飞桨

图神经网络七日打卡营

小斯妹　百度PGL团队成员

2020.11.25

昨晚课程内容的问题

陈轻歌 1小时前

deepwalk相关问题

deepwalk论文中用的加速方法是层次的softmax，而不是用的负采样吧。。。
这两种加速方法，会影响最终模型的性能吗?
毕竟是讲论文，还是尊重一下作者在论文中的表述吧。。。

赞 回复

Desmon 老师 1小时前

因为实际情况中，我们会选择用负采样的方式优化（可以看到后续的论文基本都是采用负采样的方法）。所以我们选择讲解负采样这种方式，对于大家来说比较好理解，同时实现起来也方便一些。另外，我们在讲述游走模型时，其实关注点是放在如何游走上面哈~ 晚上我会跟大家说一下，我们是有选择的对论文内容进行筛选讲解。

赞 回复 删除

昨晚的作业

AIStudio 红白黑： https://aistudio.baidu.com/aistudio/projectdetail/1265981?shared=1

红白黑

1小时前

图网络基本概念复习总结与思考，以及相关作业代码的详细注解（逐行解释，上下文联系，容易理解），算注册

路理解

或者博

图网络

今晚是

1

Notebook　文件(5)　«

一、图基础及分类

1.1 什么是图?

1.2 图具有的信息/属性

1.3 图节点的概念补充--度、出度与入度

1.4 概念理解

1.5 图的基本分类

二、同构图与异构图

2.1 同构图基本定义

*2.2 异构图的定义

*2.3 异构图的补充概念(图元（子图))

2.4 同构图与异构图的数学表示

/1260103

1　回复

3.1 Node2Vec采样算法转换代码注解

1. 这部分代码，用于随机游走后得到的路径，然后对这些路径进行吸收学习，训练图结构

In [16]

```
%%writefile userdef_sample.py
import numpy as np

# 随机节点的获取
def node2vec_sample(succ, prev_succ, prev_node, p, q):

    """
    输入：succ – 当前节点的下一个相邻节点id列表 list (num_neighbors,)
        prev_succ – 前一个节点的下一个相邻节点id列表 list (num_neighbors,)
        prev_node – 前一个节点id int
        p – 控制回到上一节点的概率 float
        q – 控制偏向DFS还是BFS float
    输出：下一个节点id int
    """
```

多看看课程讨论区

课程大纲

第一课：图学习初印象

- 图学习概述、入门路线
- 实践：环境搭建

第二课：游走类算法

- DeepWalk, node2vec, metapath2vec
- 实践：DeepWalk

第三课：图神经网络算法(一)

- GCN, GAT
- 实践：GCN, GAT

第四课：图神经网络算法(二)

- 图采样、邻居聚合
- 实践：GraphSage

第五课：GNN 进阶

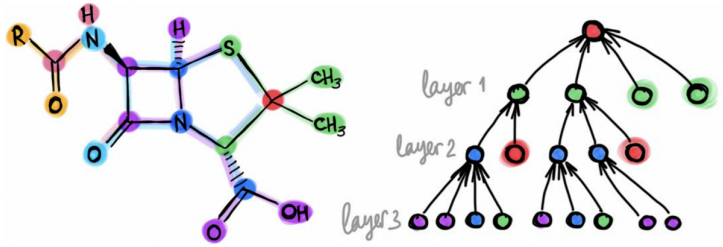
- ERNIESage, UniMP
- 实践：有趣的应用

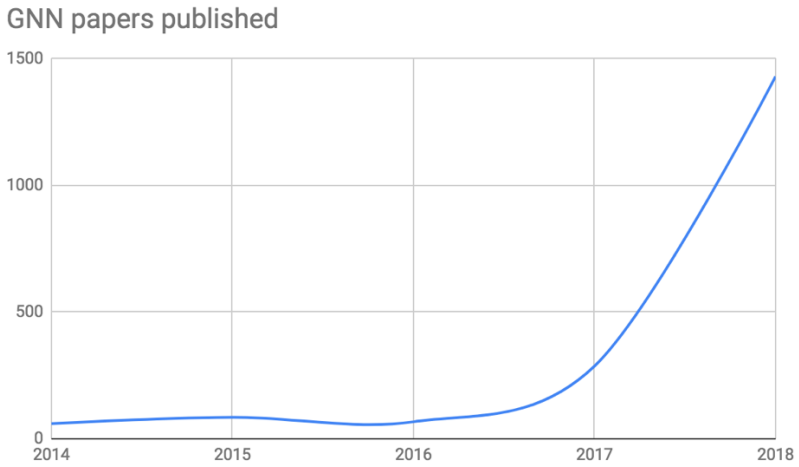
后续：新冠项目实战，带你助力疫情防控

参考材料：

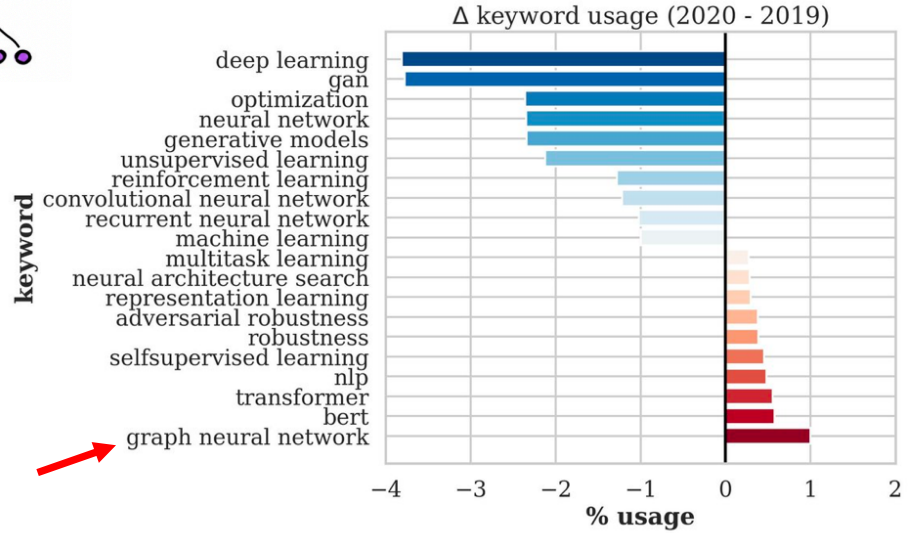
- 斯坦福CS224W课程： http://cs224w.stanford.edu
- 图学习库 PGL：https://github.com/PaddlePaddle/PGL

图神经网络的兴起：业界论文发表

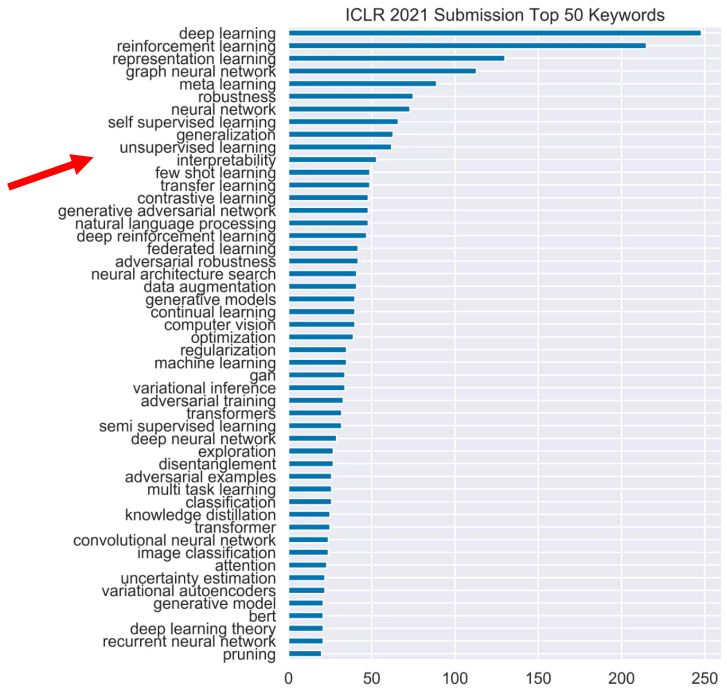




图神经网络发表论文数（2014-2018）

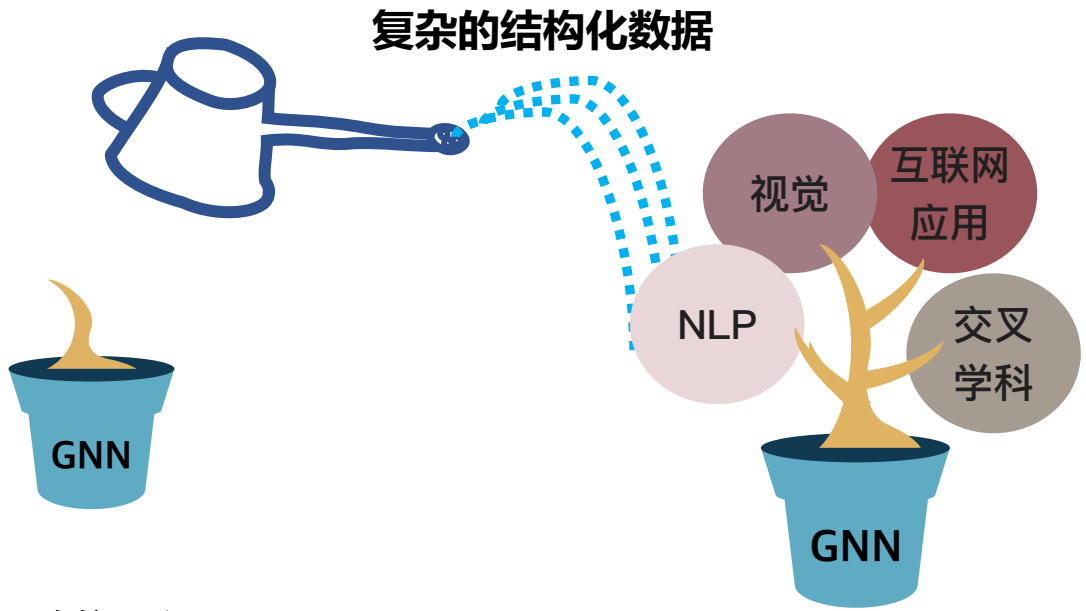


ICLR 2020 提交论文情况



ICLR 2021 提交Top50关键词

图神经网络的兴起：各领域的渗透

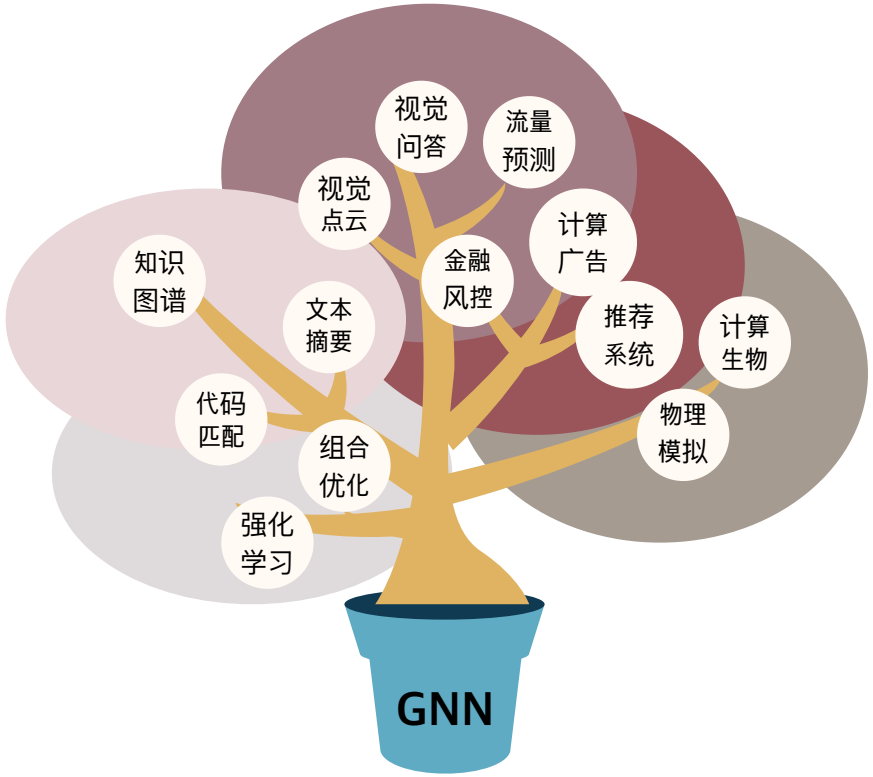


图网络结构雏形

- 涌现不同的图网络如GCN、GAT
- 提供了建模复杂结构的工具

图网络领域初探

- 在不同的领域引入复杂结构化信息，而信息增益带来收益
- 可迁移性，图网络可以统一不同领域的模型。



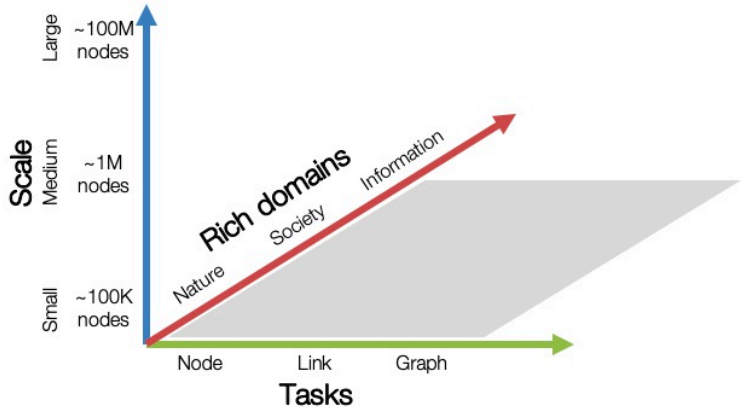
图网络应用爆炸

- 全新的解决方案：解决交叉学科的问题，生物计算、药物发现、粒子运动模拟
- 大量落地推荐系统、搜索广告、金融风控

图神经网络的兴起：权威榜单出现

OPEN GRAPH BENCHMARK

- Jure Leskovec、Yoshua Bengio、Will Hamilton、Max Welling等学者提出



图神经网络领域目前最**权威**的评测榜单：

① 规模大：最大为10亿级

② 领域多：Nature/Society/Information

③ 任务多：Node/Link/Graph

https://ogb.stanford.edu/

https://github.com/graphdeeplearning/benchmarking-gnns

Benchmarking Graph Neural Networks

Vijay Prakash Dwivedi[1]* vijaypra001@e.ntu.edu.sg

Chaitanya K. Joshi[1]* chaitanya.joshi@ntu.edu.sg

Thomas Laurent[2] tlaurent@lmu.edu

Yoshua Bengio[3,4] yoshua.bengio@mila.quebec

Xavier Bresson[1] xbresson@ntu.edu.sg

[1] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2] Department of Mathematics, Loyola Marymount University
[3] Mila, University of Montréal [4] CIFAR

http://keg.cs.tsinghua.edu.cn/cogdl/leaderboard.html

Leaderboard

CogDL provides several downstream tasks including node classification (with or without node attributes), link prediction (with or without attributes, heterogeneous or not). These leaderboards maintain state-of-the-art results and benchmarks on these tasks.

NODE CLASSIFICATION

LINK PREDICTION

GRAPH CLASSIFICATION

飞桨

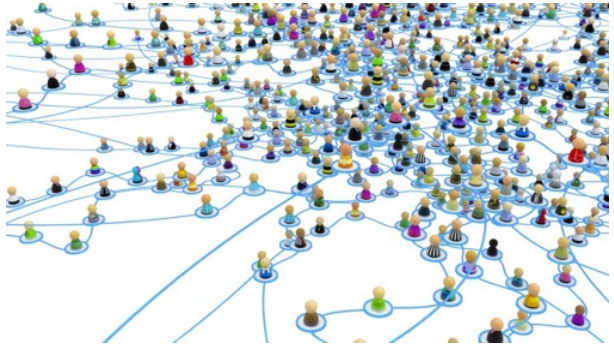
第三课 图神经网络算法(一)

小斯妹 百度PGL团队成员

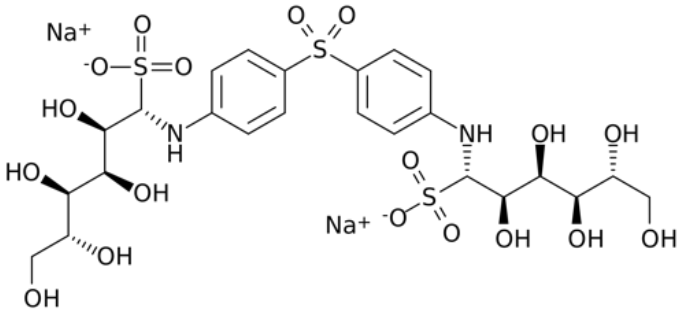
2020.11.25

简单回顾

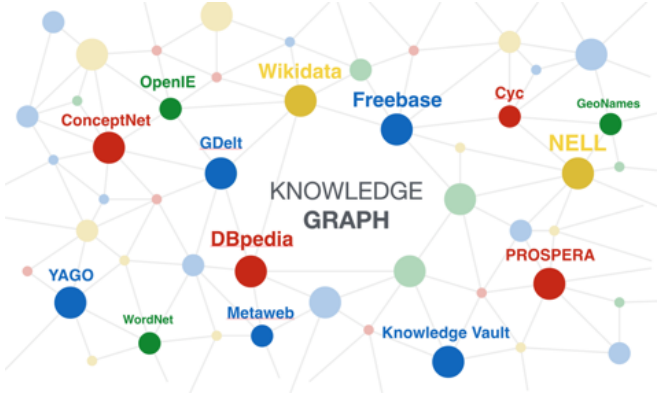
社交关系



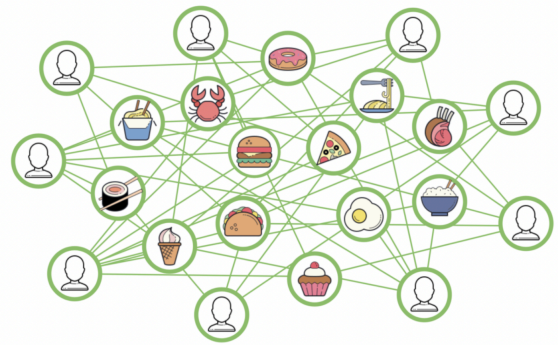
化学分子



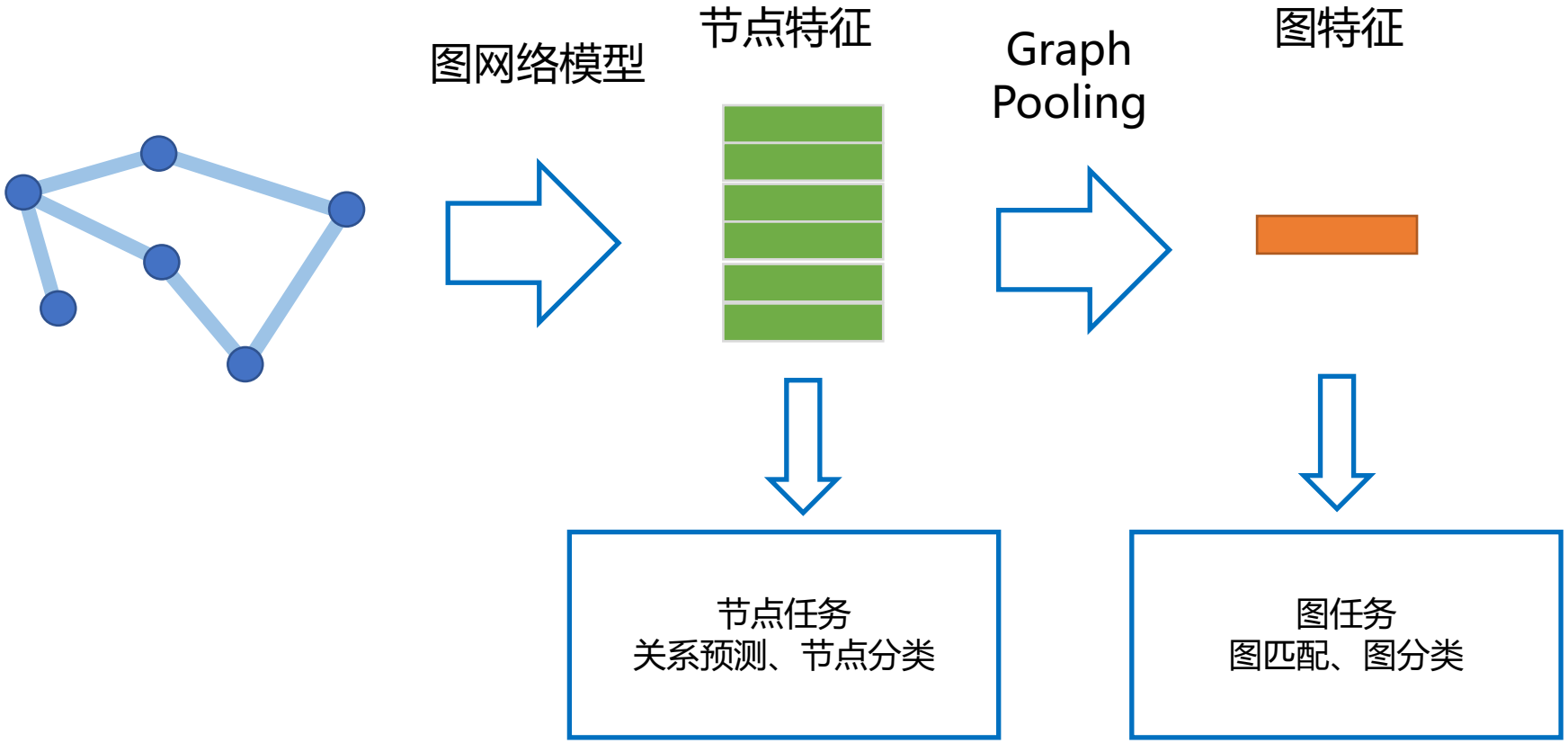
知识图谱



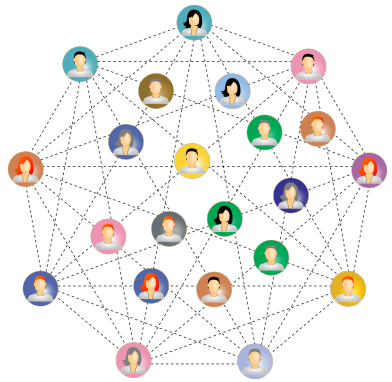
推荐系统



图网络的目的

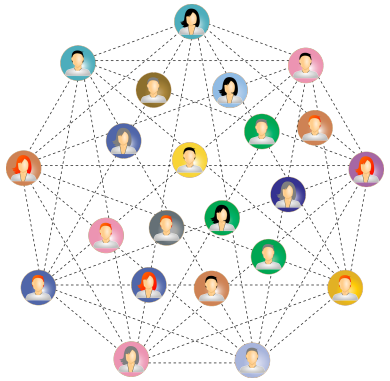


训练方式



图游走类模型

下游任务，如分类

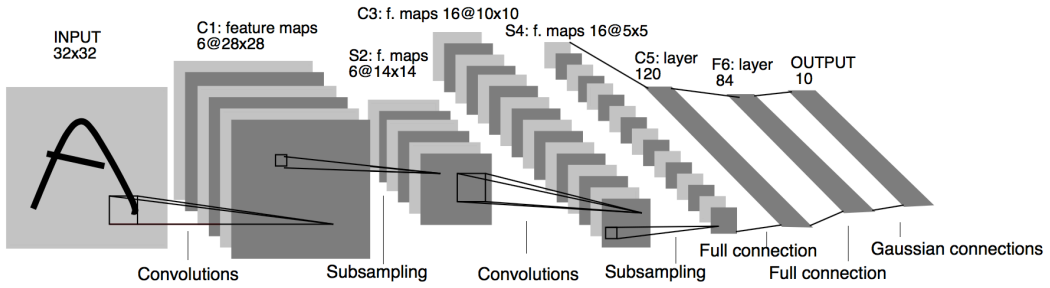


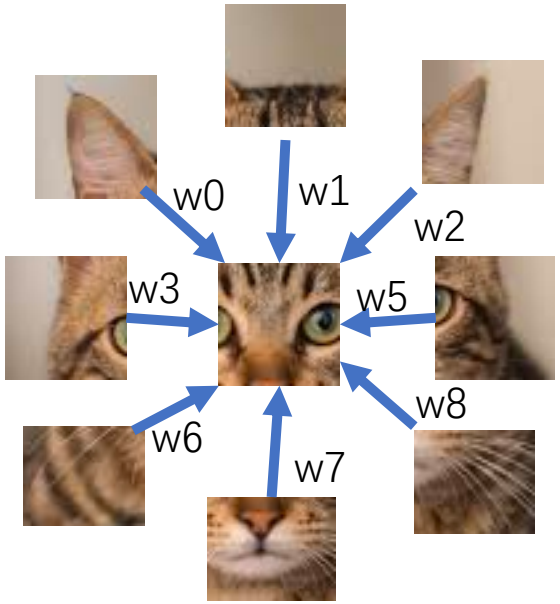
具体任务，如分类

端到端训练

回顾DNN在文本以及图像的发展

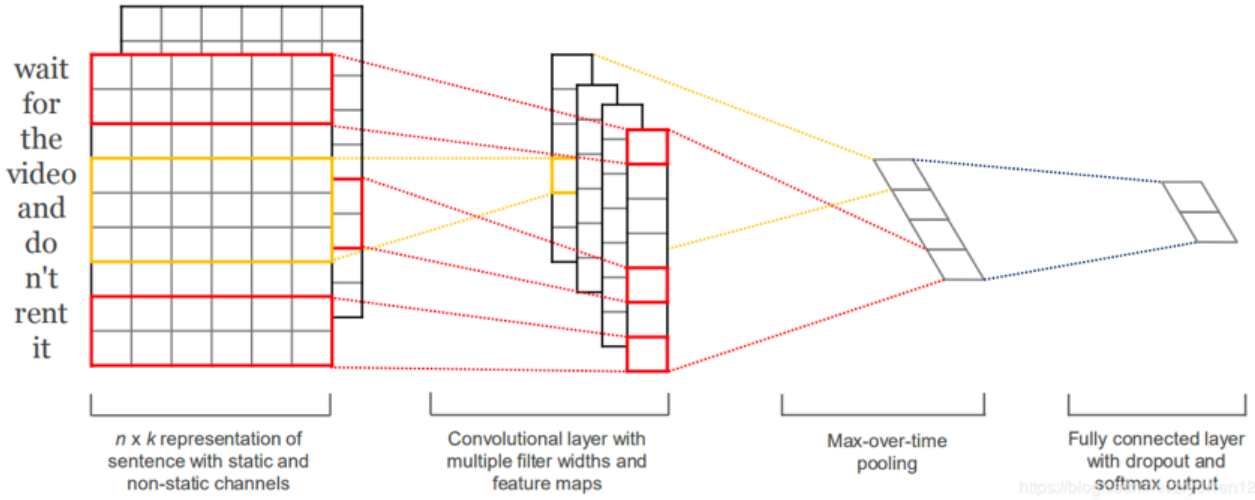
图像上的卷积网络
2D-Convolution





回顾DNN在文本以及图像的发展

文本上的卷积网络 1D-Convolution

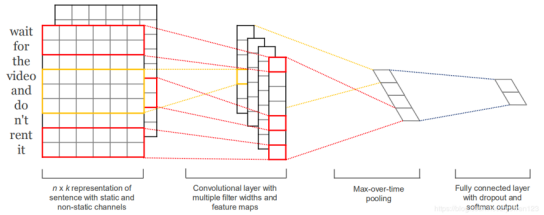


TextCNN: Convolutional Neural Networks for Sentence Classification

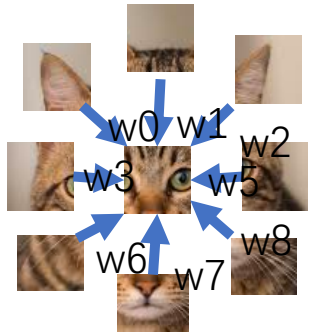
规则性数据

排列整齐的数据
Grid-Like Data

文本

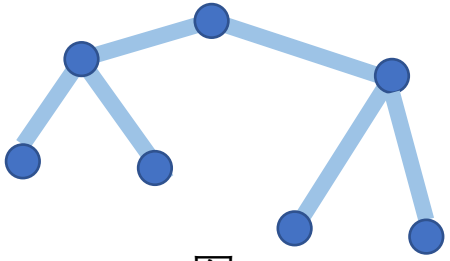


图像

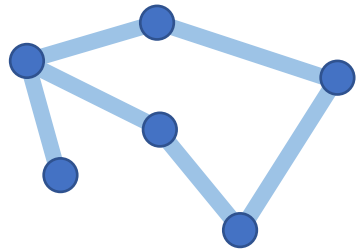


不规则数据

树

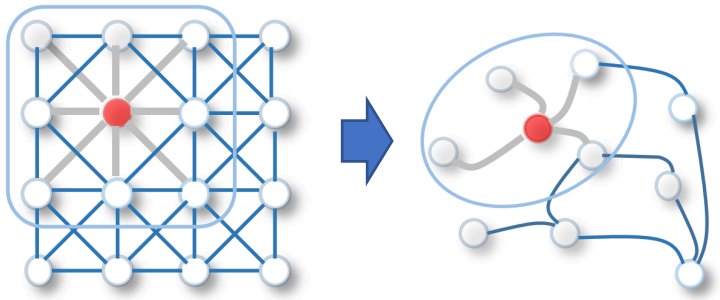


图

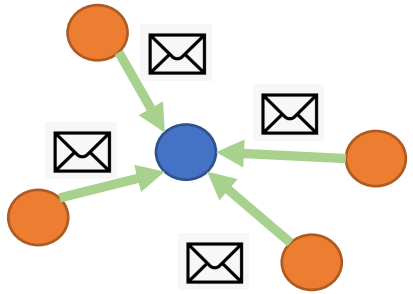


图卷积网络

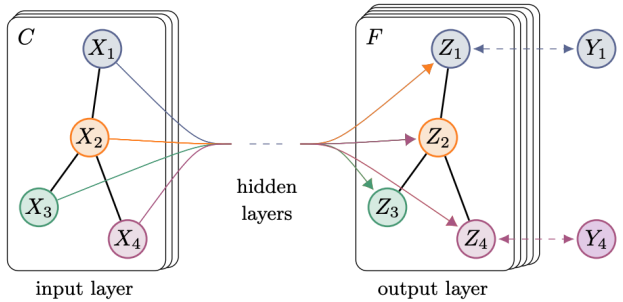
Graph Convolutional Network



如何从**图像卷积**，类比到**图结构卷积**



怎么用**消息传递**方式实现图卷积网络

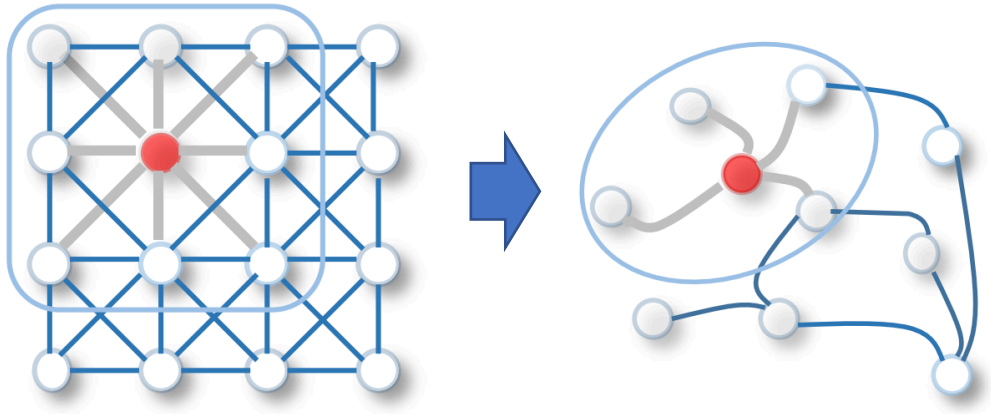


怎么用多层图网络完成**节点分类任务**

参考文献: SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS (ICLR 2017)

图卷积网络

Graph Convolutional Network

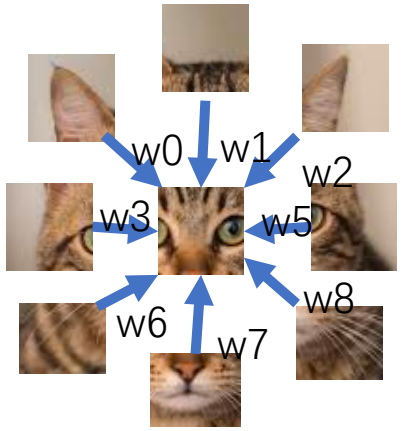


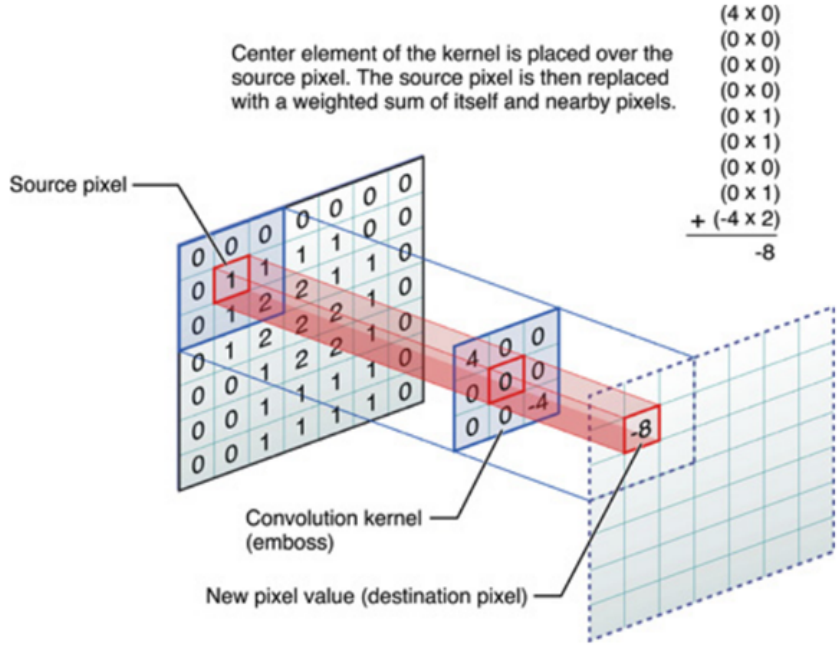
如何从**图像卷积**，类比到**图结构卷积**

图卷积网络

Graph Convolutional Network

图像上的卷积操作





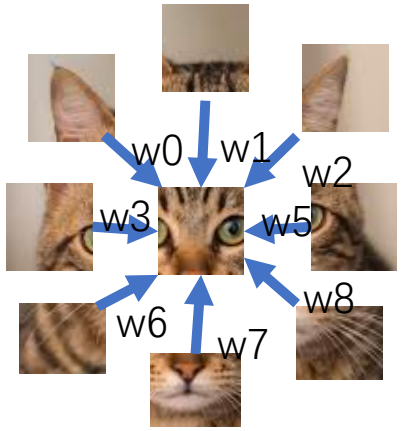
图像卷积

将一个**像素点**周围的**像素**按照不同的**权重叠加**起来。

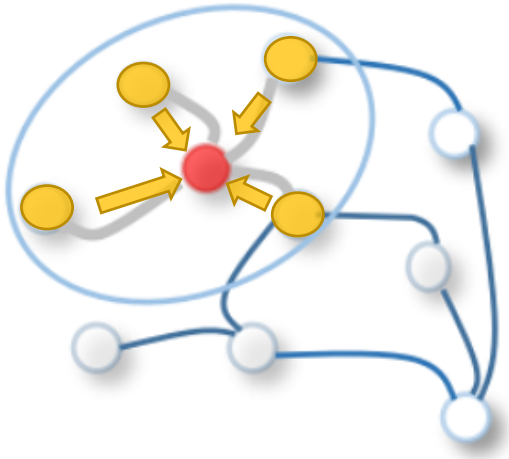
图卷积网络

Graph Convolutional Network

图像上的卷积操作



图结构上的卷积



图像卷积

将一个**像素点**周围的**像素**按照不同的**权重叠加**起来。

图结构卷积

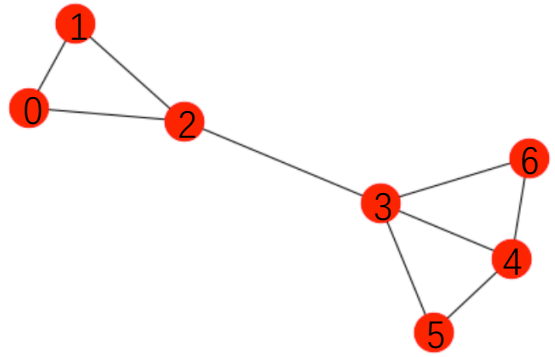
将一个**节点**周围的**邻居**按照不同的**权重叠加**起来。

图卷积网络

Graph Convolutional Network

图卷积网络的计算公式

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$



红框 是图卷积网络的**核心公式**

$\hat{A}$ 邻接矩阵

$\hat{D}^{-\frac{1}{2}}$ 与度矩阵相关

$H^{(l)}$ 每一层的节点表示

1	1	1	0	0	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0
0	0	1	1	1	1	1
0	0	0	1	1	1	1
0	0	0	1	1	1	0
0	0	0	1	1	0	1

邻接矩阵

图卷积网络

Graph Convolutional Network

图卷积网络的计算公式

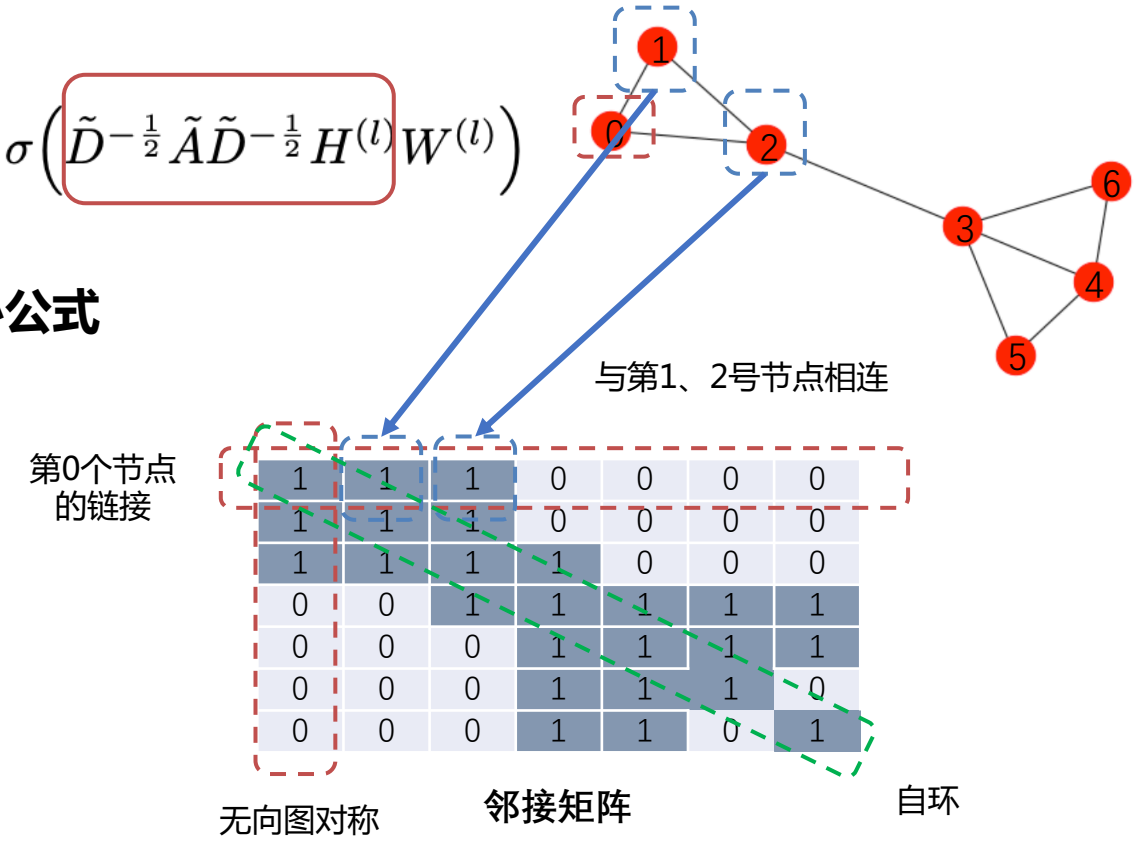
$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

红框 是图卷积网络的**核心公式**

$\hat{A}$ 邻接矩阵

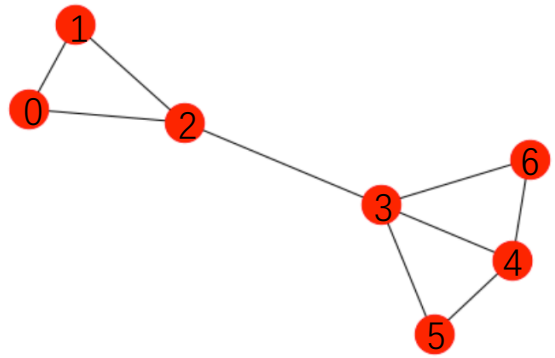
$\hat{D}^{-\frac{1}{2}}$ 与度矩阵相关

$H^{(l)}$ 每一层的节点表示



图卷积网络

Graph Convolutional Network



图卷积网络的计算公式

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

$\hat{A}$ 是邻接矩阵

1	1	1	0	0	0	0
1	1	1	0	0	0	0
1	1	1	1	0	0	0
0	0	1	1	1	1	1
0	0	0	1	1	1	1
0	0	0	1	1	1	0
0	0	0	1	1	0	1

求和

$\hat{D}$ 是度矩阵

3	0	0	0	0	0	0
0	3	0	0	0	0	0
0	0	4	0	0	0	0
0	0	0	5	0	0	0
0	0	0	0	4	0	0
0	0	0	0	0	3	0
0	0	0	0	0	0	3

$\hat{D}^{-\frac{1}{2}}$

$1/\sqrt{3}$	0	0	0	0	0	0
0	$1/\sqrt{3}$	0	0	0	0	0
0	0	$1/\sqrt{4}$	0	0	0	0
0	0	0	$1/\sqrt{5}$	0	0	0
0	0	0	0	$1/\sqrt{4}$	0	0
0	0	0	0	0	$1/\sqrt{3}$	0
0	0	0	0	0	0	$1/\sqrt{3}$

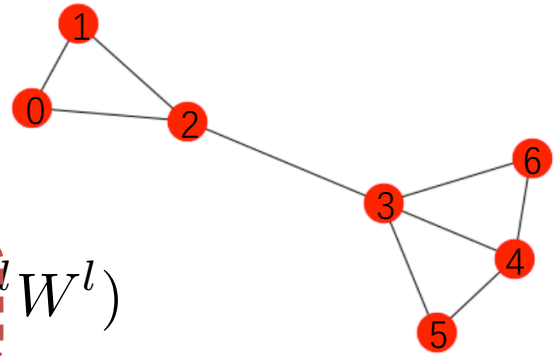
图卷积网络

Graph Convolutional Network

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

简化

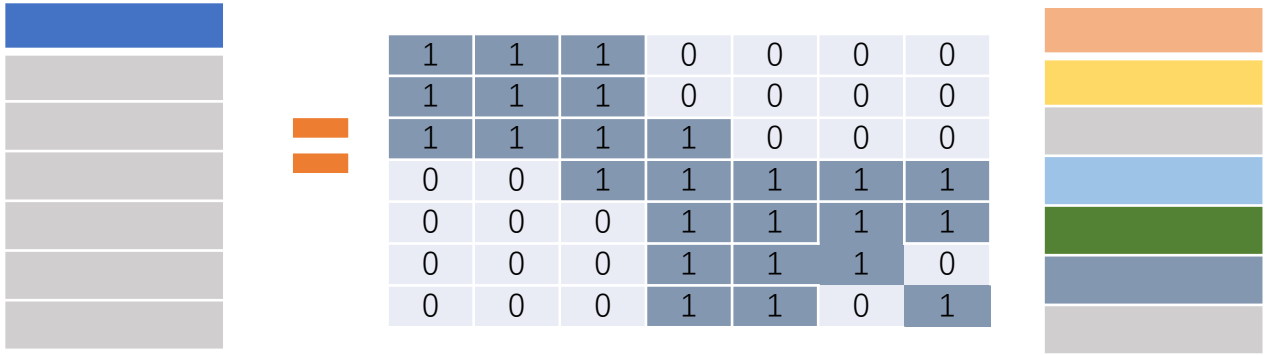
$$H^{(l+1)} = \sigma(\hat{A}H^{l}W^{l})$$



$AH^{(l)}$ 公式的物理含义是什么

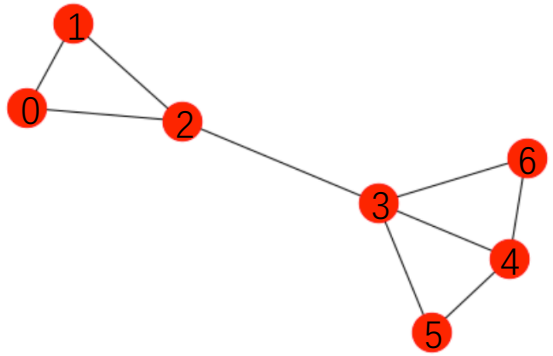
$H^{(l+1)}$ A $H^{(l)}$

第0节点表示



图卷积网络

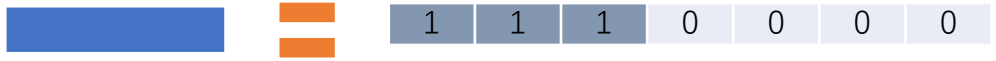
Graph Convolutional Network



$AH^{(l)}$ 公式的物理含义是什么

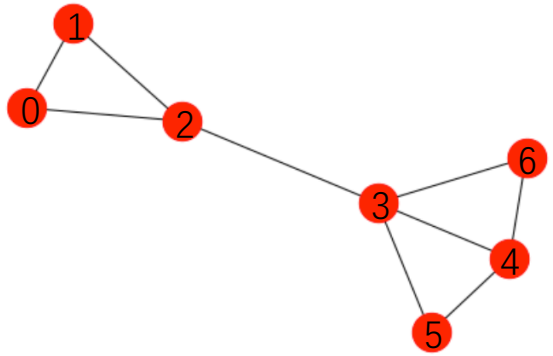
$H^{(l+1)}$ A $H^{(l)}$

第0节点表示



图卷积网络

Graph Convolutional Network



$AH^{(l)}$ 公式的物理含义是什么

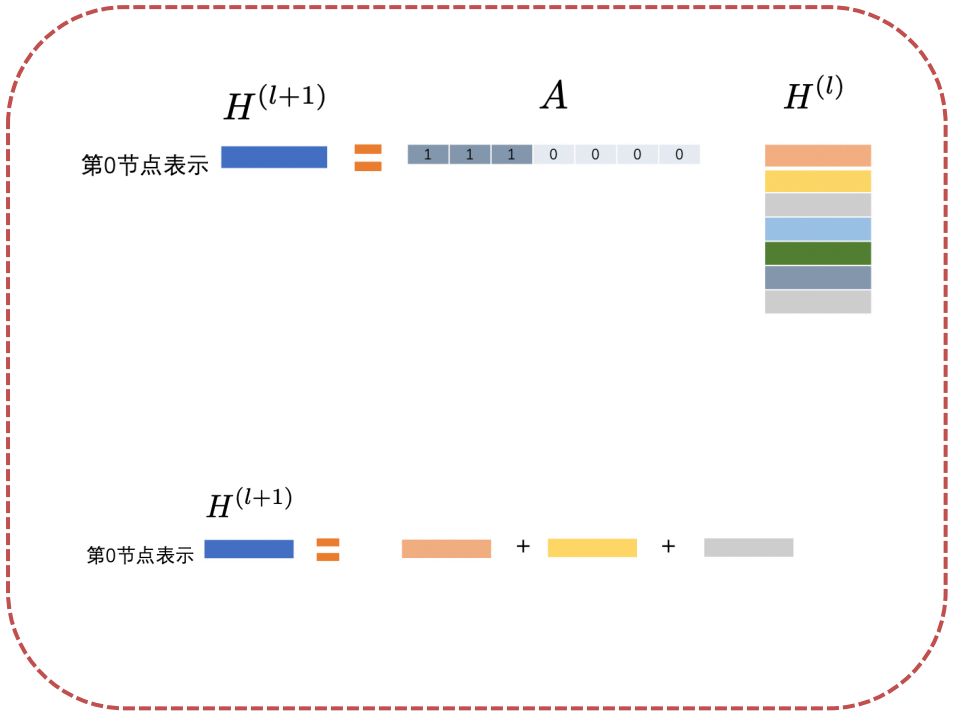
$H^{(l+1)}$

第0节点表示 

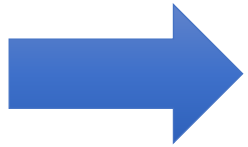
图卷积网络

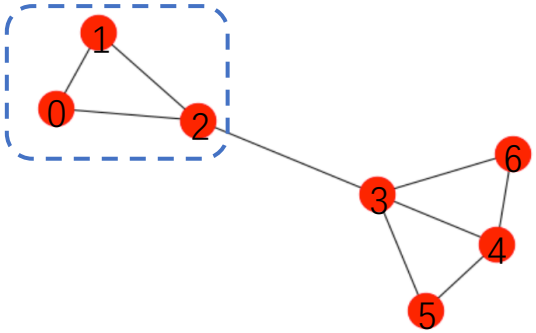
Graph Convolutional Network

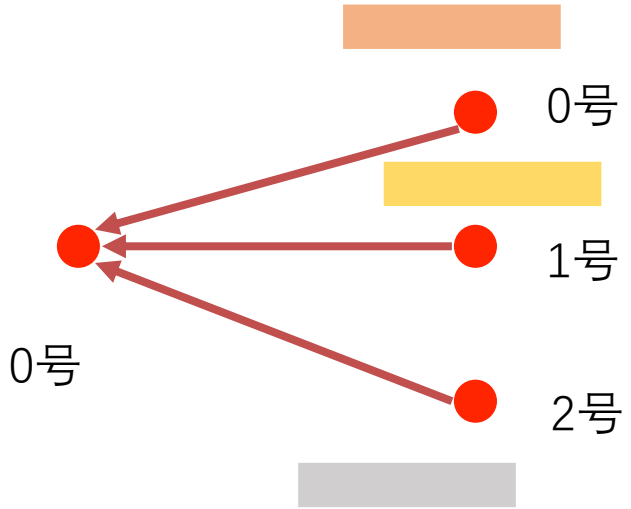
$AH^{(l)}$ 公式的物理含义是什么



消息传递
Message Passing

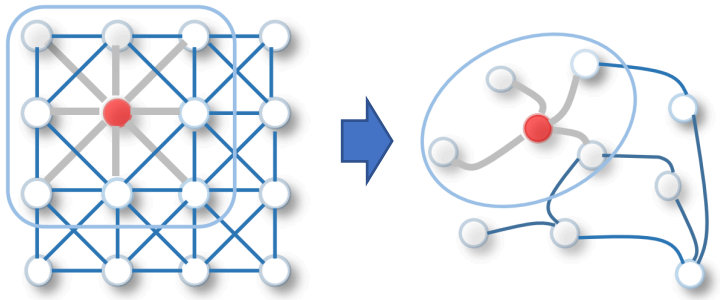




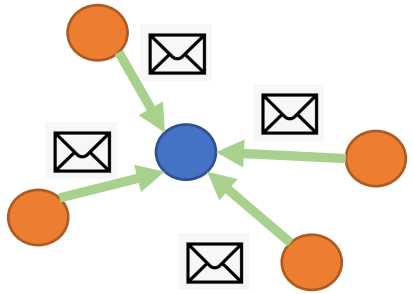


图卷积网络

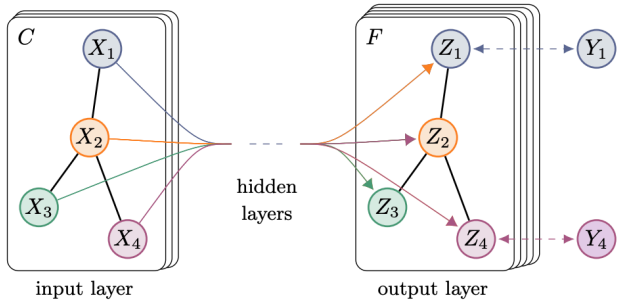
Graph Convolutional Network



如何从**图像卷积**，类比到**图结构卷积**



怎么用**消息传递**方式实现图卷积网络

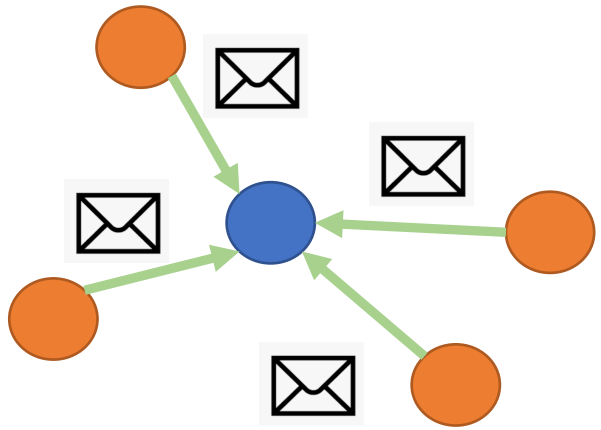


怎么用多层图网络完成**节点分类任务**

参考文献: SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS (ICLR 2017)

图卷积网络

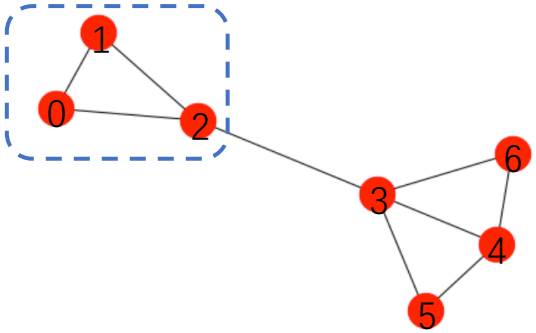
Graph Convolutional Network



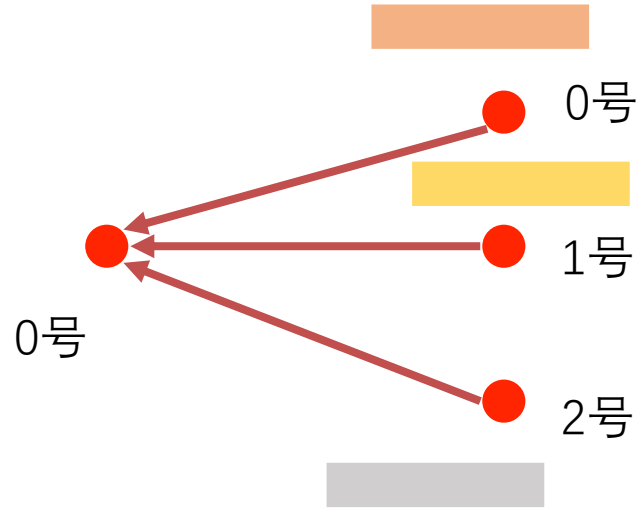
怎么用**消息传递**方式实现图卷积网络

图卷积网络

Graph Convolutional Network



消息传递
Message Passing

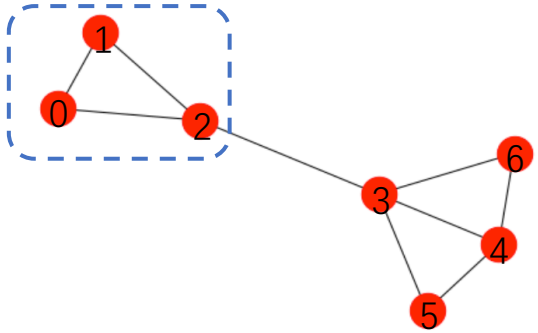


发送Send

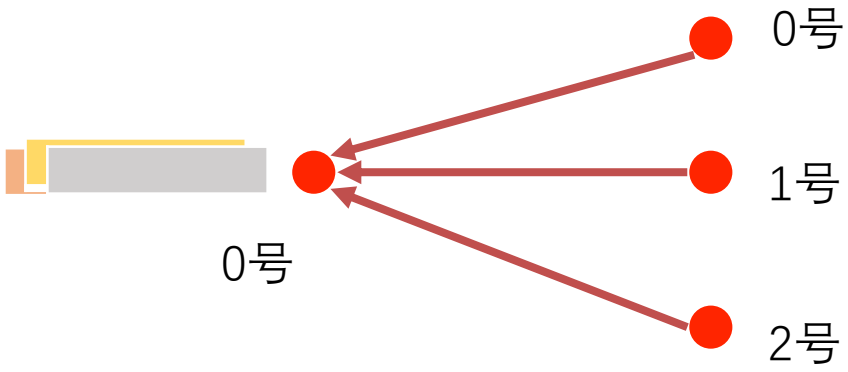
1. 边上的源节点，往目标节点发送特征

图卷积网络

Graph Convolutional Network



消息传递
Message Passing

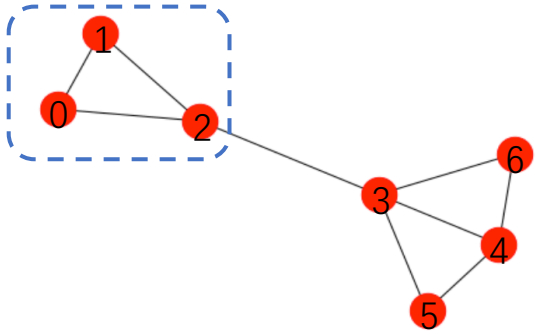


接收Recv

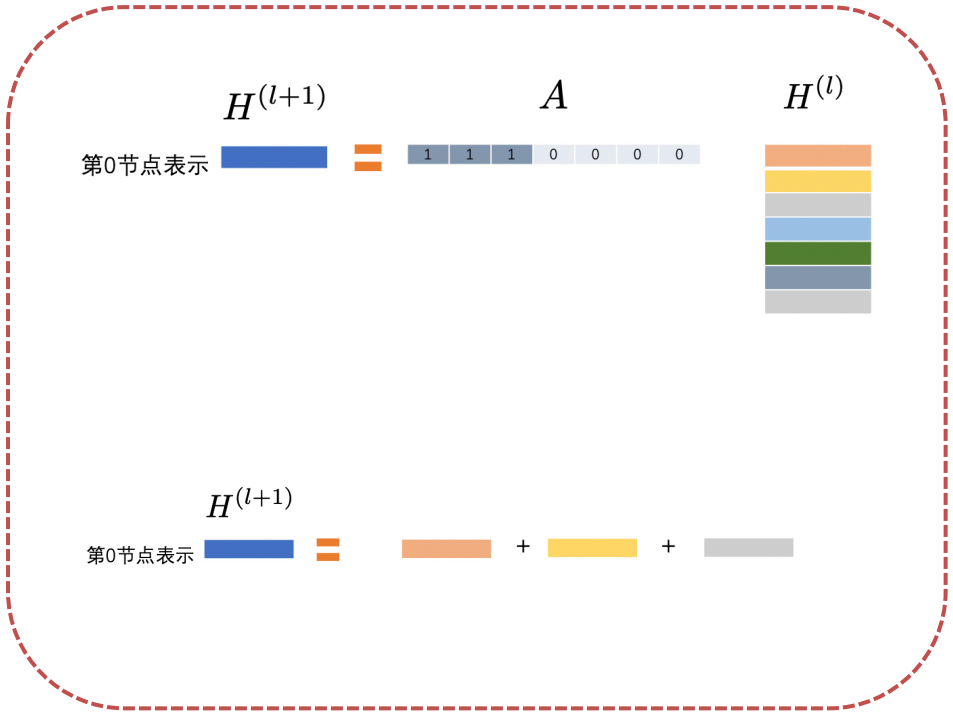
2. 目标节点对收到的特征进行聚合

图卷积网络

Graph Convolutional Network



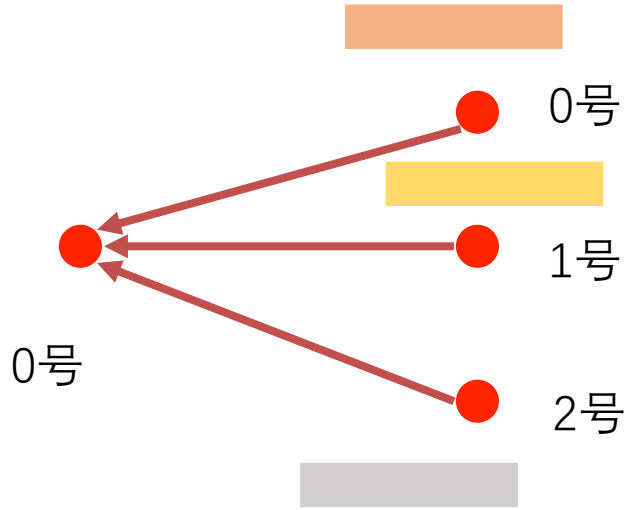
$AH^{(l)}$ 公式的物理含义是什么



消息传递
Message Passing

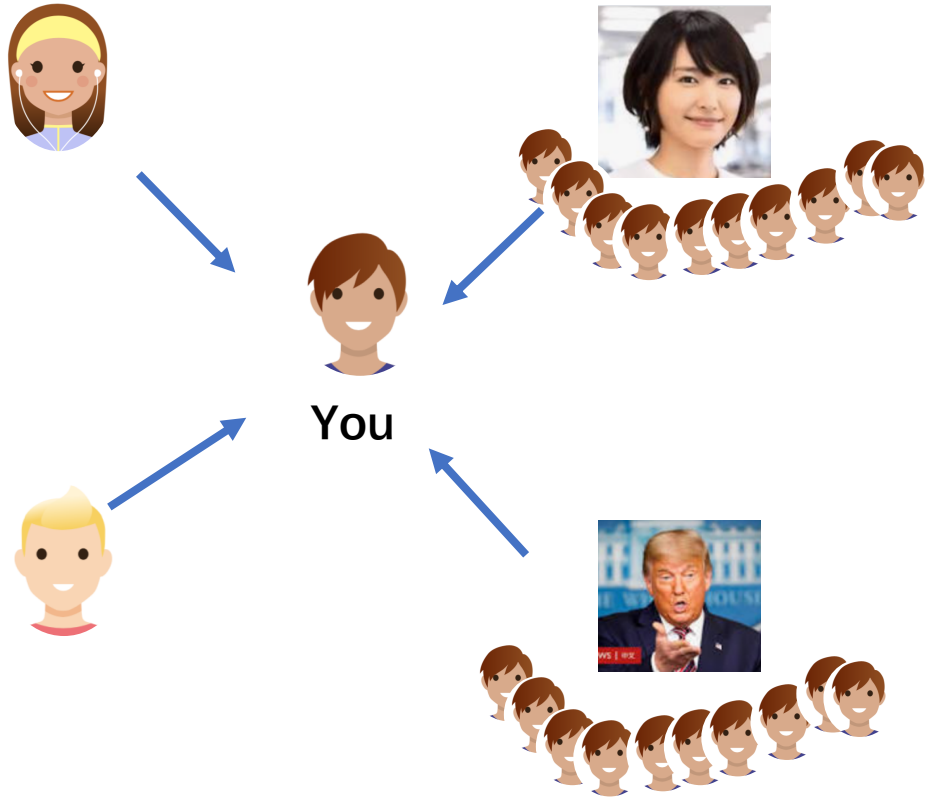
发送SEND 源节点发送特征给目标节点

接收RECV 目标节点接收特征并统计



图卷积网络

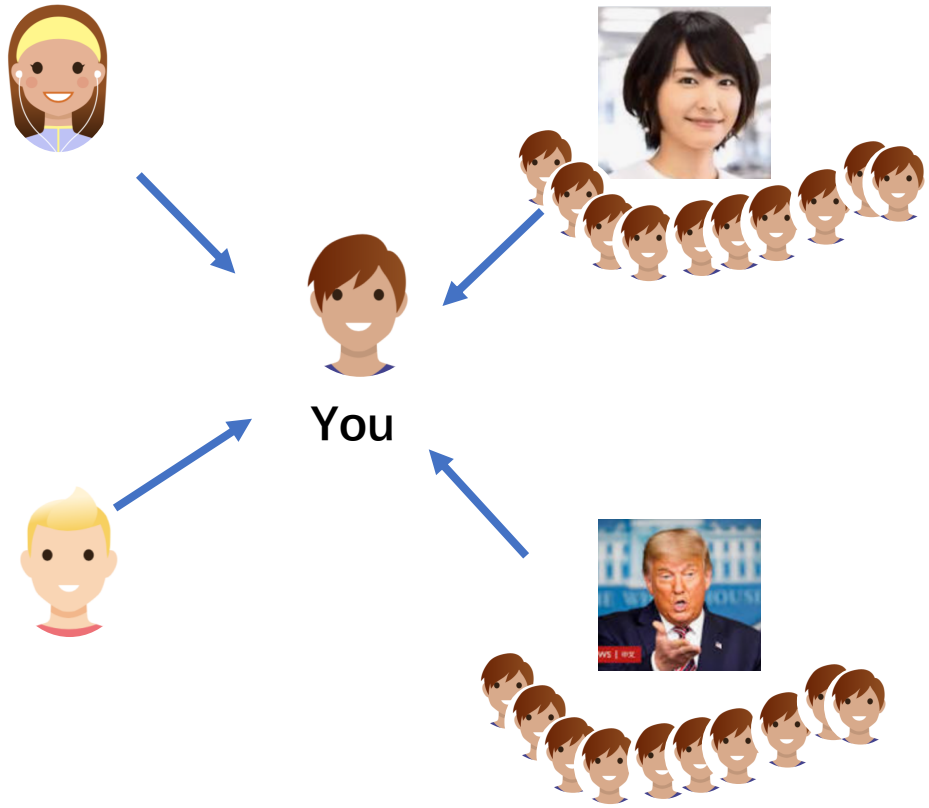
$AH^{(l)}$ 将邻居的特征接收、聚合



是否所有的人对自己的评价都是有用的？

图卷积网络

$AH^{(l)}$ 将邻居的特征接收、聚合

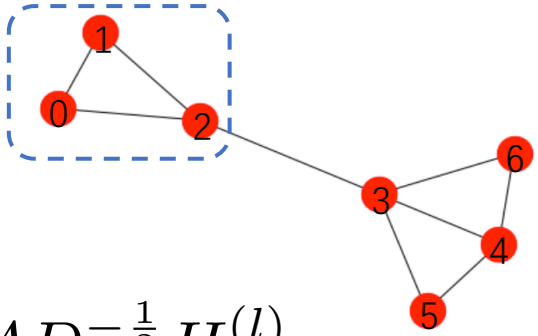


是否所有的人对自己的评价都是有用的？

如何衡量邻居的重要性 **度**

图卷积网络

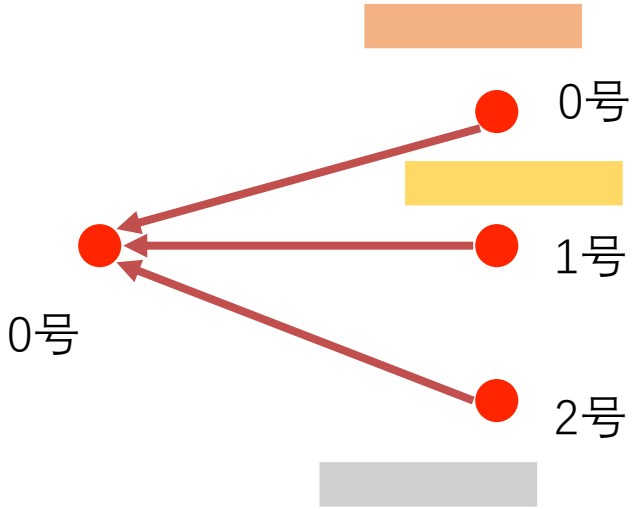
Graph Convolutional Network

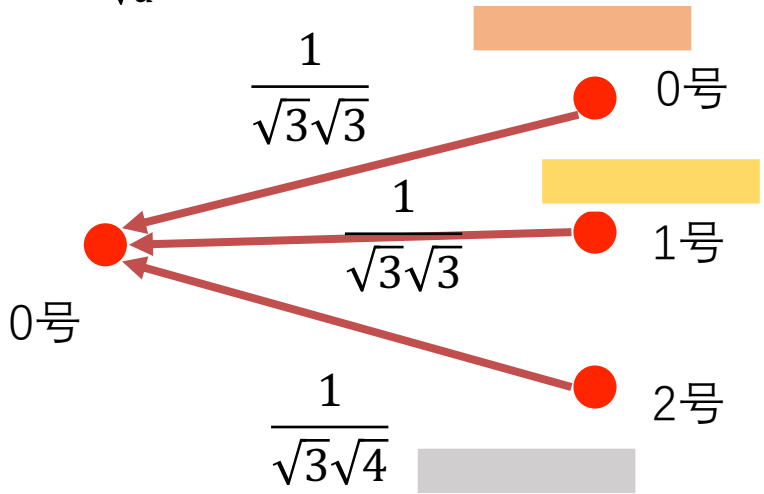


给边加权

$AH^{(l)}$ → $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}H^{(l)}$

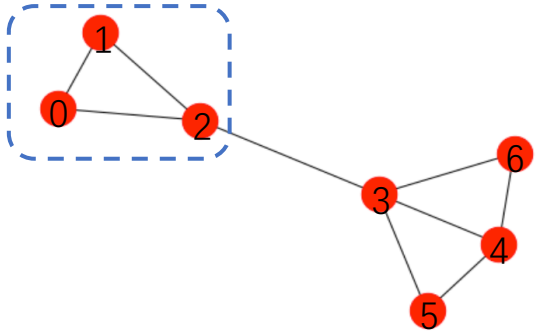
d越大，信息量越小，$\frac{1}{\sqrt{d}}$越小



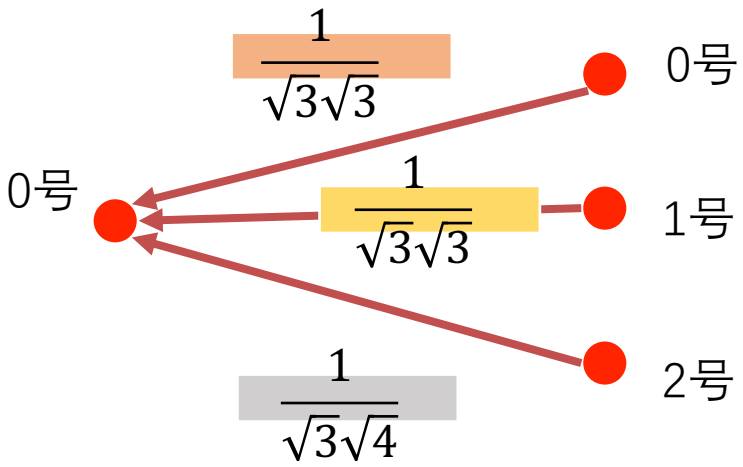


图卷积网络

Graph Convolutional Network



$$D^{-\frac{1}{2}}AD^{-\frac{1}{2}}H^{(l)}$$

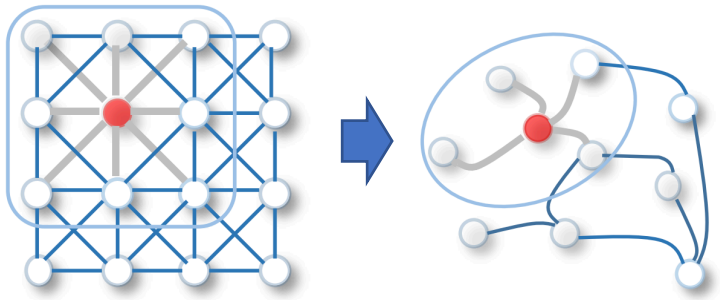


在PGL中Message Passing实现

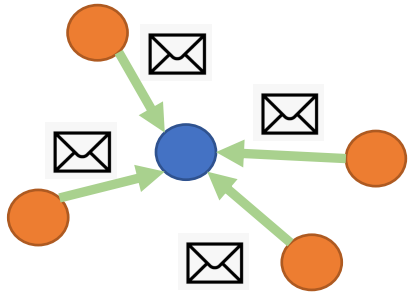
```
from paddle.fluid.layers import sequence_pool as pool

# 发送函数，从发送源src到发送目标点dst
def send(src_feat, dst_feat, edge_feat):
    return src_feat["norm"] * dst_feat["norm"] * src_feat["h"]

# 接收函数，从发送源src到发送目标点dst
def recv(message):
    return pool(message, "sum")
```

图卷积网络

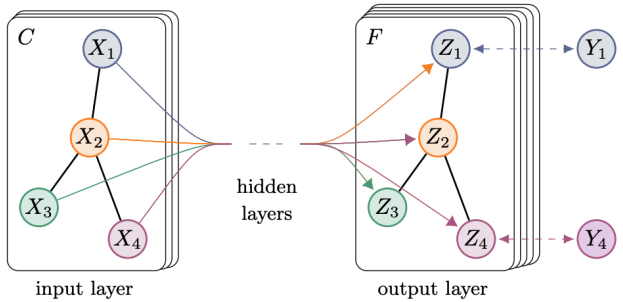
Graph Convolutional Network



如何从**图像卷积**，类比到**图结构卷积**



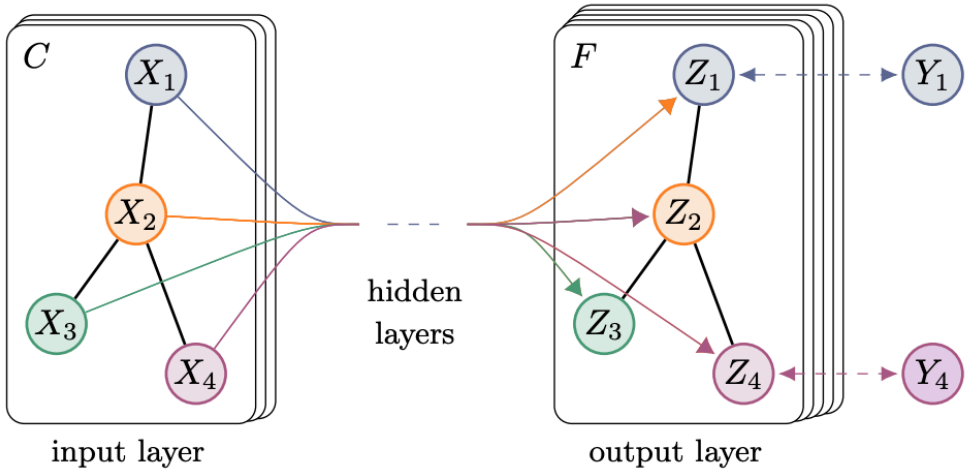
怎么用**消息传递**方式实现图卷积网络



怎么用多层图网络完成**节点分类任务**

图卷积网络

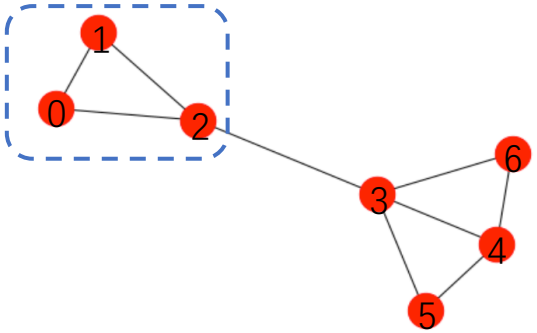
Graph Convolutional Network



怎么用多层图网络完成**节点分类任务**

图卷积网络

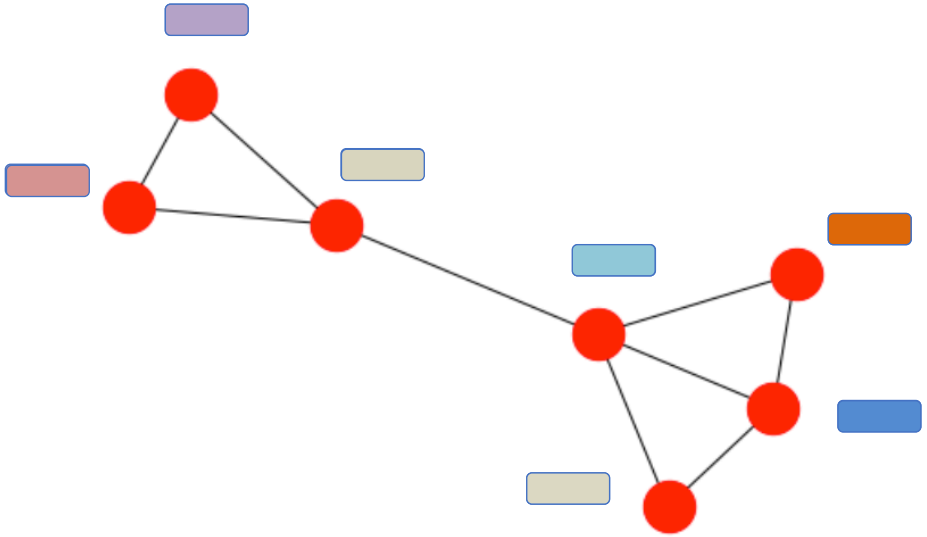
Graph Convolutional Network



GCN算法全流程

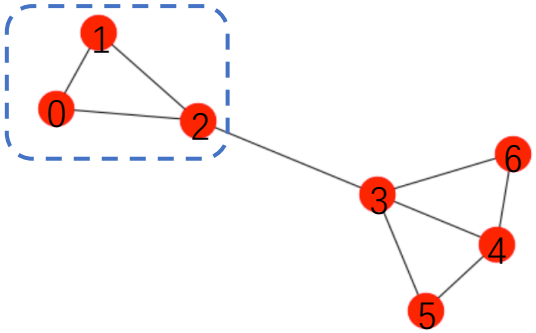
$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

第一步： $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}$ 节点间进行特征传递



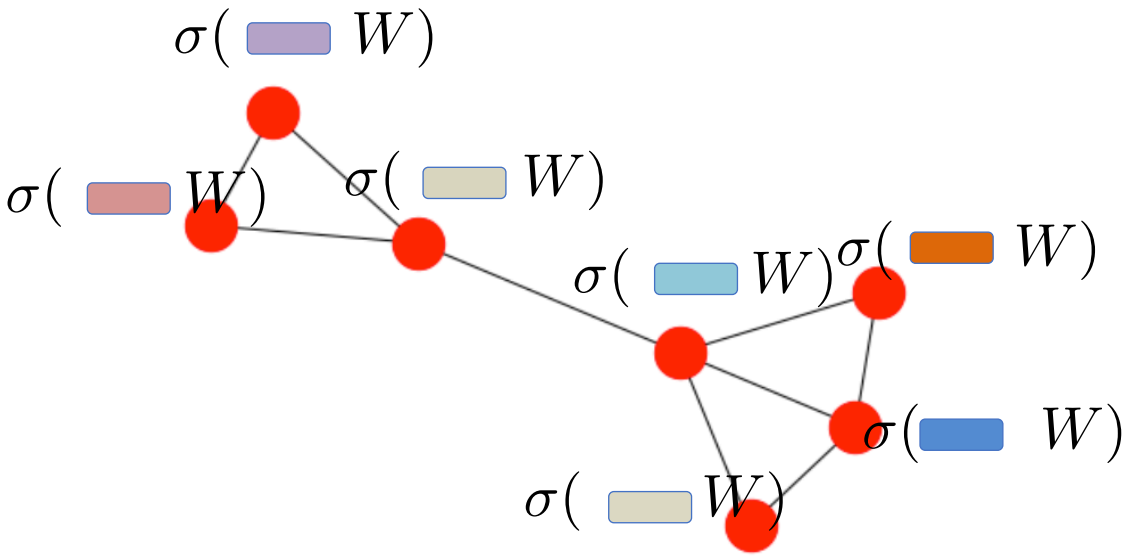
图卷积网络

Graph Convolutional Network



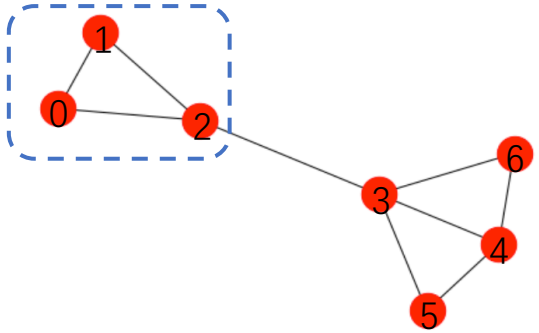
GCN算法全流程 $$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

第二步： 对每一个节点过一层DNN



图卷积网络

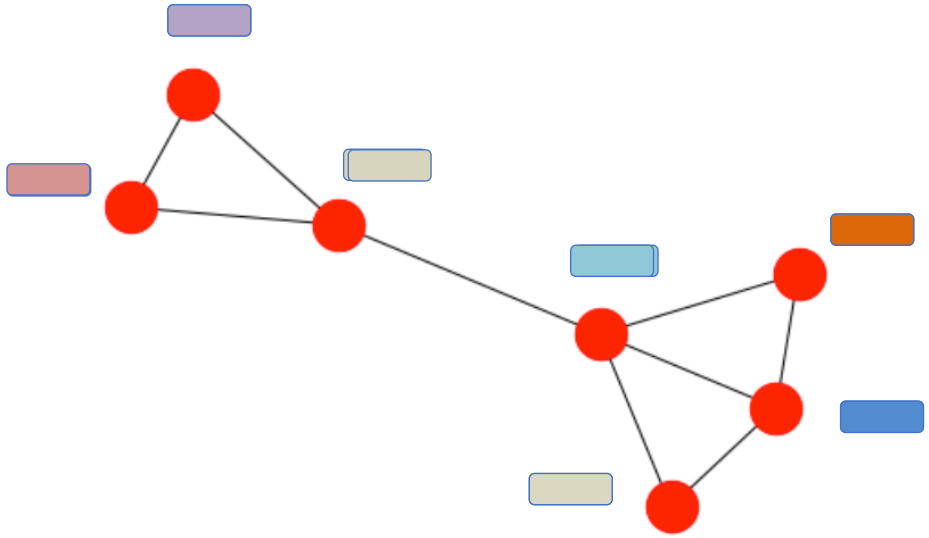
Graph Convolutional Network

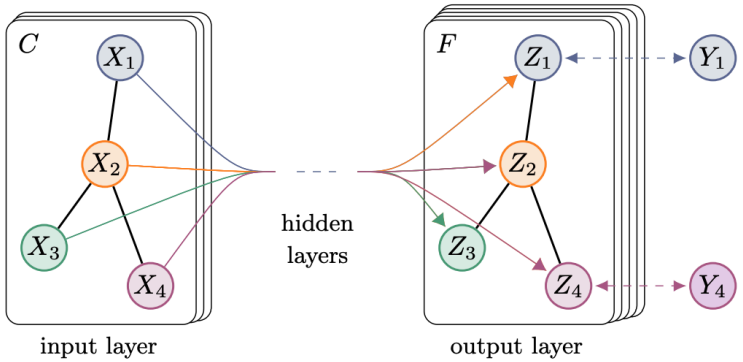


GCN算法全流程

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

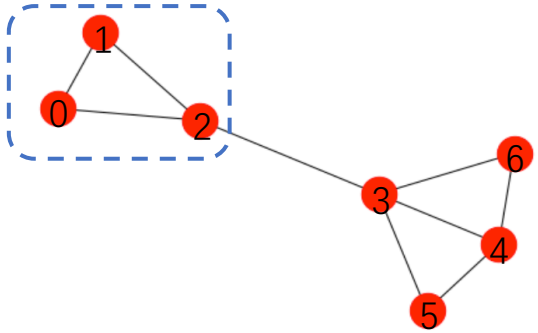
重复L次，步骤1和步骤2，实现多层图卷积网络





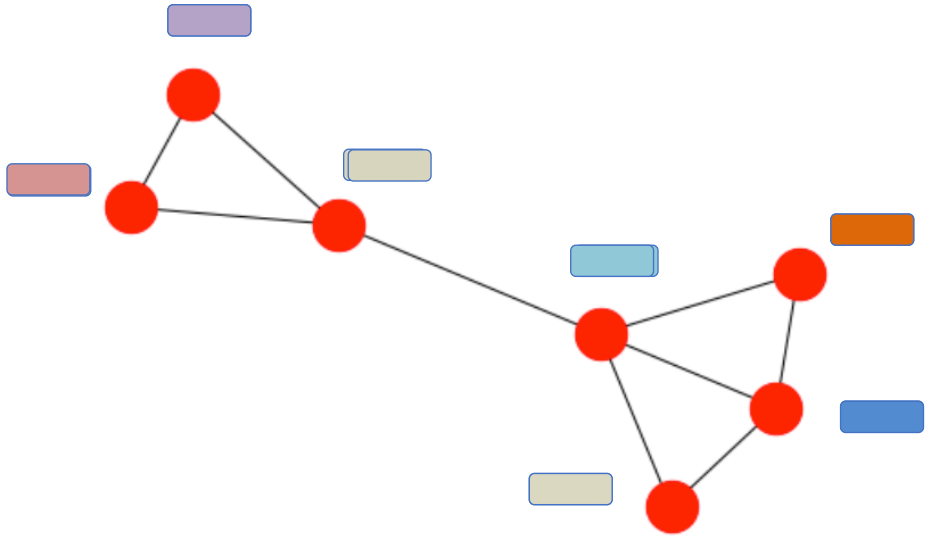
图卷积网络

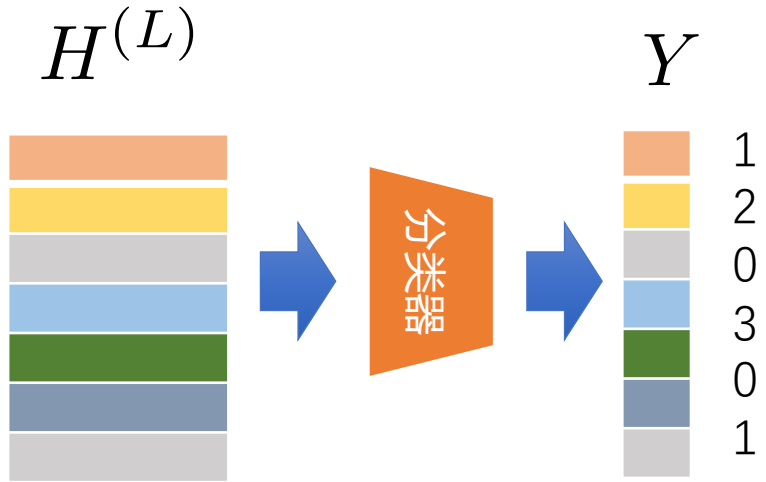
Graph Convolutional Network



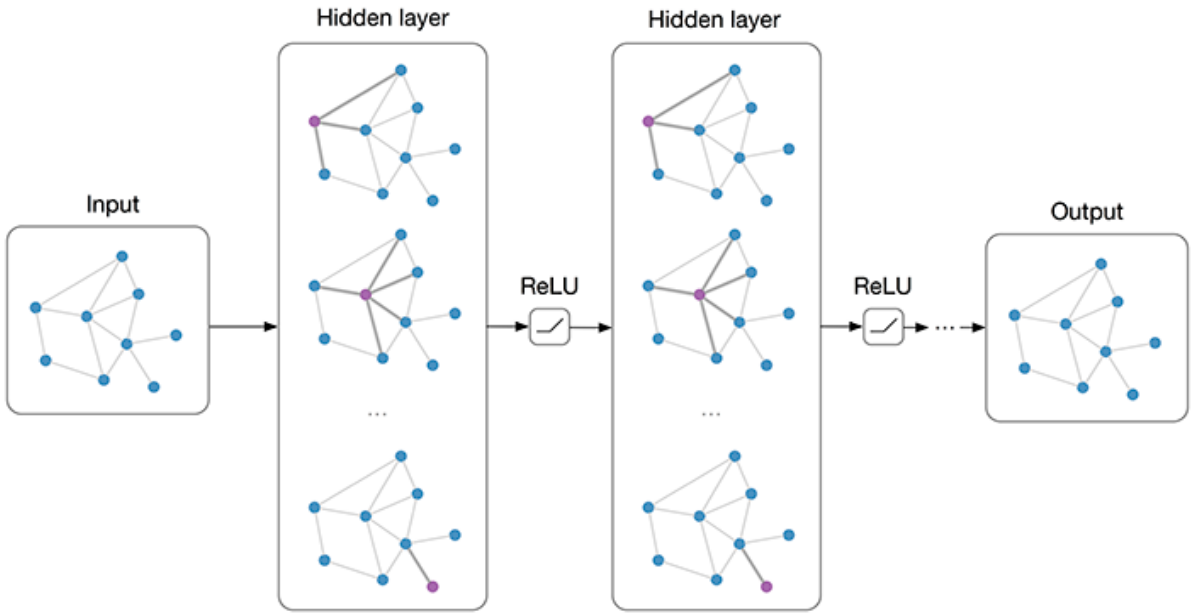
GCN算法全流程 $$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

获取的最终表示 $H^{(L)}$ 作为最终节点表示



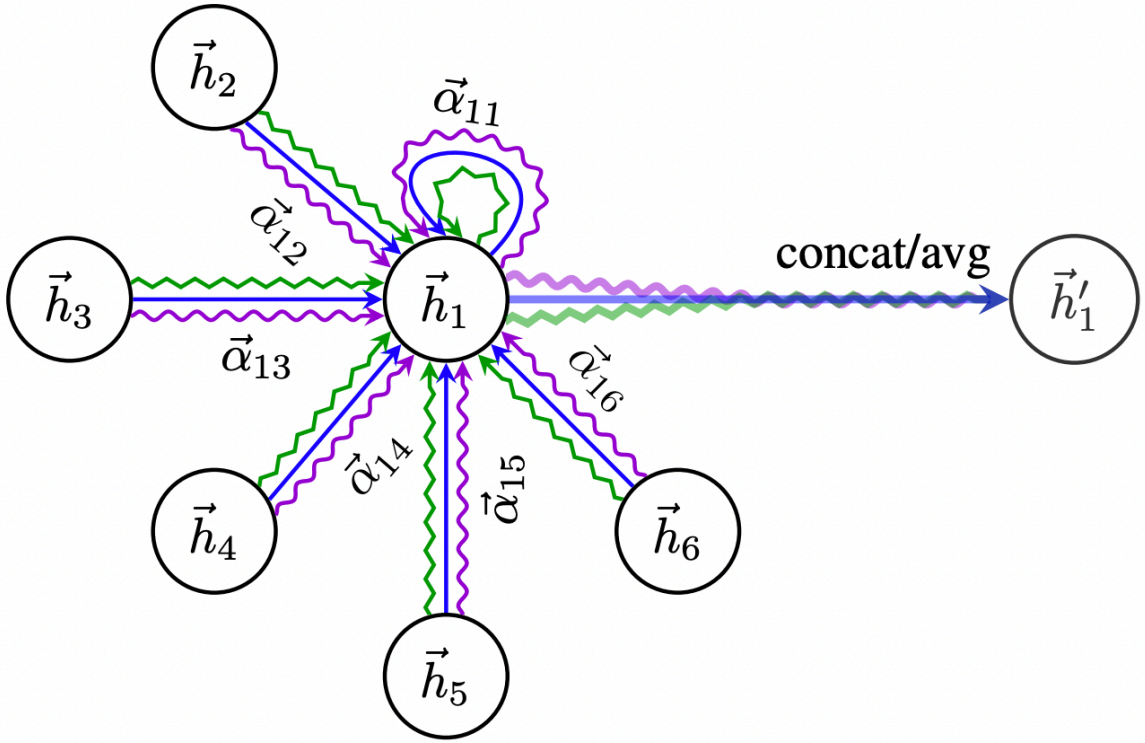


图卷积网络多种形式



图卷积网络（GCN）

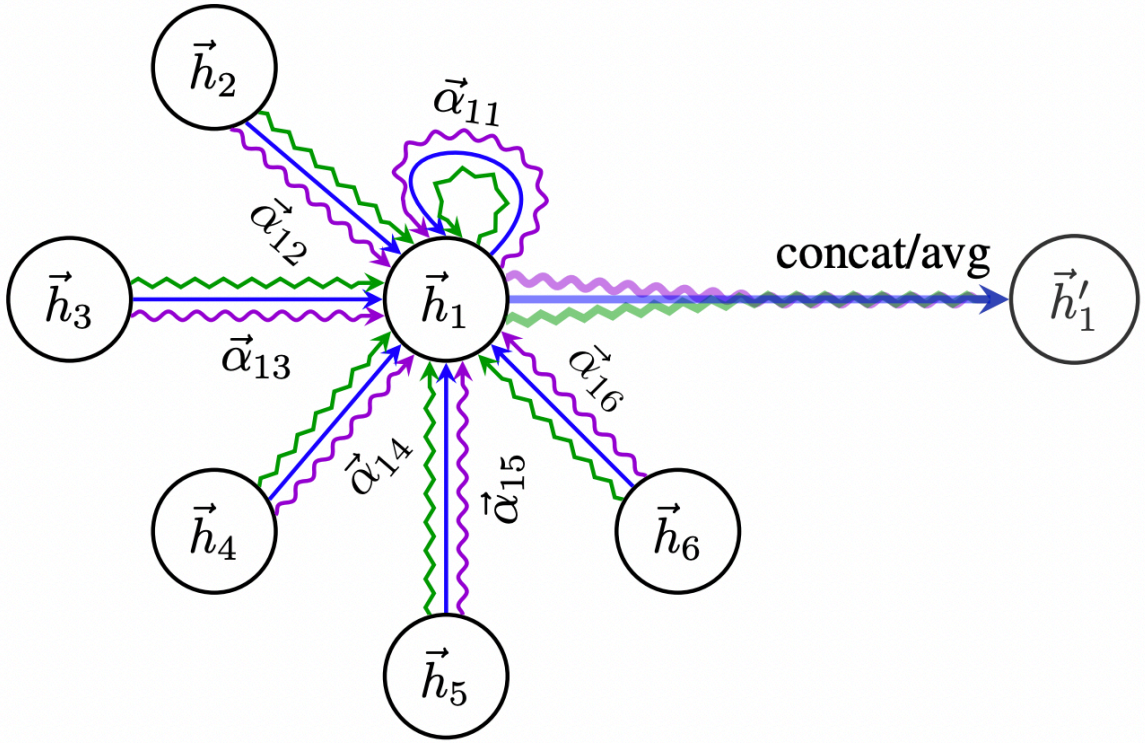
Kipf, Thomas N and Welling, Max
Semi-Supervised Classification with Graph Convolutional Networks
ICLR 2017



图注意力网络（GAT）

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò and Yoshua Bengio
Graph Attention Networks,
ICLR 2018

图卷积网络多种形式



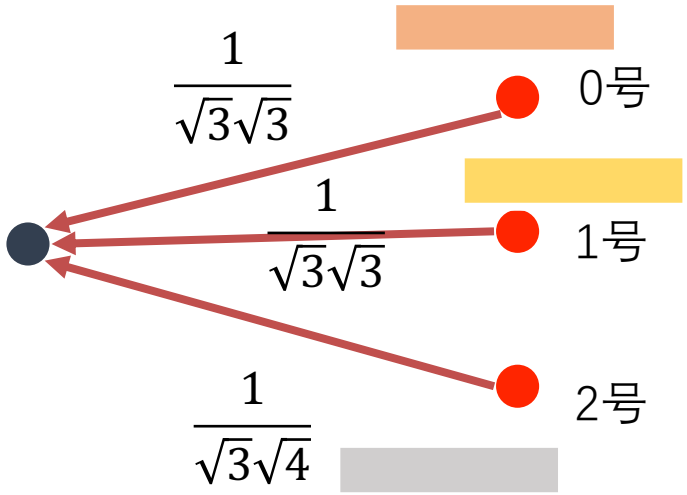
图注意力网络（GAT）

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò and Yoshua Bengio
Graph Attention Networks,
ICLR 2018

图注意力网络

Graph Attention Network

$$D^{-\frac{1}{2}}AD^{-\frac{1}{2}}H^{(l)}$$

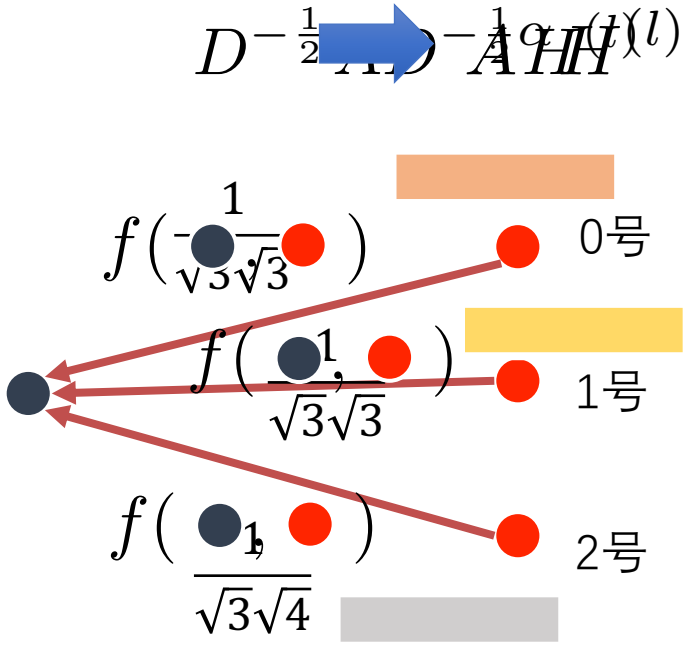


GCN中边的权重：

- 与节点的度相关
- 不可学习

图注意力网络

Graph Attention Network



GCN中边的权重：

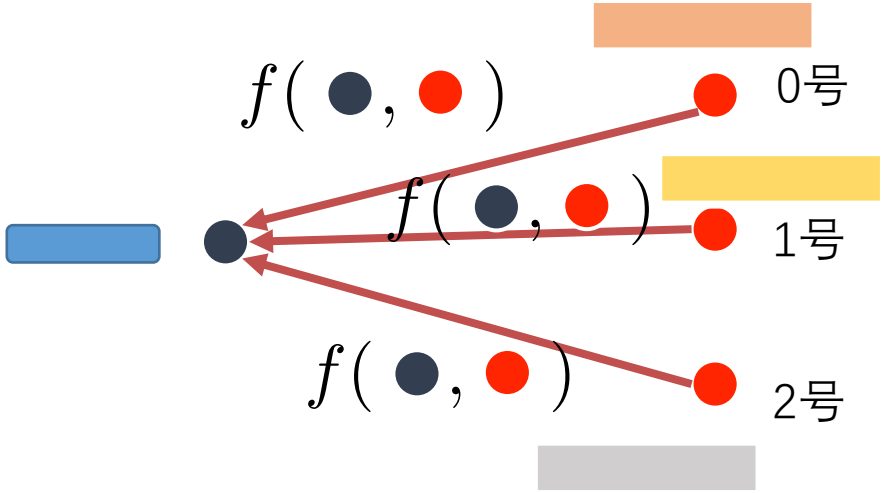
- 与节点的度相关
- 不可学习

GAT中的边权重 $f(\bullet, \bullet)$

- 权重变成节点间的函数
- 权重与两个节点相关性有关
- 可学习

图注意力网络

Graph Attention Network



attention计算方法

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$

特征聚合计算方法

$$\vec{h}'_i = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}\vec{h}_j\right)$$

图注意力网络

Graph Attention Network

飞桨

attention计算方法

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$

特征聚合计算方法

$$\vec{h}'_i = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}\vec{h}_j\right)$$

单头GAT消息传递在PGL中Message Passing实现

```
from paddle.fluid.layers import sequence_pool as pool
from paddle.fluid.layers import sqeuence_softmax as edge_softmax
import paddle.fluid.layers as L
...
# 发送函数，从发送源src到发送目标点dst
def send(src_feat, dst_feat, edge_feat):
    src_weight = L.fc(src_feat["h"], size=1, name="a1_weight")
    dst_weight = L.fc(dst_feat["h"], size=1, name="a2_weight")
    weight = L.leaky_relu(src_weight + dst_weight, alpha=0.2)
    # 计算每一条边的源节点以及目标节点相关性
    return {"weight": weight, "h":  src_feat["h"]}

# 接收函数，从发送源src到发送目标点dst
def recv(message):
    # 对相关性进行softmax 归一化
    alpha = edge_softmax(message["weight"])
    hidden = pool(alpha * message["h"], "sum")
    return hidden
```

图注意力网络

Graph Attention Network

飞桨

attention计算方法

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$

特征聚合计算方法

$$\vec{h}'_i = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}\vec{h}_j\right)$$

多头 Attention 特征聚合方法

$$\vec{h}'_i = \Big\Vert_{k=1}^{K}\sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k\mathbf{W}^k\vec{h}_j\right)$$

```
for head_no in range(n_heads):
    # 请完成单头的GAT的代码
    single_output = single_head_gat(graph_wrapper,
                                    node_feature,
                                    hidden_size,
                                    name="head_%s" % (head_no) )
    heads_output.append(single_output)
```

作业部分代码

num_nodes * feature_size -> num_heads * num_nodes * hidden_size

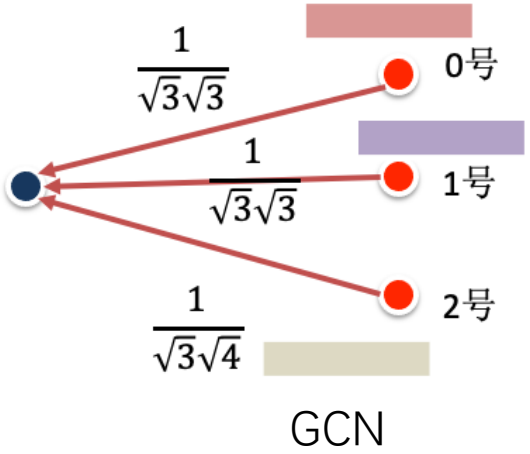
Message Passing

GCN、GAT都是基于邻居聚合的模型

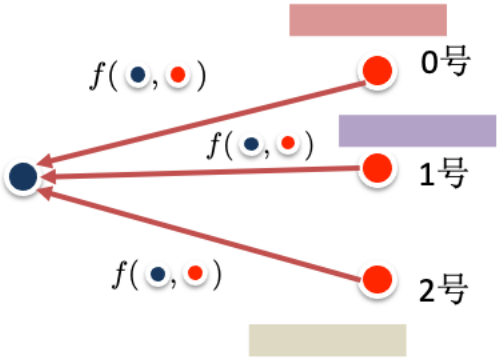
称为 **Spatial (空间) GNN**

大部分的Spatial GNN都可以用Message Passing实现

- **消息的发送**
- **消息的接收**



GCN



GAT

Message Passing

基于消息传递的 Graph Neural Network的通用公式

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u)|u \in N(v)\})$$

多层MLP

F可以是聚合函数
通常是Mean/Max/Sum

邻居消息发送

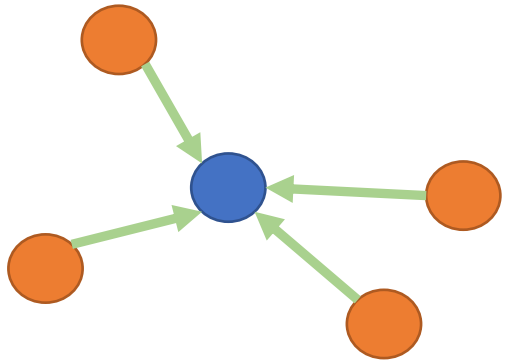
Message Passing

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u)|u \in N(v)\})$$

Message Passing流程

1. 邻居的发送信息

2. F可以看作是目标接收信息的方式



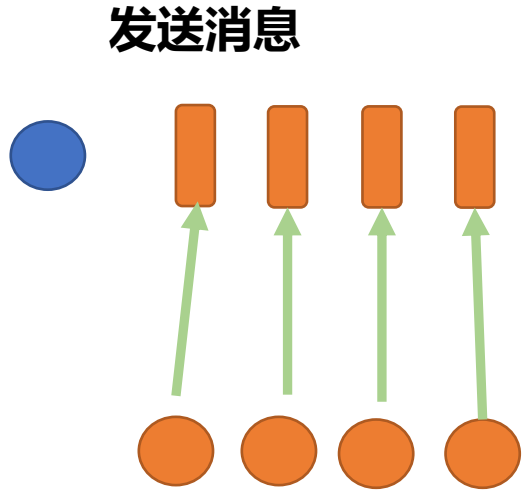
Message Passing

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u)|u \in N(v)\})$$

Message Passing流程

1. 邻居的发送信息

2. F可以看作是目标接收信息的方式



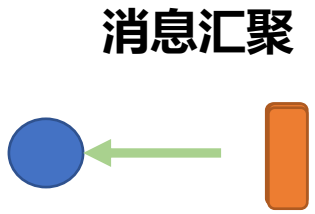
Message Passing

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u)|u \in N(v)\})$$

Message Passing流程

1. 邻居的发送信息

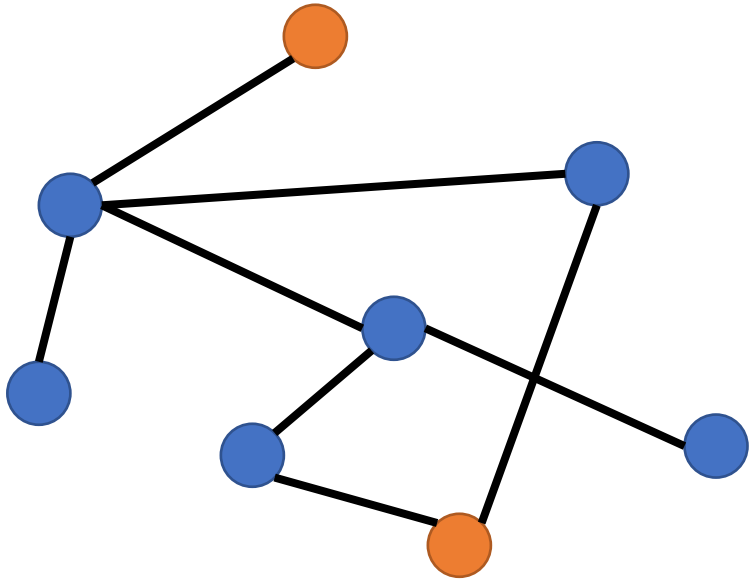
2. F可以看作是目标接收信息的方式



GCN: F = 基于度的加权求和
GAT: F = 基于 Attention 的 加权求和

Message Passing

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u) | u \in N(v)\})$$

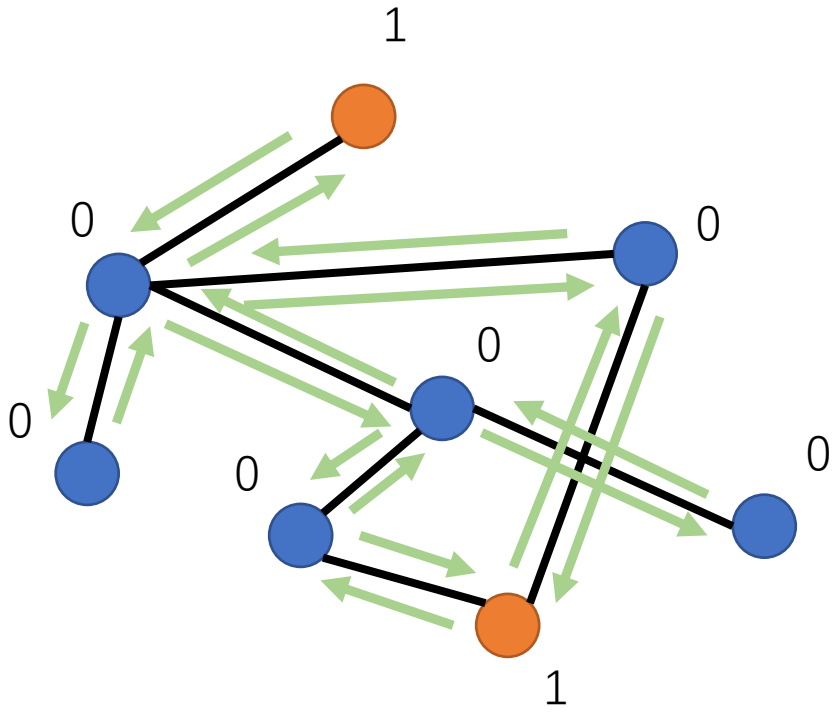


用Message Passing的范式解决问题

1. 哪些点的邻居存在橙色点（包括自己）
2. 每个节点附近平均有多少个橙色点（包括自己）
3. 哪些点的半径为2范围内存在橙色点 (包括自己)

Message Passing

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u) | u \in N(v)\})$$

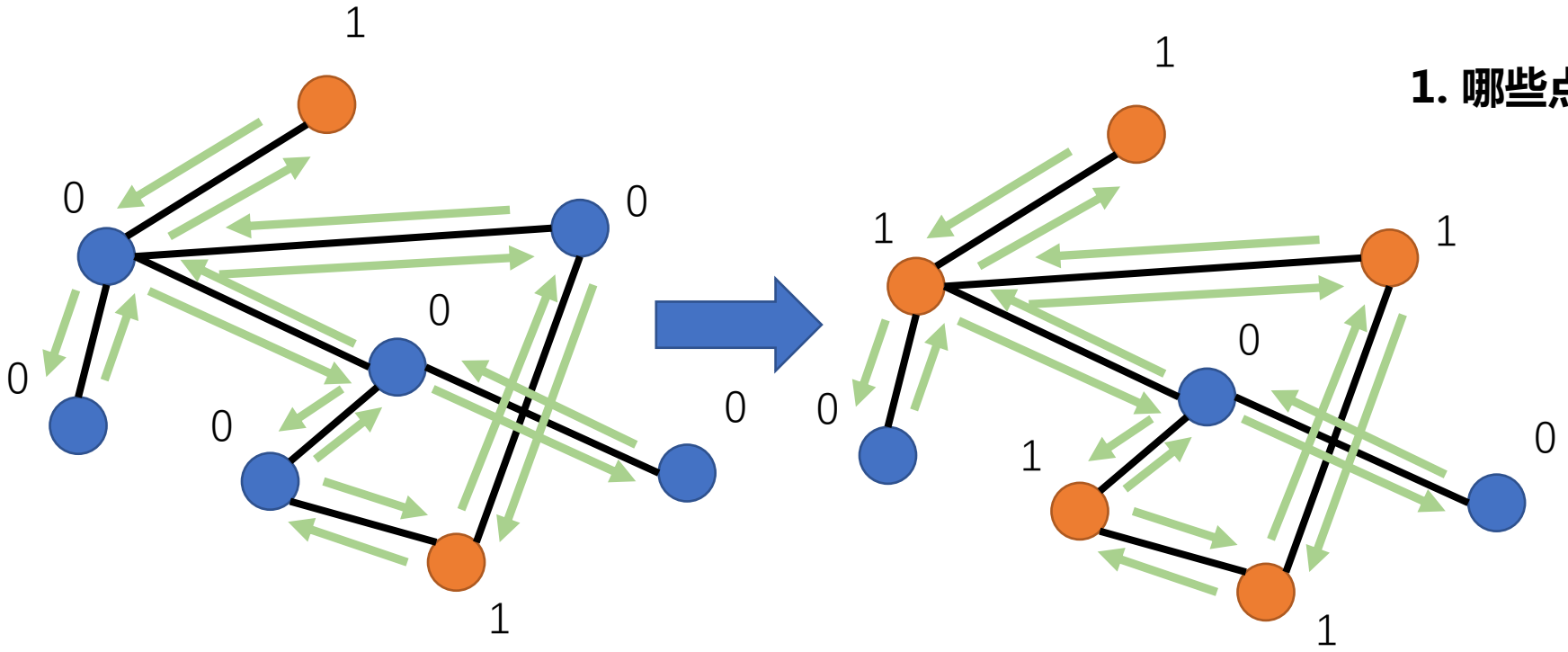


1. 哪些点的邻居存在橙色点（包括自己）

$\mathcal{F}$ 为求Max

Message Passing

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u) | u \in N(v)\})$$



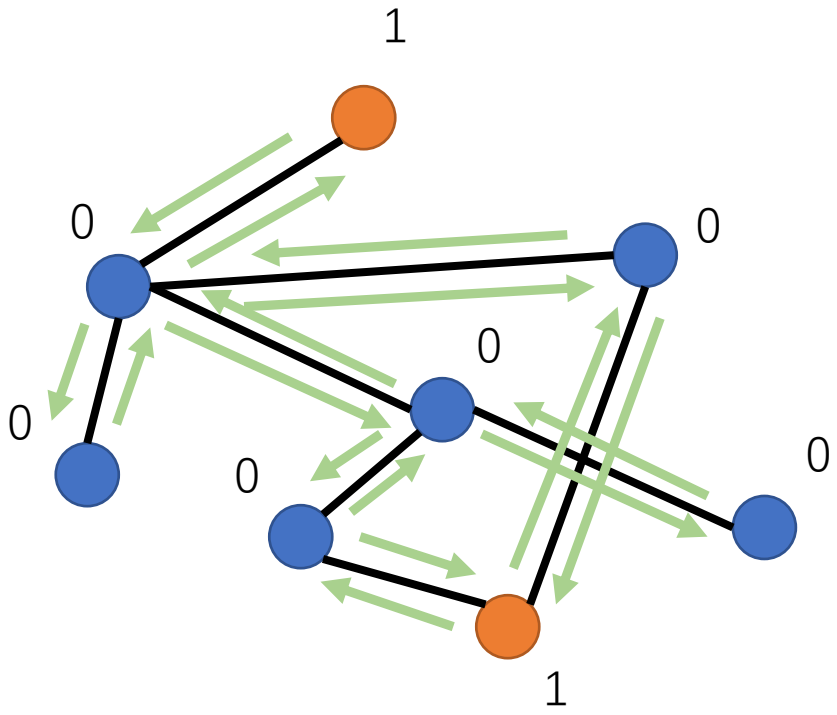
1. 哪些点的邻居存在橙色点（包括自己）

$\mathcal{F}$ 为求Max

邻居节点有1的有5个

Message Passing

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u)|u \in N(v)\})$$

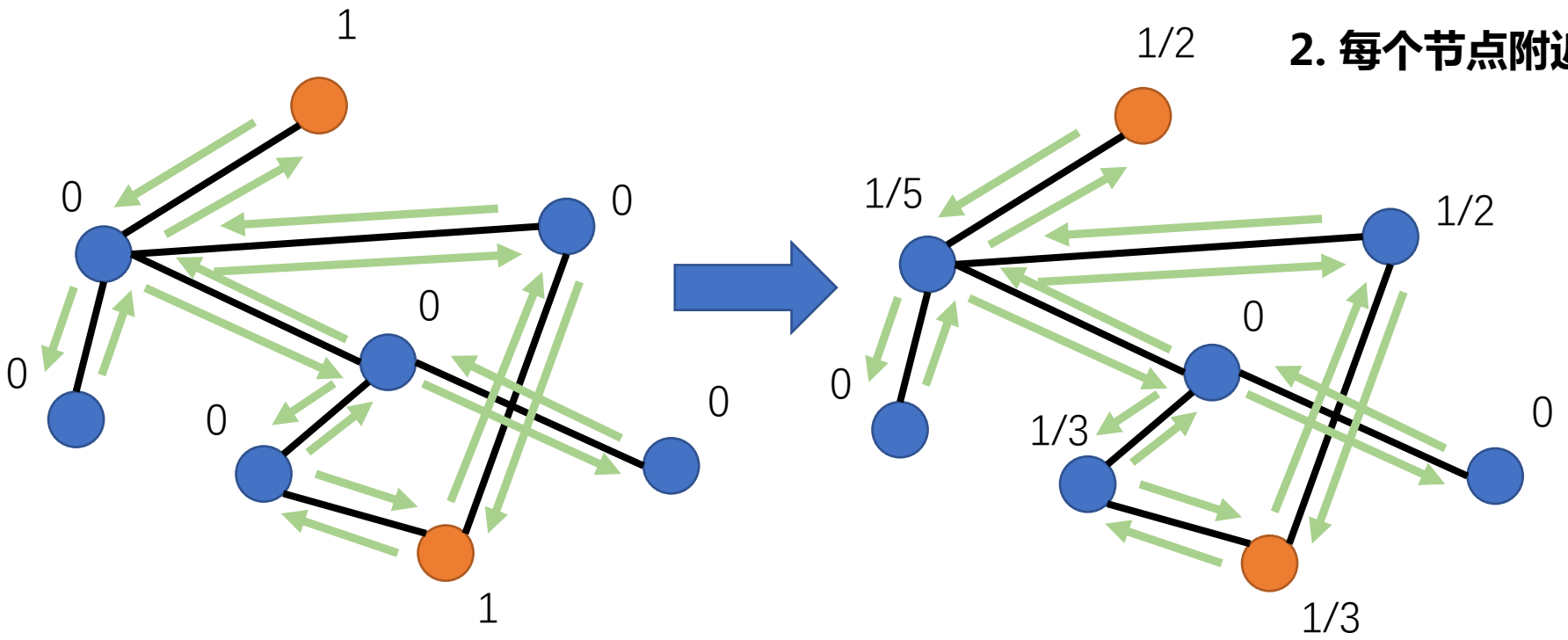


2. 每个节点附近平均有多少个橙色点(包括自己)

$\mathcal{F}$ 为求均值Mean

Message Passing

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u) | u \in N(v)\})$$

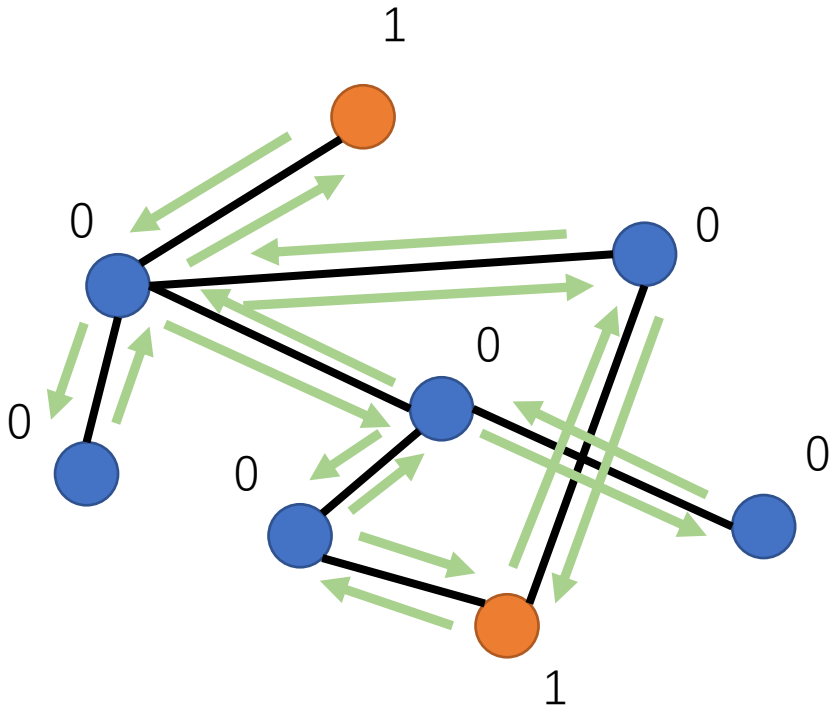


2. 每个节点附近平均有多少个橙色点(包括自己)

$\mathcal{F}$ 为求均值Mean

Message Passing

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u) | u \in N(v)\})$$

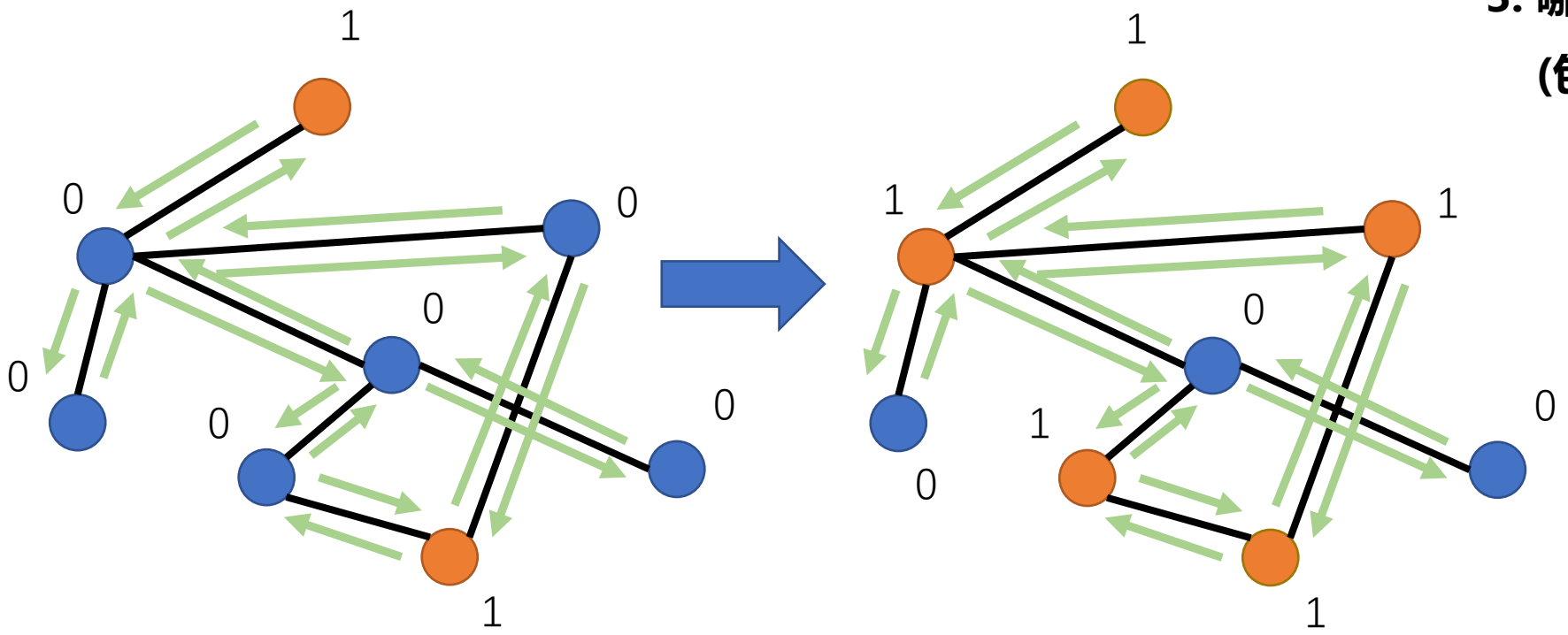


3. 哪些点的半径为2范围内存在橙色点(包括自身)

$\mathcal{F}$ 为求Max

Message Passing

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u) | u \in N(v)\})$$



3. 哪些点的半径为2范围内存在橙色点(包括自身)

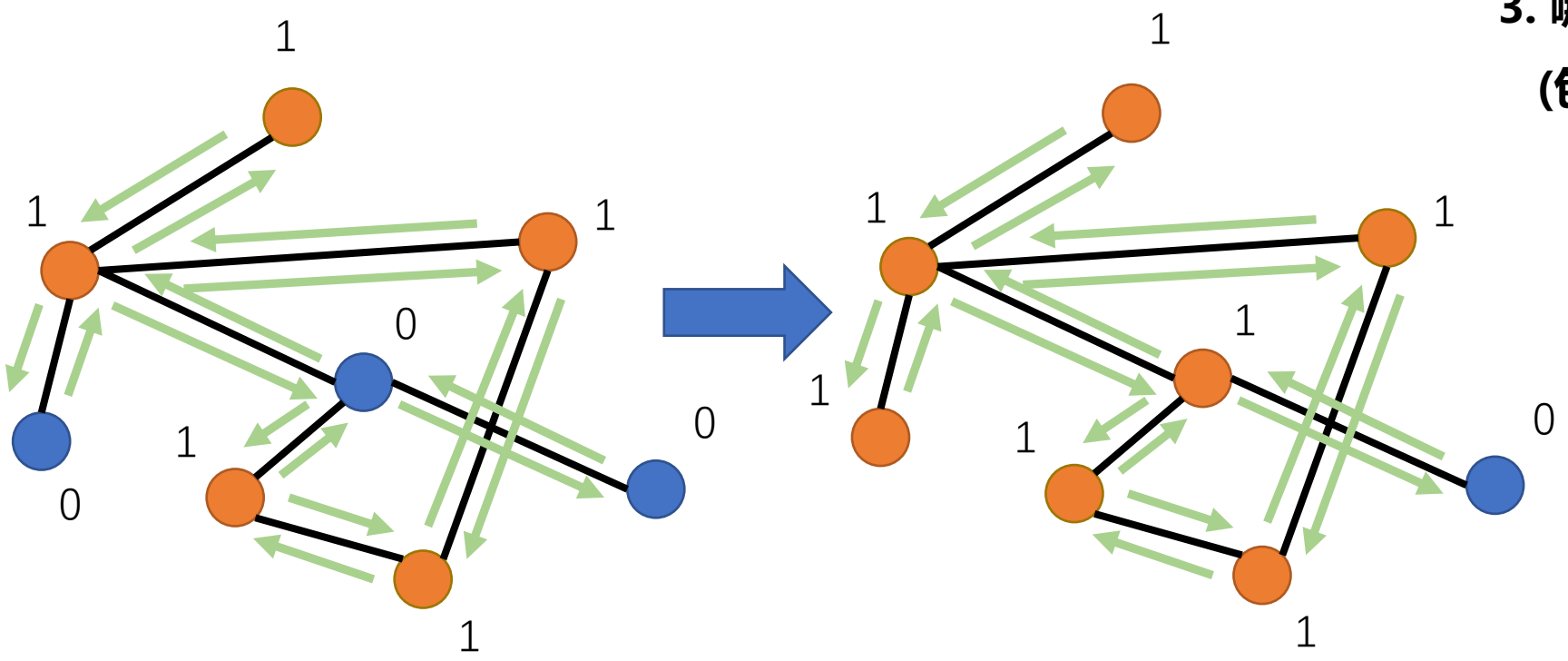
$\mathcal{F}$ 为求Max

第一次迭代

1. 一阶邻居有橙色点

Message Passing

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u) | u \in N(v)\})$$



3. 哪些点的半径为2范围内存在橙色点（包括自身）

$\mathcal{F}$ 为求Max

第一次迭代

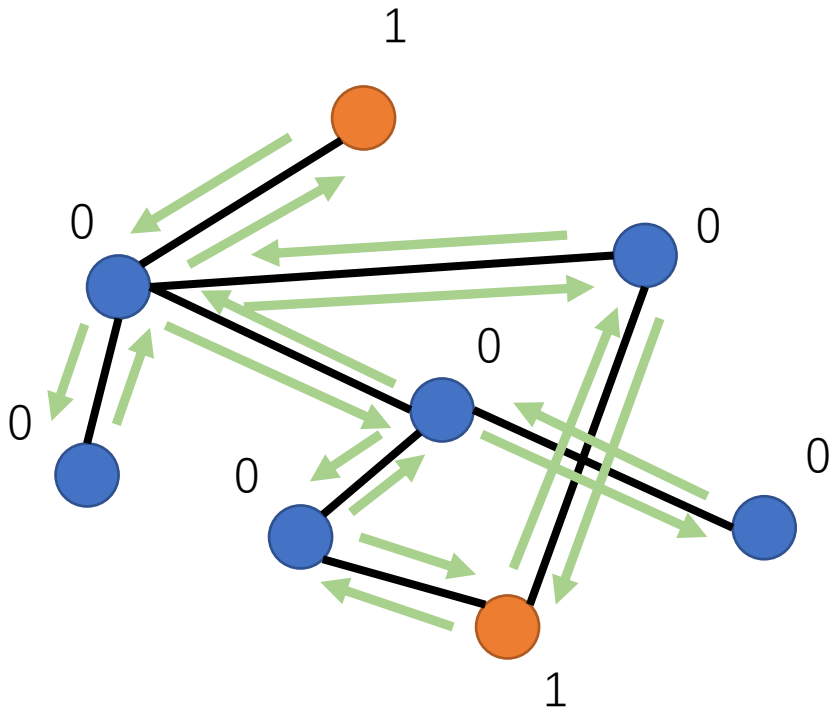
1. 一阶邻居有橙色点

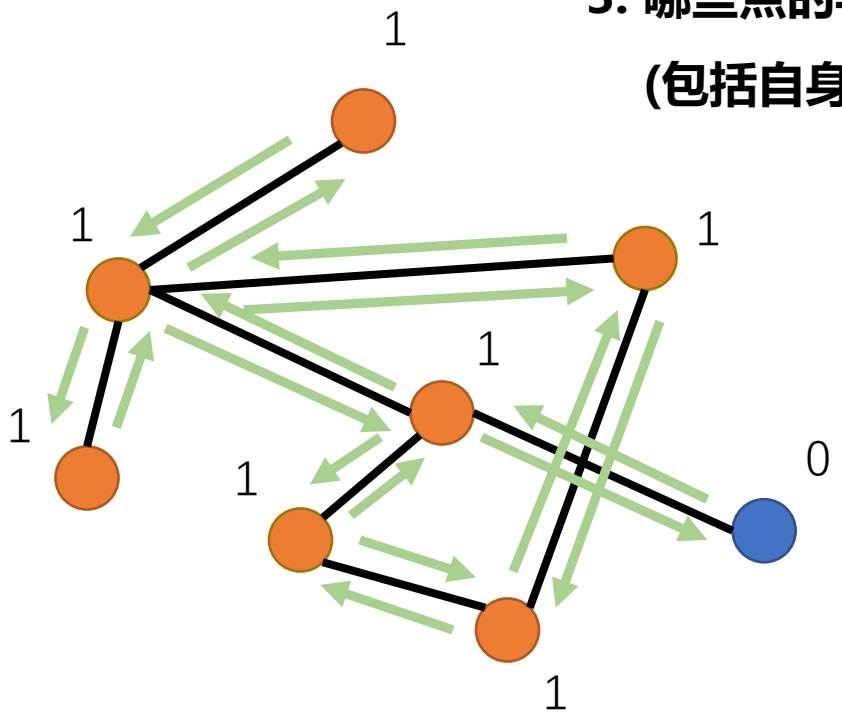
第二次迭代

2. 二阶邻居有橙色点

Message Passing

$$h_l^{(t)}(v) = f(h_l^{(t-1)}, \mathcal{F}\{h_l^{(t-1)}(u)|u \in N(v)\})$$

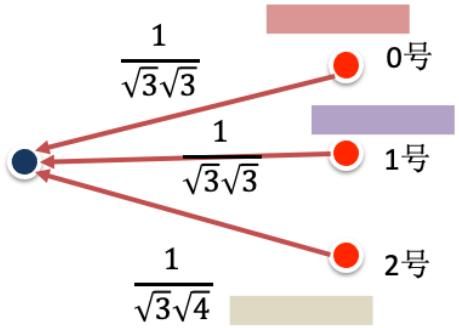




3. 哪些点的半径为2范围内存在橙色点(包括自身)

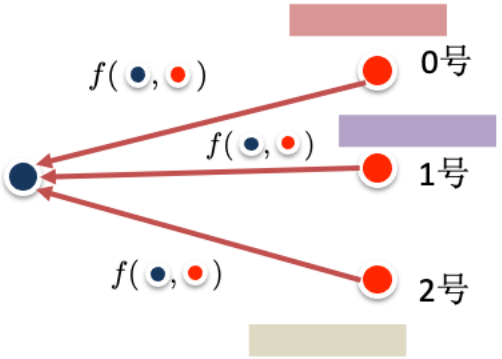
总结

1. GCN算法介绍



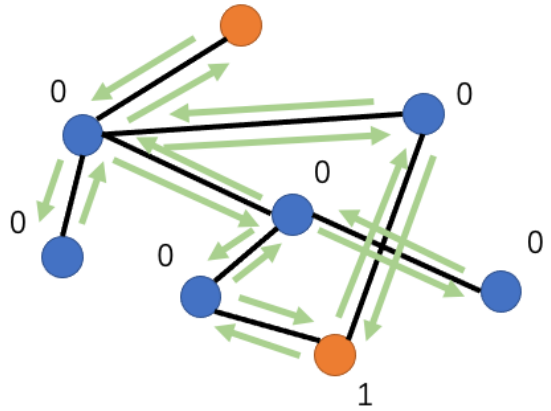
GCN

2. GAT算法介绍



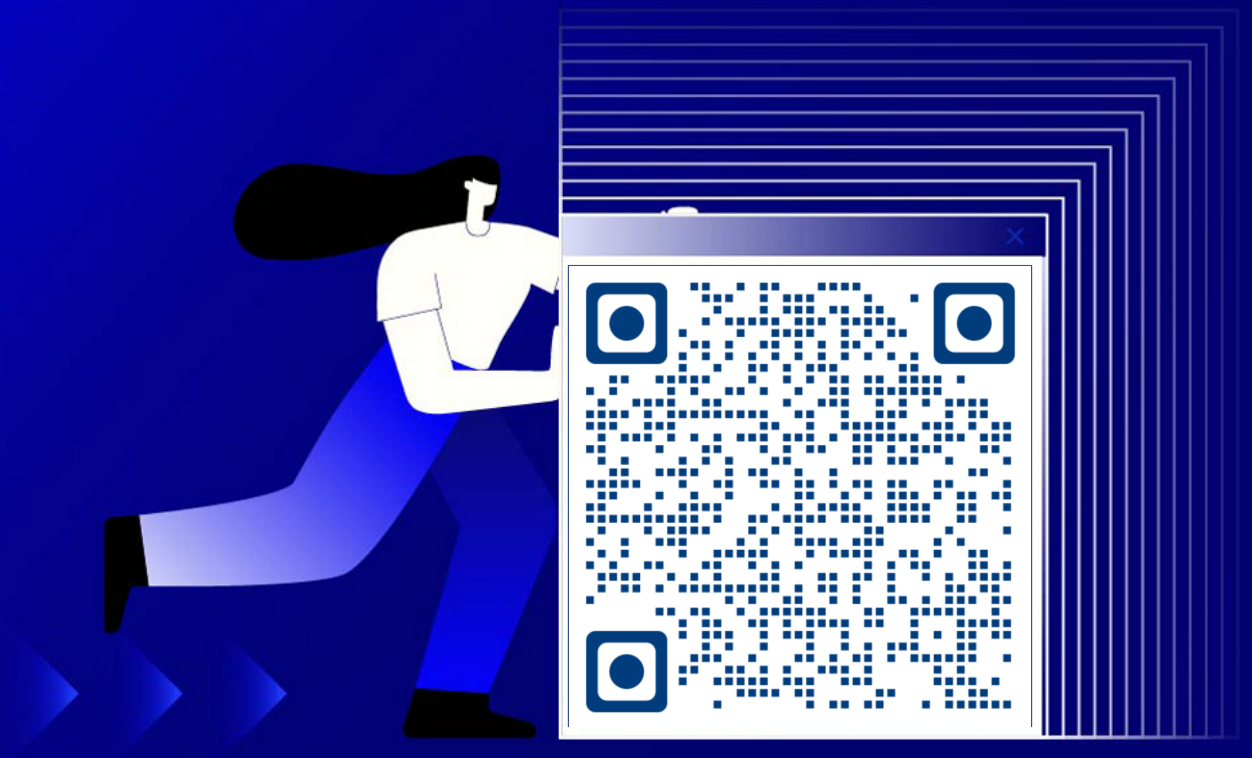
GAT

3. Message Passing 消息传递算法



课后作业

实现 GAT 模型的消息传递接口



PGL github

谢谢观看