Understanding and Enhancing Graph Neural Networks From the Perspective of Partial Differential Equations

Guangdong Feng
Hebei University of Technology, Tianjin China300401

Abstract: We understand graph neural networks from the perspective of partial differential equations. Firstly, based on the relationship between the partial differential equation and the propagation equation of graph neural networks, the topology and node features are treated as independent variables of the wave function to better combine the topological structure information of the graph with the node feature information. Secondly, the theoretical framework of the graph neural network model PGNN is established by the variable separation operation of the partial differential equation, which makes some existing models have different degrees of PGNN approximation. Finally, experiments show that the model in this paper achieves good results on commonly used citation datasets.

Keywords: Graph Neural Networks; Partial Differential Equations; Separation of Variables.

1 Introducion

Graph Neural Networks (GNNs) are a framework for graph representation learning that follows a neural message passing mechanism. In the past few years, some classical GNN models such as GCN , GIN , SGC , GAT , JKNet , DropEdge , APPNP , GCNII , GRAND , PDE-GCN , GIND , DGI etc., have been designed from different perspectives for their propagation and achieved good performance. Although PDE-GCN, GIND interpret GNNs from the perspective of partial differential equations , they do not include topological information as an input variable.

In this paper, the topology and node features are treated as independent variables of the wave function to better combine the topological structure information of the graph with the node feature information, and the theoretical framework of the graph neural network model PGNN is established through the variable separation operation of the partial differential equation, which makes some existing models are different degrees of approximation of PGNN. Experiments show that the model in this paper achieves good results on commonly used citation datasets.

2 Basic notation

Note a general undirected graph $G=(\mathcal{V},\mathcal{E},X)$, nodes set $\mathcal{V}:=\{v_1,v_1,\cdots,v_N\}$, edges set $\mathcal{E}:=\mathcal{V}\times\mathcal{V}$, Adjacent Matrix A, Node Degree Matrix D, Laplacian matrix $L=D-A\in\mathbb{R}^{N\times N}$, nodes features $X\in\mathbb{R}^{N\times M}$, weight $W\in\mathbb{R}^{M\times C}$, $W_0\in\mathbb{R}^{N\times C}$.

3 Understanding graph neural networks from the perspective of partial differential equations

The message propagation formula of a graph neural network is obtained by minimizing ASD, with A as the equilibrium parameter, and each convolutional layer of a graph neural network is understood as a time step of a discrete partial differential equation, and the message propagation of a graph neural network is understood as the propagation of messages in a high-dimensional space. In this paper, topology and node characteristics are used as independent variables of the wave function, which grows in time and diffuses in high-dimensional space.

If there is a wave function $\psi(L,X,t)=\psi_1(L,X)T(t)$ on the graph data, the partial differential equation is satisfied:

$$\begin{aligned}&\frac{\partial\psi(L,X,t)}{\partial t}=\nabla^2\psi(L,X,t)\quad\frac{\partial\psi(L,X,t)}{\partial t}=\frac{dT(t)}{dt}\psi_1(L,X)\\&\nabla^2\psi(L,X,t)=\nabla^2\psi_1(L,X)T(t)\quad\frac{d(t)}{dt}\psi_1(L,X)=\nabla^2\psi_1(L,X)T(t)\end{aligned}\tag{1}$$

Multiplying the right side of the wave function $\psi(L,X,t)$ by the W transforms and ReLU activation function the form of a graph convolutional network. It is defined as follows:

$$H=\mathrm{Re}LU[\psi(L,X,t)W]\tag{2}$$

We introduce the separation matrix $\Lambda\in\mathbb{R}^{N\times N}$, dividing both sides of the equation by $\psi(L,X,t)$, to separate the variables:

$$\frac{\frac{dT(t)}{dt}\psi_1(L,X)}{\psi(L,X,t)}=\frac{\nabla^2\psi_1(L,X)T(t)}{\psi(L,X,t)}=-\Lambda^2\tag{3}$$

Breaking up the eq 3, that is :

$$\frac{dT(t)}{dt}=-\Lambda^2T(t)\quad\nabla^2\psi_1(L,X)=-\Lambda^2\psi_1(L,X)\tag{4}$$

Although separating $\psi_1(L,X)$ from $T(t)$, L and X in $\psi_1(L,X)$ is not separated. In the following we expand $\psi_1(L,X)$ in polar coordinates.

If $\theta = \arccos(\sqrt{L}\Lambda X)$ and $r = \sqrt{L}$, then $\psi_1(L,X) = R(r)g(\theta)$, $r\sin(\theta) = \sqrt{L}\sqrt{I - \sqrt{L}\Lambda X X^T \Lambda^T \sqrt{L}}$, $r\cos(\theta) = L\Lambda X$, and so

$$\nabla^2 (R(r)g(\theta)) = -\Lambda^2 (R(r)g(\theta)) \tag{5}$$

Divide both sides of eq 5 by $R(r)g(\theta)$,

$$\frac{R''(r)}{R(r)} + \frac{R'(r)}{rR(r)} + \Lambda^2 + \frac{1}{r^2}\frac{g''(\theta)}{g(\theta)} = 0 \tag{6}$$

We introduce the constant diagonal matrix $m \in \mathbb{R}^{N \times N}$,there is:

$$g(\theta)'' = -m^2 g(\theta),\ R''(r)r^2 + R'(r)r + (\Lambda^2 r^2 - m^2)R(r) = 0 \tag{7}$$

Now, the three variables of wave function $\psi(L,X,t)$ on the graph have been separated. The solutions of $T(t)$ and $g(\theta)$ are: $T(t) = e^{-\Lambda^2 t}$, $g(\theta) = \cos(m\theta) + \sin(m\theta)$, respectively.

Introducing the Bessel function $J_m(\ )$, the analytic solution of $R(r)$ is: $R(r) = J_m(r) = \sum_{l=0}^{\infty}\frac{(-1)^l}{l!\,(l+1)!}\left(\frac{\Lambda^T r}{2}\right)^{2l+m}$. Thus, the wave function $\psi(L,X,t)$on the graph can be written as variables:

$$\psi(L,X,t) = e^{-\Lambda^2 t} J_m(\Lambda r)\left[\cos(m\theta) + \sin(m\theta)\right] \tag{8}$$

By variable substitution $r = \sqrt{L}, \theta = \arccos(\sqrt{L}\Lambda X)$, $\psi(L,X,t)$ can be written in the form with respect to variable L,X,t :

$$\begin{aligned}\psi(L,X,t) = e^{-\Lambda^2 t}\sum_{l=0}^{\infty}\frac{(-1)^l}{l!\,(l+1)!}\left(\frac{\Lambda^T\sqrt{L}}{2}\right)^{2l+m}\left[\cos\left(m\arccos(\sqrt{L}\Lambda X)\right)\right] \\ + e^{-\Lambda^2 t}\sum_{l=0}^{\infty}\frac{(-1)^l}{l!\,(l+1)!}\left(\frac{\Lambda^T\sqrt{L}}{2}\right)^{2l+m}\left[\sin\left(m\arccos(\sqrt{L}\Lambda X)\right)\right]\end{aligned} \tag{9}$$

4 An Optimization Framework PGNN

In this section, we introduce a theoretical framework for the graph neural network model PGNN by using topology and node characteristics as independent variables of the wave function and establishing a separation of variable operations by means of partial differential equations.

Note $\mathcal{B} = \Lambda^{-1}\sqrt{I - \sqrt{L}\Lambda X X^T \Lambda^T \sqrt{L}}$, $L_1 = \frac{\Lambda^T L \Lambda}{4}$, the optimization framework PGNN on topology and attribute combination form can be obtained by $\alpha \in \mathbb{R}$ weighted combination of case $m = 0$ and case $m = 1$. It is defined as follows:

$$H = \mathrm{Re}LU\left[e^{-\Lambda^2 t}\sum_{l=0}^{\infty}\frac{(-L_1)^{l+1}}{l!\,(l+1)!}\left[(\alpha - 1)W_0 - \alpha[X + \mathcal{B}]W\right] + (1-\alpha)e^{-\Lambda^2 t}W_0\right] \tag{10}$$

According to Eq 10, PGNN is a unified framework and GCN, JKNet, GAT, APPNP, GCNII is an approximation of PGNN, where $\Lambda^T L \Lambda$ corresponds to the Laplace matrix of GAT, and *m* corresponds to the order interval of the node neighbors of JKNet.

5 Evaluations

This paper follows the data partitioning of the GCN model and performs node classification experiments on three citation datasets : Cora, Citeseer and Pubmed, and reports the average node classification accuracy and standard deviation on the test set after 50 training runs. The experimental results in Table 1 show that PGNN achieves a high level of performance in the node classification task, validating the correctness of the model.

Table 1 Results of Node Classification Task in Citation Datasets Cora, Citeseer, Pubmed

Dataset	Cora	CiteSeer	Pubmed
Chebyshev	81.2±0.5	69.8±0.5	74.4±0.3
GCN	81.5±0.2	70.3±0.3	79.0±0.4
GAT	83.0±0.7	72.5±0.7	79.0±0.3
JKNet	81.1±0.2	69.8±0.3	78.1±0.1
APPNP	83.3±0.5	71.7±0.4	80.1±0.7
GCNII	85.5±0.5	73.4±0.6	80.2±0.7
PGNN	85.9±0.3	73.9±1.0	80.5±1.0

6 Evaluations and Conclusions

This paper provides a theoretical framework for the graph neural network model PGNN through the separation of variables operation of partial differential equations, which allows some existing models to be PGNN approximations of varying degrees, combining better information on the topology of the graph with information on the characteristics of the nodes.

REFERENCES:

[1] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

Kipf T N, Welling M. Semi-Supervised Classification with Graph Convolutional Networks[C]//International Conference on Learning Representations.

[2] Xu K, Hu W, Leskovec J, et al. How powerful are graph neural networks?[J]. arXiv preprint arXiv:1810.00826, 2018.

[3] Velickovic P, Cucurull G, Casanova A, et al. Graph attention networks[J]. stat, 2017, 1050(20): 10.48550.

[4] Wu F, Souza A, Zhang T, et al. Simplifying graph convolutional networks[C]//International conference on machine learning. PMLR, 2019: 6861-6871.

[5] Xu K, Li C, Tian Y, et al. Representation learning on graphs with jumping knowledge networks[C]//International conference on machine learning. PMLR, 2018: 5453-5462.

[6] Rong Y, Huang W, Xu T, et al. Dropedge: Towards deep graph convolutional networks on node classification[J]. arXiv preprint arXiv:1907.10903, 2019.

[7] Gasteiger J, Bojchevski A, Günnemann S. Predict then Propagate: Graph Neural Networks meet Personalized PageRank[J]. arXiv e-prints, 2018: arXiv: 1810.05997.

[8] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In Proc. ICML, pages 1725–1735, 2020.

[9] Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. Graph random neural networks for semi-supervised learning on graphs. In Proc. NeurIPS, pages 22092–22103, 2020.

[10] Eliasof M, Haber E, Treister E. PDE-GCN: novel architectures for graph neural networks motivated by partial differential equations[J]. Advances in neural information processing systems, 2021, 34: 3836-3849.

[11] Pazy A . Semigroups of Linear Operators and Applications to Partial Differential Equations[M]. Springer-Verlag, 1983.

[12] Yang L, Wang C, Gu J, et al. Why do attributes propagate in graph convolutional neural networks?[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(5): 4590-4598.

[13] Namata G , London B , Getoor L , et al. Query-driven Active Surveying for Collective Classification[C]// Mining and Learning with Graphs. 2012.

[14] Sen P, Namata G, Bilgic M, et al. Collective classification in network data[J]. AI magazine, 2008, 29(3): 93-93.

[15] Chen Q, Wang Y, Wang Y, et al. Optimization-induced graph implicit nonlinear diffusion[C]//International Conference on Machine Learning. PMLR, 2022: 3648-3661.

[16] Velickovic P, Fedus W, Hamilton W L, et al. DEEP GRAPH INFOMAX[J]. stat, 2018, 1050: 21.