GMAN and bag of tricks for graph classification

Hao Zhang Jiaxin Gu Pengcheng Shen

Abstract

In this technical report, we present our solutions of several OGB graph classification tasks. For ogbn-ppa, we adopt ExpC [1] as our basic model, and with bag of tricks we get the state of the art. For ogbn-molhiv and ogbn-code2, we propose a new message function with multi-head attention which achieves good performance[1].

1 Introduction

Machine Learning on graphs has attracted immense attention in recent years. OGB provides graph datasets that cover important graph machine learning tasks, diverse dataset scale and rich domains. OGB Dataset cover three fundamental graph learning task categoties: predicting the properties of nodes, links and graphs. For the graph property prediction task, it provides four datasets: ogbn-molhiv, ogbg-molpcba, ogbn-ppa, ogbn-code2. This report mainly talks about the graph property prediction task, and we propose a new method Graph Multi-head Attention Neural Network (GMAN) with bag of tricks.

2 Methods

2.1 Bag of tricks

We replace ReLU by PReLU on the baseline method ExpC, which we call it ExpC* , and bag of tricks includes PreLU, adamW, cosine learning rate, warm up strategy, virtual node [2], flag [3] and temperature:

$$out_i = \frac{\exp\left(y_i/T\right)}{\sum_i \exp\left(y_i/T\right)} \tag{1}$$

2.2 GMAN

We propose GMAN(Graph Multi-head Attention Neural Networks) which differs in the message function from GIN [2]. We consider 2 nodes with 1 edge as a sequence (sequence length is three). Then we use a multi-head attention to obtain the output of 2 nodes and 1 edge. Finally we try 2 types of message function, one is that output of neighbor node is the message (type A), and the other one is the sum of the output of neighbor node and edge which is as shown in Figure 1 (type B).

[1]https://github.com/PierreHao/YouGraph

$$\boldsymbol{m}_{uv}^{(t)} = \mathrm{MHA}\left(\boldsymbol{h}_u^{(t-1)}, \boldsymbol{h}_v^{(t-1)}, \boldsymbol{e}_{uv}^{(t-1)}\right)\tag{2}$$

$$\mathbf{h}_u^{(t)} = \mathbf{MLP1}^{(t)}\Big(\sum_{v \in \mathcal{N}(u)} \mathbf{m}_{uv}^{(t)}\Big) + \mathbf{MLP2}^{(t)} h_u^{(t-1)}\tag{3}$$

where MHA is multi-head attention, u is the target node, and v is one of the neighbor nodes, e is the edge between u and v.

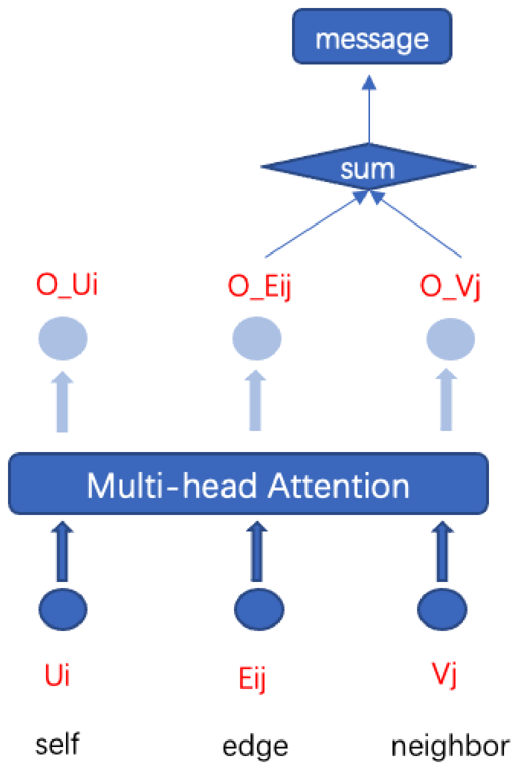

Figure 1: Multi-head Attention Convolution

3 Experiments and Results

All the final results are obtained by 10 different models with 10 different random seeds.

3.1 ogbn-ppa

In this task, we have tried bag of tricks which could improve the performance of GNN on the task graph classification. In order to test more quickly, we adopt a small model for these experiments, which is a 3 layers neural network with 128 hidden embedding dimension and they are trained with 60 epochs. The baseline is the official code of ExpC. In fact, train a large model will costs one week, it's very expensive.

methods	test accuracy	val accuracy	train accuracy
baseline	0.7065	0.6646	0.9090
ExpC*(PReLU+AdamW+CosLR)	0.7271	0.6885	0.9764
ExpC*+T	0.7424	0.7015	0.9852
ExpC*+T+Warmup	0.7451	0.7062	0.9892
ExpC*+T+Warmup+flag	0.7614	0.7193	0.9886
ExpC*+T+Warmup+flag+vn	0.7758	0.7343	0.9855

Table 1: experiments for bag of tricks on a small model

Then we have tried all the tricks on a large model, the result and the parameters are showed as follow, and we win first place.

	ExpC*
hidden dimension	512
Attention Heads	16
initial lr	0.0005
weight decay	0.00001
warmup epochs	20
layers	3
JK	cat
Temperature	10
virtual node	true
flag	true
pooling	add
epochs	200
Dropout	0.5

Table 2: Hyper-parameters of the final result

methods	test accuracy	val accuracy
ExpC	0.7976±0.0072	0.7518±0.0080
DeeperGCN+FLAG	0.7752±0.0069	0.7484±0.0052
ExpC* + bag of tricks	0.8140±0.0028	0.7811±0.0012

Table 3: final result on ogbn-ppa

3.2 ogbn-code2

We use GMAN(type B) for ogbn-code2, and with bag of tricks (T+Warmup+vn) we win first place, details as follow:

	GMAN typeB
hidden dimension	512
Attention Heads	16
initial lr	0.001
weight decay	0.00005
warmup epochs	10
layers	3
JK	cat
Temperature	10
virtual node	true
flag	false
pooling	mean
epochs	50
Dropout	0.5

Table 4: Hyper-parameters

methods	test accuracy	val accuracy
DAGNN	0.1751±0.0049	0.1607±0.0040
GCN+vn	0.1595±0.0018	0.1461±0.0013
GMAN	0.1770±0.0012	0.1631±0.0090

Table 5: result of ogbn-code2

3.3 ogbn-molhiv

OGBN-molhiv is the smallest dataset of the four graph classification tasks. We get a confusing phenomenon on this task. With GMAN(type A), we find that the performance of the 10 models has a large standard deviation (from 78.5 to 82.5). With some tricks, train and validation accuracy can be improved by 2%, but test accuracy is decreased by 2%. Now, the result is 0.8043 which is only third place, and we will keep working on this task to further improve the accuracy. I think if we solve the stability problem, maybe we will be ranked second or even first.

	GMAN typeA
hidden dimension	128
Attention Heads	16
initial lr	0.0001
weight decay	0.0
warmup epochs	0
layers	3
JK	cat
Temperature	1
virtual node	false
flag	false
pooling	mean
epochs	50
Dropout	0.0

Table 6: Hyper-parameters

methods	test accuracy	val accuracy
Neural FingerPrints	0.8232±0.0047	0.8331±0.0054
MorganFP+RandForest	0.8060±0.0010	0.8420±0.0030
GMAN	0.8043±0.0099	0.8011±0.0079

Table 7: final result on ogbn-molhiv

References

[1] Yang, M., Shen, Y., Qi, H. & Yin, B. Breaking the expressive bottlenecks of graph neural networks. *arXiv preprint arXiv:2012.07219* (2020).

[2] Xu, K., Hu, W., Leskovec, J. & Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[3] Kong, K. *et al.* Flag: Adversarial data augmentation for graph neural networks (2020). 2010.09891.