图卷积网络(GCN)新手村完全指南


Orangrass
机器学习/网球爱好者

前言

图卷积网络Graph Convolutional Network，简称GCN，最近两年大热，取得不少进展。

最近，清华大学孙茂松教授组在 arXiv 发布了论文***Graph Neural Networks: A Review of Methods and Applications***，作者对现有的 GNN 模型做了详尽且全面的综述。GCN就是GNN中的一种重要的分支。

但是对于GCN的萌新，看着这篇综述可能还是会困难重重、不知所措。

写这篇文章的目的，就是帮助萌新们掌握GCN的重要概念和理论，走出新手村。

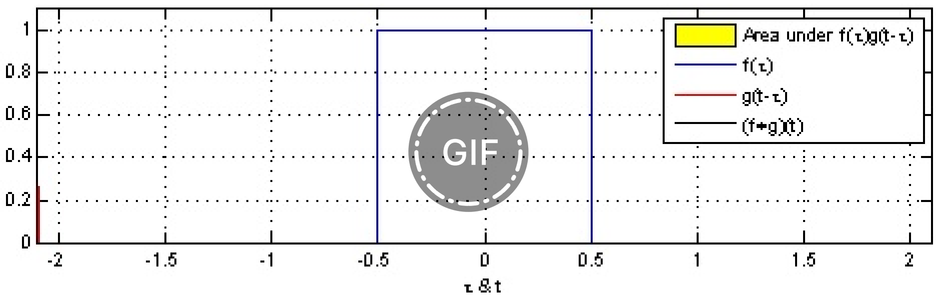
什么是Convolution

Convolution的数学定义是

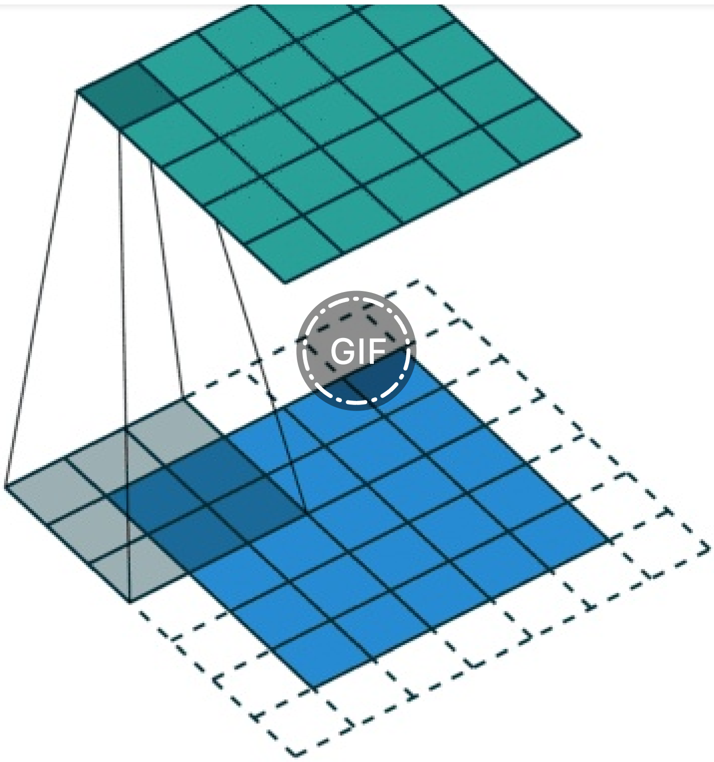
$$(f * g)(t) = \int_{\mathbb{R}} f(x)g(t-x)dx$$

一般称g为作用在f上的filter或kernel

一维的卷积示意图如下

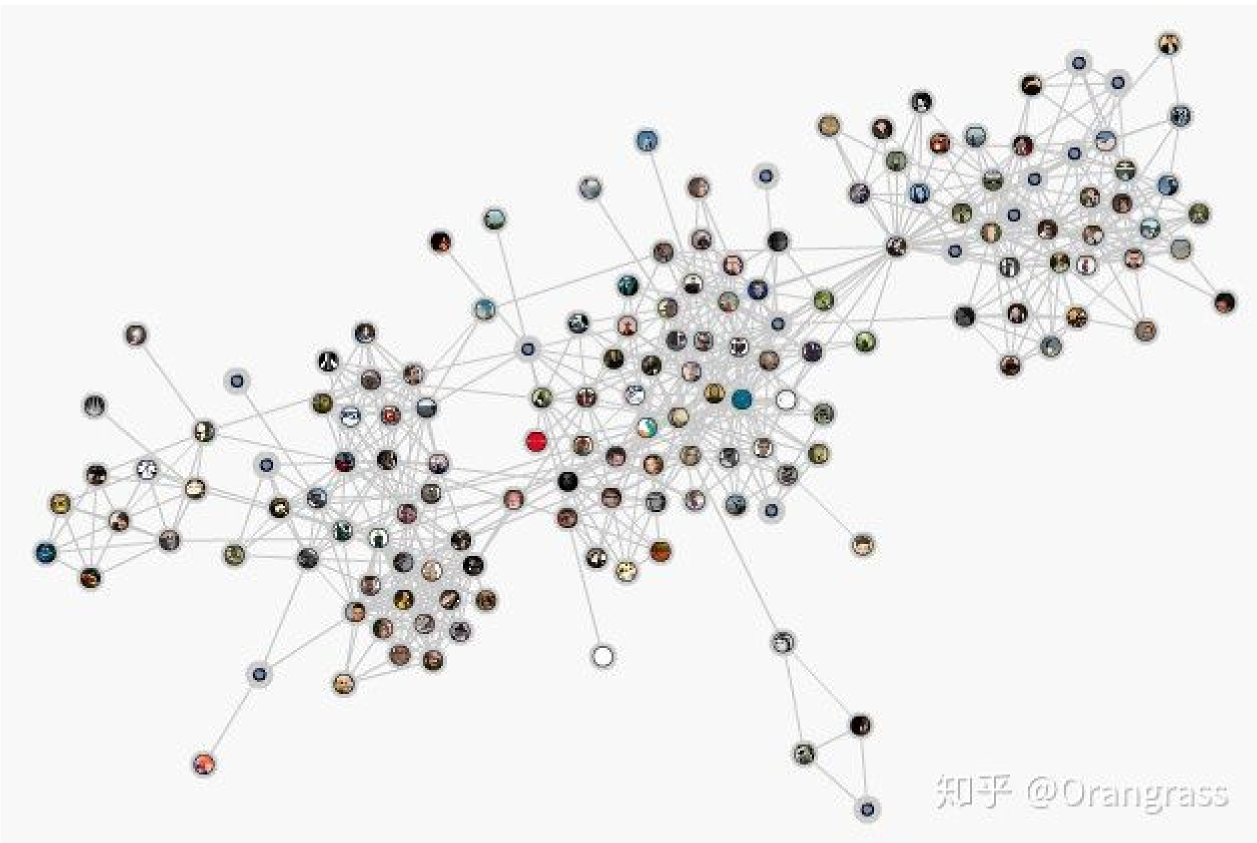


大家常见的CNN二维卷积示意图如下



在图像里面卷积的概念很直接，因为像素点的排列顺序有明确的上下左右的位置关系。

那在抽象的graph里面卷积该怎么做呢



比如这个社交网络抽象出来的graph里面，有的社交vip会关联上万的节点，这些节点没有空间上的位置关系，也就没办法通过上面给出的传统

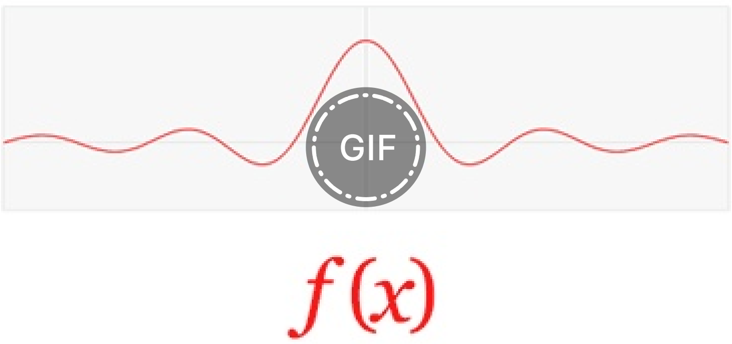
Fourier变换

为了解决graph上卷积计算的问题，我们给出第二个装备--Fourier变换。

先上结论，根据卷积定理，卷积公式还可以写成

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$$

这样我们只需要定义graph上的fourier变换，就可以定义出graph上的convolution变换。



1ucasvb.tumblr.com

好的，先来看下Fourier变换的定义

$$\mathcal{F}\{f\}(v) = \int_{\mathbb{R}} f(x) e^{-2\pi i x \cdot v} dx$$

Inverse Fourier变换则是

$$\mathcal{F}^{-1}\{f\}(x) = \int_{\mathbb{R}} f(v) e^{2\pi i x \cdot v} dv$$

根据Fourier变换及其逆变换的定义，下面我们来证明一下卷积定理

我们定义 h 是 f 和 g 的卷积，那么

$$h(z) = \int_{\mathbb{R}} f(x) g(z - x) dx$$

$$
\begin{aligned}
&= \int_{\mathbb{R}} h(z)e \quad dz \\
&= \int_{\mathbb{R}}\int_{\mathbb{R}} f(x)g(z-x)e^{-2\pi iz\cdot v}dxdz \\
&= \int_{\mathbb{R}} f(x)(\int_{\mathbb{R}} g(z-x)e^{-2\pi iz\cdot v}dz)dx
\end{aligned}
$$

带入 $y = z - x$; $dy = dz$

$$
\begin{aligned}
\mathcal{F}\{f*g\}(v) &= \int_{\mathbb{R}} f(x)(\int_{\mathbb{R}} g(y)e^{-2\pi i(y+x)\cdot v}dy)dx \\
&= \int_{\mathbb{R}} f(x)e^{-2\pi ix\cdot v}(\int_{\mathbb{R}} g(y)e^{-2\pi iy\cdot v}dy)dx \\
&= \int_{\mathbb{R}} f(x)e^{-2\pi ix\cdot v}dx\int_{\mathbb{R}} g(y)e^{-2\pi iy\cdot v}dy \\
&= \mathcal{F}\{f\}(v)\cdot\mathcal{F}\{g\}(v)
\end{aligned}
$$

最后对等式的两边同时作用 $\mathcal{F}^{-1}$ ，得到

$$f*g = \mathcal{F}^{-1}\{\mathcal{F}\{f\}\cdot\mathcal{F}\{g\}\}$$

Laplacian算子

一波未平，又来一个陌生的概念。。。

不要担心，这是出新手村之前的最后一件装备了。

一阶导数定义为

laplacian算子简单的来说就是二阶导数

$$\Delta f(x) = \lim_{h \to 0} \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

那在graph上，我们可以定义一阶导数为

$$f'_{*g}(x) = f(x) - f(y)$$

其中y是x的邻居节点

那么对应的Laplacian算子可以定义为
$$\Delta_{*g} f'(x) = \Sigma_{y \sim x} f(x) - f(y)$$

定义 D 是 $N \times N$ 的度数矩阵(degree matrix)

$$D(i,j) = \begin{cases} d_i & \text{if } i = j \\ 0 & otherwise \end{cases}$$

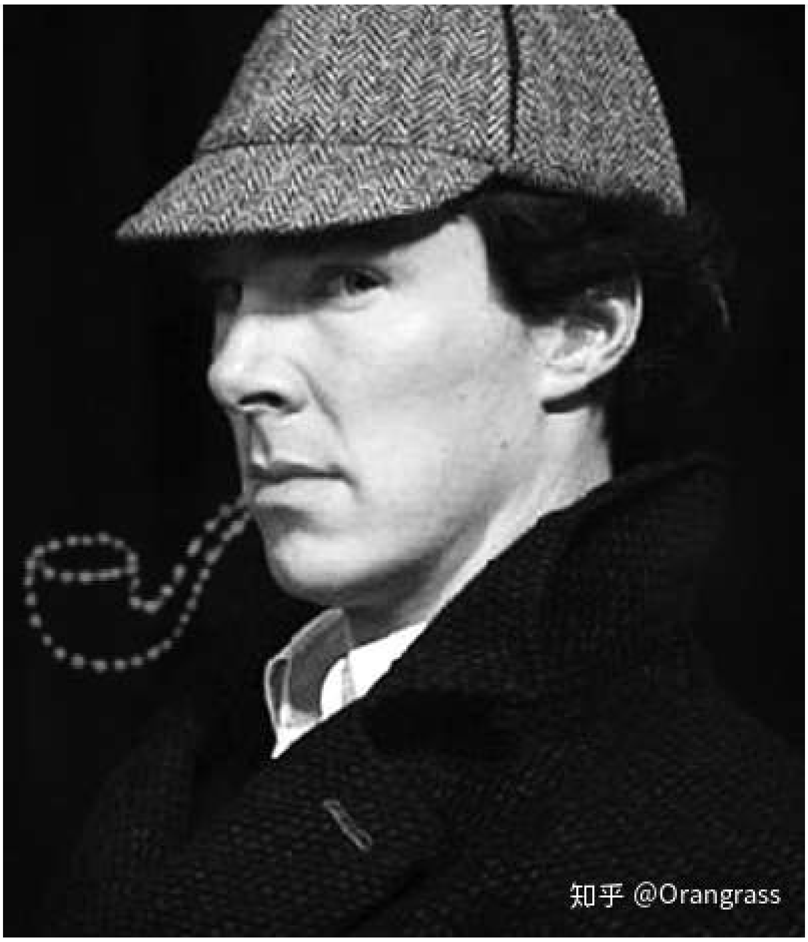
定义 A 为 $N \times N$ 邻接矩阵(adjacency matrix)

$$A(i,j) = \begin{cases} 1 & \text{if } x_i \sim x_j \\ 0 & otherwise \end{cases}$$

那么图上的Laplacian算子可以写成

$$L = D - A$$

标准化之后得到 $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$



定义Laplacian算子的目的是为了找到Fourier变换的基

比如传统Fourier变换的基 $e^{2\pi i x\cdot v}$ 就是Laplacian算法的一组特征向量

$\Delta e^{2\pi i x\cdot v}=\lambda e^{2\pi i x\cdot v}$ ，λ 是一个常数

那么图上的Fourier基就是 L 矩阵的n个特征向量 $U=[u_1\ldots u_n]$ ， L 可以分解为

$$L=U\Lambda U^T$$

其中 Λ 是特征值组成的对角矩阵

	传统Fourier变换	Graph Fourier变换
$Fourier$变换基	$e^{-2\pi ixv}$	U^T
逆$Fourier$变换基	$e^{2\pi ixv}$	U
维度	∞	点的个数n

那么Graph Fourier变换可以定义为

$l=1$

其中 $f(i)$ 可以看做是作用在第 i 个点上的signal，用向量 $x = (f(1) \dots f(n)) \in \mathbb{R}^n$ 来表示

u_l^* 是 u_l 的的对偶向量， u_l^* 是矩阵 U^T 的第 l 行， u_l 是矩阵 U 的第 l 行。

那么我们可以用矩阵形式来表示Graph Fourier变换

$$\mathcal{GF}\{x\} = U^T x$$

类似的Inverse Graph Fourier变换定义为

$$\mathcal{IGF}\{\hat{f}\}(i) = \sum_{l=0}^{n-1} \hat{f}(\lambda_l) u_l(i)$$

它的矩阵形式表达为

$$\mathcal{IGF}\{x\} = Ux$$

推导Graph Convolution

走到这里，我们已经获得了新手村的所有装备，下面就开始推导GCN的公式。

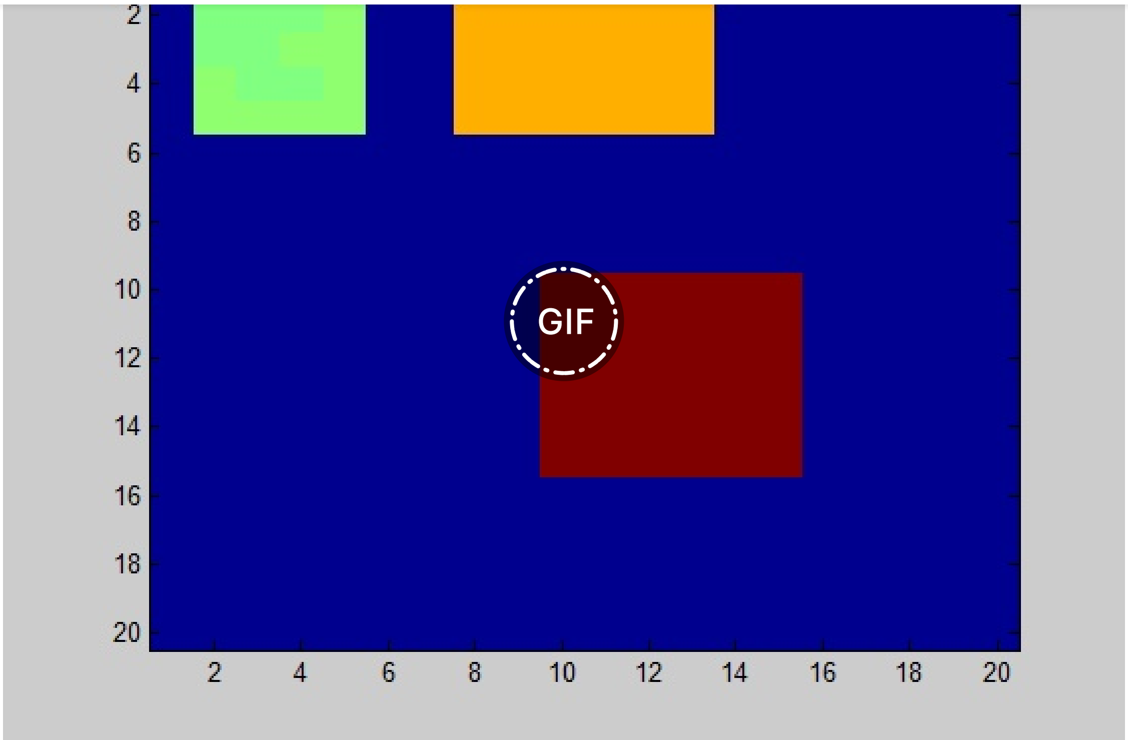
还记得我们之前提到的先上卷积定理吗

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$$

那么图的卷积公式可以表示为

$$g * x = U(U^T g \cdot U^T x)$$

作为图卷积的filter函数 g ，我们希望具有很好的局部性。就像CNN模型里的filter一样，只影响到一个像素附近的像素。那么我们可以把 g 定义成一个laplacian矩阵的函数 $g(L)$



作用一次laplacian矩阵相当于在图上传播了一次邻居节点。进一步我们可以把 $U^T g$ 看做是 $g_\theta(\Lambda)$ 一个laplacian特征值的函数，参数为 θ

改写上面的图卷积公式，我们就可以得到论文SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS

https://arxiv.org/pdf/1609.02907.pdf
arxiv.org

的公式(3)

$$g_\theta * x = U g_\theta U^T x = U g_{\theta'}(\Lambda) U^T x$$

可以看到这个卷积计算的复杂度是非常高的，涉及到求laplacian矩阵的特征向量，和大量的矩阵计算。下面我们考虑对filter函数做近似，目标是省去特征向量的求解

$$g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda})$$

其中 T_k 是Chebyshev多项式。这里可以把简单 $g_\theta(\Lambda)$ 简单看成是 Λ 的多项式。

因为 $U \Lambda^k U^T = (U \Lambda U^T)^k = L^k$

$k=0$

设定 $K=1$ 那卷积公式可以简化为

$$\begin{aligned} g_{\theta'} * x &\approx \theta(I_N + L)x \\ &= \theta(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}})x \end{aligned}$$

令 $\tilde{A} = A + I_N$ ， $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$

$$g_{\theta'} * x = \theta(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}})x$$

那么再加上激活层，我们就可以得到最终的GCN公式

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

编辑于 2019-01-11

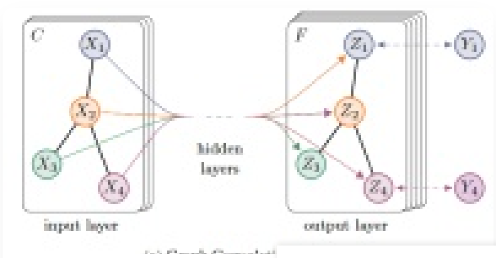
深度学习（Deep Learning）

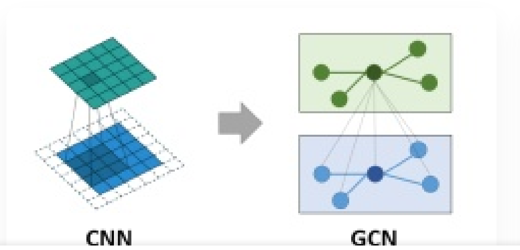
文章被以下专栏收录

Deep Learner

进入专栏

推荐阅读





图神经网络入门（

80 条评论

⇌ 切换为时间排序

写下你的评论...

精选评论（1）

知乎用户 1 年前

作者写得非常好，不过有一点小漏洞。laplace算子是一阶梯度的散度，散度可以认为是某个区域内流通量的求和，对每个graph上的点，因为没有大小，流通量就可以直接把所有跟该点相连节点的梯度相加起来。这样子推导拉普拉斯矩阵可能会更好哦。参考资料：youtube.com/watch?... 散度定义。

29 查看回复

评论（80）

Admu 1 年前

很清晰

2

知乎用户 1 年前

感谢🙏，学习了

2

屈里昂 1 年前

目前看到最好的一篇，解决了最近读那篇综述文献的疑惑，谢谢我作者！

5

stucou 1 年前

公式里没有边上的特征，怎样包含边上的特征

1

毛毛雨啦 回复 stucou 1 年前

同问

1

Waden 回复 stucou 1 年前

看看sklearn.metrics.pairwise.pairv

赞

感谢🙏

赞

知乎用户 1 年前

作者写得非常好，不过有一点小漏洞。laplace算子是一阶梯度的散度，散度可以认为是某个区域内流通量的求和，对每个graph上的点，因为没有大小，流通量就可以直接把所有跟该点相连节点的梯度相加起来。这样子推导拉普拉斯矩阵可能会更好哦。参考资料：youtube.com/watch?... 散度定义。

29

谢宇 回复 知乎用户 10 个月前

莫非阁下是流体力学出来的[捂脸]

赞

知乎用户 1 年前

你在原文中提到的也是正确的 norbertwiener.umd.edu/R...

1

杰奏 1 年前

看了这个，感觉入不了门了，哈哈[捂脸]

8

清峪沟的砍柴人 回复 杰奏 9 个月前

我也觉得入不了门。大佬有什么入门建议？相关数学咋补啊

1

杰奏 回复 清峪沟的砍柴人 9 个月前

我现在入门了，过程极其残忍

赞

展开其他 2 条回复

Knife 1 年前

图很清晰，可是公式中符号的含义都没有解释，小白表示根本看不懂。

11

刘欣悦 回复 Knife

清峪沟的砍柴人 回复 Knife 9 个月前

加1

1

阿赛 1 年前

我连卷积公式都看不懂【捂脸】

4

冬之晓 1 年前

请问人头像下面的第二句话："Laplacian算法"是不是应该改成"Laplacian算子"更合适一点？还是说"Laplacian算法"和"Laplacian算子"是一个意思？

1

Orangrass (作者) 回复 冬之晓 1 年前

应该是Laplacian算子，笔误了，谢谢指正

赞

maja 1 年前

卷积本质是加权平均，在一般图 和 3D点云里面更接近的是聚类。其实lalacian的所有理论都可以在普积累找到，大部分做PDE出身搞图像都是清楚的。清华那个教授的GCN不过是炒冷饭了，这个技术我们一直在用，详情见普聚类。

1

Genome 回复 maja 11 个月前

GCN不是清华那个教授提出的，那个教授只是写了个GCN review，怼人要有依据啊

5

RaySir 回复 Genome 11 个月前

这个世界有些人会错误地认为知识系他家的，自己学过了用过了恍惚就不许所有人学习与致用，更是要将所有教人之人拉出去打死一般。

4

展开其他 2 条回复

魔方的梦想 12 个月前

博主您好，请问在推倒图卷积的那部分写的是g*x=U(U^Tg.Ux)，如果是这样的话那么滤波器不应该是x吗？为什么会是g呢，而且如果这样写，在矩阵计算过程中UU^T不就是E吗？，我哪里理解有错吗？谢谢。

赞

赞

月下无痕夜 12 个月前

服气看得懂的 俺数学功底差远了

2

此间尘缘 回复 月下无痕夜 11 个月前

前几天想法和你一样，不过多找几篇博客就能看明白，建议把散度、拉普拉斯矩阵、谱聚类、拉普拉斯映射一起看，感觉能入门

1

Knife 11 个月前

终于搞清楚了 T T 感谢这篇文章

赞

forgetmenot 11 个月前

最后一个公式w是什么?

1

栗子 回复 forgetmenot 1 个月前

最终所要求的矩阵

赞

jaycase 11 个月前

请问 “作用一次laplacian矩阵相当于在图上传播了一次邻居节点”这句话是什么意思

赞

爱媳妇的好男人 11 个月前

文章写得很好，整体思路理清了，部分公式还需要进一步的细扣，建议作者像写论文一样对文章公式里的每个符号进行解释，瑕不掩瑜，值得一看。

赞

RaySir 回复 爱媳妇的好男人 11 个月前

论文并不需要对每个符号都进行解释啊，建议作者**像写给自家小朋友阅读一样——对文章公式里的每个符号进行解释**：这样或许会好D。

赞

井井井井井井井井 回复 RaySir 3 个月前

论文不需要解释? ? ? ? ? ? ? ? ?

匿名 11 个月前

估计要吓到一批初学者啊，对于门外汉能不能给个通俗的解释，gcn能用来干嘛？

6