Graph Neural Networks for COVID-19 Drug Discovery

Mark Cheung
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, USA
markcheu@andrew.cmu.edu

José M. F. Moura
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, USA
moura@andrew.cmu.edu

2020 IEEE International Conference on Big Data (Big Data) | 978-1-7281-6251-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/BigData50022.2020.9378164

***Abstract*—Deep learning has led to major advances in fields like natural language processing, computer vision, and other Euclidean data domains. Yet, many important fields have data defined on irregular domains, requiring graphs to be explicitly modeled. One such application is drug discovery. Recently, research has found that using graph neural network (GNN) models, given enough data, can perform better than using human-engineered fingerprints or descriptors in predicting molecular properties of potential antibiotics.**

We explore these state-of-the-art AI models on predicting desirable molecular properties for drugs that can inhibit SARS-CoV-2. We build upon the GNN models with ideas from recent breakthroughs in geometric deep learning, inspired by the topologies of the molecules. In this poster paper, we present an overview of the drug discovery framework, drug-target interaction framework, and GNNs. Preliminary results on two COVID-19 related datasets are encouraging, achieving a ROC-AUC of 0.72 for FDA-approved chemical library screened against SARS-CoV-2 *in vitro*.

***Index Terms*—Topology adaptive graph convolutional neural networks, message passing neural networks, GNN, COVID-19, SARS-CoV-2**

I. Introduction

The scientific community has united to combat the growing COVID-19 pandemic. At the time of writing, there are more than 200 vaccines in development, 12 have begun large-scale (phase 3) clinical trials, and 2 have shown to be 95% effective (Pfizer and Moderna) [1]. Given the urgency of the situation, vaccine development and approval has been accelerated from the general timeline of 10-15 years [2] to potentially under a year. Part of this acceleration is made possible due to the development of artificial intelligence (AI).

The first stage of this process is drug discovery, whereby scientists typically spend 2-5 years to identify vaccine candidates [2]. Instead of testing most drugs *in vitro* in the lab, scientists are able to leverage computational methods to screen potential drugs *in silico*, saving a lot of time and physical resources, as there can be millions of compounds to test. In the last decade, scientists have found machine learning models to be a good predictor of potential drug candidates found experimentally. More recently, it is shown that using deep learning, given enough data, can result in superior performance for antibiotic drug discovery [3]. For extensive reviews of AI applied to drug discovery, see [4], [5].

The scope of this paper is on one type of deep learning models, namely graph neural networks (GNNs), applied to COVID-19 drug discovery. In section II, we give a cursory summary of drug-target interaction framework and the deep learning approach based on GNNs. Then we describe our specific approach combining topology adaptive graph neural networks and message-passing neural network in section III. We present the experiments and preliminary but promising results on two COVID-19 related datasets in sections IV and V, respectively. Finally, we conclude in section VI.

II. Background and Related Work

A. Drug-target Interaction Framework

Drug-target interaction (DTI) is one of the most important steps in the drug discovery stage. It characterizes the binding of chemical compounds to the protein targets. Among several, drug screening and drug repurposing are two main tasks. Drug screening identifies ligand molecules that can bind to specific proteins, whereas drug repurposing finds new therapeutic uses of existing and available drugs. For both tasks, deep learning has demonstrated superior performance than traditional computational methods [6]. In addition to classic chemical fingerprints drug encoders like Morgan and RDKit-2D, there are many deep learning-based drug encoders that can be categorized into autoencoder, convolutional neural networks, recurrent neural networks, transformers, and GNNs.

B. Graph Neural Networks

Graph neural networks (GNNs) have shown promising results for drug discovery applications [7], [8]. They are motivated in part by convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which have yield significant improvements in computer vision, natural language processing, and other Euclidean domains. Unlike CNNs and RNNs, GNNs are able to extract features via the graph

This material is based upon work partially funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center; DM20-0590. This work is also partially supported by NSF grants CPS 1837607 and CIF 1513936.

structure or topology, defined by its adjacency matrix. Like a conventional CNN architecture, a GNN architecture can have convolutional, pooling, and fully connected layers. Convolutional layers extract the features from the graph. Pooling layers reduce the dimensionality and learn the hierarchical representations. For tasks like graph classification (including COVID-19 drug discovery), there may be an aggregation layer to compare graphs of different sizes. Finally, fully connected layers perform the classification. For a more in-depth review of GNNs, see [9].

III. Methods

In this paper, we focus on the convolutional layer of the GNN architecture. In this section, we describe our approach combining two GNN convolution methods: topology adaptive graph neural networks and message-passing neural network.

The general graph convolution approach is as follows. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of N vertices defined by graph signal $\mathbf{X}^{(0)} \in \mathbb{R}^{N \times C}$ (C is the number of signal dimensions) and $\mathcal{E}$ is defined by its adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. The general form of the convolutional layer, introduced by [10], is

$$\mathbf{X}^{(\ell+1)} = \sigma\left(\tilde{\mathbf{A}}\mathbf{X}^{(\ell)}\mathbf{W}^{(\ell)}\right), \tag{1}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{C \times F}$ is the trainable weight matrix, σ is the nonlinear activation function, F is the number of output features, ℓ is the layer number with input layer being $\ell = 0$. In practice, $\tilde{\mathbf{A}}$ is normalized to address vanishing and exploding gradients (for nodes with very small and large degrees, respectively). For example, using $\tilde{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ in place of $\tilde{\mathbf{A}}$, where $\tilde{\mathbf{D}}$ is the degree matrix of $\tilde{\mathbf{A}}$.

A. Topology Adaptive Graph Convolutional Neural Networks

The topology adaptive graph convolution network (TAGCN) implementation of graph convolution [11] uses the polynomial filter coefficients as learnable weights. The general form of the TAGCN graph convolutional layer is

$$\mathbf{X}^{(\ell+1)} = \sigma\left(\mathbf{X}^{(\ell)}\mathbf{W}_0^{(\ell)} + \mathbf{A}\mathbf{X}^{(\ell)}\mathbf{W}_1^{(\ell)} + \ldots + \mathbf{A}^K\mathbf{X}^{(\ell)}\mathbf{W}_K^{(\ell)}\right) \tag{2}$$

$$= \sigma\left(\sum_{k=0}^{K}\mathbf{A}^k\mathbf{X}^{(\ell)}\mathbf{W}_k^{(\ell)}\right), \tag{3}$$

where K is the degree of the graph polynomial filter and a hyperparameter of the model. We can use normalized version of $\mathbf{A}$ in (2), as we did in (1). Fig. 1 shows the visual representation of polynomial filter of degree 2. TAGCN allows learning of more complex functions with deeper models.

B. Message Passing Neural Networks

Message passing neural networks (MPNNs) are a generalization of GNNs, first formulated by [12]. Following the same definition given above, we define $\boldsymbol{h}_i^l \in \mathbb{R}^C$ be the states of node i at t-th time step and $\boldsymbol{h}_i^0 \in \mathbb{R}^C$ to be the initialized state (from splitting $\mathbf{X}^{(0)}$ by nodes).

Fig. 1: TAGCN Polynomial filter of degree 2 for blue node, signals at the red nodes are propagated to the blue one and aggregated, i.e., $\mathbf{A}^2x^{(\ell)}\mathbf{W}_2^{(\ell)} + \mathbf{A}x^{(\ell)}\mathbf{W}_1^{(\ell)} + \mathbf{I}_N x^{(\ell)}\mathbf{W}_0^{(\ell)}$

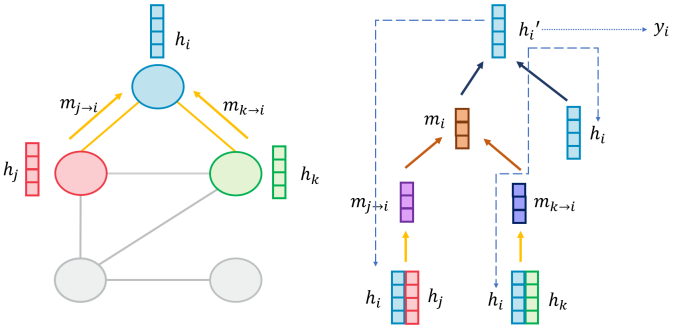


Fig. 2: Propagation and aggregation steps for node i.

One GNN layer can be expressed using message passing neural network:

$$\boldsymbol{m}_{i\to j}^{\ell+1} = f_\alpha^\ell(\boldsymbol{h}_i^\ell, \boldsymbol{h}_j^\ell, \mathbf{A}_{ij}) \qquad \text{(propagation)} \tag{4}$$

$$\boldsymbol{h}_j^{\ell+1} = f_\beta^\ell(\{\boldsymbol{m}_{i\to j}^{\ell+1} | i \in \mathcal{N}_j\}, \boldsymbol{h}_j^\ell) \qquad \text{(aggregation)} \tag{5}$$

where ℓ is the time step (or layer number in GNN), f_α^ℓ and f_β^ℓ are parametrized function like neural network, $\boldsymbol{m}_{i\to j}^{\ell+1}$ is the propagated information from node i to node j. For example, f_α^ℓ can be a MLP followed by a nonlinear activation. $\boldsymbol{m}_j^{\ell+1}$ is the aggregated information with $\mathcal{N}_j$ representing all the incoming neighbors of node j:

$$\boldsymbol{m}_i^{\ell+1} = \sum_{j \in N_i} \boldsymbol{m}_{j\to i}^{\ell+1} \tag{6}$$

To update the hidden state at $\ell + 1$ for each node, we can also use a MLP or a recurrent neural network like a GRU or LSTM for f_β:

$$\boldsymbol{h}_i^{\ell+1} = f_\beta(\boldsymbol{m}_i^{\ell+1}, \boldsymbol{h}_i^\ell) \tag{7}$$

After $\mathcal{L}$ time steps/layers, we get the final states $\boldsymbol{h}_i^{\mathcal{L}}$ and predict the graph label using a readout function R (which can also be MLPs):

$$\widehat{y} = R(\{\boldsymbol{h}_i^{\mathcal{L}} \mid \forall i \in \mathcal{V}\}) \tag{8}$$

Then apply a loss function like cross entropy loss $\mathrm{CE}(\widehat{y}, y)$. See fig. 2 for a visualization of this process.

C. Topological adaptive message passing neural networks

Our approach is applying TAGCN to MPNN to get topology adaptive message passing neural network. Instead of summing just the direct neighbors in 6, we aggregate nodes up to a degree K hops away and apply the nonlinear activation after the summation. MPNN generalization enables us to consider more variants such as passing messages on the nodes.

IV. Experiments

A. Datasets

As shown in table I, we consider two COVID-19 related datasets, with positive/negative indicating whether the sample is effective. Amu_sars_cov_2_in_vitro dataset is a list of FDA-approved compounds screened against SARS-CoV-2 in vitro, and AID1706_binarized_sars is a much longer list of compounds screened against SARS-CoV in-vitro via fluorescence. For more information and download links, see https://www.aicures.mit.edu/data.

TABLE I: COVID-19 related datasets

Filename	Negative	Positive
Amu_sars_cov_2_in_vitro	1484	88
AID1706_binarized_sars	290767	446

B. Experimental Setup

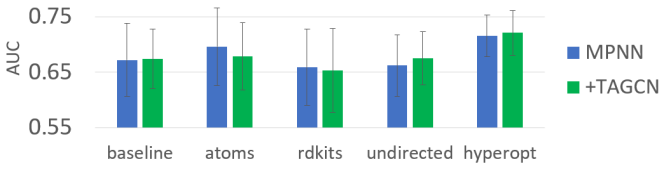
In our experiments, we use the DeepPurpose DTI framework from [6] and the MPNN model from [13] as baseline. We perform 5-fold cross-validation using scaffold split and evaluate our results using the area under the receiver operating characteristic curve (ROC-AUC). We consider several hyperparameters such as number of layers, degree of the graph polynomial filter, extra features (RDKit-2D), message passing type (atoms/edges), and class weights (for class imbalance).

V. Results

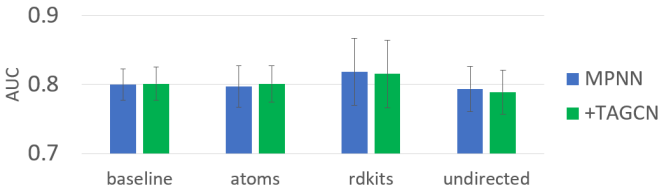
Figure 3 shows the preliminary results for Amu_sars_cov_2_in_vitro and AID1706_binarized_sars datasets. Blue and green bars represent the mean ROC-AUCs for MPNN baseline, and MPNN with TAGCN, respectively. Black lines represent the standard deviation. We have set the baseline to be edge-based message passing and tried using atom-based message passing, adding extra features (RDKit-2D), undirected message passing, and for the smaller dataset, optimized over all hyperparameters. In general, GNNs yield great performance on these datasets, achieving a ROC-AUC of 0.72 and 0.82 for Amu_sars_cov_2_in_vitro and AID1706_binarized_sars datasets, respectively. More results on both synthetic and real data are needed.

VI. Conclusion

We have presented an overview of the drug discovery framework, drug-target interaction framework, and GNNs. We have also introduced a novel approach combining two existing GNN methods. Experiments suggest that GNNs can perform well on two coronavirus datasets, and that these methods may aid in COVID-19 drug discovery.



(a) Amu_sars_cov_2 results



(b) AID1706_binarized_sars results

Fig. 3: Comparison of results in terms of ROC-AUC.

References

[1] N. Kommenda and F. Hulley-Jones, "Covid vaccine tracker: when will a coronavirus vaccine be ready?" *The Guardian*, Nov. 2020.

[2] International Federation of Pharmaceutical Manufacturers and Associations, "The complex journey of a vaccine," Jul 2019.

[3] J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, C. R. MacNair, S. French, L. A. Carfrae, Z. Bloom-Ackermann, V. M. Tran, A. Chiappino-Pepe, A. H. Badran, I. W. Andrews, E. J. Chory, G. M. Church, E. D. Brown, T. S. Jaakkola, R. Barzilay, and J. J. Collins, "A deep learning approach to antibiotic discovery," *Cell*, vol. 180, no. 4, pp. 688 – 702.e13, 2020.

[4] Y. Zhou, F. Wang, J. Tang, R. Nussinov, and F. Cheng, "Artificial intelligence in COVID-19 drug repurposing," *The Lancet Digital Health*, Sep. 2020.

[5] J. M. Levin, T. I. Oprea, S. Davidovich, T. Clozel, J. P. Overington, Q. Vanhaelen, C. R. Cantor, E. Bischof, and A. Zhavoronkov, "Artificial intelligence, drug repurposing and peer review," *Nature Biotechnology*, vol. 38, no. 10, pp. 1127–1131, Sep. 2020.

[6] K. Huang, T. Fu, L. Glass, M. Zitnik, C. Xiao, and J. Sun, "Deeppurpose: a deep learning library for drug-target interaction prediction and applications to repurposing and screening," *Computing Research Repository*, vol. abs/2004.08919, 2020.

[7] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, "Molecular graph convolutions: moving beyond fingerprints," *Journal of Computer-Aided Molecular Design*, vol. 30, no. 8, pp. 595–608, Aug. 2016.

[8] Y.-C. Lo, S. E. Rensi, W. Torng, and R. B. Altman, "Machine learning in chemoinformatics and drug discovery," *Drug Discovery Today*, vol. 23, no. 8, pp. 1538 – 1546, 2018.

[9] M. Cheung, J. Shi, O. Wright, L. Y. Jiang, X. Liu, and J. M. F. Moura, "Graph signal processing and deep learning: Convolution, pooling, and topology," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 139–149, 2020.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, (ICLR) 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[11] J. Du, S. Zhang, G. Wu, J. M. F. Moura, and S. Kar, "Topology adaptive graph convolutional networks," *Computing Research Repository*, vol. abs/1710.10370, 2017.

[12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 1263–1272.

[13] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, A. Palmer, V. Settels, T. Jaakkola, K. Jensen, and R. Barzilay, "Analyzing learned molecular representations for property prediction," *Journal of Chemical Information and Modeling*, vol. 59, no. 8, pp. 3370–3388, 2019.