《HOW POWERFUL ARE GRAPH NEURAL NETWORKS？》

《图神经网络有多强大?》

摘要：

1. **本文的背景：**图神经网络（GNN）是图表示学习的有效框架，而大部分 GNN 当时主流的做法是通过递归迭代聚合一阶邻域表征来更新节点表示的，例如 GCN、GraphSAGE，但这些方法大多是经验主义的，模型的效果并没有理论上的支持。因此，本文基于 Weisfeiler-Lehman(WL) test 的角度，从理论分析了 GNN 的性能水平。
2. **本文的贡献：**证明了 GNN 在性能上最多只能和 WL 测试一样有效，即 WL 测试是 GNN 性能的上限；并分析了 GCN 和 GraphSAGE 等主流 GNNs 在捕获图结构上的不足和特性；此外，根据所提出的理论，还设计了一种图同构网络框架(GIN)，并通过实验证明了它的判别/表达能力可以达到和 WL 测试一样的水平。
3. **主要创新点：**从（数学）理论上分析了 GNN 的性能上限，并给出了一种基于理论的新型图神经网络框架（GIN）。
4. **实验结果：**GIN 模型在图分类问题上取得了和理论上与 WL test 一样优异的效果。同样，在不同的数据集上，与其他传统的 GNN 框架相比，GIN 也取得了最优的测试结果。

Weisfeiler-Lehman 图同构测试：

Weisfeiler-Lehman(WL)算法是一种用于分析和判断两个图（G_1、G_2）是否是同构图的算法。（**同构图：**如果G_1和G_2的顶点和边数量相同，且边的连接性也相同，则可以称这两个图是同构的。）【WL 算法细节请参考：https://blog.csdn.net/qq_45312141/article/details/106783670】

而图的同构测试问题就是判断两个图的拓扑结构是否等价。这是一个非常有挑战的问题，但到目前为止，仍没有一个能在线性时间内就可以解决该问题的算法。不过，WL 图同构测试却是当前最有效的、计算效率高的一种检验方法，它可以区分一大类图。并且，它的实现也类似于 GNN 中的邻居聚合方式。

简而言之，WL 测试会反复进行如下两个步骤：

（1） 聚合节点及其邻域节点的标签；

（2） 将聚合后的标签重新散列为唯一的新标签。

而如果在某次迭代中，两个图之间的节点标签不同，那么该算法就会判定这两个图是非对称的。

图神经网络（GNN）：

GNN 的目标是以图结构数据和节点特征作为输入，以学习到节点（或图）的表示，用于分类问题等一系列任务。而当时主流的 GNN 都是采用**邻域聚合策略**，即通过聚合节点邻域的表示来迭代更新节点的表示。

而基于邻域聚合策略的 GNN 又可以拆分为以下三个模块：

- ✧ Aggregate：聚合一阶邻域特征。
- ✧ Combine：将邻居聚合的特征与当前节点特征合并，以更新当前节点特征。
- ✧ Readout（可选，针对图分类）：如果是对 graph 分类，则需要将 graph 中所有节点特征转变成 graph 特征。

在经过k次迭代聚合后，节点的表示就能够捕捉到其k跳网络邻域内的结构信息。于是，在形式上，GNN 的第k层可以表示为：

$$a_v^{(k)} = AGGREGATE^{(k)}\left(\left\{h_u^{(k-1)}: u \in \mathcal{N}(v)\right\}\right), h_v^{(k)} = COMBINE^{(k)}\left(h_u^{(k-1)}, a_v^{(k)}\right)$$

其中，$h_v^{(k)}$是节点v在第k次迭代/层时的特征向量，我们初始化$h_v^{(0)} = X_v$；$a_v^{(k)}$是节点v的邻居聚合特征向量；而$\mathcal{N}(v)$是与节点v相邻的节点集。并且，在 GNN 中，对于函数$AGGREGATE^{(k)}(\cdot)$和$COMBINE^{(k)}(\cdot)$ 的选择是至关重要的。例如，在 GraphSAGE 里的 pooling AGGREGATE 函数为：

$$a_v^{(k)} = MAX\left(\left\{ReLU\left(W \cdot h_u^{(k-1)}\right), \forall u \in \mathcal{N}(v)\right\}\right)$$

其中，W是可学习矩阵；MAX表示元素最大池化。而它的$COMBINE$步骤可以是连接（$Concat$）再加线性映射，即：

$$h_v^{(k)} = W \cdot [h_v^{(k-1)}, a_v^{(k)}]$$

而在 GCN 中，如果我们采用的是元素均值池化的方式，那么$AGGREGATE$和$COMBINE$的步骤就可以整合成如下公式：

$$h_v^{(k)} = ReLU\left(W \cdot MEAN\left\{h_u^{(k-1)}, \forall u \in \mathcal{N}(v) \cup \{v\}\right\}\right)$$

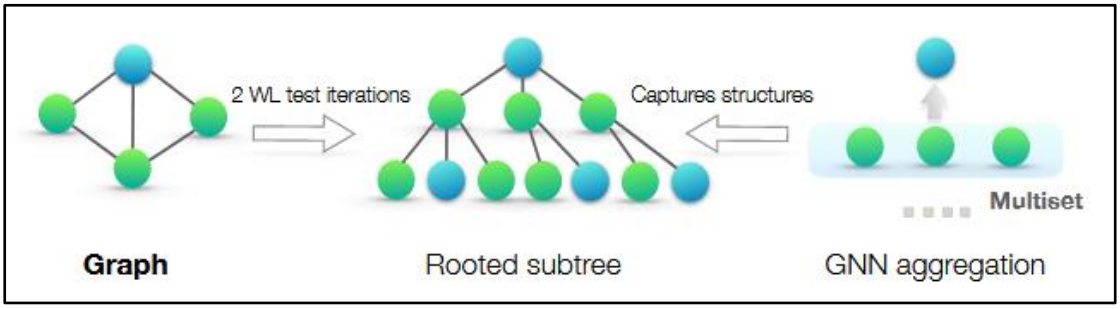
接下来，**对于节点分类问题**，我们可以直接将节点在最后一层的表示$h_v^{(k)}$用于预测；而**对于图分类问题**，我们则还需要将 graph 中所有节点特征转变成一个 graph 特征。而整个图的表示向量h_G为：

$$h_G = READOUT\left(\left\{h_v^{(k)} \middle| v \in G\right\}\right)$$

其中，$READOUT$可以是一个简单的置换不变性函数（permutation invariant function），也可以是一个更复杂的图级 pooling 函数。

Weisfeiler-Lehman 子树核：

在 WL 测试的基础上，Shervashidze 等人（2011 年）提出了 WL 子树核，用于测量两个图之间的相似性。该方法使用了 WL test 在不同迭代中节点标签的计数作为图的特征向量。直观而言，一个节点在 WL 测试的第k次迭代中的标签代表了以该节点为根、高度为k的子树结构，如下图所示。因此，WL 子树核所考虑的图特征向量在本质上是图中不同根子树的计数。



- ✧ 中间图：表示有根的子树结构，WL 测试使用它来区分不同的图；
- ✧ 右图：如果 GNN 的聚合函数捕获了邻居的 full multiset，那么 GNN 可以以递归的方式捕获有根的子树，其功能与 WL 测试一样强大；
- ✧ 左图中蓝色节点在进行 2 次 WL 测试后的标签可以用以蓝色节点为根节点的 2 层子树来表示。

这里的 **multiset（多集）**是本文定义的一个数据结构，一个广义的集合概念，它允许有重复的元素。而论文中的 multiset 就是指节点邻居的特征向量集。

理论框架：

在理想情况下，性能最强大的 GNN 可以通过将不同的图结构映射到嵌入空

间中的不同表示来区分它们。然而，这种将任意两个不同图映射到不同嵌入空间的能力也意味着要解决具有挑战性的图同构问题。**即，我们希望将同构的图映射到相同的表示中，而将非同构的图映射到不同的表示中。**而定理 2 证明了该理论的可能性：

令两个图G_1和G_2是任意两个非同构的图。如果存在一个图神经网络$\mathcal{A}: \mathcal{G} \longrightarrow \mathbb{R}^d$能将图$G_1$和$G_2$映射到不同的 embedding。那么通过 WL 图同构测试也可以确定图G_1和G_2是非同构的。

紧接着，定理 3 表明了上述图神经网络能够存在的条件：

令$\mathcal{A}: \mathcal{G} \longrightarrow \mathbb{R}^d$是一个 GNN。对于两个通过 WL 同构测试判断为非同构的两个图G_1和G_2，在 GNN 层足够多的情况下，如果下述的条件成立，则通过 GNN 可以将这两个图映射到不同的 embedding 上：

（1）$\mathcal{A}$用下面的公式迭代地聚合和更新节点特征向量：

$$h_v^{(k)} = \phi\left(h_v^{(k-1)}, f\left(\left\{h_u^{(k-1)}: u \in \mathcal{N}(v)\right\}\right)\right)$$

其中，函数$f(\cdot)$和$\phi(\cdot)$都是单射函数。

（2）对于图分类问题，$\mathcal{A}$的 graph-level $READOUT$函数也是单射的。

（本文中所有定理的证明请自行参考论文附录）

图同构网络（GIN）：

根据定理 4、5 和推论 6 的分析，本文提出了一个新型的图神经网络架构——Graph Isomorphism Network 图同构网络(GIN)。

假设X是可数的，则存在一个函数$f: X \to \mathbb{R}^n$，使得$h(X) = \sum_{x \in X} f(x)$，对于每个大小有界的多集$X \subset \mathcal{X}$都是唯一的。此外，对于某个函数$\phi$，任何多集函数$g$都可以分解为$g(X) = \phi(\sum_{x \in X} f(x))$。

进一步，可有：

假设X是可数的，则存在一个函数$f: X \to \mathbb{R}^n$，对于包括所有无理数在内的无穷多个选择ϵ，有$h(c, X) = (1+\epsilon) \cdot f(c) + \sum_{x \in X} f(x)$对于每一对$(c, X)$都是唯一的，其中$c \in \mathcal{X}$和$X \subset \mathcal{X}$是大小有界的多集。此外，对于某个函数$\phi$，任何关于这些$(c, X)$对的函数$g$都可以分解为$g(c, X) = \phi\big((1+\epsilon) \cdot f(c) + \sum_{x \in X} f(x)\big)$。

而根据通用近似定理，在图神经网络中。我们可以使用多层感知机（MLP）来建模和学习其中的f和ϕ函数。因此，在 GIN 框架下的节点表示形式为：

$$h_v^{(k)} = MLP_\phi^{(k)}\left(\left(1+\epsilon^{(k)}\right)\cdot MLP_f^{(k)}\left(h_v^{(k-1)}\right) + \sum_{u\in\mathcal{N}(v)} MLP_f^{(k)}\left(h_u^{(k-1)}\right)\right)$$

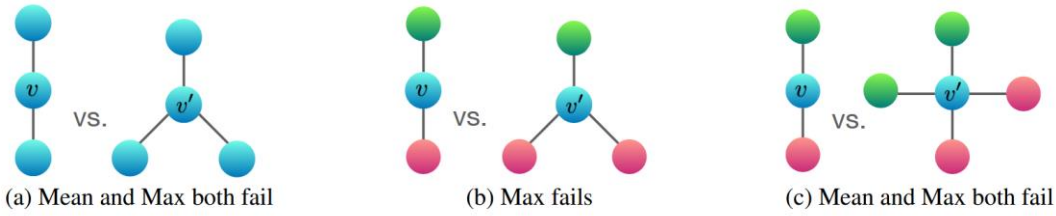
而若在第一次迭代中，如果输入特征是 one-hot 独热编码，则在求和之前我们就不需要做 MLP 了，因为它们的求和本身就是单射的。因此，上述公式可进一步简化为：

$$h_v^{(k)} = MLP^{(k)}\left(\left(1+\epsilon^{(k)}\right)\cdot h_v^{(k-1)} + \sum_{u\in\mathcal{N}(v)} h_u^{(k-1)}\right)$$

于是，通过上述 GIN 框架学习得到的节点 embeddings，我们就可以直接用于类似于节点分类、连接预测这样的任务。而**对于图分类问题**，本文则提出了一个新的$READOUT$函数：

$$h_G = CONCAT\left(sum\left(\left\{h_v^{(k)} \middle| v\in G\right\}\right) \middle| k = 0,1,2\ldots,K\right)$$

基于$MEAN$、MAX聚合的 GNN 分析：



(a) Mean and Max both fail　(b) Max fails　(c) Mean and Max both fail

Figure 3: **Examples of graph structures that mean and max aggregators fail to distinguish.** Between the two graphs, nodes v and v' get the same embedding even though their corresponding graph structures differ. Figure 2 gives reasoning about how different aggregators "compress" different multisets and thus fail to distinguish them.

如图所示，假设节点v和v'为中心节点，且它们都是通过聚合邻居特征来生成 embedding 的。而在图中我们可以看到，在 MEAN、MAX 聚合的设置下，它们并不能区分不同的结构。【如果能捕获不同结构，二者的 embedding 应该不一样】

因此，我们可以得到下述结论：

由于 mean 和 max-pooling 函数不满足单射性，无法区分某些特定结构的图，故性能会比 sum 差一点。如下图所示：

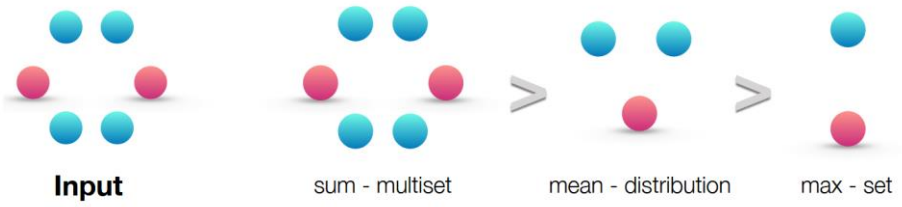


Figure 2: **Ranking by expressive power for sum, mean and max aggregators over a multiset**. Left panel shows the input multiset, *i.e.*, the network neighborhood to be aggregated. The next three panels illustrate the aspects of the multiset a given aggregator is able to capture: sum captures the full multiset, mean captures the proportion/distribution of elements of a given type, and the max aggregator ignores multiplicities (reduces the multiset to a simple set).

总之，三种不同的 aggregate 都有各自的特点：

- ✧ sum：学习全部的标签以及数量，可以学习精确的结构信息；
- ✧ mean：学习标签的比例（比如两个图标签比例相同，但是节点有倍数关系），所以主要偏向学习分布信息；
- ✧ max：学习最大标签，忽略多样，主要偏向学习有代表性的元素信息。

实验及结果：

Datasets：

9 个图分类 benchmarks、4 个生物信息学数据集（MUTAG、PTC、NCI1、PROTEINS）和 5 个社交网络数据集（OLLAB、IMDB-BINARY、IMDB-MULTI、REDDIT-BINARY、REDDIT-MULTI5K）。

实验设置：

在所有实验配置中，都应用了 5 个 GNN 层（包括输入层），而所有 MLP 都为 2 层。此外，还将批量归一化应用在了每个隐藏层上，并使用 Adam 优化器作为梯度下降算法；初始学习率为 0.01，每经过 50 个 epoch 学习率衰减 0.5%。而针对每个数据集设置的超参数是：

1）对于生物信息图，隐藏单元数∈{16，32}；对于社交图，隐藏单元数∈{64}；

2）每个批量大小∈{32，128}；

3）全连接层后的丢弃率∈{0，0.5}；

4）epoch 数，选择 10 个交叉验证精度平均值最佳的单个 epoch。

此外，实验还记录了不同类型的 GNN 的训练准确率，且所有的超参数设置都是固定的：5 个 GNN 层（包括输入层）、64 个隐藏单元、128 个小批量单元和

0.5 的丢弃率。并且为了进行比较，实验还记录了 WL 子树核的训练准确率，将其迭代次数设为 4，这与 5 层 GNN 模型的训练精度相当。

拟合能力——train 集效果

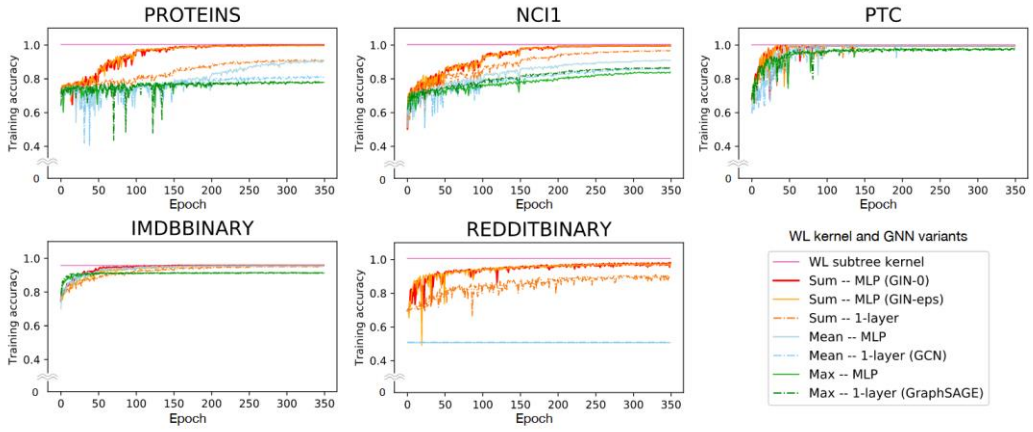


Figure 4: Training set performance of GINs, less powerful GNN variants, and the WL subtree kernel.

如上图所示，可见理论上最强大的 GNN，即$GIN-\epsilon$和$GIN-0$，都能几乎完美地拟合所有训练集。不过，在上述实验结果中，也发现$GIN-\epsilon$的显式学习与$GIN-0$的固定为 0 相比，在拟合训练数据方面没有任何地增益。但相比之下，使用均值/最大池化或单层感知器的 GNN 变体（GCN\GraphSAGE）在许多数据集上都拟合严重不足。此外，上图也证明了 WL 测试的确是 GNN 性能的上限。

泛化能力——test 集效果

	Datasets	IMDB-B	IMDB-M	RDT-B	RDT-M5K	COLLAB	MUTAG	PROTEINS	PTC	NCI1
Datasets	# graphs	1000	1500	2000	5000	5000	188	1113	344	4110
	# classes	2	3	2	5	3	2	2	2	2
	Avg # nodes	19.8	13.0	429.6	508.5	74.5	17.9	39.1	25.5	29.8
Baselines	WL subtree	73.8 ± 3.9	50.9 ± 3.8	81.0 ± 3.1	52.5 ± 2.1	78.9 ± 1.9	90.4 ± 5.7	75.0 ± 3.1	59.9 ± 4.3	**86.0 ± 1.8** *
	DCNN	49.1	33.5	–	–	52.1	67.0	61.3	56.6	62.6
	PATCHYSAN	71.0 ± 2.2	45.2 ± 2.8	86.3 ± 1.6	49.1 ± 0.7	72.6 ± 2.2	**92.6 ± 4.2** *	75.9 ± 2.8	60.0 ± 4.8	78.6 ± 1.9
	DGCNN	70.0	47.8	–	–	73.7	85.8	75.5	58.6	74.4
	AWL	74.5 ± 5.9	51.5 ± 3.6	87.9 ± 2.5	54.7 ± 2.9	73.9 ± 1.9	87.9 ± 9.8	–	–	–
GNN variants	SUM-MLP (**GIN-0**)	**75.1 ± 5.1**	**52.3 ± 2.8**	**92.4 ± 2.5**	**57.5 ± 1.5**	**80.2 ± 1.9**	**89.4 ± 5.6**	**76.2 ± 2.8**	**64.6 ± 7.0**	**82.7 ± 1.7**
	SUM-MLP (**GIN-ϵ**)	**74.3 ± 5.1**	**52.1 ± 3.6**	**92.2 ± 2.3**	**57.0 ± 1.7**	**80.1 ± 1.9**	**89.0 ± 6.0**	**75.9 ± 3.8**	63.7 ± 8.2	**82.7 ± 1.6**
	SUM-1-LAYER	74.1 ± 5.0	**52.2 ± 2.4**	90.0 ± 2.7	55.1 ± 1.6	**80.6 ± 1.9**	**90.0 ± 8.8**	**76.2 ± 2.6**	63.1 ± 5.7	82.0 ± 1.5
	MEAN-MLP	73.7 ± 3.7	**52.3 ± 3.1**	50.0 ± 0.0	20.0 ± 0.0	79.2 ± 2.3	83.5 ± 6.3	75.5 ± 3.4	**66.6 ± 6.9**	80.9 ± 1.8
	MEAN-1-LAYER (GCN)	74.0 ± 3.4	51.9 ± 3.8	50.0 ± 0.0	20.0 ± 0.0	79.0 ± 1.8	85.6 ± 5.8	76.0 ± 3.2	64.2 ± 4.3	80.2 ± 2.0
	MAX-MLP	73.2 ± 5.8	51.1 ± 3.6	–	–	–	84.0 ± 6.1	76.0 ± 3.2	64.6 ± 10.2	77.8 ± 1.3
	MAX-1-LAYER (GraphSAGE)	72.3 ± 5.3	50.9 ± 2.2	–	–	–	85.1 ± 7.6	75.9 ± 3.2	63.9 ± 7.7	77.7 ± 1.5

如上图所示，可见 1）$GIN-0$比$GIN-\epsilon$泛化能力更强：可能是因为模型在结构上更简单的缘故；2）GIN 比 WL test 效果更好：因为 GIN 进一步考虑了结构的相似性，WL test 只是 one-hot 输出，而 GIN 是将 WL test 映射到低维的 embedding 上；3）max 池化在无节点特征的图上（用度来表示特征）基本无效。

结论：

本文主要基于 graph 分类问题，在不同的数据集实验上证明了 sum 比 mean、max 的聚合效果更好，但是它并不能说明在节点分类问题上也是这样的效果，因为应用的场景在节点特征上可能会更加关注邻域特征分布，或者代表性特征。

对本文的感悟：

本文的最大贡献其实并不是 GIN 框架的提出，而是它从理论上证明了 GNN 框架在图分类问题上的上限是 WL test，打破了之前靠经验主义来设计 GNN 网络的不足，同时本文也更有力地证明了图神经网络的强大性，给该领域上的研究者们打下了一针有效的强心剂。同样，这也在告诉我们这些后来者不能忽视数学理论在计算机科学或深度学习方面的作用。总之，一定要打好数学的基础。