

图神经网络

王希豪

北京大学中国语言文学系

wangxihao@pku.edu.cn

2021 年 6 月 23 日

目录

① 概述

② 基本原理

③ 模型介绍

④ 总结

图表示学习 (Graph Representation Learning)

图可以表示各种元素和元素之间关系构成的复杂系统

- 每个结点 (Node) 表示一个元素
- 每条边 (Edge) 表示两个元素之间的关系

图表示学习 (Graph Representation Learning)

图可以表示各种元素和元素之间关系构成的复杂系统

- 每个结点 (Node) 表示一个元素
- 每条边 (Edge) 表示两个元素之间的关系



图: 社交网络

图表示学习 (Graph Representation Learning)

图可以表示各种元素和元素之间关系构成的复杂系统

- 每个结点 (Node) 表示一个元素
- 每条边 (Edge) 表示两个元素之间的关系

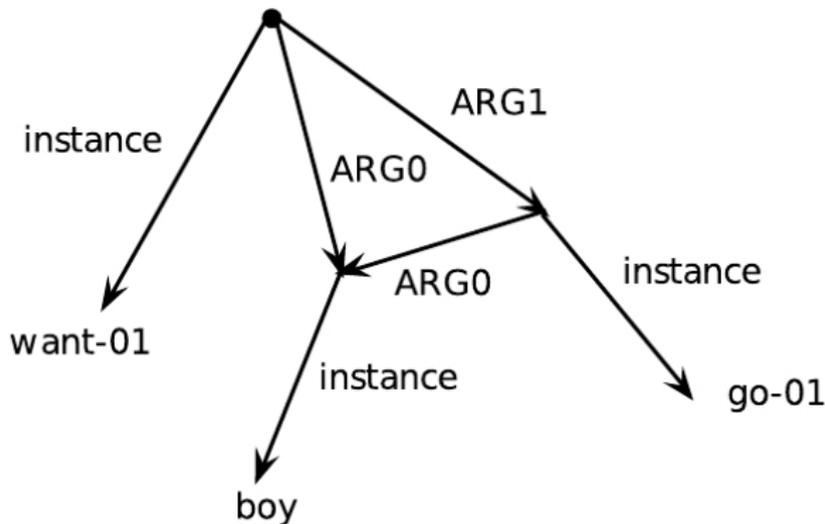
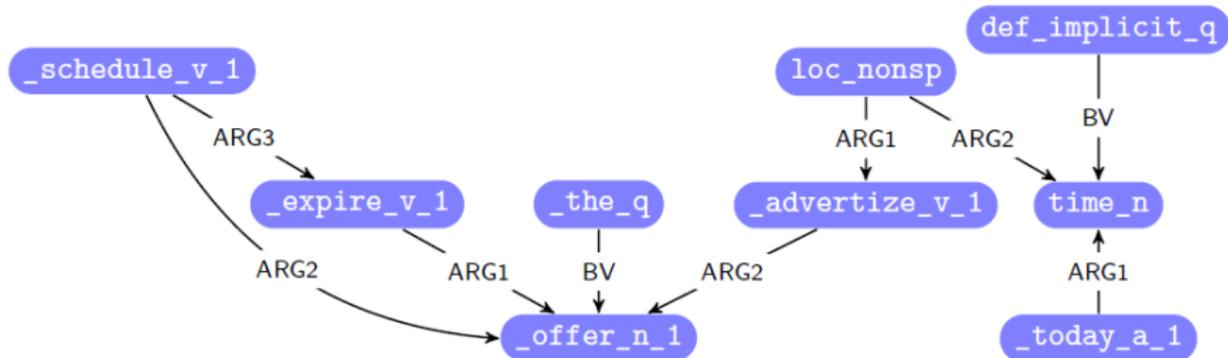


图: Abstract Meaning Representation

图表示学习 (Graph Representation Learning)

图可以表示各种元素和元素之间关系构成的复杂系统

- 每个结点 (Node) 表示一个元素
- 每条边 (Edge) 表示两个元素之间的关系



The offer advertized today is scheduled to expire.

[WSJ #0032002]

图: English Resource Semantics

图表示学习 (Graph Representation Learning)

图可以表示各种元素和元素之间关系构成的复杂系统

- 每个结点 (Node) 表示一个元素
- 每条边 (Edge) 表示两个元素之间的关系

图结构为我们分析现实世界中的复杂系统提供了数学基础

图表示学习 (Graph Representation Learning)

图可以表示各种元素和元素之间关系构成的复杂系统

- 每个结点 (Node) 表示一个元素
- 每条边 (Edge) 表示两个元素之间的关系

图结构为我们分析现实世界中的复杂系统提供了数学基础

图表示学习，是表示学习的分支，研究如何自动地从图结构数据中获取信息、提取特征，并应用到特定的任务当中。

图表示学习 (Graph Representation Learning)

图可以表示各种元素和元素之间关系构成的复杂系统

- 每个结点 (Node) 表示一个元素
- 每条边 (Edge) 表示两个元素之间的关系

图结构为我们分析现实世界中的复杂系统提供了数学基础

图表示学习，是表示学习的分支，研究如何自动地从图结构数据中获取信息、提取特征，并应用到特定的任务当中。

相关任务

- 点分类 (Node classification)
- 关系预测 (Relation prediction)
- 聚类 and 社区检测 (Cluster and community detection)
- 图的分类、回归和聚类 (Graph classification, regression, and clustering)

图表示学习的各种方法 [1]

传统方法

- 点一级的特征
 - 点的度数
 - 向心性
 -

图表示学习的各种方法 [1]

传统方法

- 点一级的特征
 - 点的度数
 - 向心性
 -
- 图一级的特征
 - 点集
 - 路径
 - 核方法
 -

图表示学习的各种方法 [1]

传统方法

- 点一级的特征
 - 点的度数
 - 向心性
 -
- 图一级的特征
 - 点集
 - 路径
 - 核方法
 -
- 近邻重合检测 (Neighborhood Overlap Detection)
 - 局部重合检测 (Local Overlap Detection)
 - 全局重合检测 (Global Overlap Detection)

图表示学习的各种方法 [1]

传统方法

- 点一级的特征
 - 点的度数
 - 向心性
 -
- 图一级的特征
 - 点集
 - 路径
 - 核方法
 -
- 近邻重合检测 (Neighborhood Overlap Detection)
 - 局部重合检测 (Local Overlap Detection)
 - 全局重合检测 (Global Overlap Detection)
- 拉普拉斯变换与谱方法
-

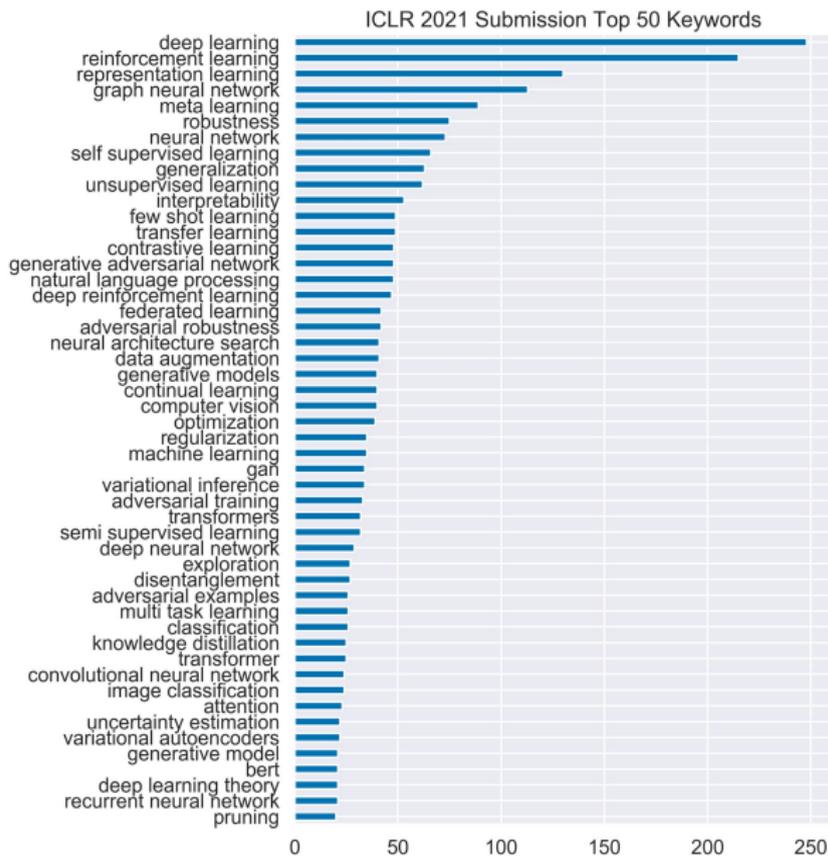
图表示学习的各种方法 [1]

传统方法

- 点一级的特征
 - 点的度数
 - 向心性
 -
- 图一级的特征
 - 点集
 - 路径
 - 核方法
 -
- 近邻重合检测 (Neighborhood Overlap Detection)
 - 局部重合检测 (Local Overlap Detection)
 - 全局重合检测 (Global Overlap Detection)
- 拉普拉斯变换与谱方法
-

新的方法: **图神经网络 (Graph Neural Network)**

图表示学习的各种方法 [1]



目录

- ① 概述
- ② 基本原理
- ③ 模型介绍
- ④ 总结

目标和难点

图神经网络 (Graph Neural Networks, GNN) 的基本目标是

- 基于图的整体结构和特征信息，获取每个结点的向量表示

目标和难点

图神经网络 (Graph Neural Networks, GNN) 的基本目标是

- 基于图的整体结构和特征信息，获取每个结点的向量表示

图神经网络研究的难点：

- 现有神经网络模型不适用
 - 卷积神经网络 (Convolutional Neural Networks, CNNs) \Rightarrow 网格数据
 - 循环神经网络 (Recurrent Neural Networks, RNNs) \Rightarrow 序列数据

目标和难点

图神经网络 (Graph Neural Networks, GNN) 的基本目标是

- 基于图的整体结构和特征信息，获取每个结点的向量表示

图神经网络研究的难点：

- 现有神经网络模型不适用
 - 卷积神经网络 (Convolutional Neural Networks, CNNs) \Rightarrow 网格数据
 - 循环神经网络 (Recurrent Neural Networks, RNNs) \Rightarrow 序列数据

\Rightarrow 需要构建新的神经网络框架

目标和难点

图神经网络 (Graph Neural Networks, GNN) 的基本目标是

- 基于图的整体结构和特征信息，获取每个结点的向量表示

图神经网络研究的难点：

- 现有神经网络模型不适用
 - 卷积神经网络 (Convolutional Neural Networks, CNNs) \Rightarrow 网格数据
 - 循环神经网络 (Recurrent Neural Networks, RNNs) \Rightarrow 序列数据

\Rightarrow 需要构建新的神经网络框架

一个合理的想法

- 直接以邻接矩阵作为输入，构建神经网络模型

置换不变性与等变性

考虑直接将邻接矩阵 \mathbf{A} 的每个列向量 (代表一个结点) 拼接起来, 输入多层感知机 (Multi-layer Perceptron, MLP), 得到整张图的特征向量:

$$\mathbf{z}_G = \text{MLP}(\mathbf{A}[1] \oplus \mathbf{A}[2] \oplus \cdots \oplus \mathbf{A}[|\mathcal{V}|])$$

置换不变性与等变性

考虑直接将邻接矩阵 \mathbf{A} 的每个列向量 (代表一个结点) 拼接起来, 输入多层感知机 (Multi-layer Perceptron, MLP), 得到整张图的特征向量:

$$\mathbf{z}_G = \text{MLP}(\mathbf{A}[1] \oplus \mathbf{A}[2] \oplus \cdots \oplus \mathbf{A}[|\mathcal{V}|])$$

缺点: 依赖点在邻接矩阵中的出现顺序

置换不变性与等变性

考虑直接将邻接矩阵 \mathbf{A} 的每个列向量 (代表一个结点) 拼接起来, 输入多层感知机 (Multi-layer Perceptron, MLP), 得到整张图的特征向量:

$$\mathbf{z}_G = \text{MLP}(\mathbf{A}[1] \oplus \mathbf{A}[2] \oplus \cdots \oplus \mathbf{A}[|\mathcal{V}|])$$

缺点: 依赖点在邻接矩阵中的出现顺序

假设 $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ 是点的特征集合, 则模型 f 需要满足

- 置换不变性 (Permutation invariance): $f(\mathbf{P}\mathbf{X}) = f(\mathbf{X})$
- 或 置换等变性 (Permutation equivariance): $f(\mathbf{P}\mathbf{X}) = \mathbf{P}f(\mathbf{X})$

置换不变性与等变性

考虑直接将邻接矩阵 \mathbf{A} 的每个列向量 (代表一个结点) 拼接起来, 输入多层感知机 (Multi-layer Perceptron, MLP), 得到整张图的特征向量:

$$\mathbf{z}_G = \text{MLP}(\mathbf{A}[1] \oplus \mathbf{A}[2] \oplus \cdots \oplus \mathbf{A}[|\mathcal{V}|])$$

缺点: 依赖点在邻接矩阵中的出现顺序

假设 $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$ 是点的特征集合, 则模型 f 需要满足

- 置换不变性 (Permutation invariance): $f(\mathbf{P}\mathbf{X}) = f(\mathbf{X})$
- 或 置换等变性 (Permutation equivariance): $f(\mathbf{P}\mathbf{X}) = \mathbf{P}f(\mathbf{X})$

扩展至边集, 则模型 f 需要满足

- 置换不变性 (Permutation invariance): $f(\mathbf{P}\mathbf{A}\mathbf{P}^{-1}) = f(\mathbf{A})$
- 或 置换等变性 (Permutation equivariance): $f(\mathbf{P}\mathbf{A}\mathbf{P}^{-1}) = \mathbf{P}f(\mathbf{A})$

置换不变性与等变性

假设 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ 是点的特征集合, 则模型 f 需要满足

- 置换不变性 (Permutation invariance): $f(\mathbf{P}\mathbf{X}) = f(\mathbf{X})$
- 或 置换等变性 (Permutation equivariance): $f(\mathbf{P}\mathbf{X}) = \mathbf{P}f(\mathbf{X})$

扩展至边集, 则模型 f 需要满足

- 置换不变性 (Permutation invariance): $f(\mathbf{P}\mathbf{A}\mathbf{P}^{-1}) = f(\mathbf{A})$
- 或 置换等变性 (Permutation equivariance): $f(\mathbf{P}\mathbf{A}\mathbf{P}^{-1}) = \mathbf{P}f(\mathbf{A})$

Zaheer et al. [2]证明了, 形式如下的模型 f 满足置换不变性

$$f(\mathbf{X}) = \phi\left(\bigoplus_{i \in \mathcal{V}} \psi(\mathbf{x}_i)\right)$$

目录

- ① 概述
- ② 基本原理
- ③ 模型介绍**
- ④ 总结

神经信息传递 (Neural Message Passing)

信息传递框架

- 每个结点 u 都有一个隐向量 $\mathbf{h}_u^{(k)}$
- 结点 u 根据其所有相邻结点 $\mathcal{N}(u)$ 的隐向量更新自身的隐向量

神经信息传递 (Neural Message Passing)

信息传递框架

- 每个结点 u 都有一个隐向量 $\mathbf{h}_u^{(k)}$
- 结点 u 根据其所有相邻结点 $\mathcal{N}(u)$ 的隐向量更新自身的隐向量

$$\begin{aligned}\mathbf{h}_u^{(k)} &= \phi^{(k)}\left(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)}\right) \\ &= \phi^{(k)}\left(\mathbf{h}_u^{(k-1)}, \bigoplus_{v \in \mathcal{N}(u)} \psi^{(k)}(\mathbf{h}_v^{(k-1)})\right)\end{aligned}$$

神经信息传递 (Neural Message Passing)

信息传递框架

- 每个结点 u 都有一个隐向量 $\mathbf{h}_u^{(k)}$
- 结点 u 根据其所有相邻结点 $\mathcal{N}(u)$ 的隐向量更新自身的隐向量

$$\begin{aligned}\mathbf{h}_u^{(k)} &= \phi^{(k)}\left(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)}\right) \\ &= \phi^{(k)}\left(\mathbf{h}_u^{(k-1)}, \bigoplus_{v \in \mathcal{N}(u)} \psi^{(k)}(\mathbf{h}_v^{(k-1)})\right)\end{aligned}$$

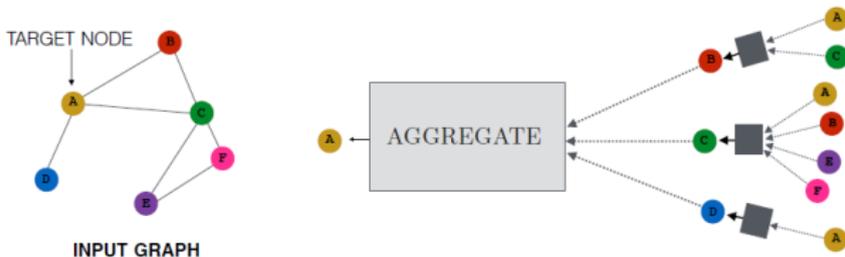


图: 信息传递图示

神经信息传递 (Neural Message Passing)

信息传递框架

- 每个结点 u 都有一个隐向量 $\mathbf{h}_u^{(k)}$
- 结点 u 根据其所有相邻结点 $\mathcal{N}(u)$ 的隐向量更新自身的隐向量

$$\begin{aligned}\mathbf{h}_u^{(k)} &= \phi^{(k)}\left(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)}\right) \\ &= \phi^{(k)}\left(\mathbf{h}_u^{(k-1)}, \bigoplus_{v \in \mathcal{N}(u)} \psi^{(k)}(\mathbf{h}_v^{(k-1)})\right)\end{aligned}$$

基础模型

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}_{\text{self}}^{(k)}\mathbf{h}_u^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)} \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{(k-1)} + \mathbf{b}^{(k)}\right)$$

神经信息传递 (Neural Message Passing)

信息传递框架

- 每个结点 u 都有一个隐向量 $\mathbf{h}_u^{(k)}$
- 结点 u 根据其所有相邻结点 $\mathcal{N}(u)$ 的隐向量更新自身的隐向量

$$\begin{aligned}\mathbf{h}_u^{(k)} &= \phi^{(k)}\left(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)}\right) \\ &= \phi^{(k)}\left(\mathbf{h}_u^{(k-1)}, \bigoplus_{v \in \mathcal{N}(u)} \psi^{(k)}(\mathbf{h}_v^{(k-1)})\right)\end{aligned}$$

基础模型

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}_{\text{self}}^{(k)}\mathbf{h}_u^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)} \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{(k-1)} + \mathbf{b}^{(k)}\right)$$

自环模型

$$\mathbf{H}^{(k)} = \sigma\left((\mathbf{A} + \mathbf{I})\mathbf{H}^{(k-1)}\mathbf{W}^{(k)}\right)$$

聚合函数 (Aggregate function)

近邻标准化 (Neighborhood Normalization)

$$\mathbf{m}_{\mathcal{N}(u)} = \frac{\sum_{v \in \mathcal{N}(u)} \mathbf{h}_v}{|\mathcal{N}(u)|}$$

聚合函数 (Aggregate function)

近邻标准化 (Neighborhood Normalization)

$$\mathbf{m}_{\mathcal{N}(u)} = \frac{\sum_{v \in \mathcal{N}(u)} \mathbf{h}_v}{|\mathcal{N}(u)|}$$

图卷积网络 (Graph convolutional networks, GCNs)

$$\mathbf{m}_{\mathcal{N}(u)} = \sum_{v \in \mathcal{N}(u)} \frac{\mathbf{h}_v}{\sqrt{|\mathcal{N}(u)| |\mathcal{N}(v)|}}$$

聚合函数 (Aggregate function)

近邻标准化 (Neighborhood Normalization)

$$\mathbf{m}_{\mathcal{N}(u)} = \frac{\sum_{v \in \mathcal{N}(u)} \mathbf{h}_v}{|\mathcal{N}(u)|}$$

图卷积网络 (Graph convolutional networks, GCNs)

$$\mathbf{m}_{\mathcal{N}(u)} = \sum_{v \in \mathcal{N}(u)} \frac{\mathbf{h}_v}{\sqrt{|\mathcal{N}(u)| |\mathcal{N}(v)|}}$$

池化 (Pooling)

$$\mathbf{m}_{\mathcal{N}(u)} = \text{MLP}_{\theta} \left(\sum_{v \in \mathcal{N}(u)} \text{MLP}_{\phi}(\mathbf{h}_v) \right)$$

聚合函数 (Aggregate function)

近邻标准化 (Neighborhood Normalization)

$$\mathbf{m}_{\mathcal{N}(u)} = \frac{\sum_{v \in \mathcal{N}(u)} \mathbf{h}_v}{|\mathcal{N}(u)|}$$

图卷积网络 (Graph convolutional networks, GCNs)

$$\mathbf{m}_{\mathcal{N}(u)} = \sum_{v \in \mathcal{N}(u)} \frac{\mathbf{h}_v}{\sqrt{|\mathcal{N}(u)| |\mathcal{N}(v)|}}$$

池化 (Pooling)

$$\mathbf{m}_{\mathcal{N}(u)} = \text{MLP}_{\theta} \left(\sum_{v \in \mathcal{N}(u)} \text{MLP}_{\phi}(\mathbf{h}_v) \right)$$

近邻注意力 (Neighborhood Attention)

$$\alpha_{u,v} = \frac{\exp(\mathbf{a}^{\top} [\mathbf{W}\mathbf{h}_u \oplus \mathbf{W}\mathbf{h}_v])}{\sum_{v' \in \mathcal{N}(u)} \exp(\mathbf{a}^{\top} [\mathbf{W}\mathbf{h}_u \oplus \mathbf{W}\mathbf{h}_{v'}])}$$

更新函数 (Update Function)

过平滑 (Over-smoothing)

- 在经过多轮迭代之后，不同结点的隐向量会趋于相同

更新函数 (Update Function)

过平滑 (Over-smoothing)

- 在经过多轮迭代之后，不同结点的隐向量会趋于相同

拼接 (Concatenation)

$$\text{UPDATE}_{\text{concat}}(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}) = \text{UPDATE}_{\text{base}}(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}) \oplus \mathbf{h}_u$$

更新函数 (Update Function)

过平滑 (Over-smoothing)

- 在经过多轮迭代之后，不同结点的隐向量会趋于相同

拼接 (Concatenation)

$$\text{UPDATE}_{\text{concat}}(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}) = \text{UPDATE}_{\text{base}}(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}) \oplus \mathbf{h}_u$$

门控更新 (Gated Update)

$$\mathbf{h}_u^{(k)} = \text{GRU}(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)})$$

更新函数 (Update Function)

过平滑 (Over-smoothing)

- 在经过多轮迭代之后，不同结点的隐向量会趋于相同

拼接 (Concatenation)

$$\text{UPDATE}_{\text{concat}}(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}) = \text{UPDATE}_{\text{base}}(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}) \oplus \mathbf{h}_u$$

门控更新 (Gated Update)

$$\mathbf{h}_u^{(k)} = \text{GRU}(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)})$$

跳跃式知识连接 (Jumping Knowledge Connections)

$$\mathbf{z}_u = f_{\text{JK}}(\mathbf{h}_u^{(0)} \oplus \mathbf{h}_u^{(1)} \oplus \dots \oplus \mathbf{h}_u^{(K)})$$

更新函数 (Update Function)

过平滑 (Over-smoothing)

- 在经过多轮迭代之后，不同结点的隐向量会趋于相同

拼接 (Concatenation)

$$\text{UPDATE}_{\text{concat}}(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}) = \text{UPDATE}_{\text{base}}(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}) \oplus \mathbf{h}_u$$

门控更新 (Gated Update)

$$\mathbf{h}_u^{(k)} = \text{GRU}(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)})$$

跳跃式知识连接 (Jumping Knowledge Connections)

$$\mathbf{z}_u = f_{\text{JK}}(\mathbf{h}_u^{(0)} \oplus \mathbf{h}_u^{(1)} \oplus \dots \oplus \mathbf{h}_u^{(K)})$$

抽样 (Sampling)

图池化 (Graph Pooling)

基础方法

$$\mathbf{z}_G = \frac{\sum_{v \in \mathcal{V}} \mathbf{z}_v}{f_n(|\mathcal{V}|)}$$

图池化 (Graph Pooling)

基础方法

$$\mathbf{z}_G = \frac{\sum_{v \in \mathcal{V}} \mathbf{z}_v}{f_n(|\mathcal{V}|)}$$

LSTM+attention

$$\mathbf{q}_t = \text{LSTM}(\mathbf{o}_{t-1}, \mathbf{q}_{t-1})$$

$$\mathbf{e}_{v,t} = f_a(\mathbf{z}_v, \mathbf{q}_t), \forall v \in \mathcal{V}$$

$$a_{v,t} = \frac{\exp(\mathbf{e}_{v,t})}{\sum_{u \in \mathcal{V}} \exp(\mathbf{e}_{u,t})}, \forall v \in \mathcal{V}$$

$$\mathbf{o}_t = \sum_{v \in \mathcal{V}} a_{v,t} \mathbf{z}_v$$

图池化 (Graph Pooling)

基础方法

$$\mathbf{z}_G = \frac{\sum_{v \in \mathcal{V}} \mathbf{z}_v}{f_n(|\mathcal{V}|)}$$

LSTM+attention

$$\mathbf{q}_t = \text{LSTM}(\mathbf{o}_{t-1}, \mathbf{q}_{t-1})$$

$$e_{v,t} = f_a(\mathbf{z}_v, \mathbf{q}_t), \forall v \in \mathcal{V}$$

$$a_{v,t} = \frac{\exp(e_{v,t})}{\sum_{u \in \mathcal{V}} \exp(e_{u,t})}, \forall v \in \mathcal{V}$$

$$\mathbf{o}_t = \sum_{v \in \mathcal{V}} a_{v,t} \mathbf{z}_v$$

图聚类 (Graph Clustering)

- 为每个类计算特征，再利用前述方法计算整体特征

目录

- ① 概述
- ② 基本原理
- ③ 模型介绍
- ④ 总结

总结

总结

- 图神经网络在许多领域都有重要的作用
 - 推荐系统
 - 化学反应预测
 - 交通状况预测
 - 知识图谱与问答系统
 - 文本分类
 - 自然语言生成
 -

总结

- 图神经网络在许多领域都有重要的作用
- 图神经网络有坚实的理论基础

总结

- 图神经网络在许多领域都有重要的作用
- 图神经网络有坚实的理论基础
- 目前的图神经网络主要基于“置换不变性”

总结

- 图神经网络在许多领域都有重要的作用
- 图神经网络有坚实的理论基础
- 目前的图神经网络主要基于“置换不变性”
- 图神经网络主要提取的还是图的结构信息

总结

- 图神经网络在许多领域都有重要的作用
- 图神经网络有坚实的理论基础
- 目前的图神经网络主要基于“置换不变性”
- 图神经网络主要提取的还是图的结构信息

未来工作

- 进一步改进图神经网络模型
- 加强图神经网络在自然语言处理任务中的应用
- 构建适合运用图神经网络技术的语言资源

- [1] William L. Hamilton. Graph representation learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 14(3):1–159.
- [2] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets, 2018.