GraphSAGE理论与实践

# 前言

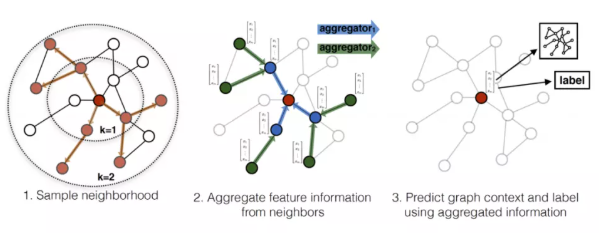
GraphSAGE是2017年提出的一种图神经网络算法，解决了GCN网络的局限性: GCN训练时需要用到整个图的邻接矩阵，依赖于具体的图结构，一般只能用在直推式学习（Transductive Learning）。GraphSAGE使用多层聚合函数，每一层聚合函数会将节点及其邻居的信息聚合在一起得到下一层的特征向量，GraphSAGE采用了节点的邻域信息，不依赖于全局的图结构。

图神经网络的任务一般有Transductive (直推式)和Inductive(归纳式)。Transductive通常指要预测的节点在训练时已经出现过，例如有一个作者关系网络，知道部分作者的类别，用整个网络训练GCN，最后预测未知类别的作者。Inductive指要预测的节点在训练时没有出现，例如用今天的图结构训练，预测明天的图。

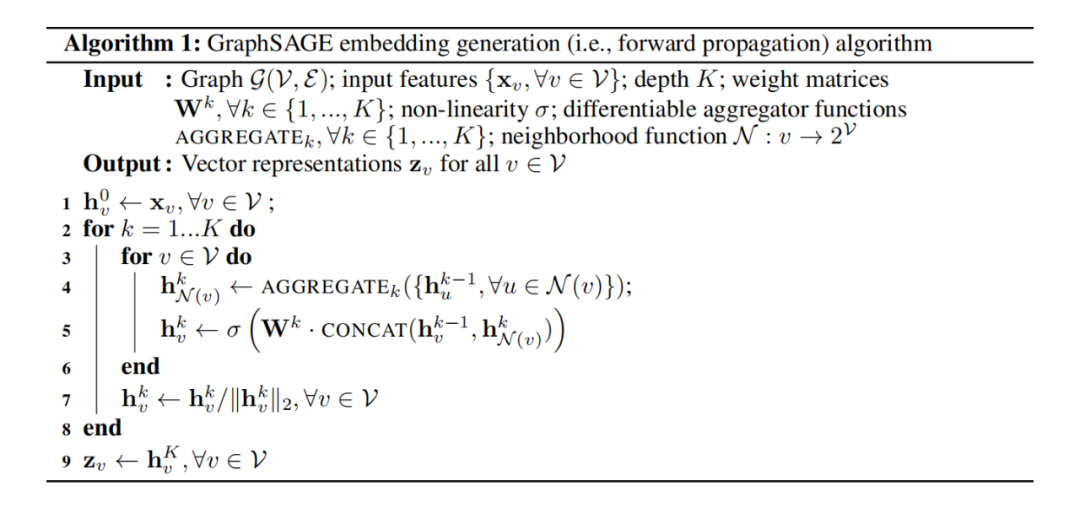
GCN利用了图的整个邻接矩阵和图卷积操作融合相邻节点的信息，因此一般用于 Transductive任务而不能用于处理Inductive任务。因此2017年GraphSAGE算法被提出，用于解决GCN的问题，《Inductive Representation Learning on Large Graphs》。

# GraphSAGE

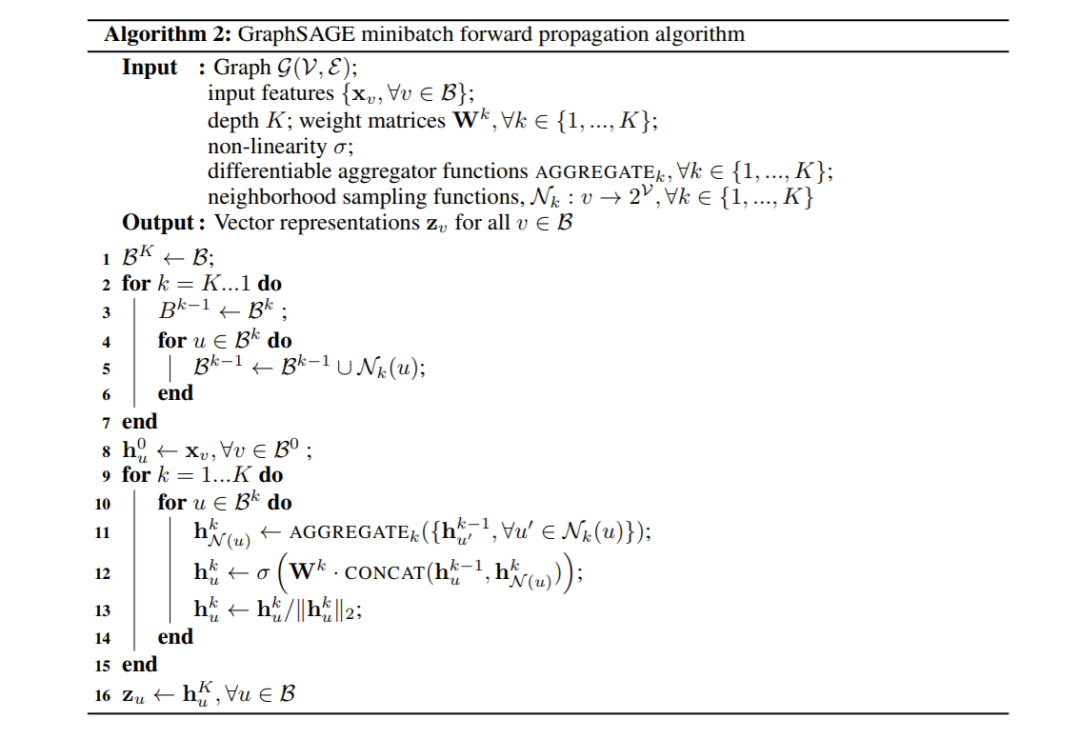
GraphSAGE 包含采样和聚合 (Sample and aggregate)，首先使用节点之间连接信息，对邻居进行采样，然后通过多层聚合函数不断地将相邻节点的信息融合在一起。用融合后的信息预测节点标签。下图展示了 GraphSAGE 的聚合过程，采用了两层聚合层。



上图中的包括两层聚合，对应的聚合函数为aggregator1和aggregator2。通过k层聚合之后，可以得到节点最终的表示向量，GraphSAGE的伪代码如下：



伪代码中的h0表示节点v的初始特征向量，包含K层聚合操作。在第k次聚合生成v节点特征向量时，会采用聚合函数把v节点的邻居信息融合在一起。这一操作也可改成 minibatch的，伪代码如下：



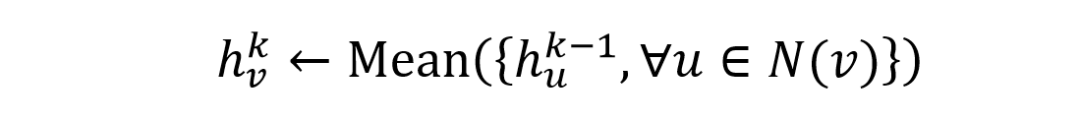
上面的伪代码中，B = BK为要生成向量的节点集合， Bk-1是深度为1的邻域，B0为深度为K的邻域，B0包含的节点最多。Nk(v)表示v节点在第k次聚合时的邻域，节点在每一层的邻域数量都不同，通过采样得到。

## GraphSAGE聚合函数

GraphSAGE提供了四种聚合节点的函数：

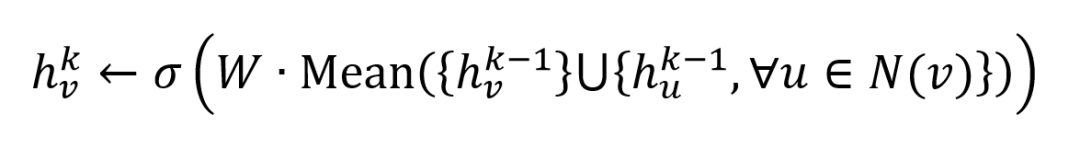
### Mean aggregator

Mean aggregator: 对节点 v 进行聚合时，对节点 v 和邻域的特征向量求均值。



### GCN aggregator

GCN aggregator: 采用了类似GCN卷积的方式进行聚合，公式和Mean aggregator类似：



### LSTM aggregator

LSTM aggregator: 作者任务LSTM有比较好的抽取特征能力，因此也使用了LSTM进行聚合，但是因为节点之间没有明显的顺序关系，因此会打乱之后放入LSTM。

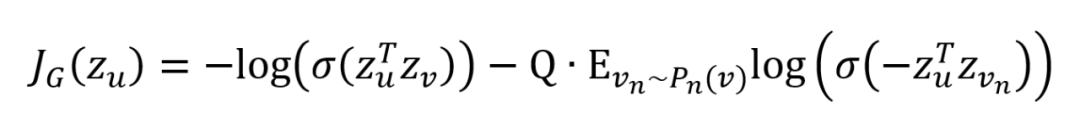
### Pooling aggregator

Pooling aggregator: 先把所有邻居节点的特征向量传入一个全连接层，然后使用 max-pooling聚合。



## GraphSAGE训练

GraphSAGE可以采用无监督训练或者有监督训练。无监督训练采用负采样算法，公式如下：

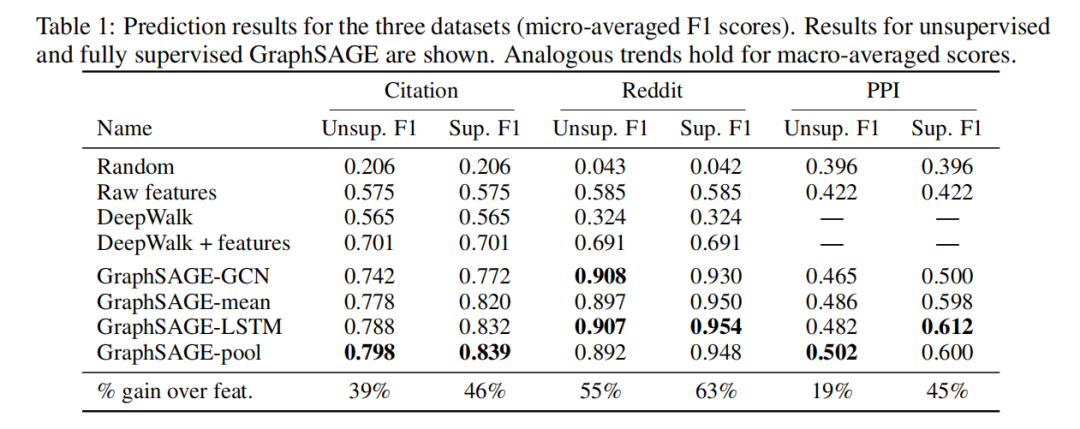


公式中的zu是经过GraphSAGE聚合之后的特征向量，节点v是节点u邻域内的节点，而Q表示负采样次数。

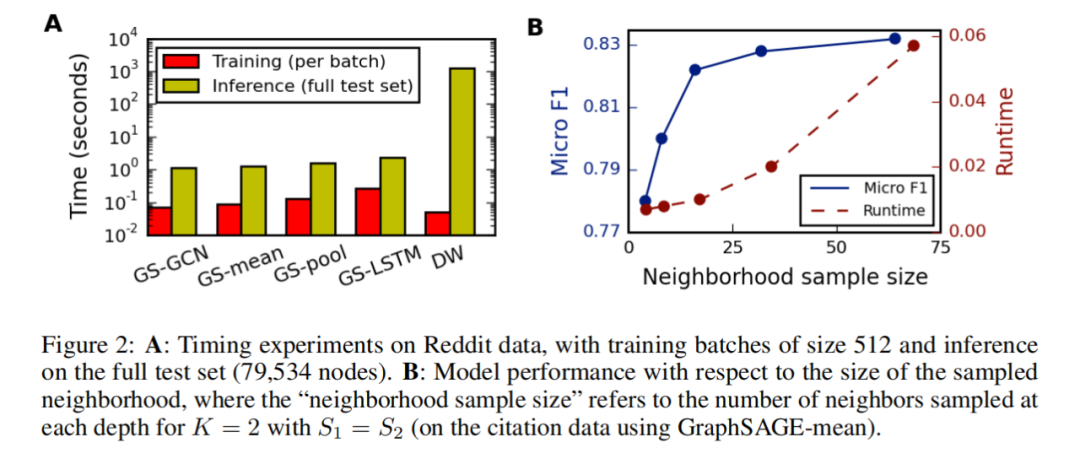
对于有监督训练可以使用任务相关的目标函数，例如节点分类时采用交叉熵损失函数。

# 实验

作者对比了不同算法的性能，也对比了GraphSAGE四种聚合方式的效果，如下表所示。



下图A是训练和测试时间的实验结果，B是采样邻域大小对性能影响的结果。



参考：

<https://mp.weixin.qq.com/s/pn4GYn9CU-sq49qg2OHkIw>