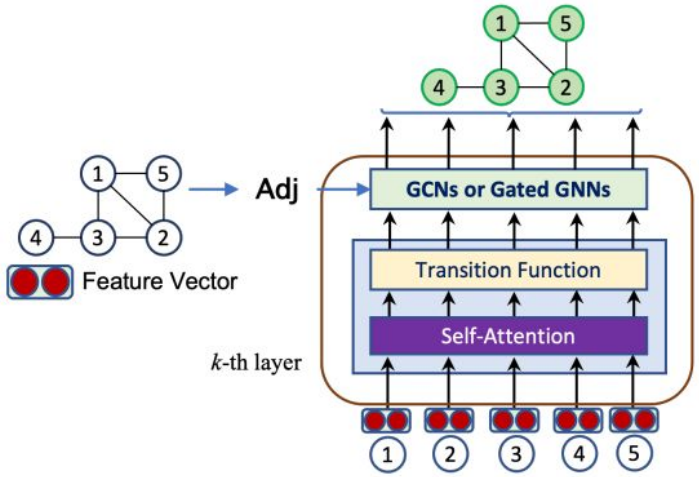
Molecular Graphs

Jongchan Kim

Motivation

Universal Graph Transformer Self Attention Network
On Mar 8, 2022.



Task	Dataset	Model	Metric Name	Metric Value	Global Rank	Result	Benchmark
Graph Classification	COLLAB	U2GNN	Accuracy	77.84%	# 17		Compare
Graph Classification	COLLAB	U2GNN (Unsupervised)	Accuracy	95.62%	# 1		Compare
Graph Classification	D&D	U2GNN (Unsupervised)	Accuracy	95.67%	# 1		Compare
Graph Classification	D&D	U2GNN	Accuracy	80.23%	# 12		Compare
Graph Classification	IMDb-B	U2GNN (Unsupervised)	Accuracy	96.41%	# 1		Compare
Graph Classification	IMDb-B	U2GNN	Accuracy	77.04%	# 7		Compare
Graph Classification	IMDb-M	U2GNN (Unsupervised)	Accuracy	89.20%	# 1		Compare
Graph Classification	IMDb-M	U2GNN	Accuracy	53.60%	# 6		Compare

Motivation

OGB dataset : Open Graph Benchmark (OGB) is a collection of realistic, large-scale, and diverse benchmark datasets for machine learning on graphs.

Leaderboard for ogbg-molhiv

The ROC-AUC score on the test and validation sets. The higher, the better.

Package: >=1.1.1

Rank	Method	Ext. data	Test ROC-AUC	Validation ROC-AUC	Contact	References	#Params	Hardware	Date
1	**PAS+FPs**	No	0.8420 ± 0.0015	0.8238 ± 0.0028	Xu Wang(4Paradigm)	Paper, Code	26,706,953	RTX3090	Feb 22, 2022
2	**HIG**	No	0.8403 ± 0.0021	0.8176 ± 0.0034	Yan Wang (Tencent Youtu Lab)	Paper, Code	1,019,408	Tesla V100 (32GB)	Dec 28, 2021
3	**DeepAUC**	No	0.8352 ± 0.0054	0.8238 ± 0.0061	Zhuoning Yuan (UIowa)	Paper, Code	3,444,509	Tesla V100 (32GB)	Oct 10, 2021
4	FingerPrint+GMAN	No	0.8244 ± 0.0033	0.8329 ± 0.0039	Jiaxin Gu	Paper, Code	1,444,110	Tesla V100 (32GB)	Jul 8, 2021
5	**Neural FingerPrints**	No	0.8232 ± 0.0047	0.8331 ± 0.0054	Shanzhuo Zhang (PaddleHelix & PGL)	Paper, Code	2,425,102	Tesla V100 (32GB)	Mar 15, 2021
6	Graphormer + FPs	No	0.8225 ± 0.0001	0.8396 ± 0.0001	Huixuan Chi (AML@ByteDance)	Paper, Code	47,085,378	Tesla V100 (32GB)	Aug 5, 2021

Dataset name : Molhiv

Task : Binary Classification

- Predict whether a molecule inhibits HIV replication or not
- Metric : ROC-AUC
  - A performance measurement for the classification problems at various threshold settings
  - Area under of ROC curve (TPR, FPR)

Use cases : Drug Discovery

- ML models has been widely used in drug discovery
- Predicting chemical, biological and physical characteristics of compounds
- Narrowing down candidates
  - Thousands of candidates : huge amount of expenses

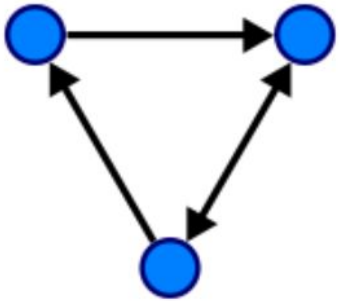
Dataset : A list of molecule**[Graph]** and ground truth

Q. What is Graph?

: Described by the set of vertices V and edges E it contains

- Vertices[Nodes] : **Atom**
- Edges : directed or **undirected : chemical bonds between atoms**

$$G = (V, E)$$



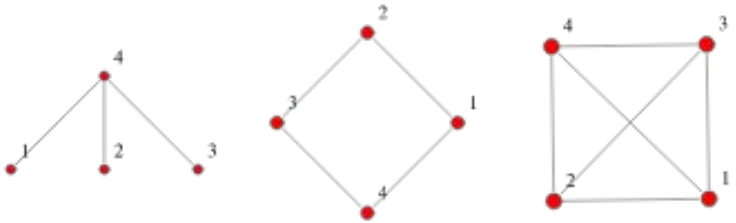
A Directed Graph (wiki)

Dataset : A list of molecule**[Graph]** and ground truth

Q. How to represent Connectivity of a Graph?

Adjacency Matrix

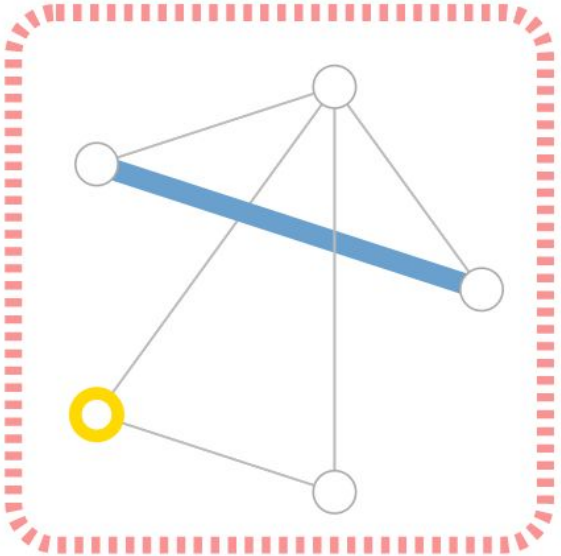
a matrix with rows and columns labeled by graph vertices, with a 1 or 0 in position according to whether and. are adjacent or not.



$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Dataset : A list of molecule**[Graph]** and ground truth

Q. How does graph data look like?
<u>Low dimensional vectors : embeddings</u>



Vertex (or node) embedding

Edge (or link) attributes and embedding

Global (or master node) embedding

Dataset : A list of molecule**[Graph]** and ground truth

- 41,127 Graphs
- Each graph has different number of atoms (nodes)
  - Handled using Zero padding
- Node feature : 9 dimensions
  - Atomic number, chirality, formal charge and so on.
- Edge feature : 3 dimensions
  - Bone type, bond stereochemistry and so on.

Method : (Embedding layer) - Self Attention + Graph Neural Network

- Embedding Layer
  1) Main Function : Embedding layer **enables us to convert each node /edge into a fixed length vector of defined size**
  2) Embeddings of nodes are used as inputs for the model

Method : Self Attention + Graph Neural Network

- Self Attention
  1) Main idea : allows the inputs to interact with each other and find out who they should pay more attention to.
  2) NLP vs Graph Molecule
     a) NLP : To understand interaction between words
     b) In molecule graph, To understand interaction between atoms

Method : Self Attention + Graph Neural Network

- Self Attention

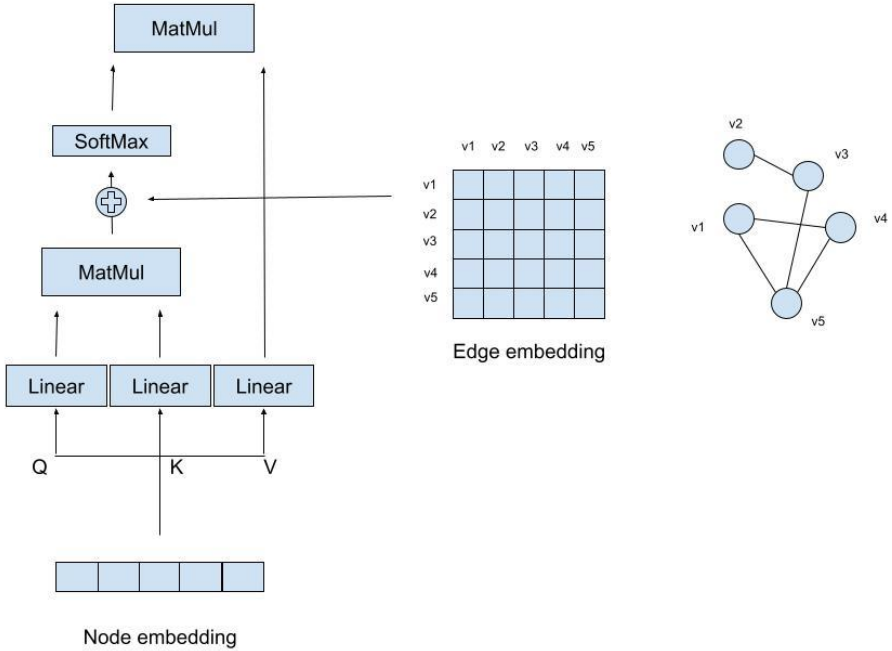
1) Node embeddings will be used to create key and query embeddings of each node

   -> Matmul(key, query)
   -> So far, we are not using edge information.
   -> Implicit relationship between every atom

2) Before doing softmax function on it, Add edge embedding adjacency matrix to it

   -> Explicitly utilize edge information of connected atoms

Method : Self Attention + Graph Neural Network



Method : Self Attention + Graph Neural Network

- Graph Neural Network
  1) Main idea : **nodes** can send and receive a **message** along with **edges(connections) with neighbors**
  2) Graphs provide a better way of dealing with abstract concepts like relationships and interactions.
  3) Goal : to learn a parametric mapping function that **embeds** nodes, subgraphs, or the entire graph **into low-dimensional continuous vector spaces.**

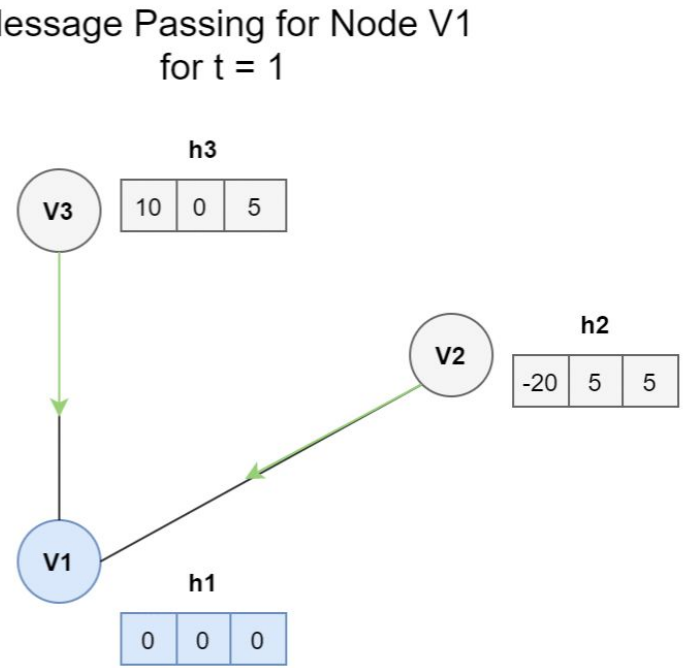
Graph Neural Network

- Each node in the graph has a hidden state -> **initialized using node embedding**

$$m_v^{t+1} = \sum_{w \in N(v)} h_w^t$$

$$h_v^{t+1} = average(h_v, m_v^{t+1})$$

ht - hidden state for each node



A very simple example of message passing architecture for node **V1**. In this case a message is a sum of neighbour's hidden states. The update function is an average between a message **m** and **h1**. Gif created by author

Graph Neural Network

- Creating a message between nodes

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

Obtaining the message from the neighbouring nodes. Modified from [3].

Average function could be used as a message function.
In this experiment Sum of hidden states of neighbors was used as my message function

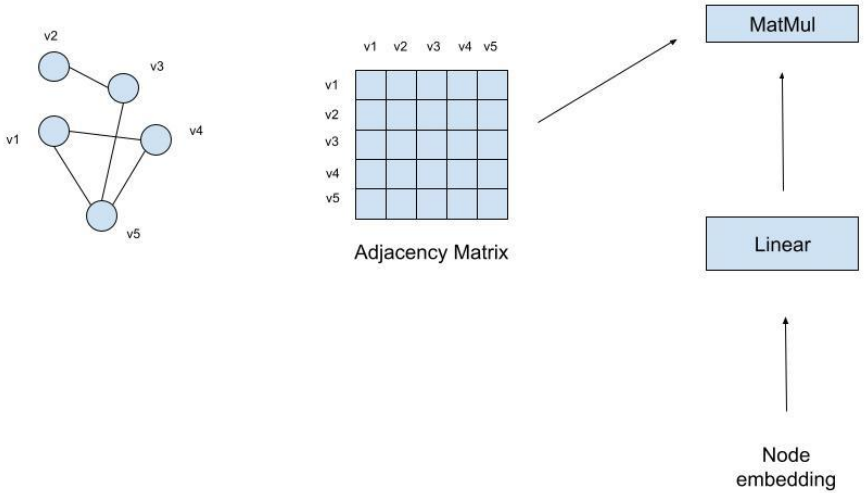
Graph Neural Network

- Updating hidden states based on the message and the old hidden states

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

- hidden state of the node Vt is obtained by
  - updating the old hidden state with the newly obtained message mv
  - **In this case,** the update function Ut is an **average** between the **previous hidden state** and **the message**

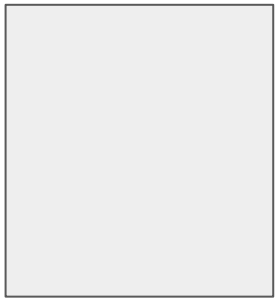
Graph Convolutional Neural (GCN)



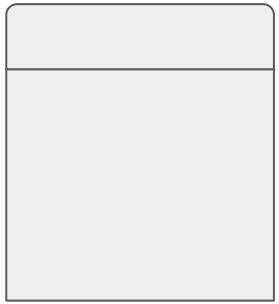
Design of Experiment

In Self Attention, we could understand the implicit relationship between every node even if it's not connected

In GNN, we have less strong process to utilize the implicit relationship between atoms that are not connected because we use only hidden states of neighbors.
But we can more strongly/directly use information about connected nodes



2 Encoder layers



1 Encoder layers + GNN

Results

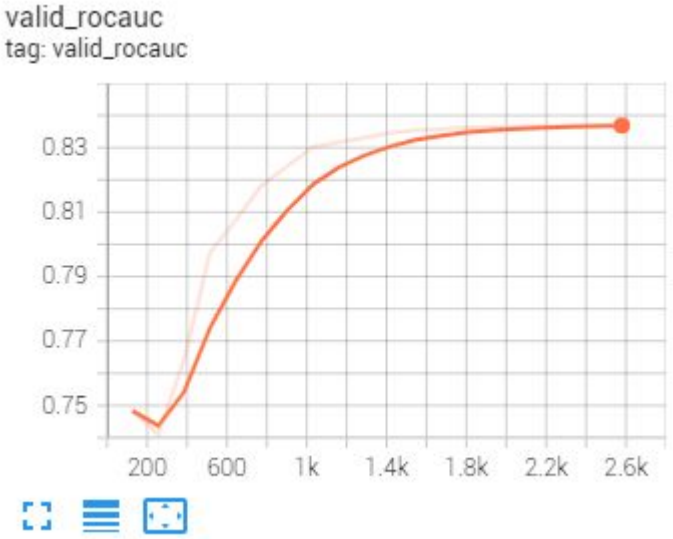
[Table. 1] Performance of models

	2 Encoder layers	1 Encoder + GNN
Test - ROCAUC	0.824	0.824
Valid - ROCAUC	0.837	0.839
Num of Parameters	532,418	456,074
Epochs	20	75

Results - Optimization

Validation ROC-AUC curve
: To see if model reaches model's maximum performance

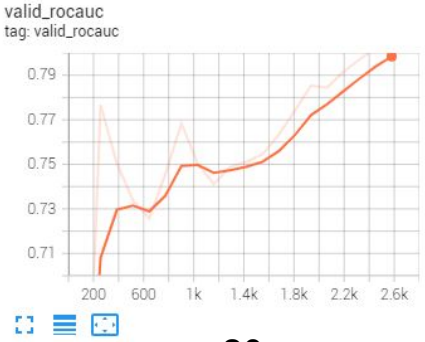
2 Self attention layer



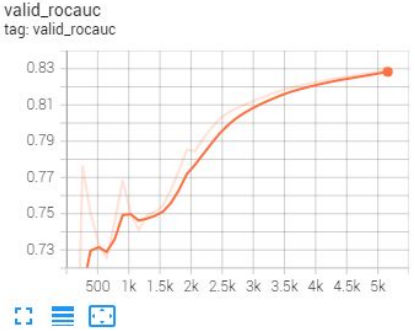
Results - Optimization

1 Self attention layer + 1 GNN layer

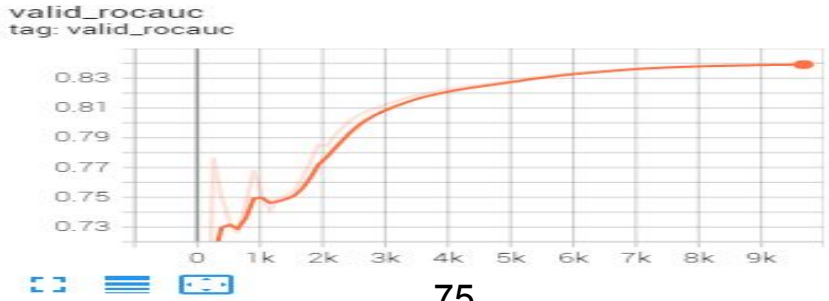
Epochs	Test ROCAUC
20	0.757
40	0.809
75	0.824



20



40

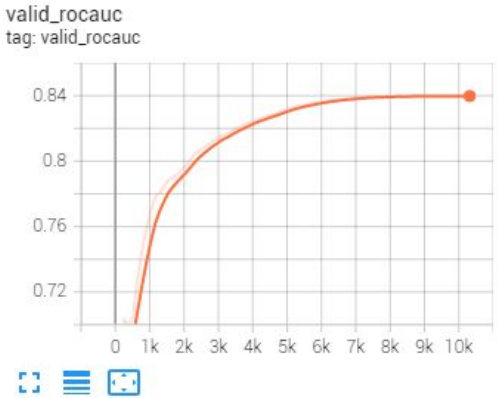


75

Results - Optimization

1 Self attention layer + 1 GNN layer

Hidden Dimension	Test ROCAUC	# of Parameters
64	0.824	456,074
256	0.825	2,043,914



Results

Rank	Method	Ext. data	Test ROC-AUC	Validation ROC-AUC	Contact	References	#Params	Hardware	Date
1	**PAS+FPs**	No	0.8420 ± 0.0015	0.8238 ± 0.0028	Xu Wang(4Paradigm)	Paper, Code	26,706,953	RTX3090	Feb 22, 2022
2	**HIG**	No	0.8403 ± 0.0021	0.8176 ± 0.0034	Yan Wang (Tencent Youtu Lab)	Paper, Code	1,019,408	Tesla V100 (32GB)	Dec 28, 2021
3	**DeepAUC**	No	0.8352 ± 0.0054	0.8238 ± 0.0061	Zhuoning Yuan (UIowa)	Paper, Code	3,444,509	Tesla V100 (32GB)	Oct 10, 2021
4	FingerPrint+GMAN	No	0.8244 ± 0.0033	0.8329 ± 0.0039	Jiaxin Gu	Paper, Code	1,444,110	Tesla V100 (32GB)	Jul 8, 2021
5	**Neural FingerPrints**	No	0.8232 ± 0.0047	0.8331 ± 0.0054	Shanzhuo Zhang (PaddleHelix & PGL)	Paper, Code	2,425,102	Tesla V100 (32GB)	Mar 15, 2021

Discussion

- This experiment has shown that using GNN on top of the self-attention encoder layers could work as a more efficient method for a graph dataset by providing the same performance with a lower number of parameters. However, this idea needs to be further proved with future studies.
- Also, as a smaller number of parameters is used, this model architecture has one of the possible advantages in minimizing overfitting problems due to reduced model complexity.

Resources

1. A gentle introduction to Graph Neural Network https://distill.pub/2021/gnn-intro/

2. Hu, Weihua, et al. "Open graph benchmark: Datasets for machine learning on graphs." *Advances in neural information processing systems* 33 (2020): 22118-22133.

Thank you