

1.图神经网络（GNN）处理既有节点特征又包含边特征的图数据

1.节点与边特征的整合

(1).消息传递机制：

GNN 通过消息传递机制来更新节点和边的表示。每个节点会从其邻居节点聚合信息，同时也考虑与这些邻居之间的边特征。例如，边函数可以定义如何从边传递消息到相邻节点，这些消息通常依赖于边的特性和连接的节点特征[1][7]。

(2).多维边特征：

近年来，许多 GNN 模型开始支持多维边特征的学习，而不仅仅是简单的二元表示。通过引入可学习的边嵌入，模型能够更全面地捕捉边的信息。例如，CEN-DGCNN 模型通过一个新颖的消息传递框架，能够同时学习节点和多维边特征，从而提高了信息传递的能力[2][3]。

(3).交替更新策略：

在 GNN 设计中，一个重要的决策是选择更新节点嵌入还是边嵌入。可以采用交替更新的方法，即在每一层中先更新节点嵌入，再利用更新后的节点信息来更新边嵌入，或者反之。这种“编织”方式允许在每次更新中同时考虑节点和边的信息，从而增强了模型的表达能力[1][2][4]。

2.特征聚合方法

(1).加权邻接矩阵：对于具有权重的边，可以使用加权邻接矩阵来描述边特征，从而在聚合过程中考虑这些权重。这样，模型能够有效地利用边强度等信息来影响节点表示的更新[3][5]。

(2).注意力机制：在某些 GNN 变体中，例如图注意力网络（GAT），为每一维特征单独学习注意力权重，使得不同类型的边特征可以被动态地聚合。这种方法能够增强模型对不同关系类型的敏感度[3][4]。

(3).辅助图构建：在处理复杂图结构时，可以构建辅助图（如线图），将边视为新的节点，通过现有 GNN 架构进行训练。这种方法允许在同一框架下同时学习节点和边的表示，进一步提升了模型性能[3][6]。

Citations:

[1] <https://less-bug.com/posts/what-is-gnn/>

[2] <https://hub.baai.ac.cn/view/31568>

[3] <https://cloud.tencent.com/developer/article/1902504>

[4] <https://www.cnblogs.com/picassooo/p/15437797.html>

[5] <https://finance.sina.cn/tech/2020-09-22/tech-iihvvpwy8175449.d.html?fromtech=1>

[6] <https://blog.csdn.net/ARPOSPF/article/details/122722340>

[7] <https://developer.volcengine.com/articles/7382257268630224905>

[8] <https://www.cnblogs.com/lemonzhang/p/13168669.html>

2.图卷积网络（GCN）处理即有节点特征与边特征的图数据策略

1.节点特征与边特征的整合

(1).节点特征聚合：

GCN 的基本思想是通过聚合邻居节点的特征来更新每个节点的表示。每一层 GCN 都会计算当前节点及其邻居节点的特征加权平均，从而形成新的节点表示。

2.边特征的利用

在许多应用中，边不仅仅是连接两个节点的简单标识，而是承载了丰富的信息。例如，在社交网络中，边可以表示用户之间的关系类型。为了有效利用这些边特征，GCN 可以采用以下几种策略：

(1).带权邻接矩阵：使用带权邻接矩阵来描述边的特征，这样在聚合过程中可以考虑边的权重[1][4]。

(2).多维边特征处理：对于具有多维特征的边，可以在聚合阶段将这些特征与邻居节点的特征结合。例如，可以为每种边特征单独学习一组注意力权重，并在聚合时将这些输出连接起来[1][2]。

(3).辅助图构建：将边视为新图中的节点，通过构建辅助图（如 line graph）来学习边和节点的表示。这种方法允许在不同层次上同时更新节点和边的嵌入[1][2]。

Citations:

[1] <https://cloud.tencent.com/developer/article/1902504>

[2]<https://ifwind.github.io/2021/07/01/Cluster-GCN%EF%BC%9A%E8%B6%85%E5%A4%A7%E5%9B%BE%E4%B8%8A%E7%9A%84%E8%8A%82%E7%82%B9%E7%89%B9%E5%BE%81%E5%AD%A6%E4%B9%A0/>

[3] <https://www.51cto.com/article/627010.html>

[4] <https://blog.csdn.net/shanwenkang/article/details/106469581>

[5] <https://www.cnblogs.com/SupremeBoy/p/12652338.html>

[6] <https://www.cnblogs.com/lemonzhang/p/13168669.html>

[7] <https://blog.csdn.net/ARPOSPF/article/details/122722340>

[8] <https://tech.sina.cn/2020-09-22/detail-iivhvpwy8175449.d.html>