

图形神经网络:方法与应用综述

周杰¹、干曲崔¹、张正艳¹、程阳¹、王¹、孙茂松

摘要—许多学习任务需要处理包含元素间丰富关系信息的图形数据。对物理系统建模、学习分子指纹、预测蛋白质界面和对疾病进行分类需要一个模型来从图形输入中学习。在从文本和图像等非结构数据中学习等其他领域，对句子的依存树和图像的场景图等提取的结构进行推理是一个重要的研究课题，也需要图形推理模型。图形神经网络(GNNs)是连接主义模型，通过图形节点之间的消息传递来捕捉图形的相关性。与标准神经网络不同，图形神经网络保留了一种状态，可以以任意深度表示其邻域的信息。尽管发现原始的神经网络很难针对固定点进行训练，但网络体系结构、优化技术和并行计算的最新进展已经使他们能够成功学习。近年来，基于图形神经网络变体的系统，如图形卷积网络(GCN)、图形注意网络(GAT)、选通图形神经网络(GGNN)，在上述许多任务中表现出突破性的性能。在本综述中，我们对现有的图神经网络模型进行了详细的回顾，对其应用进行了系统的分类，并提出了四个有待进一步研究的问题。

索引术语—深度学习、图形神经网络

一 正式介绍

图是一种数据结构，它模拟一组对象(节点)及其关系(边)。近年来，基于机器学习的图形分析研究因其强大的表达能力而受到越来越多的关注。图可以用来表示跨越不同领域的大量系统，包括社会科学(社会网络)[1]、[2]、自然科学(物理系统[3]、[4]和蛋白质-蛋白质相互作用网络[5])、知识图[6]和许多其他研究领域[7]。作为一种用于机器学习的独特的非欧几里德数据结构，图分析侧重于节点分类、链接预测和聚类。图形神经网络(GNNs)是基于深度学习的方法，在图形域上运行。由于其令人信服的性能和较高的可解释性，GNN 近年来已成为一种广泛应用的图形分析方法。在下面的段落中，我们将说明图形神经网络的基本动机。

GNNs 的第一个动机源于卷积神经网络(CNNs) [8]。氯化氮有能力

*表示同等贡献。

- 周杰, 张正艳, 程阳, (对应作者), 孙茂松, 清华大学计算机科学与技术系, 北京 100084。电子邮件: {cheng-ya14}@mails.tsinghua.edu.cn 张家界 18, 张正艳 14, {liuzy}@tsinghua.edu.cn 短信。
- 崔干渠, 清华大学物理系, 北京 100084。电子邮件: cgq15@mails.tsinghua.edu.cn。
- 王、是中国深圳腾讯公司的员工。电子邮件: {harrychli}@tencent.com 范代旺

提取多尺度局部空间特征并将其组合以构建高度表达的表示，这导致了几乎所有机器学习领域的突破，并开启了深度学习的新时代[9]。随着我们对 CNNs 和图形的深入研究，我们发现了 CNNs 的关键:本地连接、共享权重和多层的使用[9]。这些在解决图域问题中也很重要，因为 1)图是最典型的局部连通结构。2)与传统的谱图理论相比，共享权重降低了计算成本[10]。3)多层结构是处理分层模式的关键，它捕捉了各种大小的特征。然而，神经网络只能在像图像(2D 网格)和文本(1D 序列)这样的常规欧几里德数据上运行，而这些数据结构可以被视为图形的实例。因此，很容易想到找到 CNNs 对图形的推广。如图 2 所示。1、难以定义局域化卷积滤波器和汇集算子，阻碍了 CNN 从欧氏域向非欧氏域的转化。

F

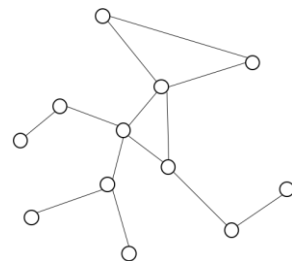
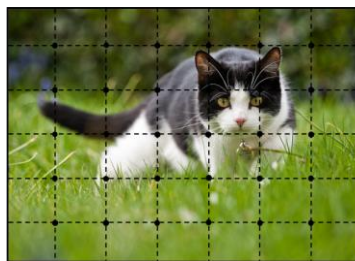


图. 1.左图:欧氏空间中的图像。右图:非欧空间中的图形

另一个动机来自图嵌入[11]–[15]，它学习在低维向量中表示图的节点、边或子图。在图形分析领域，传统的机器学习方法通常依赖于手工设计的特征，并且受限于其不灵活性和高成本。遵循表示学习的思想和单词嵌入的成功[16]，被认为是第一个基于表示学习的图形嵌入方法的 DeepWatch[17]将 SkipGram 模型[16]应用于生成的随机行走。

类似的方法如 node2vec [18]、LINE [19]和 TADW [20]也取得了突破。然而，这些方法有两个严重的缺点[12]。首先，编码器中的节点之间没有共享参数，这导致计算效率低下，因为这意味着参数的数量随着节点的数量线性增长。第二，直接嵌入方法缺乏泛化能力，这意味着它们不能处理动态图或推广到新的图。

基于神经网络和图嵌入，提出了图神经网络来集中聚集来自图结构的信息。因此，他们可以对由元素及其依赖性组成的输入和/或输出进行建模。此外，图形神经网络可以同时用 RNN 核对图形上的扩散过程进行建模。

在接下来的部分，我们解释了为什么图形神经网络值得研究的根本原因。首先，标准的神经网络，如神经网络和神经网络，不能正确处理图形输入，因为它们按照特定的顺序堆叠节点的特征。然而，图中的节点没有自然的顺序。为了完整地呈现一个图，我们应该像 CNNs 和 RNNs 一样遍历所有可能的顺序作为模型的输入，这在计算时是非常冗余的。为了解决这个问题，gnn 分别在每个节点上传播，忽略节点的输入顺序。换句话说，GNNs 的输出对于节点的输入顺序是不变的。其次，图中的一条边表示两个节点之间的依赖信息。在标准的神经网络中，依赖信息只是作为节点的特征。然而，GNNs 可以在图形结构的指导下进行传播，而不是将其用作功能的一部分。一般来说，神经网络通过邻域状态的加权和功能

更新节点的隐藏状态。第三，推理是高级人工智能的一个非常重要的研究课题，人脑中的推理过程几乎都是基于从日常经验中提取的图形。标准的神经网络已经显示出通过学习数据的分布来生成合成图像和文档的能力，而它们仍然不能从大的实验数据中学习推理图。然而，神经网络探索从场景图片和故事文档等非结构数据生成图形，这可以是进一步高级人工智能的强大神经模型。最近，事实证明，一个没有受过训练的 GNN 人用简单的架构也表现良好[21]。

对图神经网络的研究已有多综述。[22]提出了一个统一的框架 MoNet，将 CNN 架构推广到非欧几里德域(图和流形)，该框架可以推广图[2]，[23]上的几种谱方法以及流形上的一些模型[24]，[25]。[26]提供了对几何深度学习的全面回顾，介绍了它的问题、困难、解决方案、应用和未来方向。[22]和[26]集中于将卷积推广到图或流形，然而在本文中，我们只关注图中定义的问题，我们还研究了图神经网络中使用的其他机制，如门机制、注意机制和跳过连接。[27]提出了消息传递神经网络(MPNN)，它可以推广几种图神经网络和图卷积网络方法。[28]提出了非局部神经网络，它统一了几种“自我注意”式的方法。然而，在最初的论文中，该模型没有在图上明确定义。专注于特定的应用领域，[27]和[28]仅给出了如何使用它们的框架概括其他模型的例子，它们没有提供对其他图形神经网络模型的评论。[29]回顾了图形注意力模型。[30]提出了图形网络框架，该框架具有很强的推广其他模型的能力。然而，图网络模型是高度抽象的，并且[30]仅给出应用的粗略分类。

[31]和[32]是最新的 GNNs 调查论文，它们主要关注 GNN 的模型。[32]将图形神经网络分为五类:图形卷积网络、图形注意力网络、图形自动编码器、图形生成网络和图形时空网络。我们的论文与[32]的分类不同。我们在第 2.2.2 节中介绍了图卷积网络和图注意力网络，因为它们有助于传播步骤。我们在第 2.2.1 节中介绍了图形时空网络，因为模型通常用于动态图形。我们在第 2.2.3 节中介绍了图形自动编码器，因为它们是以无监督的方式训练的。最后，我们介绍了图生成网络在图生成中的应用(见第 3.3.1 节)。

在这篇文章中，我们提供了一个不同的图形神经网络模型的彻底审查，以及应用的系统分类。总之，本文对图形神经网络进行了广泛的综述，其贡献如下。

- 我们对现有的图形神经网络模型进行了详细的回顾。我们介绍原始模型、它的变体和几个通用框架。我们研究了这一领域的各种模型，并提供了一个统一的表示来表示不同模型中的不同传播步骤。通过识别相应的聚合器和更新器，可以很容易地使用我们的表示来区分不同的模型。
- 我们系统地对应用程序进行分类，并将应用程序分为结构化场景、非结构化场景和其他场景。我们为每个场景提供了几个主要的应用程序和相应的方法。
- 我们提出了四个有待进一步研究的问题。
图神经网络存在过度平滑和缩放问题。仍然没有有效的方法来处理动态图形以及建模非结构性感觉数据。我们对每个问题进行了全面的分析，并提出了未来的研究方向。

本次调查的其余部分组织如下。秒。2、我们介绍了图神经网络族中的各种模型。我们先介绍一下原框架及其局限性。然后我们展示了它的变体，试图释放这些限制。最后，我们介绍了最近提出的几种通用框架。秒。3、我们将介绍图形神经网络在结构场景、非结构场景和其他场景中的几个主要应用。秒。4、提出了图形神经网络的四个有待解决的问题以及未来的几个研究方向。最后，我们在美国证券交易委员会完成了调查。5。

2 模型

图形神经网络是非欧结构的有用工具，文献中提出了各种方法来提高模型的性能。

在第 2.1 节中，我们描述了在[33]中提出的原始图形神经网络。我们还列出了原 GNN 在表现能力和训练效率方面的局限性。在第 2.2 节中，我们介绍了图形神经网络的几种变体，旨在消除这些限制。这些变量对不同类型的图进行操作，利用不同的传播函数和高级训练方法。在第 2.3 节中，我们提出了三个通用框架，可以概括和扩展几个工作领域。详细地说，消息传递神经网络(MPNN) [27]统一了各种图神经网络和图卷积网络方法；非局部神经网络[28]统一了几种“自我注意”类型的方法。而图网络(GN) [30]几乎可以概括本文提到的每一种图神经网络变体。

在进一步进入不同的部分之前，我们给出了将在整篇论文中使用的符号。符号的详细描述可以在表 1 中找到。

2.1 图形神经网络

图神经网络(GNN)的概念最早是在[33]中提出的，它扩展了现有的用于处理图域中表示的数据的神经网络。在图中，每个节点自然由其特征和相关节点定义。GNN 的目标是学习嵌入 $h_v \in \mathbb{R}_s$ 的状态，该状态包含每个节点的邻域信息。状态嵌入 h_v 是节点 v 的 s 维向量，可用于产生输出，如节点标签。设 f 为参数函数，称为局部转移函数，在所有节点间共享，根据输入邻域更新节点状态。让 g 是本地输出函数

表 1
本文中使用的符号。

记号	描述
空间	m 维欧氏空间
$\mathbf{1}, \mathbf{1}, \mathbf{1}$	标量、矢量、矩阵
(At)人名,(越)乙	矩阵转置
在...里	N 维单位矩阵
$g_{\theta} \cdot x$	g_{θ} 和 x 的卷积
N_v / 超速 (hypervelocity 的简写)	$/v$ 节点 v 在时间步长 t 的向量 a 节点 v 的隐藏状态

高温硫化	节点 v 在时间步长 t 的隐藏状态
水资源经济价值	从节点 v 到 w 的边的特征
阿联酋航空	标签为 k 的边的特征
/	节点 v 的输出
$W_i, U_i, W_o, U_o, \dots$	计算 l, o, \dots
b_i, b_o, \dots	计算 l, o, \dots
双曲正切	双曲正切函数
李克瑞路	泄漏率函数

描述如何产生输出。然后, h_v 和 o_v 定义如下:

$$h_v = f(x_v, xco[v], hne[v], xne[v]) \quad (1)$$

$$o_v = g(h_v, x_v) \quad (2)$$

其中 $x_v, xco[v], hne[v], xne[v]$ 分别是 v 的特征、其边的特征、 v 的邻域中的节点的特征和状态。

设 H, O, X 和 XN 分别是所有状态、所有输出、所有特征和所有节点特征叠加而成的向量。那么我们有一个紧凑的形式:

$$H = F(H, X) \quad (3)$$

$$O = G(H, XN) \quad (4)$$

其中, 全局转移函数 F 和全局输出函数 G 分别是图中所有节点的 F 和 G 的堆叠版本。 H 的值是 E_q 的不动点。³ 并且是在假设 F 是收缩映射的情况下唯一定义的。

根据巴拿赫不动点定理[34]的建议, GNN 使用以下经典迭代方案来计算状态:

$$H^{t+1} = F(H^t, X) \quad (5)$$

其中 H_t 表示 h 的第 t 次迭代。动力系统方程。⁵ 指数快速收敛到方程的解。³ 对于任何初始值 $H(0)$ 。请注意, f 和 g 中描述的计算可以解释为前馈神经网络。

当我们有了 GNN 的框架, 下一个问题就是如何学习 f 和 g 的参数。有了监控的目标信息(特定节点的电视), 损失可以写成如下形式:

$$\text{损耗} = \sum_{i=1}^p X(t_i, o_i) \quad (6)$$

其中 p 是监督节点的数量。学习算法基于梯度下降策略, 由以下步骤组成。

- 状态 h_{tv} 由 E_q 迭代更新。¹ 直到时间 t 。他们接近方程的不动点解。³: $H(T) \approx H$ 。
- 权重 W 的梯度是根据损失计算的。
- 权重 W 根据最后一步中计算的梯度进行更新。

局限性虽然实验结果表明 GNN 是一个强大的结构数据建模架构, 但仍有一些原始 GNN 的局限性。首先, 针对固定点

迭代更新节点的隐藏状态是低效的。如果放松不动点的假设, 我们可以设计一个多层 GNN 来获得节点及其邻域的稳态表示。其次, GNN 在迭代中使用相同的参数, 而大多数流行的神经网络在不同的层中使用不同的参数, 这是一种分层的特征提取方法。此外, 节点隐藏状态的更新是一个连续的过程, 它可以像 GRU 和 LSTM 一样受益于 RNN 核。第三, 在边缘上也有一些信息特征不能在原始 GNN 中有效地建模。例如, 知识图中的边具有关系的类型, 通过不同边的消息传播应该根据它们的类型而不同。此外, 如何学习边缘的隐藏状态也是一个重要的问题。最后, 如果我们关注节点的表示而不是图, 则不适合使用不动点, 因为不动点中的表示的分布将在值上更加平滑, 并且用于区分每个节点的信息较少。

2.2 图神经网络的变体

在这一小节中, 我们介绍了图神经网络的几种变体。第 2.2.1 节重点关注在不同图形类型上运行的变体。这些变体扩展了原始模型的表示能力。第 2.2.2 节列出了对传播步骤的几种修改(卷积、门机制、注意机制和跳过连接), 这些模型可以学习更高质量的表示。第 2.2.3 节描述了使用先进的训练方法来提高训练效率的变体。图表神经网络的不同变体的概述可以在图. 2。

2.2.1 图形类型

在最初的 GNN [33]中, 输入图由带有标签信息的节点和无向边组成, 这是最简单的图格式。然而, 世界上有许多不同的图形。在这一小节中, 我们将介绍一些为不同类型的图形建模而设计的方法。

有向图图的第一个变体是有向图。可视为两条有向边的无向边表明两个节点之间存在关系。然而, 有向边比无向边能带来更多的信息。例如, 在一个边从头实体开始到尾实体结束的知识图中, 头实体是尾实体的父类, 这表明我们应该区别对待父类和子类的信息传播过程。DGP [35]使用两种权重矩阵, W_p 和 W_c , 来合并更精确的结构信息。传播规则如下所示:

$$h_t = \sigma(D_p \mathbf{1} \text{ app} \sigma(D_c \mathbf{1} \text{ acht } 1W_c)W_p) \quad (7), \text{ 其中}$$

$D_p^{-1}A_p, D_c^{-1}A_c$ 分别是父母和子女的归一化邻接矩阵。

异构图图的第二个变体是异构图, 其中有几种节点。处理异构图最简单的方法是将每个节点的类型转换成一个与原始特征连接的热点特征向量。此外, 图形概念[36]将元路径的概念引入到异构图形的传播中。使用元路径, 我们可以根据邻居的节点类型和距离对其进行分组。对于每个邻居组, GraphInception 将其视为同构图中的一个子图来进行传播, 并将来自不同同构图的传播结果连接起来, 以进行集合节点表示。最近, [37]提出了利用节点级和语义级关注的异构图关注网络(HAN)。该模型能够同时考虑节点重要性和元路径。

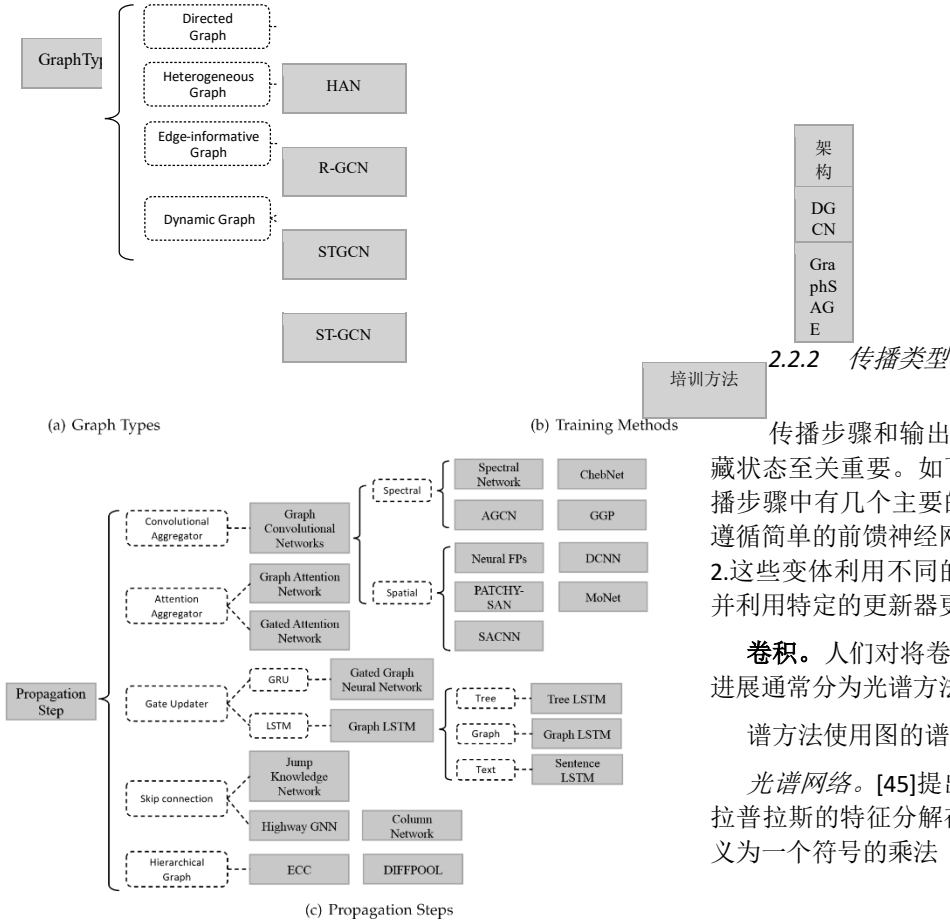
带边信息的图在图的另一个变体中, 每条边都有附加信息, 如边的权重或类型。我们列出了两种处理这类图的方法: 首先, 我们可以将图转换为二部图, 其中原始边也成为节点, 并且一条原始边被分成两条新边, 这意味着在边节点和开始/结束

节点之间有两条新边。G2S [38]的编码器对邻居使用以下聚合函数:

$$\mathbf{h}_v^t = \rho\left(\frac{1}{|\mathcal{N}_v|} \sum_{u \in \mathcal{N}_v} \mathbf{W}_r(\mathbf{r}_v^t \odot \mathbf{h}_u^{t-1}) + \mathbf{b}_r\right) \quad (8)$$

其中 \mathbf{W}_r 和 \mathbf{b}_r 是不同类型边(关系)的传播参数。其次,我们可以为不同类型的边的传播采用不同的权重矩阵。当关系的数量非常大时, \mathbf{r} -GCN [39]引入两种正则化来减少用于建模关系数量的参数的数量:基分解和块对角分解。有了基础

序列模型,如序列到序列模型或 CNNs。不同的是,结构 RNN [42]和圣 GCN [43]同时收集空间和时间信息。他们用动态连接扩展了静态图结构,因此可以在扩展图上应用传统的神经网络。

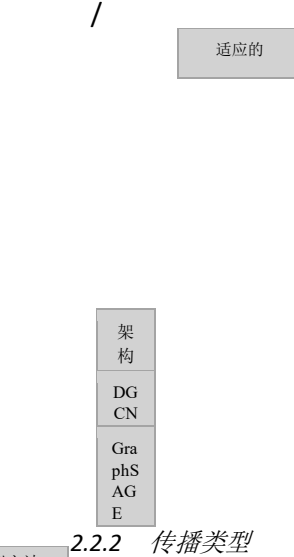


图。2.图形神经网络变体综述。分解,每个 \mathbf{W}_r 定义如下:

$$\mathbf{W}_r = \mathbf{X} \mathbf{a}_r \mathbf{b}_r^T \quad (9)$$

这里,每个 \mathbf{W}_r 是基变换 $\mathbf{V}_b \in \mathbb{R}^{d_{in} \times d_{out}}$ 与系数 \mathbf{a}_r 的线性组合。在块对角分解中, \mathbf{r} -GCN通过一组低维矩阵的直接和来定义每个 \mathbf{W}_r ,这比第一个矩阵需要更多的参数。

动态图形图形的另一个变体是动态图形,它具有静态图形结构和动态输入信号。为了捕捉这两种信息,DCRNN [40]和 STGCN [41]首先通过 GNNs 收集空间信息,然后将输出馈送到



传播步骤和输出步骤在模型中对于获得节点(或边)的隐藏状态至关重要。如下所列,在原始图形神经网络模型的传播步骤中有几个主要的修改,而研究人员通常在输出步骤中遵循简单的前馈神经网络设置。不同 GNN 变异体的比较见表 2.这些变体利用不同的聚合器从每个节点的邻居收集信息,并利用特定的更新器更新节点的隐藏状态。

卷积。人们对将卷积推广到图域越来越感兴趣。这方面的进展通常分为光谱方法和非光谱(空间)方法。

谱方法使用图的谱表示。

光谱网络。[45]提出了光谱网络。卷积运算是通过计算图拉普拉斯的特征分解在傅立叶域中定义的。这个运算可以定义为一个符号的乘法

表 2
图形神经网络的不同变体。

Name	Variant	Aggregator	Updater
Spectral Methods	ChebNet	$\mathbf{N}_k = \mathbf{T}_k(\tilde{\mathbf{L}})\mathbf{X}$	$\mathbf{H} = \sum_{k=0}^K \mathbf{N}_k \Theta_k$
	1 st -order model	$\mathbf{N}_0 = \mathbf{X}$ $\mathbf{N}_1 = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{X}$	$\mathbf{H} = \mathbf{N}_0\Theta_0 + \mathbf{N}_1\Theta_1$
	Single parameter	$\mathbf{N} = (\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{X}$	$\mathbf{H} = \mathbf{N}\Theta$
	GCN	$\mathbf{N} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}$	$\mathbf{H} = \mathbf{N}\Theta$
Non-spectral Methods	Neural FPs	$\mathbf{h}_{\mathcal{N}_v}^t = \mathbf{h}_v^{t-1} + \sum_{k=1}^{\mathcal{N}_v} \mathbf{h}_k^{t-1}$	$\mathbf{h}_v^t = \sigma(\mathbf{h}_{\mathcal{N}_v}^t \mathbf{W}_L^{\mathcal{N}_v})$
	DCNN	Node classification: $\mathbf{N} = \mathbf{P} \cdot \mathbf{X}$ Graph classification: $\mathbf{N} = \mathbf{1}^T \mathbf{N} \mathbf{P} \cdot \mathbf{X} / N$	$\mathbf{H} = f(\mathbf{W}^c \odot \mathbf{N})$
	GraphSAGE	$\mathbf{h}_{\mathcal{N}_v}^t = \text{AGGREGATE}_t(\{\mathbf{h}_u^{t-1}, \forall u \in \mathcal{N}_v\})$	$\mathbf{h}_v^t = \sigma(\mathbf{W}^t \cdot [\mathbf{h}_v^{t-1} \parallel \mathbf{h}_{\mathcal{N}_v}^t])$
Graph Attention Networks	GAT	$\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_v + \sum_{k \in \mathcal{N}_v} \alpha_{vk} \mathbf{W}\mathbf{h}_k]))$ $\alpha_{vk} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_v + \mathbf{W}\mathbf{h}_k]))}{\sum_{j \in \mathcal{N}_v} \exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_v + \mathbf{W}\mathbf{h}_j]))}$ $\mathbf{h}_{\mathcal{N}_v}^t = \sigma(\sum_{k \in \mathcal{N}_v} \alpha_{vk} \mathbf{W}\mathbf{h}_k)$ <p>Multi-head concatenation:</p> $\mathbf{h}_{\mathcal{N}_v}^t = \left\ _{m=1}^M \sigma\left(\sum_{k \in \mathcal{N}_v} \alpha_{vk}^m \mathbf{W}^m \mathbf{h}_k\right)\right\ $ <p>Multi-head average: $\mathbf{h}_{\mathcal{N}_v}^t = \sigma\left(\frac{1}{M} \sum_{m=1}^M \sum_{k \in \mathcal{N}_v} \alpha_{vk}^m \mathbf{W}^m \mathbf{h}_k\right)$</p>	$\mathbf{h}_v^t = \mathbf{h}_{\mathcal{N}_v}^t$
Gated Graph Neural Networks	GGNN	$\mathbf{h}_{\mathcal{N}_v}^t = \sum_{k \in \mathcal{N}_v} \mathbf{h}_k^{t-1} + \mathbf{b}$	$\mathbf{z}_v^t = \sigma(\mathbf{W}^z \mathbf{h}_{\mathcal{N}_v}^t + \mathbf{U}^z \mathbf{h}_v^{t-1})$ $\mathbf{r}_v^t = \sigma(\mathbf{W}^r \mathbf{h}_{\mathcal{N}_v}^t + \mathbf{U}^r \mathbf{h}_v^{t-1})$ $\widetilde{\mathbf{h}}_v^t = \tanh(\mathbf{W}\mathbf{h}_{\mathcal{N}_v}^t + \mathbf{U}(\mathbf{r}_v^t \odot \mathbf{h}_v^{t-1}))\mathbf{h}_v^t$ $\mathbf{h}_v^t = (1 - \mathbf{z}_v^t) \odot \mathbf{h}_v^{t-1} + \mathbf{z}_v^t \odot \widetilde{\mathbf{h}}_v^t$
Graph LSTM	Tree LSTM (Child sum)	$\mathbf{h}_{\mathcal{N}_v}^t = \sum_{k \in \mathcal{N}_v} \mathbf{h}_k^{t-1}$	$\mathbf{i}_v^t = \sigma(\mathbf{W}^i \mathbf{x}_v^t + \mathbf{U}^i \mathbf{h}_{\mathcal{N}_v}^t + \mathbf{b}^i)$ $\mathbf{f}_{vk}^t = \sigma(\mathbf{W}^f \mathbf{x}_v^t + \mathbf{U}^f \mathbf{h}_k^{t-1} + \mathbf{b}^f)$ $\mathbf{o}_v^t = \sigma(\mathbf{W}^o \mathbf{x}_v^t + \mathbf{U}^o \mathbf{h}_{\mathcal{N}_v}^t + \mathbf{b}^o)$ $\mathbf{c}_k^t = \tanh(\mathbf{W}^c \mathbf{x}_k^t + \mathbf{U}^c \mathbf{h}_k^{t-1} + \mathbf{b}^c)$ $\mathbf{u}_v^t = \mathbf{i}_v^t \odot \mathbf{u}_v^t + \sum_{k \in \mathcal{N}_v} \mathbf{f}_{vk}^t \odot \mathbf{c}_k^{t-1}$ $\mathbf{h}_v^t = \mathbf{o}_v^t \odot \tanh(\mathbf{c}_v^t) \mathbf{u}_v^t = \tanh(\mathbf{W}^u \mathbf{x}_v^t + \mathbf{U}^u \mathbf{h}_{\mathcal{N}_v}^t + \mathbf{b}^u) \mathbf{c}_v^t$

Tree LSTM (N-ary)	$\mathbf{h}_{\mathcal{N}_v}^{ti} = \sum_{l=1}^K \mathbf{U}_l^i \mathbf{h}_{vl}^{t-1} \mathbf{h}$ $\mathbf{h}_{\mathcal{N}_v}^{to} = \sum_{l=1}^K \mathbf{U}_l^o \mathbf{h}_{vl}^{t-1}$ $\mathbf{h}_{\mathcal{N}_v}^{tu} = \sum_{l=1}^K \mathbf{U}_l^u \mathbf{h}_{vl}^{t-1}$ $\mathbf{h}_{\mathcal{N}_v}^{tf} = \sum_{l=1}^K \mathbf{U}_l^f \mathbf{h}_{vl}^{t-1} \mathbf{h}$	$\mathbf{i}_{tv} = \sigma(\mathbf{W}_i \mathbf{x}_{tv} + \mathbf{h}_{tiN_v} + \mathbf{b}_i) \mathbf{f}_{vkt} =$ $\mathbf{t}_v^t = \sigma(\mathbf{W}^o \mathbf{x}_v^t + \mathbf{h}_{\mathcal{N}_v}^{to} + \mathbf{b}^o)$ $\mathbf{t}_v^t = \mathbf{i}_v^t \odot \mathbf{u}_v^t + \sum_{l=1}^K \mathbf{f}_{vl}^t \odot \mathbf{c}_{vl}^{t-1}$ $\mathbf{h}_v = \mathbf{o}_v \odot \tanh(\mathbf{c}_v) \sigma(\mathbf{W}_f \mathbf{x}_{tv}$ $+ \mathbf{h}_{tfN_v} + \mathbf{b}_f) \mathbf{o}_v = \tanh(\mathbf{W}^u \mathbf{x}_v +$ $\mathbf{h}_{N_v} + \mathbf{b}^u)$ \mathbf{c}
Graph LSTM in [44]	$\mathbf{h}_{\mathcal{N}_v}^{ti} = \sum_{k \in \mathcal{N}_v} \mathbf{U}_{m(v,k)}^i \mathbf{h}_k^{t-1}$ $\mathbf{h}_{\mathcal{N}_v}^{to} = \sum_{k \in \mathcal{N}_v} \mathbf{U}_{m(v,k)}^o \mathbf{h}_k^{t-1}$ $\mathbf{h}_{\mathcal{N}_v}^{tu} = \sum_{k \in \mathcal{N}_v} \mathbf{U}_{m(v,k)}^u \mathbf{h}_k^{t-1}$	$\mathbf{i}_{tv} = \sigma(\mathbf{W}_i \mathbf{x}_{tv} + \mathbf{h}_{tiN_v} + \mathbf{b}_i) \mathbf{f}_{vkt} = \sigma(\mathbf{W}_f \mathbf{x}_{tv}$ $\mathbf{t}_v^t = \sigma(\mathbf{W}^o \mathbf{x}_v^t + \mathbf{h}_{\mathcal{N}_v}^{to} + \mathbf{b}^o) +$ $\mathbf{U}_{fm(v,k)} \mathbf{h}_{tk-1} + \mathbf{b}_f) \mathbf{o}_v = \tanh(\mathbf{W}^u \mathbf{x}_v +$ $\mathbf{h}_{N_v} + \mathbf{b}^u)$ \mathbf{c} $\mathbf{t}_v^t = \mathbf{o}_v^t \odot \tanh(\mathbf{c}_v^t)$ $\mathbf{t}_v^t = \mathbf{i}_v^t \odot \mathbf{u}_v^t + \sum_{k \in \mathcal{N}_v} \mathbf{f}_{vk}^t \odot \mathbf{c}_k^{t-1} \mathbf{h}$

nal $\mathbf{x} \in \mathbb{R}^N$ (每个节点的标量), 带有由 $\theta \in \mathbb{R}^N$ 参数化的滤波器 $\mathbf{g}_\theta = \text{diag}(\theta)$:

$$\mathbf{g}_\theta \star \mathbf{x} = \mathbf{U} \mathbf{g}_\theta(\lambda) \mathbf{U}^T \mathbf{x} \quad (10)$$

其中 \mathbf{U} 是归一化图拉普拉斯 $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ $\mathbf{U} \mathbf{A} \mathbf{U}^T$ 的特征向量矩阵 (\mathbf{D} 是度矩阵, \mathbf{A} 是图的邻接矩阵), 其特征值 λ 的对角矩阵。

该操作导致潜在的高强度计算和非空间局部化滤波器。[46] 试图通过引入具有平滑系数的参数化来使光谱滤波器空间局部化。

ChebNet. [47] 表明, $\mathbf{g}_\theta(\lambda)$ 可以用切比雪夫多项式 $\mathbf{T}_k(\mathbf{x})$ 的截断展开来近似, 直到 K th 阶。因此操作是:

$$\mathbf{g}_\theta \star \mathbf{x} \approx \sum_{k=0}^K \mathbf{x} \theta_k \mathbf{T}_k(\mathbf{L}) \mathbf{x} \quad (11)$$

用 $\mathbf{L} \backslash \backslash = \frac{2}{\lambda_{max}} \mathbf{L} - \mathbf{I}_N$. λ_{max} 表示 \mathbf{L} 的最大特征值。 $\theta \in \mathbb{R}^K$ 现在是切比雪夫系数的向量。切比雪夫多项式定义为 $\mathbf{T}_k(\mathbf{x}) = 2\mathbf{x} \mathbf{T}_{k-1}(\mathbf{x}) - \mathbf{T}_{k-2}(\mathbf{x})$, $\mathbf{T}_0(\mathbf{x}) = 1$, $\mathbf{T}_1(\mathbf{x}) = \mathbf{x}$ 。可以观察到, 该操作是 K -局部化的, 因为它是拉普拉斯算子中的 K th 阶多项式。[48] 提出了 **ChebNet**。它使用这个 K -局部卷积来定义一个卷积神经网络, 该网络可以消除计算拉普拉斯特征向量的需要。

GCN. [2] 将逐层卷积运算限制在 $K = 1$, 以减轻节点度分布非常宽的图在局部邻域结构上的过拟合问题。它进一步逼近 $\lambda_{max} \approx 2$, 并且该等式简化为:

$$\mathbf{g}_\theta \star \mathbf{x} \approx \theta'_0 \mathbf{x} + \theta'_1 (\mathbf{L} - \mathbf{I}_N) \mathbf{x} = \theta'_0 \mathbf{x} - \theta'_1 \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$$

带有两个自由参数 θ'_0 和 θ'_1 。在用 $\theta = \theta'_0 = -\theta'_1$ 约束参数数量后, 我们可以获得以下表达式:

$$\mathbf{g}_\theta \star \mathbf{x} \approx \theta \left(\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \right) \mathbf{x} \quad (13)$$

请注意, 叠加该运算符可能会导致数值不稳定和爆炸性消失梯度, [2] 引入了重正化技巧: $\mathbf{L} \leftarrow \mathbf{L} / \text{D}(\mathbf{L})$, 其中 $\mathbf{A} \leftarrow \mathbf{A} / \mathbf{D}(\mathbf{A})$, $\mathbf{D} \leftarrow \mathbf{D} / \mathbf{D}(\mathbf{D})$ 。最后, [2] 将信号 $\mathbf{x} \in \mathbb{R}^N \times \mathbb{C}$ 的定义推广到具有 C 输入通道和 F 滤波器的特征映射, 如下所示:

$$\mathbf{z} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{x} \mathbf{D}^{-\frac{1}{2}} \quad (14)$$

其中 $\theta \in \mathbb{R}^{C \times F}$ 为滤波器参数矩阵, $\mathbf{z} \in \mathbb{R}^N \times \mathbb{C}$ 为卷积信号矩阵。

所有这些模型都使用原始的图结构来表示节点之间的关系。然而, 不同节点之间可能存在隐含关系, 自适应图卷积网络 (AGCN) 被提出来学习潜在的关系 [49]。AGCN 学习一个“剩余”图拉普拉斯, 并将其添加到原始拉普拉斯矩阵中。结果, 它在几个图形结构的数据集上被证明是有效的。

此外, [50] 提出了一种基于高斯过程的贝叶斯方法 (GGP) 来解决半监督学习问题。它显示了模型和光谱过滤方法之间的相似之处, 这可以从另一个角度给我们一些启示。

然而, 在上面提到的所有谱方法中, 学习的滤波器依赖于拉普拉斯特征基, 拉普拉斯特征基依赖于图结构, 即, 在特定结构上训练的模型不能直接应用于具有不同结构的图。

非谱方法直接在图上定义卷积, 对空间上的近邻进行操作。非谱方法的主要挑战是用不同大小的邻域定义卷积运算, 并保持中枢神经系统的局部不变性。

神经 FPs。[51]对不同程度的节点使用不同的权重矩阵,

$|Nv$

$$X = HTV \ 1 + X \ hti \ 1$$

$$i=1 \quad (15)$$

$$\mathbf{h}_v^t = \sigma(\mathbf{x}\mathbf{W}_t^{|\mathcal{N}_v|})$$

其中, \mathbf{Wt} 是层 t 上度数为 $|\mathbf{Nv}|$ 的节点的权重矩阵。该方法的主要缺点是不能应用于节点度较大的大规模图。

DCNN. [23]提出了扩散卷积神经网络(DCNNs)。转移矩阵用于定义离散余弦变换神经网络中节点的邻域。对于节点分类,它有

$$\mathbf{H} = f(\mathbf{W}^c \odot \mathbf{P}^* \mathbf{X}) \quad (16)$$

其中 X 是输入特征的 $N \times F$ 张量 (N 是节点数, F 是特征数)。 P 是包含幂级数 $\{P, P_2, \dots, \text{矩阵 } p \text{ 的 } PK\}$ 。 P 是图邻接矩阵 a 的度归一化转移矩阵。每个实体被转换成一个扩散卷积表示, 它是一个由图在 F 特征上扩散的 K 跳定义的 $K \times F$ 矩阵。然后将由一个 $K \times F$ 的权重矩阵和一个非线性激活函数 F 来定义。最后 $H(N \times K \times F)$ 表示图中每个节点的扩散表示。

对于图的分类，DCNN 只是取节点表示的平均值，

$$\mathbf{H} = f\left(\mathbf{W}^c \odot \mathbf{1}_N^T \mathbf{P}^* \mathbf{X} / N\right) \quad (17)$$

这里的 $\mathbf{1N}$ 是 $\mathbf{1}$ 的 $N \times 1$ 向量。DCNN 还可以应用于边缘分类任务，这需要将边缘转换为节点并增加邻接矩阵。

DGCN. [52]提出了对偶图卷积网络(DGCN)来共同考虑图的局部一致性和全局一致性。它使用两个卷积网络来捕获局部/全局一致性,并采用无监督损失来集成它们。第一个卷积网络和 Eq 一样。14.第二个网络用正的逐点互信息(PPMI)矩阵代替邻接矩阵:

$$\mathbf{P}' = \rho(\mathbf{D}_P^{-\frac{1}{2}} \mathbf{X}_P \mathbf{D}_P^{-\frac{1}{2}} \mathbf{X}_P \mathbf{D}_P^{-\frac{1}{2}}) \quad (1)$$

$$hh\theta) \quad (18)$$

其中 XP 是 PPMI 矩阵, DP 是 XP 的对角度矩阵。

东拼西凑-SAN。 PATCY-SAN 模型[53]为每个节点提取并规范化了一个恰好由 k 个节点组成的邻域。然后归一化邻域作为卷积运算的感受野。

LGCN。LGCN [54]还利用氯化萘作为聚合剂。它对节点的邻域矩阵进行最大池化,得到前 k 个特征元素,然后应用一维 CNN 计算隐藏表示。

莫奈。[22]提出了一个关于非欧几里德域的空间域模型 (MoNet)，它可以推广以前的几种技术。流形上的测地线 CNN (GCNN) [24]和各向异性 CNN (ACNN) [25]或者图上的 GCN [2]和 DCNN [23]可以被公式化为 MoNet 的特定实例。

GraphSAGE。[1]提出了 *GraphSAGE*，一个通用的归纳框架。该框架通过从节点的局部邻域中采样和聚集特征来生成嵌入。

$$\begin{aligned} \mathbf{h}^t \mathbf{N} \mathbf{v} &= \text{AGGREGATE}^t(\{\mathbf{h}_u^{t-1}, \forall u \in \mathcal{N}_v\}) \\ \mathbf{h}_v^t &= \sigma(\mathbf{W}^t \cdot [\mathbf{h}_v^{t-1} \parallel \mathbf{h}_{\mathcal{N}_v}^t]) \end{aligned} \quad (19)$$

$$t_v = \sigma(\mathbf{W}^t \cdot [\mathbf{h}_v^{t-1} \parallel \mathbf{h}_{\mathcal{N}_v}^t]) \quad \mathbf{h}_v^t$$

然而, [1]没有利用等式 19 中的全部邻居集合, 而是通过均匀采样利用固定大小的邻居集合。而[1]提出了三个聚合函数。

- 均值聚合器。它可以看作是直推 GCN 框架[2]的卷积运算的近似，因此 GCN 变型的归纳版本可以由下式导出

$$\mathbf{h}_v^t = \sigma(\mathbf{W} \cdot \mathbf{MEAN}$$

均值聚合器不同于其他聚合器，因为它不执行等式 19 中连接 h 和 $htNv$ 的连接操作。它可以被视为一种“跳过连接”的形式[55]，并可以实现更好的性能。

- **LSTM 聚合器。**[1]还使用了一个基于 LSTM 的聚合器，它具有更大的表达能力。然而，LSTMs 以顺序方式处理输入，因此它们不是置换不变的。[1]通过置换节点的邻居，使 LSTMs 适应在无序集上操作。
- **汇集聚合器。**在池聚合器中，每个邻居的隐藏状态通过一个完全连接的层提供，然后对节点的邻居集应用最大池操作。
$$\mathbf{h}tNv = \text{最大值}(\{\sigma W_{\text{pool}} \mathbf{u}_{u-1} + \mathbf{b}\}, \forall u \in \mathcal{N}_v\}) \quad (21)$$

请注意，这里可以使用任何对称函数来代替最大池操作。

最近, 结构感知卷积和结构——已经提出了感知卷积神经网络[56]。单变量函数用作过滤器, 它们可以处理欧几里德和非欧几里德结构化数据。

大门。有几项工作试图在传播步骤中使用 GRU [57] 或 LSTM [58] 等门机制, 以减少前 GNN 模型中的限制, 并改善信息在图结构中的长期传播。

[59] 提出了门控图神经网络, 该网络在传播步骤中使用门控递归单元(GRU), 在固定数量的步骤 T 中展开递归, 并使用时间反向传播来计算梯度。

具体来说,传播模型的基本循环是

$$\begin{aligned} \mathbf{a}_v^t &= \mathbf{A}_v^T [\mathbf{h}_1^{t-1} \dots \mathbf{h}_N^{t-1}]^T + \mathbf{b} \\ \mathbf{r}_v^t &= \sigma(\mathbf{W}^r \mathbf{a}_v^t + \mathbf{U}^r \mathbf{h}_v^{t-1}) \\ \mathbf{h}_v^t &= \tanh(\mathbf{W}^h \mathbf{a}_v^t + \mathbf{U}^h \mathbf{h}_v^{t-1}) \end{aligned} \quad (22)$$

$$\mathbf{r}_v^t = \sigma(\mathbf{W}^r \mathbf{a}_v^t + \mathbf{U}^r \mathbf{h}_v^{t-1}) \quad (22)$$

$$\mathbf{h}f_v^t = \tanh(\mathbf{W}_a \mathbf{U} \mathbf{r} \mathbf{h})$$

节点 \mathbf{v} 首先聚集来自其邻居的消息，其中 $\mathbf{A}_{\mathbf{v}}$ 是图邻接矩阵 \mathbf{A} 的子矩阵，并表示节点 \mathbf{v} 与其邻居的连接。类似 GRU 的更新函数结合了来自其他节点的信息和来自前一时间步的信息来更新每个节点的隐藏状态。 \mathbf{a} 收集节点 \mathbf{v} 的邻域信息， \mathbf{z} 和 \mathbf{r} 是更新和复位门。

在基于树或图的传播过程中，LSTMs 也以类似于 GRU 的方式使用。

[60] 提出了对基本 LSTM 体系结构的两个扩展:子和树-LSTM 和 N 元树-树模型。像在标准 LSTM 单元中一样,每个树-LSTM 单元(用 v 索引)包含输入和输出门 iv 和 ov 、存储单元 cv 和隐藏状态 hv 。树 LSTM 单元包含一个遗忘门 fvk , 而不是一个单一的遗忘门, 允许该单元选择性地合并来自每个孩子的信息。子和树-LSTM 转移方程如下:
$$h_v^{t-1} = \sum h_k^{t-1}$$

信息。子和树-LSTM 转移方程如下: $\mathbf{h}_v^{t-1} = \sum \mathbf{h}_k^{t-1}$

$$\begin{aligned}
& k \in \mathcal{N}_v \\
& \mathbf{i}_v^t = \sigma(\mathbf{W}^i \mathbf{x}_v^t + \mathbf{U}^i \widetilde{\mathbf{h}_v^{t-1}} + \mathbf{b}^i) \\
& \mathbf{f}_{vk}^t = \sigma(\mathbf{W}^f \mathbf{x}_v^t + \mathbf{U}^f \mathbf{h}_k^{t-1} + \mathbf{b}^f) \\
& \mathbf{o}_v^t = \sigma(\mathbf{W}^o \mathbf{x}_v^t + \mathbf{U}^o \widetilde{\mathbf{h}_v^{t-1}} + \mathbf{b}^o) \\
& \mathbf{u}_v^t = \tanh(\mathbf{W}^u \mathbf{x}_v^t + \mathbf{U}^u \widetilde{\mathbf{h}_v^{t-1}} + \mathbf{b}^u) \\
& \mathbf{t}_v^t = \mathbf{i}_v^t \odot \mathbf{u}_v^t + \sum_{k \in \mathcal{N}_v} \mathbf{f}_{vk}^t \odot \mathbf{c}_k^{t-1} \text{ 分} \\
& \mathbf{h}_v^t = \mathbf{o}_v^t \odot \tanh(\mathbf{c}_v^t)
\end{aligned} \tag{23}$$

xtv 是标准 LSTM 设置中时间 t 的输入向量。

如果树的分支因子最多为 K ，并且一个节点的所有子节点都是有序的，即。它们可以从 1 到 K 进行索引，然后可以使用 N 元树 LSTM。对于节点 v ， \mathbf{h}_{vk}^t 和 \mathbf{c}_{vk}^t 分别表示其第 k 个子节点在时间 t 的隐藏状态和存储单元。过渡方程如下：

$$\begin{aligned}
& \mathbf{i}_v^t = \sigma(\mathbf{W}^i \mathbf{x}_v^t + \sum_{l=1}^K \mathbf{U}_l^i \mathbf{h}_{vl}^{t-1} + \mathbf{b}^i) \\
& \mathbf{f}_{vk}^t = \sigma(\mathbf{W}^f \mathbf{x}_v^t + \sum_{l=1}^K \mathbf{U}_{kl}^f \mathbf{h}_{vl}^{t-1} + \mathbf{b}^f) \\
& \mathbf{o}_v^t = \sigma(\mathbf{W}^o \mathbf{x}_v^t + \sum_{l=1}^K \mathbf{U}_l^o \mathbf{h}_{vl}^{t-1} + \mathbf{b}^o) \\
& \mathbf{t}_v^t = \tanh(\mathbf{W}^u \mathbf{x}_v^t + \sum_{l=1}^{Ku} \mathbf{U}_l^u \mathbf{h}_{vl}^{t-1} + \mathbf{b}^u) \\
& \mathbf{t}_v^t = \mathbf{i}_v^t \odot \mathbf{u}_v^t + \sum_{l=1}^K \mathbf{f}_{vl}^t \odot \mathbf{c}_{vl}^{t-1} \text{ 分} \\
& \mathbf{h}_v^t = \mathbf{o}_v^t \odot \tanh(\mathbf{c}_v^t)
\end{aligned} \tag{24}$$

为每个子 k 引入单独的参数矩阵，使得模型能够学习到比子和树 LSTM 更细粒度的表示，这些表示以单元子的状态为条件。

这两种类型的树形结构可以很容易地适用于图表。[61]中的图结构 LSTM 是应用于图的 N 元树 LSTM 的一个例子。然而，它是一个简化的版本，因为图中的每个节点最多有 2 条输入边(来自其父节点和兄弟节点的前身)。[44]提出了基于关系抽取任务的图形 LSTM 的另一种变体。图和树的主要区别在于图的边都有标号。和[44]利用不同的权重矩阵来表示不同的标签。

$$\begin{aligned}
& \mathbf{i}_v^t = \sigma(\mathbf{W}^i \mathbf{x}_v^t + \sum_{k \in \mathcal{N}_v} \mathbf{U}_{m(v,k)}^i \mathbf{h}_k^{t-1} + \mathbf{b}^i) \\
& \mathbf{f}_{vk}^t = \sigma(\mathbf{W}^f \mathbf{x}_v^t + \mathbf{U}_{m(v,k)}^f \mathbf{h}_k^{t-1} + \mathbf{b}^f) \\
& \mathbf{o}_v^t = \sigma(\mathbf{W}^o \mathbf{x}_v^t + \sum_{k \in \mathcal{N}_v} \mathbf{U}_{m(v,k)}^o \mathbf{h}_k^{t-1} + \mathbf{b}^o) \\
& \mathbf{t}_v^t = \tanh(\mathbf{W}^u \mathbf{x}_v^t + \sum_{k \in \mathcal{N}_v} \mathbf{U}_{m(v,k)}^u \mathbf{h}_k^{t-1} + \mathbf{b}^u) \\
& \mathbf{c}_v^t = \mathbf{i}_v^t \odot \mathbf{u}_v^t + \sum_{k \in \mathcal{N}_v} \mathbf{f}_{vk}^t \odot \mathbf{c}_k^{t-1}
\end{aligned} \tag{25}$$

$$\mathbf{h}_v^t = \mathbf{o}_v^t \odot \tanh(\mathbf{c}_v^t)$$

其中 $m(v, k)$ 表示节点 v 和之间的边标签 k 。

[62]提出了句子 LSTM (S-LSTM)来改进文本编码。它将文本转换成图形，并利用图形 LSTM 来学习表示。S-LSTM 在许多自然语言处理问题上表现出很强的代表性。[63]提出了一种图形 LSTM 网络来解决语义对象解析任务。它利用置信度划分方案自适应地选择起始节点并确定节点更新顺序。它遵循同样的思想，将现有的 LSTMs 推广到图形结构的数据中，但是有一个特定的更新序列，而我们上面提到的方法对节点的顺序是不可知的。

注意。注意机制已经成功应用于许多基于序列的任务，如机器翻译[64]-[66]，机器阅读[67]等。[68]提出了一种图形注意网络(GAT)，它将注意机制纳入传播步骤。它遵循自我关注策略，通过关注邻居来计算每个节点的隐藏状态。

[68]定义了单个图形注意层，并通过堆叠该层来构建任意图形注意网络。该层通过以下方式计算节点对 (i, j) 的注意机制中的系数：

$$\begin{aligned}
& \exp \text{ leakyrelu } aT[\text{Whiwh}_j]) \\
& \alpha_{ij} = \text{Pk} \in \mathcal{N}_i \exp(\text{Likeyrelu}(aT[\text{Whykwhk}]))
\end{aligned} \tag{26}$$

其中 α_{ij} 是节点 j 到 i 的关注系数， \mathcal{N}_i 代表图中节点 i 的邻域。图层的输入节点要素集是 $\mathbf{h} = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N\}$ ， $\mathbf{h}_i \in \mathbb{R}^F$ ，其中 N 是节点数， F 是每个节点的特征数，该层产生一组新的节点特征(可能具有不同的基数 F_0)， $\mathbf{h}' = \{\mathbf{h}'_1, \mathbf{h}'_2, \dots, \mathbf{h}'_N\}$ ， $\mathbf{h}'_i \in \mathbb{R}^{F'}$ ，作为其输出。 $\mathbf{W} \in \mathbb{R}^{F_0 \times F}$ 是应用于每个节点的共享线性变换的权重矩阵， $\mathbf{a} \in \mathbb{R}^{2F_0}$ 是单层前馈神经网络的权重向量。它由 softmax 函数归一化，并应用 LeakyReLU 非线性(负输入斜率 $\alpha = 0.2$)。

然后，每个节点的最终输出特性可通过以下方式获得(应用非线性 σ 后)：

$$\mathbf{h}'_i = \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \mathbf{h}_j \right) \tag{27}$$

此外，该层利用类似于[66]的多头注意力来稳定学习过程。它应用 K 个独立的注意机制来计算隐藏状态，然后连接它们的特征(或计算平均值)，得到以下两种输出表示：

$$\begin{aligned}
& \mathbf{h}'_i = \parallel \sigma \left(\sum_{k=1}^K \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j \right) \quad (28) \\
& \mathbf{h}'_i = \sigma \left(\frac{1}{K} \sum_{k=1}^K \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j \right) \quad (29)
\end{aligned}$$

其中 α_{ijk} 是由第 k 个注意机制计算的归一化注意系数。

[68]中的注意结构具有几个性质：(1)节点-邻居对的计算是可并行的，因此操作是有效的；(2)通过给邻居指定任意权重，它可以应用于不同程度的图节点；(3)它可以很容易地应用于归纳学习问题。

除 GAT 外，门控注意网络(GAAN) [69]也使用多头注意机制。但它用一种自我注意机制，从不同的头部收集信息，来代替 GAT 的平均操作。

跳过连接。许多应用程序展开或堆叠图形神经网络层，旨在获得更好的结果，因为更多的层(即 k 层)使每个节点从 k 跳之外的邻居那里聚集更多的信息。然而，在许多实验中已经观察到，更深的模型不能提高性能，更深的模型甚至可能表现得更差[2]。这主要是因为更多的层也可以传播来自指数增长的扩展邻域成员的噪声信息。

解决这个问题的一个简单的方法，剩余网络[70]，可以从计算机视觉社区中找到。但是，即使有剩余连接，在许多数据集上，具有更多层的格网表现不如 2 层 GCN 网[[2]。

[71]提出了一条 GCN 公路，它使用类似于[72]公路网的分层闸门。图层的输出与其输入相加，并带有选通权重：

$$\begin{aligned} \mathbf{T}(\mathbf{h}^t) &= \sigma(\mathbf{W}^t \mathbf{h}^t + \mathbf{b}^t) \\ \mathbf{h}^{t+1} &= \mathbf{h}^{t+1} \odot \mathbf{T}(\mathbf{h}^t) + \mathbf{h}^t \odot (1 - \mathbf{T}(\mathbf{h}^t)) \end{aligned} \quad (30)$$

\mathbf{h}

通过增加高速公路大门，性能在 [71]中讨论的特定问题中的 4 层。[73]中提出的柱网(CLN)也利用了公路网。但是它有不同功能来计算门控权重。

[74]研究邻域聚集方案的特性和结果限制。提出了能够学习自适应、结构感知表征的跳跃知识网络。跳转知识网络为最后一层的每个节点从所有中间表示(其“跳转”到最后一层)中进行选择，这使得模型根据需要为每个节点调整有效邻域大小。[74]在实验中使用串联、最大池和 LSTM 注意三种方法来聚集信息。跳跃知识网络在社会、生物信息学和引用网络的实验中表现良好。它还可以与图形卷积网络、图形搜索和图形注意力网络等模型相结合，以提高它们的性能。

分层池。在计算机视觉领域，卷积层之后通常是汇集层，以获得更一般的特征。与这些池层类似，大量工作集中在设计图形上的分层池层。复杂的大规模图通常具有丰富的层次结构，这对于节点级和图级分类任务非常重要。

为了探索这种内在特征，边缘条件卷积[75]设计了具有递归下采样操作的汇集模块。下采样方法是基于通过拉普拉斯算子的最大特征向量的符号将图形分成两个分量。

DIFFPOOL [76]提出了一个可学习的层次聚类模块，通过在每一层中训练一个分配矩阵：

$$\mathbf{S}(\mathbf{l}) = \text{softmax}(\mathbf{GNNl}, \text{pool}(\mathbf{A}(\mathbf{l}), \mathbf{X}(\mathbf{l}))) \quad (31)$$

其中 $\mathbf{X}(\mathbf{l})$ 为节点特征， $\mathbf{A}(\mathbf{l})$ 为 \mathbf{l} 层粗化邻接矩阵。

2.2.3 培训方法

原始的图卷积神经网络在训练和优化方法上有几个缺点。具体来说，GCN 需要完整的图拉普拉斯，这对于大型图来说是计算消耗。此外，一个节点在 $\mathbf{L1}$ 层的嵌入是通过其所有邻居在 $\mathbf{L1}$ 层的嵌入递归计算的。因此，单个节点的感受野相对于层数成指数增长，因此计算单个节点的梯度成本很高。最后，GCN 是针对固定图形独立训练的，缺乏归纳学习的能力。

采样。GraphSAGE [1]是对原始 GCN 的全面改进。为了解决上述问题，GraphSAGE 用可学习的聚合函数代替了全图 Laplacian，聚合函数是执行消息传递和推广到不可见节点的关键。如等式 19 所示，它们首先聚合邻域嵌入，与目标节点的嵌入相连接，然后传播到下一层。通过学习聚合和传播函数，GraphSAGE 可以为看不见的节点生成嵌入。此外，GraphSAGE 使用邻居采样来减轻感受野的扩展。

PinSage [77]提出了基于重要性的抽样方法。通过模拟从目标节点开始的随机行走，该方法选择具有最高归一化访问计数的前 T 个节点。

FastGCN [78]进一步改进了采样算法。FastGCN 不是对每个节点的邻居进行采样，而是直接对每一层的感受野进行采样。FastGCN 使用重要性抽样，其重要性因子计算如下：

$$q(v) \propto \frac{1}{|\mathcal{N}_v|} \sum_{u \in \mathcal{N}_v} \frac{1}{|\mathcal{N}_u|} \quad (32)$$

与上述固定采样方法相比，[79]引入了参数化和可训练的采样器，以前一层为条件执行逐层采样。此外，这种自适应采样器可以找到最佳采样重要性，同时减少方差。

继强化学习之后，SSE [80]为 GNN 训练提出了随机定点梯度下降。该方法将嵌入更新视为值函数，将参数更新视为值函数。在训练过程中，算法将交替采样节点更新嵌入和采样标记节点更新参数。

感受野控制。[81]提出了一种基于控制变量的 GCN 随机近似算法，它利用节点的历史激活作为控制变量。这种方法将感受野限制在 1hop 邻域内，但使用历史隐藏状态作为可承受的近似值。

数据扩充。[82]集中于 GCN 的局限性，包括 GCN 需要许多附加的标记数据进行验证，并且还受到卷积滤波器的局部性的影响。为了解决这些局限性，作者提出了联合训练 GCN 和自训练 GCN 来扩大训练数据集。前一种方法寻找训练数据的最近邻，而后一种方法遵循类似 boosting 的方式。

无监督培训。神经网络通常用于监督或半监督学习问题。最近，有一种趋势是将自动编码器扩展到图形领域。图形自动编码器旨在通过无监督的训练方式将节点表示成低维向量。

图形自动编码器(GAE) [83]首先使用 GCNs 对图形中的节点进行编码。然后用一个简单的解码器重构邻接矩阵，并根据原始邻接矩阵和重构矩阵之间的相似度计算损失。

$$\mathbf{Z} = \text{GCN}(\mathbf{X}, \mathbf{A})$$

$$\mathbf{T} \quad (33)$$

$$\mathbf{Ae} = \rho(\mathbf{ZZ})$$

[83]还以变分方式训练 GAE 模型，该模型被命名为变分图形自动编码器(VGAE)。此外，伯格等人。在推荐系统中使用 GAE，并提出了图形卷积矩阵完成模型(GC-MC) [84]，该模型在移动数据集上优于其他基线模型。

敌对地调整 图表 自动编码器

(ARGA) [85]使用生成对抗网络(GANs)来规范基于 GCN 的图形自动编码器，以遵循先验分布。还有几个图形自动编码器，如 NetRA [86]，DNDR [87]，SDNE [88]和 DRNE [89]，但是，它们的框架中没有使用 GNNs。

2.3 一般框架

除了图形神经网络的不同变体之外，还提出了几个通用框架，旨在将不同的模型集成到一个框架中。[27]提出了消息传递神经网络(MPNN)，它统一了各种图神经网络和图卷积网络方法。[28]提出了非局部神经网络。它统一了几种“自我关注”式的方法[66]、[68]、[90]。[30]提出了图形网络(GN)，它统一了MPNN和NLNN方法以及许多其他变体，如交互网络[4]，[91]，神经物理引擎[92]，通信网络[93]，结构2 电子商务[7]，[94]，GGNN[59]，关系网络[95]，[96]，深集[97]和点网络[98]。

2.3.1 消息传递神经网络

[27]提出了一个在图上进行监督学习的通用框架，称为消息传递神经网络。MPNN 框架抽象了几种最流行的图结构数据模型之间的共性，例如图卷积中的谱方法[45] [2]，[48]和非谱方法[51]，门控图神经网络[59]，相互作用网络[4]，分子图卷积[99]，深度张量神经网络[100]等等。

该模型包含两个阶段，消息传递阶段和读出阶段。消息传递阶段(即传播步骤)运行 T 个时间步骤，并根据消息函数 M_t 和顶点更新函数 U_t 来定义。使用消息 m_{vw} ，隐藏状态 \mathbf{h}_v^t 的更新功能如下：

$$\mathbf{m}_v^{t+1} = \sum_{w \in \mathcal{N}_v} M_t(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw})$$

$$\mathbf{h}_v^{t+1} = U_t(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}) \quad (34)$$

其中 \mathbf{e}_{vw} 表示从节点 v 到 w 的边的特征。读出阶段使用读出函数计算整个图形的特征向量

$$\mathbf{y}' = R(\{\mathbf{h}_v^T | v \in G\}) \quad (35)$$

其中 T 表示总时间步长。消息功能 M_t 、顶点更新功能 U_t 和读出功能 R 可以具有不同的设置。因此，MPNN 框架可以通过不同的功能设置推广几种不同的模型。这里我们给出了一个推广广义神经网络的例子，其他模型的功能设置可以在[27]中找到。GGNNs 的功能设置如下：

$$M_t(\mathbf{h}_v^t, \mathbf{h}_w^t, \mathbf{e}_{vw}) = \mathbf{A}_{\mathbf{e}_{vw}} \mathbf{h}_w^t$$

$$U_t = GRU(\mathbf{h}_v^t, \mathbf{m}_v^{t+1}) \quad (36)$$

$$R = \sum_{v \in V} \sigma(i(\mathbf{h}_v^T, \mathbf{h}_v^0)) \odot (j(\mathbf{h}_v^T))$$

其中 $\mathbf{A}_{\mathbf{e}_{vw}}$ 是邻接矩阵，每个边标签 \mathbf{e} 对应一个邻接矩阵。GRU 是在[57]中介绍的门控循环单位。 i 和 j 是函数 r 中的神经网络。

2.3.2 非局部神经网络

[28]提出了非局部神经网络(NLNN)，用于捕捉深度神经网络的长期依赖性。非局部运算是计算机视觉中经典非局部平均运算[101]的推广。非本地操作将某个位置的响应计算为所有位置的特征的加权和。这组位置可以在空间、时间或时空中。因此，神经网络可以被视为不同的“自我关注”风格的方法的统一[66]，[68]，[90]。我们将首先介绍非本地操作的一般定义，然后介绍一些具体的实例。

在非局部平均运算[101]之后，一般非局部运算被定义为：

$$\mathbf{h}_i' = \frac{1}{C(\mathbf{h})} \sum_{\forall j} f(\mathbf{h}_i, \mathbf{h}_j) g(\mathbf{h}_j) \quad (37)$$

其中 i 是输出位置的索引， j 是列举所有可能位置的索引。 $f(\mathbf{h}_i, \mathbf{h}_j)$ 计算 i 和 j 之间的标量，代表它们之间的关系。 $g(\mathbf{h}_j)$ 表示输入 \mathbf{h}_j 的变换，因子 $\frac{1}{C(\mathbf{h})}$ 用于归一化结果。

有几个不同 f 和 g 设置的实例。为简单起见，[28]使用线性变换作为函数 g 。这意味着 $g(\mathbf{h}_j) = \mathbf{W}g\mathbf{h}_j$ ，其中 $\mathbf{W}g$ 是一个学习的权重矩阵。下面我们列出函数 f 的选择。

高斯。根据非局部平均[101]和双边滤波器[102]，高斯函数是自然选择。因此：

$$f(\mathbf{h}_i, \mathbf{h}_j) = e^{\mathbf{h}_i^T \mathbf{h}_j} \quad (38)$$

这里 $\mathbf{h}_i^T \mathbf{h}_j$ 是点积相似度， $C(\mathbf{h}) = \sum_j f(\mathbf{h}_i, \mathbf{h}_j)$ 。

嵌入式高斯。通过计算嵌入空间中的相似性来扩展高斯函数很简单，这意味着：

$$f(\mathbf{h}_i, \mathbf{h}_j) = e^{\theta(\mathbf{h}_i)^T \phi(\mathbf{h}_j)} \quad (39) \text{ 其中 } \theta(\mathbf{h}_i)$$

$$= \mathbf{W}\theta\mathbf{h}_i, \phi(\mathbf{h}_j) = \mathbf{W}\phi\mathbf{h}_j \text{ 和 } C(\mathbf{h}) =$$

$$\sum_j f(\theta(\mathbf{h}_i), \phi(\mathbf{h}_j))$$

可以发现，[[66]提出的自我注意是嵌入高斯型的一个特例。对于给定的 i ， $\frac{1}{C(\mathbf{h})} f(\mathbf{h}_i, \mathbf{h}_j)$ 成为沿维度 j 的软最大计算。所以 $\mathbf{h}_0 = \text{softmax}(\mathbf{h}^T \mathbf{W} \theta \mathbf{W} \phi \mathbf{h}) \mathbf{g}(\mathbf{h})$ ，这与[66]中的自我注意形式相匹配。

点积。函数 f 也可以实现为点积相似度：

$$f(\mathbf{h}_i, \mathbf{h}_j) = \theta(\mathbf{h}_i)^T \phi(\mathbf{h}_j) \quad (40)$$

这里的因子 $C(\mathbf{h}) = \mathbf{N}$ ，其中 \mathbf{N} 是 \mathbf{h} 中的位置数。

拼接。这里我们有：

$$f(\mathbf{h}_i, \mathbf{h}_j) = \text{ReLU}(\mathbf{w}_f^T [\theta(\mathbf{h}_i) \parallel \phi(\mathbf{h}_j)]) \quad (41)$$

其中 \mathbf{w}_f 是将向量投影到标量的权重向量， $C(\mathbf{h}) = \mathbf{N}$ 。

[28]将上述非本地操作包装到非本地块中，如下所示：

$$\mathbf{z}_i = \mathbf{W}_z \mathbf{h}_i' + \mathbf{h}_i, \quad (42)$$

其中 \mathbf{h}_i' 在等式 37 中给出，并且“+ \mathbf{h}_i ”表示剩余连接[70]。因此，非局部块可以被插入到任何预先训练的模型中，这使得该块更加适用。

2.3.3 图形网络

[30]提出了图形网络(GN)框架，该框架概括和扩展了各种图形神经网络、MPNN 和 NLNN 方法[27]、[28]、[33]。我们首先介绍[30]中的图形定义，然后描述核心 GN 计算单元 GN 块及其计算步骤，最后介绍其基本设计原理。

图形定义。在[30]中，一个图被定义为一个三元组 $G = (\mathbf{u}, \mathbf{H}, \mathbf{E})$ (这里我们使用 \mathbf{H} 而不是 \mathbf{V} 来表示符号的一致性)。 \mathbf{u} 是

一个全局属性, $H = \{h_i\}_{i=1:N_v}$ 是节点集(基数为 N_v), 其中每个 h_i 是一个节点的属性。 $E = \{(ek, rk, sk)\}_{k=1:N_e}$ 是边的集合(基数为 N_e), 其中每个 ek 是边的属性, rk 是接收节点的索引, sk 是发送节点的索引。

GN 块。 GN 块包含三个“更新”函数 ϕ 和三个“聚合”函数 ρ ,

$$\begin{aligned} e0k &= \phi_e(ek, hrk, hsk, u) & \bar{e}'_i &= \rho^{e \rightarrow h}(E'_i) \\ h'_i &= \phi^h(\bar{e}'_i, h_i, u) & e0 &= \rho_{e \rightarrow u}(E0) \\ u' &= \phi^u(\bar{e}', \bar{h}', u) & \bar{h}' &= \rho^{h \rightarrow u}(H') \end{aligned} \quad (43)$$

其中 $E0 = \{(e0k, rk, sk)\}_{rk=i, k=1:N_e}$, $H0 = \{h0i\}_{i=1:N_v}$, $E0 = \{(e0k, rk, sk)\}_{k=1:N_e}$ 。 ρ 函数必须对其输入的排列保持不变, 并且应该采用可变数量的参数。

计算步骤。 GN 块的计算步骤如下:

- 1) ϕ_e 是应用的每边缘, 和争论 (ek, hrk, hsk, u) , 并返回 $e0k$ 。每个节点 i 的每沿输出结果集为 $E'_i = \{(e'_k, r_k, s_k)\}_{r_k=i, k=1:N_e}$ 。和 $E' = \bigcup_i E'_i = \{(e'_k, r_k, s_k)\}_{k=1:N_e}$ 是集合所有每边沿输出的。
- 2) $\rho_{e \rightarrow h}$ 应用于 $Ei0$, 并将投影到顶点 i 的边的边更新聚合到 $e0i$ 中, $e0i$ 将用于下一步的节点更新。
- 3) ϕ_h 被应用于每个节点 i , 以计算更新的节点属性 $h0i$ 。得到的每节点输出集为, $H0 = \{h0i\}_{i=1:N_v}$ 。
- 4) $\rho_{e \rightarrow u}$ 应用于 $e0$, 并将所有边缘更新聚合到 $E0$ 中, 然后将在下一步的全局更新中使用。
- 5) $\rho_{h \rightarrow u}$ 应用于 $H0$, 并将所有节点更新聚合到 $h0$ 中, 然后将在下一步的全局更新中使用。
- 6) 每个图应用一次 ϕ_u , 并计算全局属性 $u0$ 的更新。

注意这里的命令并没有严格执行。例如, 可以从全局到每个节点, 再到边缘更新。 ϕ 和 ρ 函数不必是神经网络, 尽管本文只关注神经网络的实现。

设计原则。 图形网络的设计基于三个基本原则: 灵活的表示、可配置的块内结构和可组合的多块体系结构。

- **灵活的表述。** GN 框架支持灵活的属性表示以及不同的图形结构。全局、节点和边缘属性可以使用任意的表示格式, 但是实值向量和张量是最常见的。人们可以简单地根据任务的特定需求来定制 GN 块的输出。例如, [30]列出了几个边缘聚焦的[103]、[104]、节点聚焦的[3]、[4]、[92]、[105]和图形聚焦的[4]、[27]、[96] GNs。就图结构而言, 该框架既可应用于图结构显式的结构场景, 也可应用于应推断或假设关系结构的非结构场景。
- **可配置的块内结构。** GN 块中的函数及其输入可以有不同的设置, 以便 GN 框架在块内结构配置中提供灵活性。例如, [104]和[3]使用完整的 GN 块。它们的 ϕ 实现使用神经网络, 它们的 ρ 函数使用元素求和。基

于不同的结构和功能设置, 各种模型(如 MPNN、非线性神经网络和其他变体)可以通过 GN 框架来表达。更多细节可以在[30]中找到。

- **可组合的多块架构。** 可以组成 GN 块来构建复杂的体系结构。任意数量的 GN 块可以用共享或非共享参数按顺序组成。[30]利用 GN 块构造编码-解码结构和基于 GN 的递归结构。这些架构在图 2 中进行了演示。3. 构建基于 GN 的架构的其他技术也很有用, 例如跳过连接、LSTM 或 GRU 风格的门控方案等。

3 应用

图形神经网络已经在有监督、半监督、无监督和强化学习设置的广泛问题领域中被探索。在这一节中, 我们简单地将应用程序分为三种场景:(1)数据具有显式关系结构的结构场景, 如物理系统、分子结构和知识图; (2)关系结构不明确的非结构场景包括图像、文本等; (3)生成模型、组合优化问题等其他应用场景。请注意, 我们只列出几个有代表性的应用程序, 而不是提供一个详尽的列表。这些应用的概要见表 3。

3.1 结构场景

在接下来的小节中, 我们将介绍 GNN 在结构化场景中的应用, 其中数据自然地在图结构中执行。例如, GNNs 被广泛用于社交网络预测[1], [2], 流量预测[71], [144], 推荐系统[77], [84]和图形表示[76]。具体来说, 我们正在讨论如何用对象关系图来建模现实世界的物理系统, 如何预测分子的化学性质和蛋白质的生物相互作用性质, 以及在知识图中推理知识库外(OOKB)实体的方法。

3.1.1 物理学

模拟真实世界的物理系统是理解人类智能的最基本的方面之一。通过将对象表示为节点, 将关系表示为边, 我们可以以一种简化但有效的方式对对象、关系和物理进行基于 GNN 的推理。

[4]提出了交互网络来对各种物理系统进行预测和推断。该模型将对象和关系作为输入, 推理它们的相互作用, 并应用效果和物理动力学来预测新的状态。它们分别对以关系为中心和以对象为中心的模型进行建模, 使得在不同的系统中进行归纳更加容易。在通信网[93]中, 交互没有被显式建模。相反, 通过平均所有其他代理的隐藏向量来获得交互向量。VAIN [90]进一步将注意方法引入代理交互过程, 既保持了复杂性优势, 又保持了计算效率。

视觉交互网络[91]可以通过像素进行预测。它从每个对象的两个连续输入帧中学习状态代码。然后, 在通过交互网络块添加它们的交互效果之后, 状态解码器将状态代码转换为下一步的状态。

[3]提出了一个基于图形网络的模型，它可以执行状态预测或归纳推理。推理模型以部分观察到的信息为输入，构建隐含系统分类的隐藏图。

$$\mathbf{h}_{\mathcal{N}_v}^t = \sum_{u \in \mathcal{N}(v)} \text{CONCAT} \quad (44)$$

其中 \mathbf{e}_{uv} 是 edge (u, v) 的边缘特征。然后通过以下方式更新节点表示

$$\mathbf{h}_v^{t+1} = \sigma(\mathbf{W}_t^{\text{deg}(v)} \mathbf{h}_{\mathcal{N}_v}^t) \quad (45)$$

其中 $\text{deg}(v)$ 是节点 v 的度数， $\mathbf{W}_t \mathbf{N}$ 是每个时间步长 t 和节点度数 n 的学习矩阵。

[99]进一步明确地独立模拟原子和原子对，以强调原子相互作用。它引入了边缘表示 \mathbf{e}_{uv} 而不是聚集函数，即。

$\mathbf{h}_v^{t+1} = \text{ReLU}(\mathbf{W}_1(\text{ReLU}(\mathbf{W}_0 \mathbf{h}_u^t), \mathbf{h}_{\mathcal{N}_v}^t))$ 节点更新功能是

$$\mathbf{h}_v^{t+1} = \text{ReLU}(\mathbf{W}_1(\text{ReLU}(\mathbf{W}_0 \mathbf{h}_u^t), \mathbf{h}_{\mathcal{N}_v}^t)) \quad (46)$$

3.1.2 化学和生物学

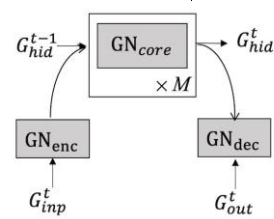
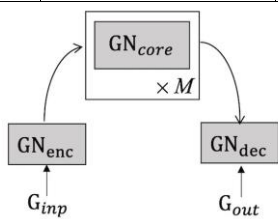
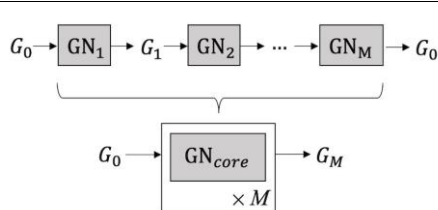
分子指纹计算分子指纹，即代表分子的特征向量，是计算机辅助药物设计的核心步骤。传统的分子指纹是手工制作和固定的。通过将 GNN 应用于分子图，我们可以获得更好的指纹。

[51]提出了通过 GCN 和求和计算子结构特征向量以获得整体表示的神经图指纹。聚合函数是

表 3
图形神经网络的应用。

Area	Application	Algorithm	Deep Learning Model	References
Text	Text classification	GCN	Graph Convolutional Network	[1], [23], [48] [2], [22], [46]
		GAT	Graph Attention Network	[68]
		DGCNN	Graph Convolutional Network	[106]
		Text GCN	Graph Convolutional Network	[107]
		Sentence LSTM	Graph LSTM	[62]
	Sequence Labeling (POS, NER)	Sentence LSTM	Graph LSTM	[62]
	Sentiment classification	Tree LSTM	Graph LSTM	[60]
	Semantic role labeling	Syntactic GCN	Graph Convolutional Network	[108]
	Neural machine translation	Syntactic GCN	Graph Convolutional Network	[109], [110]
		GGNN	Gated Graph Neural Network	[38]
	Relation extraction	Tree LSTM	Graph LSTM	[111]
		Graph LSTM	Graph LSTM	[44], [112]
		GCN	Graph Convolutional Network	[113]
	Event extraction	Syntactic GCN	Graph Convolutional Network	[114], [115]
	AMR to text generation	Sentence LSTM	Graph LSTM	[116]
		GGNN	Gated Graph Neural Network	[38]
Image	Multi-hop reading comprehension	Sentence LSTM	Graph LSTM	[117]
	Relational reasoning	RN	MLP	[96]
		Recurrent RN	Recurrent Neural Network	[118]
		IN	Graph Neural Network	[4]
	Social Relationship Understanding	GRM	Gated Graph Neural Network	[119]
	Image classification	GCN	Graph Convolutional Network	[120], [121]
		GGNN	Gated Graph Neural Network	[122]
		DGP	Graph Convolutional Network	[35]
		GSNN	Gated Graph Neural Network	[123]
	Visual Question Answering	GGNN	Gated Graph Neural Network	[119], [124], [125]
	Object Detection	RN	Graph Attention Network	[126], [127]
	Interaction Detection	GPNN	Graph Neural Network	[128]
		Structural-RNN	Graph Neural Network	[42]
	Region Classification	GCNN	Graph CNN	[129]
	Semantic Segmentation	Graph LSTM	Graph LSTM	[63], [130]
		GGNN	Gated Graph Neural Network	[131]
		DGCNN	Graph CNN	[132]
		3DGNN	Graph Neural Network	[133]

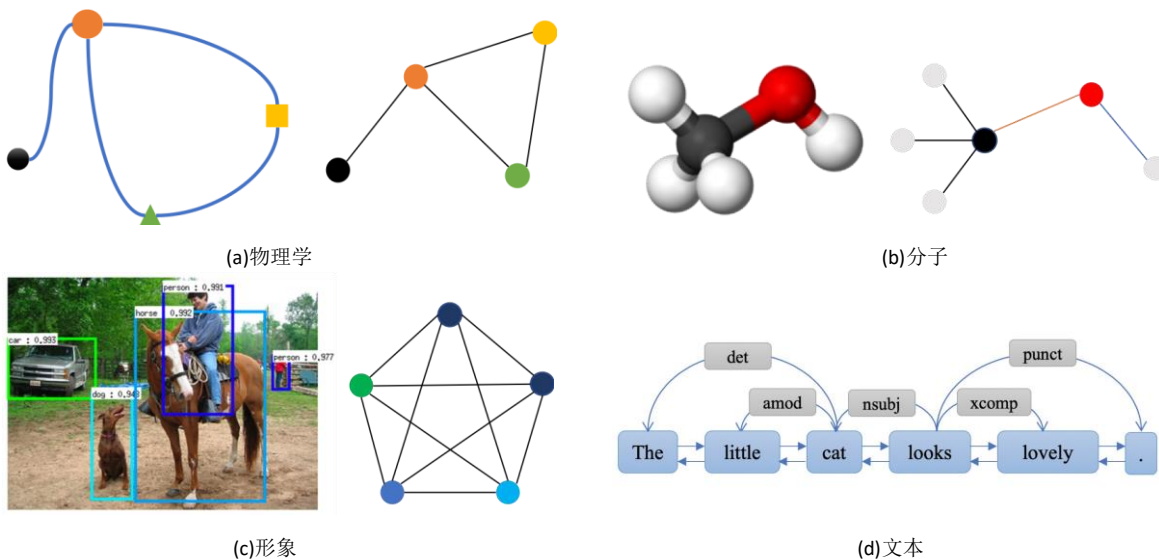
Science	Physics Systems	IN	Graph Neural Network	[4]
		VIN	Graph Neural Network	[91]
		GN	Graph Networks	[3]
	Molecular Fingerprints	NGF	Graph Convolutional Network	[51]
		GCN	Graph Convolutional Network	[99]
		GCN	Graph Convolutional Network	[5]
	Protein Interface Prediction	GCN	Graph Convolutional Network	[5]
	Side Effects Prediction	Decagon	Graph Convolutional Network	[134]
Knowledge Graph	Disease Classification	PPIN	Graph Convolutional Network	[135]
	KB Completion	GNN	Graph Neural Network	[6]
Knowledge Graph	KG Alignment	GCN	Graph Convolutional Network	[136]
Combinatorial Optimization		structure2vec	Graph Convolutional Network	[7]
		GNN	Graph Neural Network	[137]
		GCN	Graph Convolutional Network	[138]
		AM	Graph Attention Network	[139]
Graph Generation		NetGAN	Long short-term memory	[140]
		GraphRNN	Recurrent Neural Network	[137]
		Regularizing VAE	Variational Autoencoder	[141]
		GCPN	Graph Convolutional Network	[142]
		MolGAN	Relational-GCN	[143]

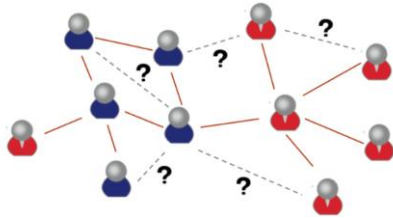


顺序控制块编码-处理-解码

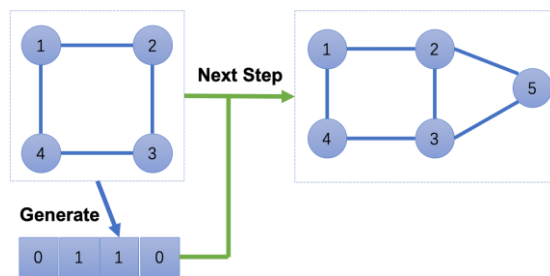
经常性政府大楼

图. 3. 由 GN 块组成的架构示例。顺序处理架构；编码-处理-解码结构；循环架构。





(e) 社交网络
图 4. 应用场景



(f) 世代

而边缘更新功能是

$$\mathbf{e}_{uv}^{t+1} = \text{ReLU}(\mathbf{W}_4(\text{ReLU}(\mathbf{e}_{uv}^t, \text{ReLU}(\mathbf{W}_3(\text{htv}, \text{htu})))) \quad (47)$$

蛋白质界面预测[5]侧重于蛋白质界面预测，这是一个具有挑战性的问题，在药物发现和设计中有重要应用。所提出的基于 GCN 的方法分别学习配体和受体蛋白残基表示，并将其合并用于成对分类。

GNN 也可以用于生物医学工程。多种药物副作用预测。他们的工作模拟了药物和蛋白质的相互作用网络，并分别处理不同类型的边缘。

3.1.3 知识图

[6]利用神经网络解决知识库构建中的知识库外实体问题(KBC)。[6]中的 OOKB 实体直接连接到现有实体，因此 OOKB 实体的嵌入可以从现有实体中聚合。该方法在标准 KBC 设置和 OOKB 设置下均获得了令人满意的性能。

[136]利用 GCNs 解决跨语言知识图对齐问题。该模型将不同语言的实体嵌入到一个统一的嵌入空间中，并根据嵌入的相似性对它们进行对齐。

3.2 非结构性场景

在本节中，我们将讨论非结构化场景的应用，如图像、文本、编程源代码[59]、[145]和多代理系统[90]、[93]、[103]。由于篇幅限制，我们只对前两种情况进行详细介绍。粗略地说，有两种方法可以将图神经网络应用于非结构场景：(1)结合来自其他领域的结构信息来提高性能，例如使用来自知识图的信息来减轻图像任务中的零镜头问题；(2)推断或假设场景中的关系结构，然后应用该模型来解决图形上定义的问题，例如[62]中的方法，该方法将文本建模为图形。

3.2.1 形象

图像分类图像分类是计算机视觉领域中一项非常基础和重要的任务，备受关注，并拥有许多著名的数据集，如 ImageNet [146]。图像分类的最新进展得益于大数据和 GPU 计算的强大

蛋白质-蛋白质相互作用网络，[135]利用图表

乳腺癌亚类型分类的卷积和关系网络。[134]还提出了一个基

能力，这使得我们可以在不从图像中提取结构信息的情况下训练分类器。然而，零镜头和少镜头学习在图像分类领域变得越来越流行，因为大多数模型可以在足够的数据下实现相似的性能。有几个工作利用图形神经网络将结构信息纳入图像分类。首先，知识图可以用作额外的信息来指导零短路识别分类[35]，[121]。[121]构建了一个知识图，其中每个节点对应于一个对象类别，并将节点的单词嵌入作为输入，用于预测不同类别的分类器。由于过平滑效应随着卷积结构的深度而发生，在[121]中使用的 6 层 GCN 将洗掉表示中的许多有用信息。为了解决 GCN 传播中的平滑问题，[35]设法使用具有更大邻域的单层 GCN，该邻域包括图中的单跳和多跳节点。并证明了利用现有的零镜头分类器构建零镜头分类器的有效性。

除了知识图，数据集中图像之间的相似性也有助于少拍学习[120]。[120]提出建立基于相似度的加权全连通图像网络，并在图中传递信息进行少镜头识别。由于大多数知识图对于推理来说都很大，[123]基于对象检测的结果选择一些相关的实体来构建子图，并将广义神经网络应用于所提取的图来进行预测。此外，[122]提出构建一个新的知识图，其中实体是所有的类别。并且，他们定义了三种类型的标签关系：超纵坐标、正相关和负相关，并直接在图中传播标签的置信度。

视觉推理计算机视觉系统通常需要结合空间和语义信息进行推理。所以为推理任务生成图是很自然的。

典型的视觉推理任务是视觉问答(VQA)，[124]分别构建图像场景图和问题句法图。然后应用遗传神经网络训练嵌入，预测最终答案。尽管对象之间存在空间联系，[147]还是根据问题构建了关系图。有了知识图，[119]，[125]可以执行更精细的关系探索和更可解释的推理过程。

视觉推理的其他应用包括对象检测、交互检测和区域分类。在目标检测[126]、[127]中，GNNs 用于计算 RoI 特征；在交互检测[42]、[128]中，GNNs 是人对象之间的消息传递工具；在区域分类[129]中，神经网络对连接区域和类别的图进行推理。

语义分割语义分割是图像理解的关键步骤。这里的任务是为图像中的每一个像素分配一个唯一的标签(或类别)，这可以被认为是一个密集的分类问题。然而，图像中的区域往往不是网格状的，需要非局部信息，导致传统 CNN 的失败。一些作品利用图形结构的数据来处理它。

[63]提出了图 LSTM 模型，通过以基于距离的超像素地图的形式构建图，并应用 LSTM 在全球传播邻域信息，来对长期相关性和空间联系进行建模。后续工作从编码层次信息的角度对其进行了改进[130]。

此外，3D 语义分割(RGBD 语义分割)和点云分类利用更多的几何信息，因此很难用 2D CNN 建模。[133]构建一个 K 近邻(KNN)图，并使用 3D GNN 作为传播模型。在展开几个步骤后，预测模型将每个节点的隐藏状态作为输入，并预测其语义标签。

由于总是有太多的点，[131]通过构建超图并为它们生成嵌入来解决大规模 3D 点云分割。为了分类超节点，[131]利用 GGNN 和图形卷积。

[132]提出通过边缘模拟点相互作用。它们通过输入终端节点的坐标来计算边缘表示向量。然后通过边聚合更新节点嵌入。

3.2.2 文本

图形神经网络可以应用于几个基于文本的任务。它可以应用于句子级别的任务(例如。文本分类)以及单词级任务(例如。序列标记)。下面我们将介绍几种主要的文本应用。

文本分类文本分类是自然语言处理中一个重要的经典问题。经典的 GCN 模型[1], [2], [22], [23], [46], [48]和 GAT 模型[68]被用来解决这个问题，但是它们只使用文档之间的结构信息，并且它们不使用太多的文本信息。[106]提出了一种基于 graphCNN 的深度学习模型，首先将文本转换为单词图，然后使用[53]中的图卷积运算来卷积单词图。[62]提议用句子 LSTM 来编码文本。它把整个句子看作一个单一的状态，由单个单词的子状态和整个句子级的状态组成。它使用全局句子级表示进行分类任务。这些方法要么将文档或句子视为单词节点图，要么依赖文档引用关系来构建该图。[107]将文档和单词视为构建语料库图(因此是异构图)的节点，并使用文本 GCN 学习单词和文档的嵌入。情感分类也可以看作是一个文本分类问题，树 LSTM 方法是由[60]提出的。

序列标注由于神经网络中的每个节点都有其隐藏状态，如果我们将句子中的每个单词都视为一个节点，我们可以利用隐藏状态来解决序列标注问题。[62]利用句子 LSTM 来标记序列。在词性标注和 NER 任务上进行了实验，取得了良好的效果。

语义角色标注是序列标注的另一项任务。[108]提出了一个句法 GCN 来解决这个问题。句法 GCN 是 GCN [2]的一个特殊变体，它作用于有标记边的直接图。它集成了边向门，让模型调节单个依赖边的贡献。句法依存关系树上的句法格网被用作句子编码器来学习句子中单词的潜在特征表示。[108]还揭示了 GCNs 和 LSTMs 在任务中的功能互补。

神经机器翻译神经机器翻译任务通常被认为是序列到序列的任务。[66]引入了注意机制，并取代了最常用的循环层或

卷积层。事实上，转换器假设语言实体之间是完全连通的图结构。

GNN 的一个流行应用是将句法或语义信息结合到 NMT 任务中。[109]在句法感知 NMT 任务中使用句法 GCN。[110]使用句法 GCN 将关于源句的表语结构的信息(即语义角色表示)合并，并比较仅将句法或语义信息或这两种信息合并到任务中的结果。[38]在句法感知 NMT 中利用广义神经网络。它通过将边转化为附加节点，将句法依赖图转化为一种新的结构，称为李维图，因此边标签可以表示为嵌入。

关系抽取提取文本中实体之间的语义关系是一项重要且研究充分的任务。一些系统将此任务视为两个独立任务的管道，命名为实体识别和关系提取。[111]提出了一个端到端的关系提取模型，使用双向顺序和树形结构的 LSTM-RNNs。[113]提出了一种图卷积网络的扩展，它是为关系抽取定制的，并对输入树应用了修剪策略。

跨句子 N 元关系抽取检测跨多个句子的 N 个实体之间的关系。[44]探索了基于图 LSTMs 的跨句 n 元关系抽取的一般框架。它将输入图形分割成两个 DAG，而重要的信息可能会在分割过程中丢失。[112]提出了图态 LSTM 模型。它保留了原始的图形结构，并通过允许更多的并行化来加速计算。

事件提取事件提取是一项重要的信息提取任务，用于识别文本中特定类型事件的实例。[114]研究了一种基于依赖树的卷积神经网络(确切地说是句法 GCN)来执行事件检测。[115]提出了一个联合多事件提取(JMEE)框架，通过引入语法快捷弧来联合提取多个事件触发器和参数，以增强流向基于注意力的图形卷积网络的信息流，从而对图形信息进行建模。

其他应用 GNNs 也可以应用于许多其他应用。有几个工作集中在文本生成任务的 AMR。在该领域已经提出了基于句子的 LSTM 方法[116]和基于广义神经网络的方法[38]。[60]使用树 LSTM 来模拟两个句子的语义关联性。[117]利用句子 LSTM 解决多级阅读理解问题。另一个重要方向是关系推理，提出了关系网络[96]、交互网络[4]和递归关系网络[118]来解决基于文本的关系推理任务。上面引用的工作不是一个详尽的列表，我们鼓励我们的读者找到更多他们感兴趣的图形神经网络的工作和应用领域。

3.3 其他场景

除了结构性和非结构性场景之外，还有一些图形神经网络发挥重要作用的其他场景。在这一小节中，我们将介绍生成图模型和使用 GNNs 的组合优化。

3.3.1 生成模型

现实世界图的生成模型因其重要的应用，包括建模社会交互、发现新的化学结构和构建知识图而受到极大关注。由于深度学习方法具有强大的学习图的隐式分布的能力，近年来神经图生成模型激增。

NetGAN [140]是最早建立神经图生成模型的工作之一，该模型通过随机行走生成图。它将图生成问题转化为行走生成问题，以特定图的随机行走为输入，利用遗传神经网络结构训练行走生成模型。生成的图保留了原图的重要拓扑性质，

但在生成过程中节点数无法改变，与原图相同。GraphRNN [148]设法通过逐步生成每个节点的邻接向量来生成图的邻接矩阵，这可以输出具有不同节点数量的所需网络。

MolGAN [143]不是按顺序生成邻接矩阵，而是立即预测离散图结构(邻接矩阵)，并利用置换不变鉴别器来解决邻接矩阵中的节点变化问题。此外，它还应用了一个奖励网络，用于基于 RL 的优化，以获得所需的化学性质。此外，[141]提出了约束变分自动编码器，以确保生成的图的语义有效性。而且，GCPN [142]通过强化学习纳入了特定领域的规则。

[149]提出了一种模型，该模型顺序地生成边和节点，并利用图神经网络来提取当前图的隐藏状态，该隐藏状态用于在顺序生成过程中决定下一步的动作。

3.3.2 组合最优化

图的组合优化问题是一组 NP 难问题，受到了各个领域科学家的广泛关注。一些具体的问题，如旅行商问题和最小生成树问题，都有不同的启发式解法。最近，使用深度神经网络来解决这样的问题已经成为一个热点，并且一些解决方案由于其图形结构而进一步利用图形神经网络。

[150]首先提出了一种解决 TSP 的深度学习算法。他们的方法由两部分组成：用于参数化奖励的指针网络[151]和用于训练的策略梯度[152]模块。这项工作已被证明与传统方法相当。然而，指针网络是为像文本这样的顺序数据设计的，而顺序不变的编码器更适合这种工作。

[153]和[139]通过包括图形神经网络改进了上述方法。前一项工作首先从 structure2vec [94]中获取节点嵌入，然后将它们输入到一个 Q 学习模块中进行决策。后者构建了一个基于注意力的编解码系统。通过用基于注意力的解码器代替强化学习模块，它对于训练更有效。这些工作取得了比以往算法更好的性能，证明了图神经网络的表示能力。

[137]集中于二次分配问题，即。测量两个图形的相似度。基于 GNN 的模型独立地学习每个图的节点嵌入，并使用注意机制来匹配它们。这种方法提供了有趣的良好表现，即使在标准的基于放松的技术似乎受到影响的制度。

四 未解决的问题

虽然神经网络在不同的领域取得了巨大的成功，但值得注意的是，GNN 模型不足以在任何条件下为任何图形提供令人满意的解。在这一部分，我们将提出一些有待进一步研究的问题。

浅结构传统的深神经网络可以堆叠数百层以获得更好的性能，因为更深的结构具有更多的参数，这显著提高了表达能力。然而，图形神经网络总是很浅，大多数不超过三层。如 [82]中的实验所示，堆叠多个 GCN 层将导致过度平滑，也就是说，所有顶点都将收敛到相同的值。虽然一些研究人员已经设法解决了这个问题[59]，[82]，但它仍然是 GNN 的最大局限。设计真正的深 GNN 对未来的研究是一个令人兴奋的挑战，并将对理解 GNN 做出相当大的贡献。

动态图另一个具有挑战性的问题是如何处理具有动态结构的图。静态图是稳定的，因此它们可以被可行地建模，而动态图引入了变化的结构。当边缘和节点出现或消失时，GNN 不能自适应地改变。充满活力的 GNN 正在被积极研究，我们相信这是 GNN 将军的稳定性和适应性的一个重要里程碑。

非结构性场景虽然我们已经讨论了 GNN 在非结构性场景中的应用，但是我们发现没有从原始数据生成图形的最佳方法。在图像领域，一些工作利用 CNN 获得特征地图，然后对它们进行上采样以形成作为节点的超像素[63]，而其他工作直接利用一些对象检测算法来获得对象节点。在文本领域 [129]，一些工作使用句法树作为句法图，而另一些采用完全连通图。因此，找到最佳的图形生成方法将提供 GNN 可以做出贡献的更广泛的领域。

可扩展性如何在像社交网络或推荐系统这样的网络规模条件下应用嵌入方法几乎是所有图形嵌入算法的致命问题，GNN 也不例外。扩展 GNN 很困难，因为在大数据环境中，许多核心步骤都是计算密集型的。关于这种现象有几个例子：首先，图形数据不是规则的欧几里得，每个节点都有自己的邻域结构，因此不能应用批处理。然后，当有数百万个节点和边时，计算图拉普拉斯也是不可行的。此外，我们需要指出，缩放决定了一个算法是否能够应用于实际应用。一些工作已经提出了解决这个问题的方法[77]，我们正在密切关注进展。

5 结论

在过去的几年里，图形神经网络已经成为图形领域机器学习任务的强大而实用的工具。这一进步归功于表达能力、模型灵活性和训练算法的进步。在本调查中，我们对图神经网络进行了全面的回顾。对于 GNN 模型，我们介绍了它的变体，按照图形类型、传播类型和训练类型进行分类。此外，我们还总结了几种通用框架来统一表示不同的变体。在应用程序分类方面，我们将 GNN 应用程序分为结构化场景、非结构化场景和其他场景，然后对每个场景中的应用程序进行详细的回顾。最后，我们提出了四个开放的问题，指出了图形神经网络的主要挑战和未来的研究方向，包括模型深度、可扩展性、处理动态图形和非结构场景的能力。

引用

- [1] W. 长度。汉密尔顿，z. 英，还有 j. Leskovec, “大型图上的归纳表示学习”，NIPS 2017, 页。1024–1034, 2017.
- [2] T. 名词 (noun 的缩写)。Kipf 和 m. 韦林, “图卷积网络的半监督分类”, ICLR 2017, 2017.
- [3] A. 桑切斯-冈萨雷斯。Heess, j. T. 斯普林伯格, j. Merel, m. Riedmiller, r. 哈德塞尔和 p. 巴塔格利亚, “图形网络作为推理和控制的机器学习物理引擎”, arXiv 预印本 arXiv:1806.01242, 2018.
- [4] 页 (page 的缩写)。巴塔格利亚。Pascanu, m. 赖, d. J. Rezende 等人, “用于学习物体、关系和物理的交互网络”, 发表于 NIPS 2016, 2016, 页。4502–4510.
- [5] A. Fout, j. 伯德, b. Shariat, 还有 a. 宾虚, “使用图形卷积网络的蛋白质界面预测”, 在 NIPS 2017, 2017, 页。6530–6539.
- [6] T. 滨口, h. Oiwa, m. Shimbo, 还有 y. 松本, “知识库外实体的知识转移:一种图形神经网络方法”, 载于 IJCAI 2017, 2017, 页。1802–1808.
- [7] H. 戴, e. B. 哈利勒, y. 张, b. Dilkina, 和 l. 宋, “学习图的组合优化算法”, arXiv 预印本 arXiv:1704.01665, 2017.

- [8] Y. 乐村, I. Bottou. 本吉奥, 还有 p. 基于梯度的学习应用于文档识别. 86, 不是. 11 页. 2278–2324, 1998.
- [9] Y. 乐村, y. 本吉奥, 还有 g. 韩丁, “深度学习”, 《自然》, 2015 年, 第一卷. 521, 没有. 7553, p. 436, 2015.
- [10] F. R. 钟和 f. C. 格雷厄姆, 谱图论. 美国数学足球. 1997 年, 没有. 92.
- [11] 页 (page 的缩写). 崔, x. 王, j. 裴, 和 w. 朱, “网络嵌入研究综述”, IEEE 知识与数据工程学报, 2018.
- [12] W. 长度. 汉密尔顿, r. 英, 还有 j. 莱斯科维克, “图的表示学习: 方法和应用. 电气和电子工程师协会数据(基础)工程公报, 卷. 40 页. 52–74, 2017.
- [13] D. 张, j. 阴, x. 朱, 和丙. 张, “网络表征学习: 一项调查”, IEEE 大数据事务, 2018.
- [14] H. 蔡, v. W. 郑, 还有 k. C. -C. 常, “图形嵌入的综合调查: 问题、技术和应用”, 《知识与数据工程学报》, 第一卷. 30 不. 9 页. 1616–1637, 2018.
- [15] 页 (page 的缩写). Goyal 和 e. 费拉拉, “图形嵌入技术, 应用和性能: 调查”, 基于知识的系统, 卷. 151 页. 78–94, 2018.
- [16] T. Mikolov, k. 陈, g. 科拉多和 j. 迪恩, “向量空间中单词表示的有效估计”, arXiv 预印本 arXiv:1301.3781, 2013.
- [17] B. Perozzi, r. Al-Rfou, and S. Skiena, “Deepwalk: 在线学习社会表征”, SIGKDD 2014.ACM, 2014, pp. 701–710.
- [18] A. 格罗弗和 j. Leskovec, “node2vec: 网络的可扩展特征学习”, SIGKDD. ACM, 2016, pp. 855–864.
- [19] J. 唐, m. 曲, m. 王, m. 张, j. 颜, 又问. 梅, “线: 大规模信息网络嵌入”, 载于 WWW 2015, 2015, pp. 1067–1077.
- [20] C. 杨, z. 刘, d. 赵, m. 孙, 和 e. Y. 常, “富文本信息的网络表征学习”. in IJCAI 2015, 2015, pp. 2111–2117.
- [21] T. 川本, m. 椿, 还有 t. 小渊惠三, “图分割中的图神经网络的平均场理论”, 神经生理学 2018, 2018, 页. 4366–4376.
- [22] F. 蒙蒂, d. Boscaini, j. Masci, e. Rodola. 斯沃博达, 和 m. 米 (meter 的缩写)). 布朗斯坦, “使用混合模型中枢神经系统对图形和流形的几何深度学习”, CVPR, 2017 年, 页. 5425–5434, 2017.
- [23] J. 阿特伍德和 d. 陶斯利, “扩散-卷积神经网络”, 在 NIPS 2016, 2016, 页. 1993–2001.
- [24] J. Masci, d. Boscaini, m. 布朗斯坦和 p. 范德赫因斯特, “黎曼流形上的测地线卷积神经网络”, ICCV 研讨会, 2015 年, 2015 页. 37–45.
- [25] D. Boscaini, j. Masci, e. 罗多拉和 m. 布朗斯坦, “学习各向异性卷积神经网络的形状对应”, 在 NIPS 2016, 2016, 页. 3189–3197.
- [26] 米 (meter 的缩写)). 米 (meter 的缩写)). j. 布朗斯坦. 纽约州布鲁纳市. 乐村, a. Szlam 和 p. Vandergheynst, “几何深度学习: 超越欧几里得数据”, IEEE SPM 2017, vol. 34 不. 4 页. 18–42, 2017.
- [27] J. Gilmer, s. 南. Schoenholz, p. F. 莱利, o. Vinyals, 和 g. E. 达尔, “量子化学的神经信息传递”, arXiv 预印本 arXiv:1704.01212, 2017.
- [28] X. 王, r. Girshick, a. 古普塔, 还有 k. 何, “非局部神经网络”, arXiv 预印本 arXiv:1711.07971, 卷. 10, 2017.
- [29] J. B. 李, r. A. 罗西, s. 金, n. K. 艾哈迈德, 还有 e. Koh, “图表中的注意力模型: 调查”, arXiv 预印本 arXiv:1807.07984, 2018.
- [30] 页 (page 的缩写). W. 巴塔格利亚. B. 哈姆里克, v. Bapst, a. 桑切斯-冈萨雷斯, 动词 (verb 的缩写). Zambaldi. 马林诺夫斯基. Tacchetti, d. Raposo, a. 桑托罗. 福克纳等人, “关系归纳偏差、深度学习和图形网络”, arXiv 预印本 arXiv:1806.01261, 2018.
- [31] Z. 张, p. 崔, 和 w. 朱, “图形的深度学习: 综述”, arXiv 预印本 arXiv:1812.04202, 2018.
- [32] Z. 吴, s. 潘, f. 陈, g. 长, c. 张和 p. 南. 于, “图形神经网络综合研究”, arXiv 预印本 arXiv:1901.00596, 2019.
- [33] F. 斯卡塞利, m. 哥里, a. C. Tsoi, m. Hagenbuchner 和 g. 蒙法迪尼, “图形神经网络模型”, TNN 电气和电子工程师协会, 2009 年, 卷. 20, 不. 1 页. 61–80, 2009.
- [34] 米 (meter 的缩写)). A. 哈姆西和 w. A. 度量空间和不动点理论导论. 约翰·威利父子公司, 2011 年, 第一卷. 53.
- [35] 米 (meter 的缩写)). Kampffmeyer, y. 陈, x. 梁, h. 王, y. 张和 e. 页 (page 的缩写). 邢, “零镜头学习的知识图传播再思考”, arXiv 预印本 arXiv:1805.11724, 2018.
- [36] Y. 张, y. 熊, x. 孔. 李, j. Mi, 还有 y. 朱, “异构信息网络中的深度集体分类”, 载于 WWW 2018, 2018, 页. 399–408.
- [37] X. 王, h. 冀, c. 石, b. 王, y. 叶, p. 崔, 和 p. 南. 于, 《异质图形注意力网络》, WWW 2019, 2019.
- [38] D. 贝克, g. 哈夫里, 还有 t. 科恩, “使用门控图形神经网络的图形到序列学习”, 载于《美国公民自由法案》2018 年, 2018 年, 页. 273–283.
- [39] 米 (meter 的缩写)). Schlichtkrull, t. 名词 (noun 的缩写). Kipf, p. Bloem, r. 范登伯格, 我. 蒂托夫和 m. 韦林, “用图形卷积网络建模关系数据”, ESWC, 2018 年. Springer, 2018, pp. 593–607.
- [40] Y. 李, r. 于, c. Shahabi, 和 y. 刘, “扩散卷积递归神经网络: 数据驱动的交通预测”, arXiv 预印本 arXiv:1707.01926, 2017.
- [41] B. 于, h. 阴, 和 z. 朱, “时空图卷积网络: 交通预测的深度学习框架”, arXiv 预印本 arXiv:1709.04875, 2017.
- [42] A. Jain, a. R. Zamir, s. 萨瓦雷斯和 a. 萨克斯纳, “结构神经网络: 时空图的深度学习”, CVPR, 2016 年, 2016 页. 5308–5317.
- [43] 南. 颜, y. 熊, 还有马超. 林, “基于骨架的动作识别的时空图卷积网络”, 第三十二届人工智能会议, 2018.
- [44] 名词 (noun 的缩写). 彭, h. 潘, c. 怪癖, k. Toutanova, 和 w. -t. Yih, “用图 lstms 提取跨句 n 元关系”, arXiv 预印本 arXiv:1708.03743, 2017.
- [45] J. 西布鲁纳. Zaremba, a. Szlam, 和 y. 勒昆, “图上的谱网络和局部连通网络”, ICLR 2014, 2014.
- [46] 米 (meter 的缩写)). Henaff, j. 布鲁纳和 y. 图结构数据上的深度卷积网络. arXiv: Learning, 2015.
- [47] D. K. 哈蒙德, p. Vandergheynst 和 r. Gribonval, “通过谱图理论的图的小波”, 应用和计算调和分析, 卷. 30 不. 2 页. 129–150, 2011.
- [48] 米 (meter 的缩写)). Defferrard, x. 布列松和 p. Vandergheynst, “具有快速局部化谱滤波的图上的卷积神经网络”, NIPS 2016, 页. 3844–3852, 2016.
- [49] R. 李, s. 王, f. 朱, 还有 j. 黄, “自适应图卷积神经网络”, 2018, 2018.
- [50] Y. C. Ng, n. 科伦坡和 r. 席尔瓦, “带图高斯过程的贝叶斯半监督学习”, 载于《神经科学》2018 年版, 2018 年版, 页. 1690–1701.
- [51] D. K. Duvenaud, d. Maclaurin, j. Aguileraiparraguirre, r. Gomezbombarelli, t. D. Hirzel, a. Aspurguzik, 和 r. 页 (page 的缩写). 亚当斯, “用于学习分子指纹的图上的卷积网络”, NIPS 2015, 页. 2224–2232, 2015.
- [52] C. 庄与 q. 马, “基于图的双半监督分类的双图卷积网络”, WWW 2018, 2018.
- [53] 米 (meter 的缩写)). Niepert. 艾哈迈德和 k. 库茨科夫, “学习图的卷积神经网络”, ICML 2016, 2016, 页. 2014–2023.
- [54] H. 高, z. 王, 和 s. 季, “大规模可学习图卷积网络”, 载于《SIGKDD 学报》. ACM, 2018, pp. 1416–1424.
- [55] K. 他, x. 张, s. 任, 和 j. 孙, “深剩余网络中的身份映射”, 2016. Springer, 2016, pp. 630–645.
- [56] J. 昌, j. 顾, l. 王, g. 孟, s. 项, 和丙. 潘, “结构感知卷积神经网络”, 载于《神经生理学》2018 年版, 2018 年版, 页. 11–20.
- [57] K. Cho, b. Van Merriënboer. Gulcehre, d. Bahdanau, f. 布加雷斯, h. 施文克, 还有 y. Bengio, “使用 rnn 编码器-解码器学习短语表示, 用于统计机器翻译”, EMNLP 2014, 页. 1724–1734, 2014.
- [58] 南. Hochreiter 和 j. 斯密德胡伯, “长短期记忆”, 神经计算, 第一卷. 9, 没有. 8 页. 1735–1780, 1997.
- [59] Y. 李, d. 塔罗, m. Brockschmidt 和 r. 南. Zemel, “门控图序列神经网络”, arXiv: Learning, 2016.
- [60] K. 南. 泰, r. Socher 和 c. D. 曼宁, “从树形结构的长短期记忆网络中改进的语义表征”, IJNLP 2015, 页. 1556–1566, 2015.
- [61] 动词 (verb 的缩写). Zayats 和 m. 奥斯坦多夫, “使用图结构 lstm 在 reddit 上进行会话建模”, TACL, 2018 年, 第一卷. 6 页. 121–132, 2018.
- [62] Y. 张问. 刘, 还有 l. 宋, “句子-状态 lstm 用于文本表示”, 美国公民自由联盟 2018 年, 第 1 卷. 1 页. 317–327, 2018.

- [63] X. 梁, x. 沈, j. 冯, l. 林, 还有 s. 颜, “基于图 lstm 的语义对象解析”, 2016, 页. 125–143, 2016.
- [64] D. Bahdanau, k. Cho 和 y. 本吉奥, “联合学习对齐和翻译的神经机器翻译”, ICLR 2015, 2015.
- [65] J. Gehring, m. 奥莉, d. 格兰吉尔和 y. 名词 (noun 的缩写)。多芬, “一个用于神经机器翻译的卷积编码器模型”, 美国公民自由联盟 2017 年, 卷. 1 页. 123–135, 2017.
- [66] A. Vaswani. Shazeer. Parmar, l. 琼斯, j. Uszkoreit, a. 名词 (noun 的缩写)。戈麦斯, 还有 l. 凯泽, “关注是你所需要的全部”, NIPS 2017, 页. 5998–6008, 2017.
- [67] J. 程, l. 董, 和 m. Lapata, “机器阅读的长期短期记忆网络”, EMNLP 2016, 页. 551–561, 2016.
- [68] 页 (page 的缩写)。Velickovic, g. 库库鲁尔, a. 卡萨诺瓦, a. 罗梅罗, p. 莉雅和 y. 本吉奥, “图形注意力网络”, ICLR 2018, 2018.
- [69] J. 张, x. 史, j. 谢, h. 马, 我. 王者, 还有 d. -Y. 杨, “Gaan: 大型时空图学习的门控注意力网络”, arXiv 预印本 arXiv:1803.07294, 2018.
- [70] K. 他, x. 张, s. 任, 和 j. 孙, “用于图像识别的深度残差学习”, 2016, 页. 770–778, 2016.
- [71] A. 拉希米, t. 科恩和 t. 鲍德温, “通过图形卷积网络的半监督用户地理定位”, 美国公民自由联盟 2018 年, 卷. 1 页. 2009–2019, 2018.
- [72] J. G. 零零星星, r. K. Srivastava, j. Koutnik, 和 j. Schmidhuber, “经常性的公路网.”. ICML, 2016 年, 页. 4189–4198, 2016.
- [73] T. Pham, t. Tran, d. Phung, 和 s. 文卡特什, “用于集体分类的柱网”, AAAI 2017, 2017.
- [74] K. 徐, c. 李, y. 田, t. 索诺贝, k. Kawarabayashi 和 s. 杰格尔卡, “跳跃知识网络图的表征学习”, ICML, 2018 年, 页. 5449–5458, 2018.
- [75] 米 (meter 的缩写)). 西蒙诺夫斯基和 n. Komodakis, “图的卷积神经网络中的动态边缘条件滤波器”, 在 IEEE 计算机视觉和模式识别会议记录中, 2017 年, 页. 3693–3702.
- [76] Z. 应, j. 你, c. 莫里斯, x. 任, w. 汉密尔顿和 j. Leskovec, “具有可区分池的分层图形表示学习”, 载于《神经病学》2018 年版, 2018 年版, 页. 4805–4815.
- [77] R. 应, r. 他, k. 陈, p. 西 Eksombatchai. 长度. 汉密尔顿和 j. Leskovec, “网络规模推荐系统的图形卷积神经网络”, SIGKDD 2018, 2018.
- [78] J. 陈, t. 马, 还有 c. 肖, “Fastgcn:通过重要抽样的图卷积网络的快速学习”, arXiv 预印本 arXiv:1801.10247, 2018.
- [79] W. 黄, t. 张, y. 荣, 还有 j. 黄, “面向快速图形表示学习的自适应抽样”, 载于《神经生理学学报》, 2018 年, 第 10-12 页. 4563–4572.
- [80] H. 戴, z. Kozareva, b. 戴, a. 斯摩拉, 还有 l. 宋, “图的迭代算法稳态学习”, 国际机器学习会议, 2018 年, 页. 1114–1122.
- [81] J. 陈, j. 朱, 和 l. 宋, “带方差缩减的图卷积网络的随机训练”. 《ICML 2018》, 2018, 页. 941–949.
- [82] q. 李, z. 韩, 还有 x. -M. 吴, “用于半监督学习的图卷积网络的更深入见解”, arXiv 预印本 arXiv:1801.07606, 2018.
- [83] T. 名词 (noun 的缩写)。Kipf 和 m. 韦林, “可变图形自动编码器”, arXiv 预印本 arXiv:1611.07308, 2016.
- [84] R. 范登贝尔赫, t. 名词 (noun 的缩写)。Kipf, 和 m. 韦林, “图形卷积矩阵完成”, arXiv 预印本 arXiv:1706.02263, 2017.
- [85] 南. Pan, r. 胡, g. 长, j. 姜, l. 姚, 还有 c. 张, “用于图形嵌入的对抗性正则化图形自动编码器”, arXiv 预印本 arXiv:1802.04407, 2018.
- [86] W. 于, c. 郑, w. 程, c. C. Aggarwal. 宋, b. 宗, h. 陈, 和 w. 王, “学习深度网络表示与对抗正则化自动编码器”, SIGKDD 2018, 2018.
- [87] 南. 曹, w. 鲁, 又问. 徐, “用于学习图形表示的深度神经网络”, 2016, 2016.
- [88] D. 王, p. 崔, 和 w. 朱, “结构深层网络嵌入”, SIGKDD 2016, 2016.
- [89] K. 屠, p. 崔, x. 王, p. 南. 余, 和 w. 朱, “正则等价的深度递归网络嵌入”, SIGKDD 2018, 2018.
- [90] Y. Hoshen, “虚荣:注意力多智能体预测建模”, NIPS 2017, 2017, 页. 2701–2711.
- [91] 名词 (noun 的缩写)。沃特斯, d. 佐兰, t. 韦伯, p. 巴塔格利亚. 帕斯卡努, 和 a. Tacchetti, “视觉交互网络:从视频中学习物理模拟器”, NIPS 2017, 2017, pp. 4539–4547.
- [92] 米 (meter 的缩写)). B. 昌, t. 乌尔曼, a. 托里芭和 j. B. Tenenbaum, “一种基于合成对象的物理动力学学习方法”, arXiv 预印本 arXiv:1612.00341, 2016.
- [93] 南. Sukhbaatar, r. Fergus 等人, “通过反向传播学习多智能体通信”, 发表于 NIPS 2016, 2016, 第. 2244–2252.
- [94] H. 戴, b. 戴, 和 l. 宋, “结构化数据潜变量模型的判别嵌入”, 2016, 2016, 页. 2702–2711.
- [95] D. Raposo, a. 桑托罗特区. r. 巴雷特. Pascanu, t. Lillicrap 和 p. 巴塔格利亚, “从纠缠的场景表象中发现物体及其关系”, arXiv 预印本 arXiv:1702.05068, 2017.
- [96] A. 桑托罗特区. Raposo, d. G. m. 巴雷特. 马林诺夫斯基. Pascanu, p. 巴塔格利亚和 t. Lillicrap, “用于关系推理的简单神经网络模块”, 发表于 NIPS 2017, 2017, pp. 4967–4976.
- [97] 米 (meter 的缩写)). Zaheer, s. Kottur, s. Ravanbakhsh, b. Poczos, r. R. Salakhutdinov 和 a. J. 斯摩拉, “深集”, NIPS 2017, 2017, 页. 3391–3401.
- [98] C. R. 齐, h. 苏, k. 莫, 还有 l. J. 圭巴斯, “点网:三维分类和分割点集的深度学习”, CVPR, 2017 年, 第一卷. 1, 没有. 2, p. 4, 2017.
- [99] 南. 科尔斯, k. 麦克罗斯基, m. Berndl, v. 潘德和 p. 赖利, “分子图卷积:超越指纹”, 计算机辅助分子设计杂志, 第一卷. 30 不. 8 页. 595–608, 2016.
- [100] K. T. 舒特, f. Arbabzadah, s. Chmiela, k. R. M. 尤勒 和 a. Tkatchenko, “来自深度张量神经网络的量子化学见解”, 《自然通讯》, 第一卷. 8, p. 13890, 2017.
- [101] A. Buades, b. Coll, 和 j. -M. 莫瑞尔, “图像去噪的非局部算法”, CVPR, 2005 年, 第一卷. 2. IEEE, 2005, pp. 60–65.
- [102] C. 托马西和 r. 曼杜奇, “灰度和彩色图像的双边滤波”, 计算机视觉 1998. IEEE, 1998, pp. 839–846.
- [103] T. Kipf, e. Fetaya, k. -C. 王, m. 韦林, 还有 r. Zemel, “交互系统的神经关系推理”, arXiv 预印本 arXiv:1802.04687, 2018.
- [104] J. B. 哈姆里克, k. R. 艾伦, v. Bapst, t. 朱, k. R. 麦基, j. B. Tenenbaum 和 p. W. 巴塔格利亚, “人类和机器的物理构造的关系归纳偏差”, arXiv 预印本 arXiv:1806.01203, 2018.
- [105] T. 王, r. 廖, j. Ba, 和 s. Fidler, “Nervenet:用图形神经网络学习结构化政策”, 2018.
- [106] H. 彭, j. 李, y. 他, y. 刘, m. 包, l. 王, y. 宋, 又问. 杨, “递归正则化深度图的大规模层次文本分类——CNN”, 载于 WWW 2018, 2018, pp. 1063–1072.
- [107] 长度. 姚, c. 毛, 还有 y. 罗, “用于文本分类的图形卷积网络”, arXiv 预印本 arXiv:1809.05679, 2018.
- [108] D. Marcheggiani 和我. 蒂托夫, “用图形卷积网络编码句子进行语义角色标注”, 载于《电子自然语言处理学报》, 2017 年, 第 10-12 页. 1506–1515.
- [109] J. 巴斯丁斯, 我. 蒂托夫, w. 阿齐兹, d. Marcheggiani, 和 k. 西曼, “用于句法感知神经机器翻译的图形卷积编码器”, EMNLP 2017, 页. 1957–1967, 2017.
- [110] D. Marcheggiani, j. 巴斯丁斯和我. 蒂托夫, “利用图卷积网络的神经机器翻译中的语义”, arXiv 预印本 arXiv:1804.08313, 2018.
- [111] 米 (meter 的缩写)). Miwa 和 m. Bansal, “在序列和树结构上使用 lstrms 的端到端关系提取”, arXiv 预印本 arXiv:1601.00770, 2016.
- [112] 长度. 宋, y. 张, z. 王, 和丁. 吉尔迪, “使用图状态 lstm 的 N 元关系提取”, arXiv 预印本 arXiv:1808.09101, 2018.
- [113] Y. 张, p. 齐, 和 c. D. 曼宁, “修剪的依赖树上的图形卷积改善了关系提取”, arXiv 预印本 arXiv:1809.10185, 2018.
- [114] T. H. 阮和 r. Grishman, “用于事件检测的带有参数感知池的图形卷积网络”, 2018.
- [115] X. 刘, z. 罗, 还有 h. 黄, “通过基于注意力的图形信息聚合联合提取多个事件”, arXiv 预印本 arXiv:1809.09078, 2018.
- [116] 长度. 宋, y. 张, z. 王, 和丁. 吉尔迪, “amr 到文本生成的图形到序列模型”, arXiv 预印本 arXiv:1805.02473, 2018.

- [117] 长度。宋, z. 王, m. 于, y. 张, r. 弗洛里安和 d. 吉尔迪, “用图形神经网络探索多跳阅读理解的图形结构段落表示”, arXiv 预印本 arXiv:1809.02040, 2018。
- [118] R. B. Palm, u. 帕凯和欧. 温瑟, “循环关系网络”, 神经科 2018, 2018。
- [119] Z. 王, t. 陈, j. 任, w. 于, h. 程, 还有 l. 林, “用知识图进行深度推理以理解社会关系”, arXiv 预印本 arXiv:1807.00504, 2018。
- [120] 动词 (verb 的缩写)。加西亚和 j. 布鲁纳, “用图形神经网络进行少量学习”, arXiv 预印本 arXiv:1711.04043, 2017。
- [121] X. 王, y. 叶, 和 a. 古普塔, “通过语义嵌入和知识图的零镜头识别”, CVPR 2018, 2018, 页. 6857–6866。
- [122] C. 李, w. 方, c. 是的, 还有你. F. 王, “基于结构化知识图的多标签零射学习”, 《学报》, 2018 年, 页. 1576–1585。
- [123] K. 圣马力诺, r. Salakhutdinov 和 a. 古普塔, “你知道的越多:使用知识图进行图像分类”, 载于《CVPR 论文集》, 2017 年, 第 10-11 页. 20–28。
- [124] D. 特妮, l. 刘, 和 a. 动词 (verb 的缩写)。丹·亨格尔, “用于视觉问题回答的图形结构表示”, 载于《CVPR 论文集》, 2017 年, 页. 3233–3241。
- [125] 米 (meter 的缩写)。Narasimhan, s. Lazebnik, 和 a. G. Schwing, “开箱即用:用图形卷积网络进行推理, 用于事实性视觉问题回答”, 载于《神经生理学学报》, 2018 年, 第 100-100 页. 2654–2665。
- [126] H. 胡, j. 顾, z. 张, j. 戴, 和 y. 魏, “用于对象检测的关系网络”, 载于 2018 年第一卷. 2, 没有. 3, 2018。
- [127] J. 顾, h. 胡, l. 王, y. 魏, 和 j. 戴, “用于对象检测的学习区域特征”, arXiv 预印本 arXiv:1803.07066, 2018。
- [128] 南. 齐, w. 王, b. 贾, j. 沈, 还有 s. -C. 朱, “通过图形解析神经网络学习人机交互”, arXiv 预印本 arXiv:1808.07962, 2018。
- [129] X. 陈, l. -J. 李 l. 飞和 a. Gupta, “卷积之外的迭代视觉推理”, arXiv 预印本 arXiv:1803.11189, 2018。
- [130] X. 梁, l. 林, x. 沈, j. 冯, s. 颜, 和 e. 页 (page 的缩写)。邢, “可解释结构-进化 lstm”, 2017, 2017, 页. 2175–2184。
- [131] 长度。兰德鲁和 m. Simonovsky, “用超图进行大规模点云语义分割”, arXiv 预印本 arXiv:1711.09869, 2017。
- [132] Y. 王, y. 孙, z. 刘, s. E. Sarma, m. 米 (meter 的缩写)。布朗斯坦和 j. 米 (meter 的缩写)。所罗门, “用于在点云上学习的动态图表 cnn”, arXiv 预印本 arXiv:1801.07829, 2018。
- [133] X. 齐, r. 廖, j. 贾, s. Fidler 和 r. Urtasun, “用于 rgb-d 语义分割的 3d 图形神经网络”, CVPR, 2017 年, 2017, 页. 5199–5208。
- [134] 米 (meter 的缩写)。Zitnik, m. 阿格沃尔和 j. Leskovec, “用图形卷积网络模拟多相副作用”, arXiv 预印本 arXiv:1802.00543, 2018。
- [135] 南. Rhee, s. Seo, 还有 s. Kim, “乳腺癌亚型分类的关系网络和局部图卷积滤波的混合方法”, arXiv 预印本 arXiv:1711.05859, 2017。
- [136] Z. 王, 问. 吕, x. 兰, 还有 y. 张, “基于图卷积网络的跨语言知识图对齐”, 电子自然语言处理 2018, 2018, 页. 349–357。
- [137] A. 诺瓦克, s. Villar. 南. 班德拉和 j. 布鲁纳, “用图形神经网络学习二次分配的修订说明”, 在 DSW 电气和电子工程师协会 2018 年.IEEE, 2018, pp. 1–5。
- [138] Z. 李问. 陈, 与 v. 科尔顿, “带图卷积网络和导向树搜索的组合优化”, 神经科学 2018, 2018, 页. 537–546。
- [139] W. 库尔和 m. 韦林, “注意力解决你的 tsp”, arXiv 预印本 arXiv:1803.08475, 2018。
- [140] O. Shchur, d. Zugner, a. Bojchevski 和 s. 冈曼, “内甘:通过随机漫步生成图形”, 载于《ICML 论文集》, 2018 年, 页. 609–618。
- [141] T. 马, j. 陈, 还有 c. 肖, “通过正则化变分自动编码器的语义有效图的约束生成”, 神经科学 2018, 2018, 页. 7113–7124。
- [142] J. 你, b. 刘, r. 应, v. 潘德和 j. Leskovec, “用于目标导向分子图生成的图形卷积策略网络”, arXiv 预印本 arXiv:1806.02473, 2018。
- [143] 名词 (noun 的缩写)。曹德和 t. Kipf, “Molgan:小分子图的隐式生成模型”, arXiv 预印本 arXiv:1805.11973, 2018。
- [144] Z. 崔, k. 亨利克森, r. 柯, 还有 y. 王, “交通图卷积递归神经网络:用于网络规模交通学习和预测的深度学习框架”, 2018。
- [145] 米 (meter 的缩写)。Allamanis, m. Brockschmidt 和 m. Khademi, “学习用图形表示程序”, arXiv 预印本 arXiv:1711.00740, 2017。
- [146] O. Russakovsky, j. 邓, h. 苏, j. 克劳斯, s. Satheesh, s. 马, z. 黄, a. Karpathy, a. Khosla, m. 伯恩斯坦等人, “Imagenet 大规模视觉识别挑战”, IJCV, 2015 年, 第一卷. 115, 没有. 3 页. 211–252, 2015。
- [147] W. Norcliffebrown, s. Vafeias, 和 s. Parisot, “学习可解释的视觉问题回答的条件图形结构”, 载于《神经生理学学报》, 2018 年, 第 10-12 页. 8334–8343。
- [148] J. 你, r. 应, x. 任, w. 汉密尔顿和 j. 莱斯科维克, “图形神经网络:用深度自回归模型生成逼真的图形”, ICML, 2018 年, 2018 页. 5694–5703。
- [149] Y. 李 o. Vinyals, c. 戴尔, r. Pascanu, 和 p. 巴塔格利亚, “学习图形的深层生成模型”, arXiv 预印本 arXiv:1803.03324, 2018。
- [150] 爱达荷 (Idaho 的缩写)。贝洛, h. Pham, q. 动词 (verb 的缩写)。乐, m. 诺鲁齐和 s. 本吉奥, “具有强化学习的神经组合优化”, 2017。
- [151] O. Vinyals, m. Fortunato, 和 n. Jaitly, “指针网络”, NIPS 2015, 2015, pp. 2692–2700。
- [152] R. 南. 莎顿和 a. G. 巴尔托, 《强化学习:导论》。麻省理工学院出版社, 2018。
- [153] E. 哈利勒, h. 戴, y. 张, b. Dilkina, 和 l. 宋, “图的组合优化算法学习”, NIPS 2017, 2017, 页. 6348–6358。

6 修订历史

- 版本 2(2019 年 1 月 2 日)。添加 NuerIPS 2018 论文。
- 版本 3(2019 年 3 月 7 日)。添加两位合著者, 我们感谢他们的善意建议和贡献。
- 版本 4(2019 年 7 月 10 日)。根据最近的论文添加更多的模型和应用, 并更新图表和参考文献。