A Survey on The Application of Graph Neural Networks for Brain Networks Research

Wei Dai
wdai26@emory.edu
Emory University

ABSTRACT

Many structural data are graphical by nature. For example, the brain MRI images can be represented by graphs, where each node represents a Region of Interest (ROI) in the brain and the edges represent connections between the ROIs. Such brain graphs are useful in identifying differences between the samples and have gained popularity in brain-related disease prediction. Since the introduction of Graph Neural Network, it has shown great potential in helping disease predictions. In this survey, we review the variants of GNNs, with a focus on their applications on brain disease predictions.

KEYWORDS

graph neural networks, healthcare, brain networks, machine learning

1 INTRODUCTION

A graph is a popular form of structured data as it captures multiple objects and their relationships simultaneously. They are widely used for representing complex systems of related entities [9]. For example, using Amazon product co-purchasing network [1], one can model the relationships between any of the two products through nearly three million labels. Brain networks, on the other hand, lie on the smaller side. Study shows human brains under fMRI images can be divided into multiple functional regions, called Region of Interests (ROI) [6]. Depending on the division criteria, the number of areas differs in size. However, most division methods, like Automated Anatomical Labeling (AAL) [17], and Freesurfer-generated cortical/subcortical gray matter regions [3], divide the human brain into less than 100 regions, a number much smaller than most other graph datasets. As a result, each brain graph features a small number of nodes, making it easy for modeling and predictions.

Recently, Graph Neural Networks is getting more and more attention due to its predicting power [20]. Graph Convolutional networks first emerged as a promising way to utilize graphical data and extract useful information [11].

As GNN shows excellent potential in brain network research, by far, there has been no known survey about GNN that puts particular focus on the area of brain networks. Several research surveys have been written on the area of brain networks [7, 14]. Similarly, there are comprehensive reviews for graph neural networks [19, 25]. However, no known review has related the two popular topics. Our paper will be the first to offer a glimpse into this particular topic.

Brain networks have several traits worth designers of graph neural networks to give special adaptations. For example, in a particular dataset, the node count for a sample is usually fixed. Plus, each node corresponds to a specific Region of Interest, which gives it special meanings. Furthermore, the connections between nodes (ROIs) are weighted, which offers another dimensionality of data that most graph datasets don't have. Last, as we mentioned above, the graph size for each sample is small, allowing more flexibility in GNN design as compared to large datasets like Amazon co-purchase dataset. These traits, if properly used, can be of great help in model design. Therefore, it is necessary to survey this topic.

Our problem is formulated as follows: Suppose we have a weighted brain network $G = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V} = \{v_i\}_{i=1}^{n}$ is the node (ROI) set of size n, $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ is the edge set, and $W \in \mathbb{R}^{n \times n}$ is the weighted adjacency matrix describing connection strengths between ROIs, we wish to find a disease prediction y through a GNN model M [5].

Applying GNN on brain networks is a relatively nascent topic, so there have been attempts in numerous different directions to optimize the GNN model's performance. This paper discusses selected publicly available brain datasets and their related research. We also review current variants of graph neural networks, exploring their performance on the brain networks. We cover popular GNN variants, including Graph Convolutional Network [11], Graph Attention Networks [18], DiffPool [21] and Principal Neighbourhood Aggregation [4]. We also cover brain-optimized networks such as BrainNetCNN [10] and BrainGNN [13]. We highlight their similarities in terms of propagation, message passing and loss functions. We evaluate their performances based on their calculation complexity, model complexity and prediction accuracy. We wish to find trends that show good potential for further optimizations, as well as identify directions that probably will not work so well.

Conference'22, January 2022, Atlanta, GA, USA
2022.

2 GENERIC GNN MODELS

2.1 Spectral Graph Convolutional Networks

The original graph convolutional network was motivated by CNN. One of successful first attempts was introduced by Bruna et al. in 2013, a year after the covolutional neural network was introduced [2]. In the paper, they presented two construction methods, one based upon a hierarchical clustering of the domain, and another based on the spectrum of the graph Laplacian. The latter became the base of many subsequent works on graph neural networks. In their work, they first revisited the idea of graph Laplacian

$$\mathcal{L} = I - D^{-1/2} W D^{-1/2} \tag{1}$$

Given this formula, they defined the smoothness functional $||\nabla x||_W^2$ at a node i as

$$||\nabla x||_W^2 = \sum_i \sum_j W_{ij} [x(i) - x(j)]^2 \ , \tag{2}$$

This leads to a question: how can we maximize the smoothness vector? It turns out the smoothest vector is always an eigenvector of the laplacian L. If we know the eigenvector matrix V, the transformation can be defined as:

$$x_{k+1,j} = h\left(V \sum_{i=1}^{f_{k-1}} F_{k,i,j} V^T x_{k,i} \right) \ (j = 1 \dots f_k) \ , \tag{3}$$

where $F_{k,i,j}$ is a diagonal matrix and h is an activation function. The construction looks quite simple and straightforward. However, as pointed out by Zhang et al., this construction require $O(n^3)$ time to calculate the eigenvectors, a costly computation for large graphs [23].

To address this computational cost issue, many variants has been proposed. One of the most famous variant is called GCN proposed by Thomas N. Kipf and Max Welling at ICLR 2017 [11]. They simplified the calculation by defining the activation as

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \ . \tag{4}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of a graph with added self connections, I_N is the identity matrix, and $\tilde{D}$, $W^{(l)}$ are weight matrices to be trained. Since the eigenvector computation is avoided, the computational cost is greatly reduced.

In recent years, attention mechanisms are getting increasing popularity. Since attention mechanisms puts unequal focus on each inputs, it performs better under noisy data and accepts inputs of almost any variable sizes. Many attempts to apply attention mechanisms on graph neural networks. One popular model is Graph Attention Network (GAT) designed by Velickovic et al. [18]. In their proposed model, the coefficient is calculated as

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T [\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j] \right) \right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{LeakyReLU}\left(\vec{\mathbf{a}}^T [\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k] \right) \right)} \tag{5}$$

In the formula, $\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j, \mathbf{W}\vec{h}_k$ are weight matrices assigned to each node, indicating "attention," or importance, of each node. This *self-attention* mechanism, similar to Recurrent Neural Networks, allows the model to focus on important nodes. An advantage of this attention model over other similar works is its computational complexity. Since the equation above does not involve eigen-decompositions or other costly matrix operations, the computation cost of a feature is linear with respect to the number of nodes and edges.

2.2 Spatial Graph Convolutional Networks

Since the spectral models are dependent on the laplacian matrix, there are certain limitations. For example, if the eigenfunctions of the two graphs are different, it is hard to generalize the model from one graph to another [23]. As a result, spatial models that are independent on the eigenfunctions are proposed.

Spatial graph neural network models also originated from the the classic convolutional neural networks (CNN) [12]. CNN models primarily deals with grid-like data, such as images [23]. These models are usually unsuitable for graphical data, as the neighborhood nodes and spatial order usually differ from sample to sample. In order to solve this issue, Gao et al. proposed a model called Learnable Graph Convolutional Layer (LGCL) [8]. In their paper, the propagation rule is formulated as

$$X_l = g(X_l, A, k)$$

where the A is the adjacency matrix, g(·) is performs the k-largest node selection to transform generic graphs to data of grid-like structures. After X_l is formulated as a grid structure, the model then performs a regular 1-D CNN,and c(·) denotes a regular 1-D CNN that aggregates neighboring information and outputs a new feature vector for each node:

$$x_{l+1} = c(X_l)$$

Through the k-largest node selection, the data is generalized into a matrix of fixed size, making it easier to generalize.

3 BRAIN SPECIFIC GNN MODELS

As we discussed above, the brain networks as many features that could be utilized by the neural networks. For example, with prior clinical knowledge, we know there may be some latent features in the brain networks that are difficult to be captured by standard neural networks. To capture these latent features, Suk et al. proposed a latent feature representation with a stacked auto-encoder (SAE) [16]. However, the work provides limited improvement on the classification model itself. One of the first specialized model on the brain networks is *BrainNetCNN* [10] proposed by Kawahara et al. in 2017.

3.1 BrainNetCNN

The most important improvement in BrainNetCNN is the three kinds of layers they proposed. They introfuced edge-to-edge, edge-to-node and node-to-graph layers, claiming it will better leverage topological locality of structural brain networks than other models. Graphical data go through all three kinds of layers sequentially, eventually feeding into a fully connected layer for classification or regression tasks. In the *edge-to-edge* (E2E) layer, each edge, represented by a position in the adjcency matrix, is learned and expanded. The output is defined as a filtered adjacency matrix

$$A_{i,j}^{l+1,n} = \sum_{m=1}^{M^l} \sum_{k=1}^{|\Omega|} r_k^{l,m,n} A_{i,k}^{l,m} + c_k^{l,m,n} A_{k,j}^{l,m} \tag{6}$$

where c and r are learnable weights of the nth filter.

Subsequent to E2E filters are the *edge-to-node* (E2N) layers. In this layer, the adjacency matrices are squashed into nodes representations, defined as follows

$$a_i^{l+1,n} = \sum_{m=1}^{M^l} \sum_{k=1}^{|\Omega|} r_k^{l,m,n} A_{i,k}^{l,m} + c_k^{l,m,n} A_{k,j}^{l,m} \tag{7}$$

The right hand side of this exactly the same as the one in the E2E layer. The left side, however, is a one-dimentional vector with size equal to the size of the node, instead of a 2D vector with the same size as the adjacency matrix.

The *node-to-graph* layer, as its name suggests, further reduces the dimension from node representation to graph representation

$$a^{l+1,n} = \sum_{m=1}^{M^l} \sum_{k=1}^{|\Omega|} w_i^{l,m,n} a_i^{l,m} \tag{8}$$

This reduces the result a from a vector of node size to a single scalar.

With this design, the author state that they can achieve a consistently better result than all baselines using brain network datasets.

3.2 BrainGNN

In 2021, Li et. al proposed another interesting brain network-specialized GNN model called *BrainGNN* [13]. They utilized one of the most important prior knowledge of the brain networks: region of interests (ROIs). As mentioned in the introduction section, each node in a brain network is assigned a specific ROI, and each ROI has a specific clinical meaning. Therefore, utilizing such feature would be great for training and interpretation of the results.

In order to leverage this information, they proposed *Ra-GConv* layer, defined as follows

$$vec(W_i^{(l)}) = f_{MLP}^{(l)}(r_i) = \Theta_2^{(l)} relu(\Theta_1^{(l)} r_i) + b^{(l)} \tag{9}$$

where r_i is node i's regional information. Θ_1, Θ_2 are weight parameters in MLP and $b^{(l)}$ is the bias term.

The *Ra-GConv* layer, combined with dropout layers in between and MLP layers on each ends, defines the *BrainGNN* model. Since it utilizes the ROI information, it outperforms the previous model, *BrainNetCNN*, in the two classification task, as stated by Li et al. Another advantage is that it provides interpretability, as each node (ROI) corresponds to a specific region in a brain.

3.3 BrainNNExplainer

BrainNNExplainer, proposed by Cui et al., makes an attempt utilizing the edge weight in graphs, a feature of brain networks unused by all models mentioned above [5]. They first construct the message passing vector

$$m_{ij}^{(l)} = MLP_{\Theta}\left(\left[h_i^{(l)}; h_j^{(l)}; w_{ij}\right]\right)$$

where h_i is node i's embedding and $w_i j$ is the edge weight of the edge from node i to j. Through this simple concatenation, the edge weight is utilized.

Other models, including CGTS [15], MGCN-GAN [22] and DMBN [24], are also designed to train specifically on brain networks. Most works are on improving the loss terms, which will be skipped due to page limit.

4 FUTURE DIRECTIONS

The area of applying GNN on brain networks is relatively nascent. While many generic GNNs can directly apply on graphs, they do not utilize features of brain networks. Multiple models has been proposed to utilize specific features, but many of the ideas still have space to improve. For example, *BrainNNExplainer* utilizes edge weight through a simple concatenation, which may not be the best use of this information. Other potentially useful features, such as the increase and decrease of edge weights from nodes to nodes, are not leveraged. The region of interst, while could provide very useful prior knowledge, are used mainly for interpretation and regularization. Another challenge on the brain networks is that most datasets have a relatively small sample sizes. Models that works well on large datasets may not work well on smaller ones. What makes it worse is that brain networks are usually noisy, making it even harder for deep learning models to leverage their information. Potential future directions include using semi-supervised methods to expand the sample size, as s larger sample size generally leads to a better performance. Graph neural networks itself has limited learning powers as well. Cycles are one of the most basic graph features, but they are not captured by any of the current GNN models. The feature propagation technique may need to be improved to better capture this information.

REFERENCES

[1] K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. 2016. The extreme classification repository: Multi-label datasets and code. http://manikvarma.org/downloads/XC/XMLRepository.html

[2] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* (2013).

[3] Bokai Cao, Liang Zhan, Xiangnan Kong, S Yu Philip, Nathalie Vizueta, Lori L Altshuler, and Alex D Leow. 2015. Identification of discriminative subgraph patterns in fMRI brain networks in bipolar affective disorder. In *International Conference on Brain Informatics and Health*. Springer, 105–114.

[4] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. 2020. Principal neighbourhood aggregation for graph nets. *arXiv preprint arXiv:2004.05718* (2020).

[5] Hejie Cui, Wei Dai, Yanqiao Zhu, Xiaoxiao Li, Lifang He, and Carl Yang. 2021. BrainNNExplainer: An Interpretable Graph Neural Network Framework for Brain Network based Disease Analysis. *arXiv preprint arXiv:2107.05097* (2021).

[6] Nico U.F. Dosenbach, Damien A. Fair, Alexander L. Cohen, Bradley L. Schlaggar, and Steven E. Petersen. 2008. A dual-networks architecture of top-down control. *Trends in Cognitive Sciences* 12, 3 (2008), 99–105. https://doi.org/10.1016/j.tics.2008.01.001

[7] Farzad V. Farahani, Waldemar Karwowski, and Nichole R. Lighthall. 2019. Application of Graph Theory for Identifying Connectivity Patterns in Human Brain Networks: A Systematic Review. *Frontiers in Neuroscience* 13 (2019). https://doi.org/10.3389/fnins.2019.00585

[8] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. 2018. Large-scale learnable graph convolutional networks. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1416–1424.

[9] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687* (2020).

[10] Jeremy Kawahara, Colin J. Brown, Steven P. Miller, Brian G. Booth, Vann Chau, Ruth E. Grunau, Jill G. Zwicker, and Ghassan Hamarneh. 2017. BrainNetCNN: Convolutional neural networks for brain networks; towards predicting neurodevelopment. *NeuroImage* 146 (2017), 1038–1049. https://doi.org/10.1016/j.neuroimage.2016.09.046

[11] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[13] Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H. Staib, Pamela Ventola, and James S. Duncan. 2021. BrainGNN: Interpretable Brain Graph Neural Network for fMRI Analysis. *Medical Image Analysis* 74 (2021), 102233. https://doi.org/10.1016/j.media.2021.102233

[14] Jin Liu, Min Li, Yi Pan, Wei Lan, Ruiqing Zheng, Fang-Xiang Wu, and Jianxin Wang. 2017. Complex brain network analysis and its applications to brain disorders: a survey. *Complexity* 2017 (2017).

[15] Abubakhari Sserwadda and Islem Rekik. 2021. Topology-guided cyclic brain connectivity generation using geometric deep learning. *Journal of Neuroscience Methods* 353 (2021), 108988.

[16] Heung-Il Suk, Seong-Whan Lee, and Dinggang Shen. 2015. Latent feature representation with stacked auto-encoder for AD/MCI diagnosis. *Brain Structure and Function* 220, 2 (2015), 841–859.

[17] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot. 2002. Automated Anatomical Labeling of Activations in SPM Using a Macroscopic Anatomical Parcellation of the MNI MRI Single-Subject Brain. *NeuroImage* 15, 1 (2002), 273–289. https://doi.org/10.1006/nimg.2001.0978

[18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[19] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[20] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[21] Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. *arXiv preprint arXiv:1806.08804* (2018).

[22] Lu Zhang, Li Wang, and Dajiang Zhu. 2020. Recovering brain structural connectivity from functional connectivity via multi-gcn based generative adversarial network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 53–61.

[23] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. 2019. Graph convolutional networks: a comprehensive review. *Computational Social Networks* 6, 1 (2019), 1–23.

[24] Wen Zhang, Liang Zhan, Paul Thompson, and Yalin Wang. 2020. Deep representation learning for multimodal brain networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 613–624.

[25] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001