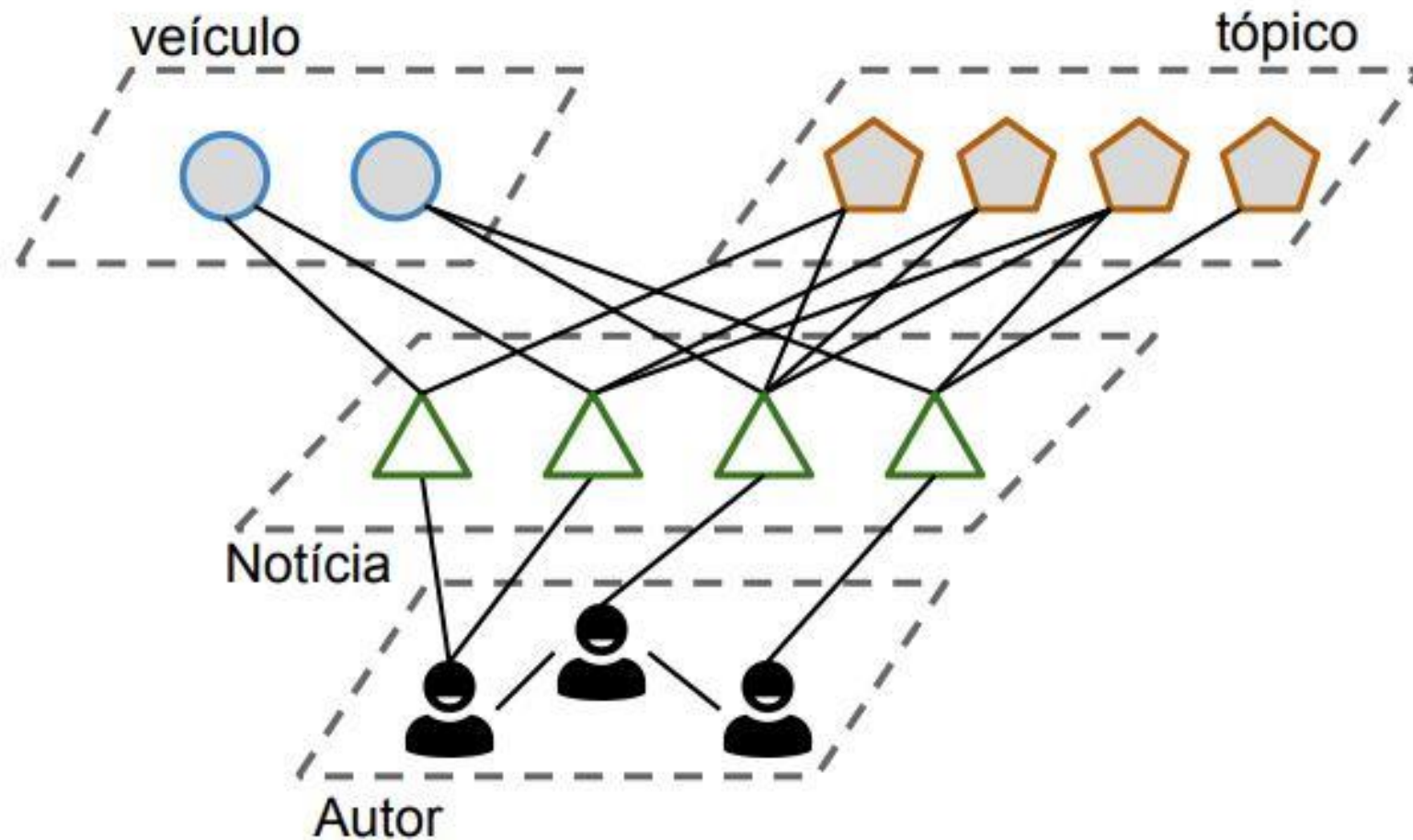
Exemplo(s) de aplicação (Detecção de Crimes)



Exemplo(s) de aplicação (Classificação de Grafos)



Exemplo(s) de aplicação (Detecção Notícias Falsas)



Comparação com outros algoritmos

Redes Neurais Convolucionais (CNNs) e Redes Neurais Recorrentes (RNNs):

- GNNs são projetadas para dados estruturados em forma de gráfico, enquanto CNNs são mais adequadas para dados com grade (como imagens) e RNNs para dados sequenciais (como texto ou séries temporais).
- GNNs capturam informações sobre a estrutura do grafo (conexões entre nós) durante o processo de aprendizado, enquanto CNNs e RNNs não têm essa capacidade inata.

Comparação com outros algoritmos

Métodos de Classificação e Regressão Tradicionais:

- GNNs podem superar métodos tradicionais em tarefas de classificação e regressão em grafos, especialmente quando a estrutura do grafo é importante para a tarefa.
- Em tarefas onde os atributos dos nós e arestas em um grafo são informativos, GNNs tendem a superar métodos tradicionais que não levam em consideração a estrutura do grafo.

Comparação com outros algoritmos

Métodos de Processamento de Grafos Tradicionais:

- GNNs são geralmente mais flexíveis e poderosas do que métodos tradicionais de processamento de grafos, como PageRank ou algoritmos de caminho mais curto, porque podem aprender representações de nós e arestas diretamente dos dados.
- No entanto, em algumas circunstâncias específicas ou para certas métricas de desempenho, métodos tradicionais podem ser mais eficientes computacionalmente.

Perguntas?

Referências

- <https://www-sciencedirect.ez26.periodicos.capes.gov.br/science/article/pii/S0957417422011654>

<https://forms.gle/UCyb8dj9Fhsw5BKU7>

Obrigado!