Graph Convolutional Networks (GCNs)

for Molecular Property Prediction in Drug Development

Undergraduate Research Thesis

Presented in partial fulfillment of the requirements for graduation with honors research distinction in Computer Science in the undergraduate colleges of The Ohio State University

by

Yifan Song

Project Advisor: Professor Xia Ning

The Ohio State University

April 2020

Abstract

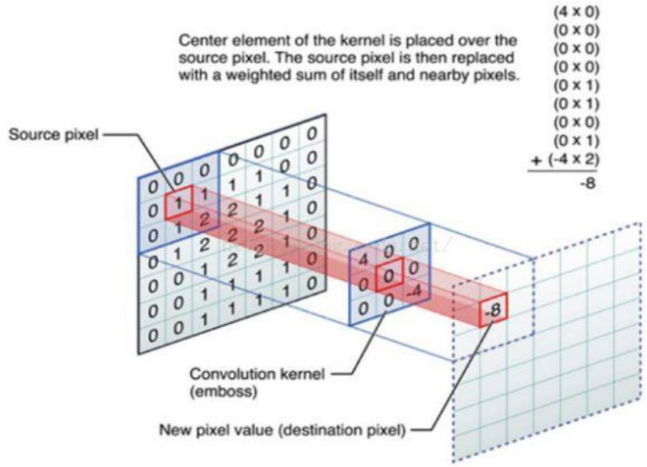
Molecular property prediction is key to drug development. The rising of deep learning techniques provides new possibilities to learn the molecular properties directly from chemical data. In particular, graph convolutional networks have been introduced into the field and made significant enhancements compared to traditional methods. The first part of this paper serves as a study to explore and evaluate this emerging method while the second part demonstrates that graph convolution networks can be further improved by incorporating attention mechanism, another influential deep learning idea.

1 Introduction

Drug discovery and development is a resource and time-consuming process. In modern drug development, machine learning tools are involved to reduce the huge cost. However, traditional machine learning methods do not always have a good performance in analyzing the complex relations and structures among a large collection of molecular compounds. Thus, new attempts of using deep learning tools have risen recently in drug development. Despite the fact that deep learning has achieved substantial success in areas such as computer vision, image analysis and language processing, its application in biomedical informatics or drug development is still limited. This research project is aimed to study and make progress in applying deep neural networks to drug development.

More specifically, one emerging approach is to use convolutional neural networks (CNNs), a special architecture of deep neural networks. Conventional CNNs work well on structured data such as images. For molecular compounds, which may have special topologies, an extended architecture named graph convolutional networks (GCN) is introduced. GCN is a methodology to extend convolution technology on graph, a non-Euclidean structure. Molecules, composed of atoms and bonds, would be typically represented as a graph, in which atoms and bonds can be considered as nodes and edges, respectively. This research project will be focused on using GCNs to better analyze chemical graphs and predict molecular biological activities and chemical properties. The technique can be used in the early stages of drug development to identify promising drug candidates.

(a)



(b) ***(c)***

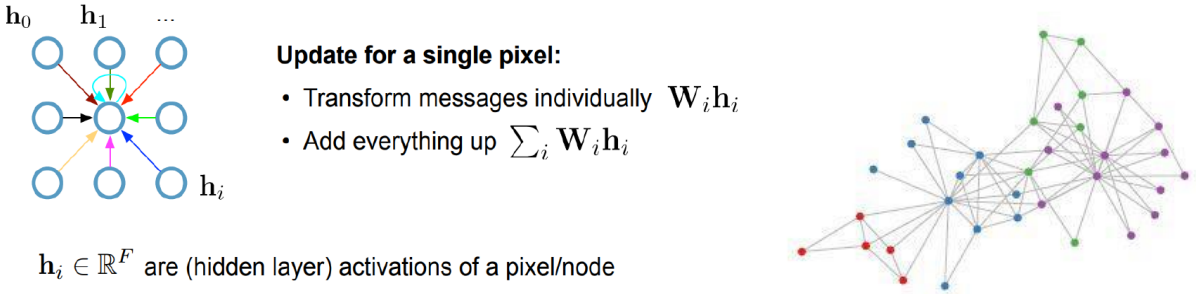


Figure 1*. (a) Illustration of convolution computation; (b) Convolution layer in CNN; (c) Typical graph, a non-Euclidean structure*

2 Related Work

2.1 Graph Convolution

GCN was first proposed where graph convolutional operations were defined in the Fourier domain [1]. It involves intense computations as the eigen-decomposition of the Laplacian matrix

of the graph is needed. Later, smooth parametric spectral filters, in particular Chebyshev polynomials [2], has been introduced in convolution computation to achieve localization in the spatial domain thus enhance the computational efficiency [3]. Recently, Thomas N. Kipf [4] has further simplified by providing a first-order approximation of the Chebyshev polynomials. The derivation finally leads to a one-step neighbors localization and achieves a simple but direct formula with state-of-the-art performance.

2.2 Application of GCN for Molecular Graphs

Duvenaud et. al. [5] first proposed a method to generate data-driven fingerprints using neural networks and tested on molecular property prediction tasks. A real-value vector was learned to represent a molecule, called neural fingerprints. The representation of a molecule was obtained by aggregating representations of all atoms. For each atom, the vector was learned in a convolutional procedure that the neighborhood information was aggregated to update the center atom. Schutt et al. [6] also proposed a neural network for predicting molecular total energy. Similarly, the atom energy was obtained by interaction passes from its neighbors and the final molecular energy was obtained by summing up all atoms. Later, Gilmer et al. [7] reformulated various previous works into a common framework called message-passing neural network.

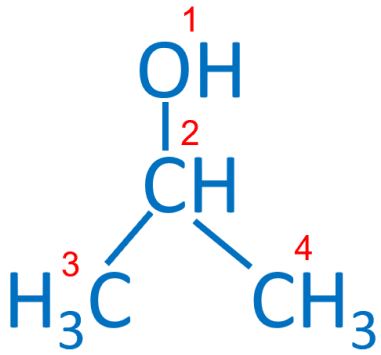
In addition to node (atom) features, some researchers also took edge (bond) information into account for GCN. Kearnes et. al. [8] proposed a network, named “Weave”, that considering both atom representations and pair-wise representations, i.e. edge features, in the convolution process and applied this model in the molecular property classification tasks. Shang et. al. [9] later first

incorporate attention mechanism into GCN and proposed an edge attention based graph convolutional network. The author considered bond feature as relation and trained the attention score for each edge feature. The edge attention is then shared across all multi-relational graphs and the final representation of molecular is calculated by concatenating each relation or taking a weighted average. The model is evaluated on both classification and regression tasks for molecular property prediction.

3 Method

3.1 Graph Representation of Molecular

A graph is denoted by $G = (V, E)$, where V is a finite set of nodes with $|V| = N$, and E is a finite set of edges with $|E| = M$. Then A would represent the adjacency matrix of $N \times N$ where $A_{ij} = 1$ if there exists an edge between node i and j, otherwise 0. A molecular can be represented as a graph with atoms as nodes and bonds as edges. For each node, there is a feature vector corresponding to the atom features such as atom type, number of hydrogens attached and aromaticity. Note that the adjacency matrix used in this paper's model only represents the connectivity between atoms (either 0 or 1) which doesn't include any edge (bond) features. In addition, the elements on the diagonal of the adjacency matrix are all 1 which represents a self-loop on each atom, i.e. each atom is connected to itself. This setting is used to facilitate the computation in GCN. Figure 2 illustrates an example of graph representation for Isopropyl Alcohol. The figure shows the adjacency matrix according to the molecular graph with self-loops, as well as sample atom features on each node.



$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$X_1 = \begin{bmatrix} 8 \\ 1 \\ 0 \end{bmatrix},\ X_2 = \begin{bmatrix} 6 \\ 1 \\ 0 \end{bmatrix},\ X_3 = \begin{bmatrix} 6 \\ 3 \\ 0 \end{bmatrix},\ X_4 = \begin{bmatrix} 6 \\ 3 \\ 0 \end{bmatrix}$$

Atom type (one-hot encoding)
Number of Hydrogens
Aromaticity (1: aromatic, 0: non-aromatic)

Figure 2*. Molecular graph representation of Isopropyl Alcohol (2-Propanol, CH_3-CHOH-CH_3), adjacency matrix and node features*

3.2 GCN

As mentioned in section 2, Graph convolutional networks (GCN) can be understood as a kind of message passing framework, which updates node features from its neighbors in a graph structure. According to the formula introduced by Kipf [4], GCN updates each node state as follow:

$$H^{(l+1)} = \sigma(AH^lW^l)$$

where σ is the activation function and *ReLU* will be used for the implementation of this project, H^l is the node feature vector of l^{th} layer and H^1 is the initial feature vector, A is the adjacency matrix, and W is the convolution weight. Thus, the message passing process for molecular can be interpreted as the interaction between an atom and its connected atoms. Suppose there are two

adjacent nodes 1 and 2 for a center node 0, the formula of updating node 0 can be expanded as follow:

$$H_0^{l+1} = \sigma\left(H_0^l w_0^l + H_1^l w_0^l + H_2^l w_0^l\right)$$

which means the center node feature is updated by summing up all adjacent features multiplies by a same weight then pass an activation function. After passing graph convolution layers, the atom feature vectors have been updated by graph convolution and a readout layer is now needed to transform atom feature vectors into the target property. We first calculate the molecular feature vector by simply taking the sum of atom feature vectors. For a property prediction (classification) task, the molecular feature vector will then pass a fully connected layer to get the result as follow:

$$Z = \sigma(MLP(\sum H_i))$$

where Z is the readout value, σ is the activation function and *ReLU* will be used here, *MLP* represents multi-layer perceptron, and H_i is the i^{th} atom feature vector.

The GCN model typically yields a good performance compare to traditional models and computationally efficient as it requires a small number of parameters. However, there're still limits in the model by lacking the following terms: 1) the importance of each atom for a certain property; 2) the interaction strength between each atom pairs. It is expected that the model would be improved by providing these two terms which will be discussed following.

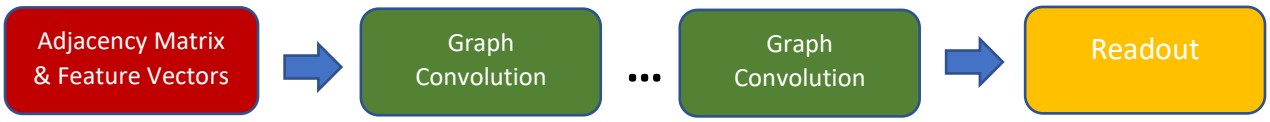


Figure 3. *GCN Architecture: first parsed the molecular into adjacency matrix and feature vectors, then passed through one or more graph convolution layers, and finally predicted property through readout layer*

(a) ***(b)***

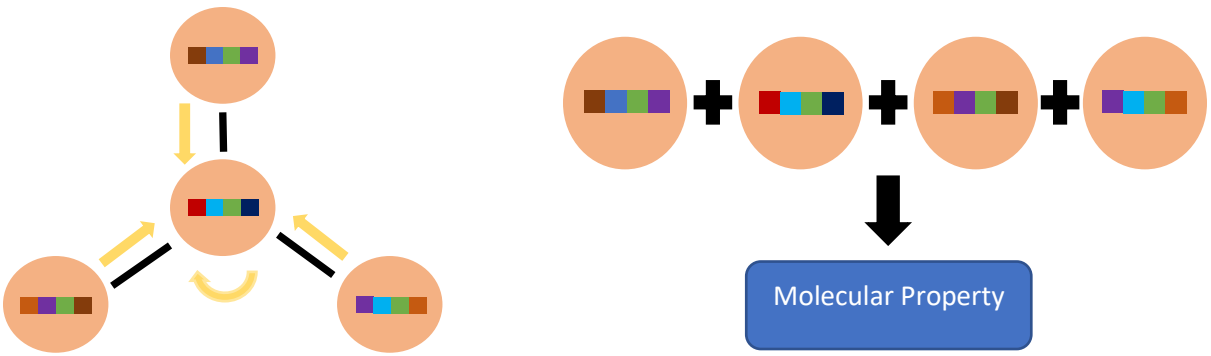


Figure 4. *(a) graph convolution layer structure (example of center node); (b) readout layer structure*

3.3 Attention Mechanism on GCN

Attention mechanism is one of the most influential ideas of deep learning. It was first applied in the field of natural language processing of machine translation [10]. For example, when

translating a sentence A to sentence B, the first word in B is not only depending on the first word in A but all words. Then, the attention can be understood as the importance of each word in A that affects the translation at this timestamp, i.e. how much we should pay attention to each word. Hence, the attention mechanism leverages the network to find the best word to be translated. The attention mechanism can then be applied to GCN to solve the two limits in the previous section.

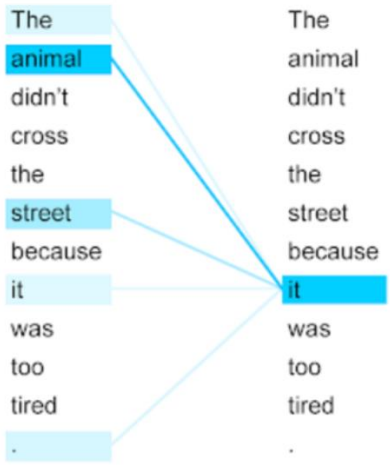


Figure 5. *A classical example of attention mechanism visualization: showing the word which "it" represents, the darker color indicates a heavier attention*

The first term can be solved directly by adding attention in the readout layer. The above GCN generates the molecular feature vector by simply taking the sum of atom feature vectors. However, different atoms would have different strength or importance for a certain property. Therefore, the attention mechanism can be applied in this case to learn the importance on each atom then do a weighted sum to calculate the molecular feature vector as follow:

$$M = \sum a_i H_i$$

where M is the molecular feature vector ($M = \sum H_i$ in the previous case), a_i is the attention of i^{th} atom, and H_i is the i^{th} atom feature vector. Then the next question would be how to get the values of attentions. In this case, we simply use the final state of each atom feature vectors to obtain attentions, the atom feature vector here is called the "query" vector. As we use the atom features to obtain the attention of each atom, the attention here is also a self-attention. A typical way to calculate attention from certain vector is to pass the vector through a multi-layer perceptron then an activation function. The activation function used here is *Softmax* which aims to normalize a set of vectors to a probability distribution that sums to 1. Hence, we can interpret the attention as the importance (proportion) that each atom contributes to the target property. The exact formula of attention can be got as follow:

$$a_i = Softmax(MLP(H_i)) = \frac{e^{MLP(H_i)}}{\sum e^{MLP(H_i)}}$$

where *MLP* represents multi-layer perceptron, and the attention is then calculated through a *Softmax* function.

The same idea can then be extended to the convolution process to target the second issue. Still using the expanded example of convolution in the previous section, the update formula of center node 0 with two adjacent nodes is as follow:

$$H_0^{l+1} = \sigma\left(a_{00}^l H_0^l w_0^l + a_{01}^l H_1^l w_0^l + a_{02}^l H_2^l w_0^l\right)$$

The only difference is with the general GCN is that there is an attention term to multiply with the original feature vector and weight vector. The attention term a_{ij}^l means the attention between center node *i* and its adjacent node *j* in layer *l*. To obtain the attention, the query vector used here is the concatenation of feature vector of atom *i* and feature vector of atom *j* as the attention

represents some measurement of the relation between node *i* and *j*. Still using the MLP formula and Softmax activation function, the attention can be calculated as follow:

$$a_{ij} = Softmax(MLP(H_i, H_j)) = \frac{e^{MLP(H_i, H_j)}}{\Sigma_{j \in N}^{N} e^{MLP(H_i, H_j)}}$$

However, in this case, the attention cannot be interpreted as the importance rate of a center node with certain adjacent node; in molecular field, the attention should be interpreted as the interaction strength between center node and certain adjacent node pair compare to other adjacent nodes.

In the next session, we'll add the two attention terms separately to the original GCN model and compare their performances.

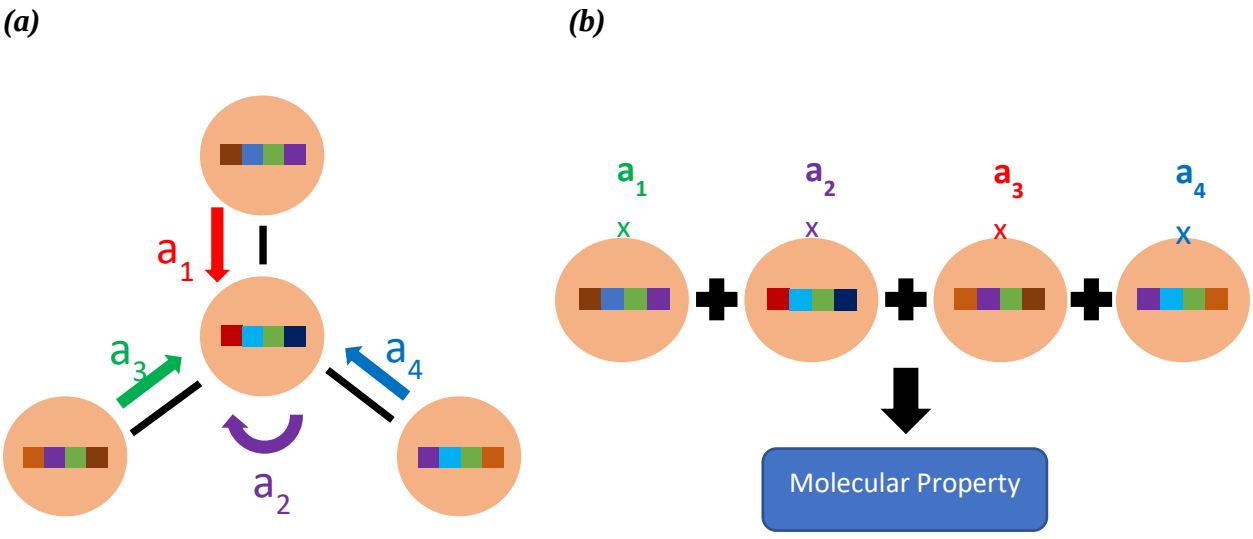


***Figure 6**. Attention mechanism applied on graph convolution layer (a) and readout layer (b)*

4 Experiment

4.1 Dataset

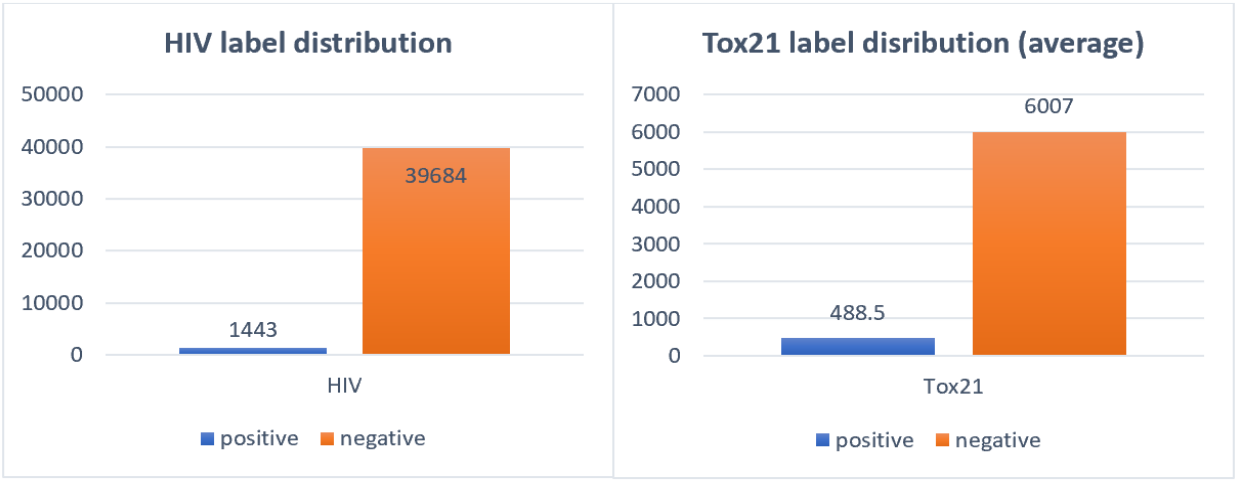
Two datasets, Tox21 [11] and HIV [12], are utilized in the experiment to evaluate the performance of GCN and GCN with attention.

Tox21: The dataset comes from the Toxicology in the 21st century research initiative. It contains 7831 environmental compounds and drugs as well as their biological outcomes of 12 pathway assays that measure various nuclear receptor or oxidative stress responses that are either positive or negative, e.g., androgen receptor, estrogen receptor, and mitochondrial membrane potential. The dataset has a mean of 18.57 and a median of 16 atoms in each molecular, the number for bonds is 19.29 and 17 respectively.

HIV: The dataset was introduced by the Drug Therapeutics Program (DTP) AIDS Antiviral Screen, which tested the ability to inhibit HIV replication for over 41,127 compounds. Screening results were evaluated and placed either active or inactive. The dataset has a mean of 25.51 and a median of 23 atoms in each molecular, the number for bonds is 27.47 and 25 respectively.

The model will be built for a binary classification problem for the two datasets, the result of a predicted property would be either positive (1) or negative (0). Both two datasets are imbalanced with fewer positive samples. The label distribution is shown in the following figures.

(a) ***(b)***



(c)

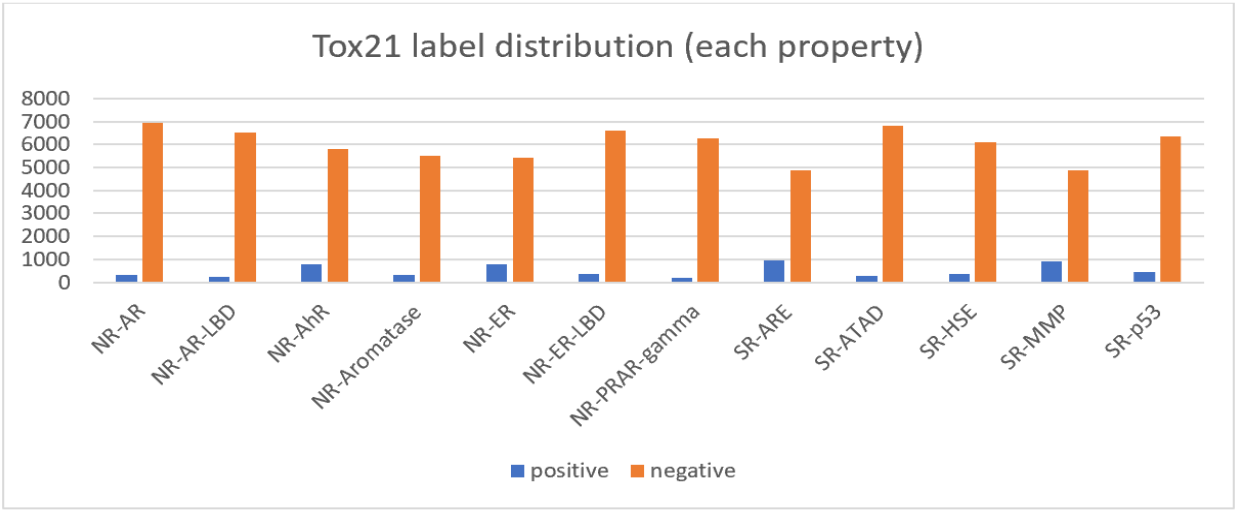


Figure 7. *Label distribution for the datasets*

4.2 Setup

For each dataset, the dataset is randomly and equally split into five subsets to do a 5-fold cross validation. To generate a predictive model, the data is often split into training data and testing

data to evaluate the performance of the model for a “first-seen” data. However, there would exist variability for a one-round split as changing another way of splitting may produce a very different result. To solve this problem, K-fold cross validation [13] is a typical method to reduce variability, as well as problems like overfitting and selection bias, to generate a more generalized result. In 5-fold validation, one piece of subsample is used as testing data and the remaining 4 subsamples are training data; the training-testing process repeats 5 times and each subsample is used as testing data once.

As the dataset is strongly imbalanced with much fewer positive samples, a resampling process will be applied to the training set before training – down-sample the number of negative samples to 1:1 of the positive ones, i.e. randomly drop negative samples to the same number of positive ones.

The initial atom features are extracted with a Python open-source toolkit for molecular analysis – RDKit [14]. The features include:

- One hot encoding of the atom type, supported atom types include ``C``, ``N``, ``O``, ``S``, ``F``, ``Si``, ``P``, ``Cl``, ``Br``, ``Mg``, ``Na``, ``Ca``, ``Fe``, ``As``, ``Al``, ``I``, ``B``, ``V``, ``K``, ``Tl``, ``Yb``, ``Sb``, ``Sn``, ``Ag``, ``Pd``, ``Co``, ``Se``, ``Ti``, ``Zn``, ``H``, ``Li``, ``Ge``, ``Cu``, ``Au``, ``Ni``, ``Cd``, ``In``, ``Mn``, ``Zr``, ``Cr``, ``Pt``, ``Hg``, ``Pb``.
- One hot encoding of the atom degree, supported possibilities include 0 - 10.
- One hot encoding of the number of implicit Hs on the atom, supported possibilities include 0 - 6.
- Formal charge of the atom.

- Number of radical electrons of the atom.
- One hot encoding of the atom hybridization, supported possibilities include ``SP``, ``SP2``, ``SP3``, ``SP3D``, ``SP3D2``.
- Whether the atom is aromatic or not.
- One hot encoding of the number of total Hs on the atom, supported possibilities include ``0 - 4``.

The other parameter settings for the experiment - batch size: 128, optimizer: Adam, learning rate: 1e-3, loss function: cross entropy with logits, number of epochs: 100, early stop: 10

4.3 Result

The first experiment would like to discover how the number of graph convolution layers and the number of hidden features in the graph convolution layer will affect the performance.

(a) ***(b)***

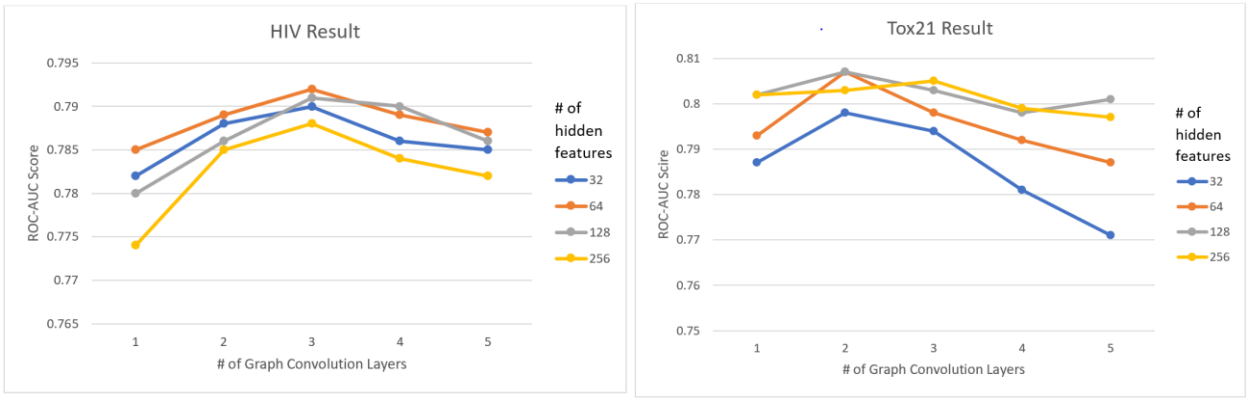


Figure 8. *Result for the first experiment*

In the above figure, we compare the performance of models with different numbers of graph convolution layers and different number of hidden features based on ROC-AUC score. From the experiment on both datasets, we may conclude that the ROC-AUC score first rises then drops when increasing the number of graph convolution layers. This finding is consistent with Kipf and Welling and one possible reason for this is overfitting. For HIV dataset, 3 convolution layers yield the best and the number for Tox21 dataset is 2. Another finding is about the number of hidden features. The model is having a better performance when the number of hidden features is 64 or 128, comparing to 32 or 256. Considering the number of initial node features is 74, a number of hidden features closer to the number of initial features might provide a better performance. In the next experiment comparing GCN and GCN with attentions, the parameters (# of convolution layers and # of hidden features) which provide the best performance in the first experiment will be used.

In the second experiment, we would like to compare the performance for the three models discussed in the previous section, regular GCN, GCN with attention on the readout layer, and GCN with attention on the convolution layer. There are also two base models, Random Forest and SVM, included in the experiment. Random Forest and SVM were two popular methods in predicting molecular properties before introducing deep learning techniques into the field. The settings of baseline models can be referenced to DeepChem [15], an open source tool developed by Stanford.

	HIV	Tox21
Random Forest	0.781	0.736
Kernel SVM	0.747	0.752
GCN	0.792	0.807
GCN + readout attention	0.795	0.812
GCN + convolution attention	**0.813**	**0.834**

From the results showing above, we may conclude that the GCN technique is outperforming the traditional (baseline) models. The effect of attention mechanism varies between the way of adding it: adding the attention mechanism in the readout layer will increase the ROC-AUC score for a very little amount but adding the attention mechanism in the convolution layer will improve the performance significantly. This is under expectation since attention in readout layer can be simply understood as a weight-and-sum but applying the attention mechanism in the convolution layer takes the interaction strength between atoms into account which is a true supplement compared to the original GCN.

5 Conclusion and Discussion

We have introduced the emerging technique in molecular property prediction for drug development, GCN, and proved that the new technique outperforms the traditional methods significantly. We have also introduced the attention mechanism, another influential idea of deep learning, which may be applied and merged with GCN to provide a way of improving the rising

technique. We have shown that adding attention mechanism in the graph convolution process can improve the performance by considering the interaction strength between atoms. However, in real cases, the molecular property is not an effect based on every single atom, nor the interaction between nodes. Sometimes, a pair or a subgroup of atoms can be considered as a substructure that functions together to affect the molecular property and interact with other substructures. If we can detect substructures in the GCN network and apply attention mechanism to represent the interaction effect between substructures, then it is expectable to further enhance the performance of GCN for molecular property prediction in drug use.

Reference

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

[2] Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163

[3] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In Advances in Neural Information Processing Systems. 3844–3852.

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

[5] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. In Advances in neural information processing systems, pages 2224–2232.

[6] Kristof Schutt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet. 2017. A continuous-filter convolutional neural network for modeling quantum interactions. In Advances in Neural Information Processing Systems.

[7] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. arXiv preprint arXiv:1704.01212.

[8] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. 2016. Molecular graph convolutions: moving beyond fingerprints. Journal of computer-aided molecular design, 30(8):595–608

[9] Chao Shang, Qinqing Liu, Ko-Shin Chen, Jiangwen Sun, Jin Lu, JinfengYi, and Jinbo Bi. 2018. Edge attention-based multi-relational graph convolutional networks. arXiv preprint arXiv:1802.04944

[10] Vaswani A, Shazeer N, Parmar N, et al. 2017. Attention is all you need. Advances in Neural Information Processing Systems. 5998-6008.

[11] Tox21 Challenge. https://tripod.nih.gov/tox21/challenge/

[12] AIDS Antiviral Screen Data. https://wiki.nci.nih.gov/display/NCIDTPdata/AIDS+Antiviral+Screen+Data

[13] Airola, A.; Pahikkala, T.; Waegeman, W.; De Baets, Bernard; Salakoski, T. 2011. An experimental comparison of cross-validation techniques for estimating the area under the ROC curve. Computational Statistics & Data Analysis. 55 (4): 1828–1844.

[14] RDKit toolkit. http://www.rdkit.org

[15] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. 2018. MoleculeNet: a benchmark for molecular machine learning. Chemical Science 9, 2 (2018), 513–530.