DSSV 2022
Data Science, Statistics & Visualisation

Training Graph Neural Networks via Self-Supervised Learning: Experiments and Analysis

Ho, Ching-Ru (r09946006@ntu.edu.tw)
Data Science Degree Program, National Taiwan University, Taiwan.

Introduction

Learning from unlabeled data has become popular in machine learning recently. One of such learning idea is self-supervised learning (SSL), which trains models by leveraging information from data themselves without guidance from labels or other external information.

Generally, self-supervised learning methods can be grouped into four approaches: contrastive learning, clustering, distillation, and redundancy reduction. These approaches are widely using in computer vision and natural language processing, and outperform other supervised models in many benchmark tasks. However, self-supervised learning still remains a challenge in graph representation learning.

In this work, we will apply three self-supervised learning approaches to training models on graph data, and conduct simulation experiments by training our models on four real-world datasets under various experimental settings. These experimental settings included data augmentation methods, different number of layers, various batch sizes, and so on. Besides, we will briefly describe possible explanation of results about self-supervised learning in graph neural network (GNN).

Methodology

Dataset (via TUDataset)	• Molecules: MUTAG (188 graphs), NCI1 (4110 graphs) • Bioinformatics: PROTEIN (1113 graphs), DD (1178 graphs)
Self-supervised Learning Approach	• Contrastive Learning: SimCLR • Distillation Learning: Simsiam • Redundancy Reduction: Barlow Twins
Data Augmentation Operator	• Node dropping: Randomly delete node and its connections. • Edge perturbation: Randomly delete or add new edges. • Attribute masking: Randomly substitute the attribute of node. • Subgraph: Randomly sample the subgraph component. • (with ratio 0.3)
Mini-batch Size	• 64, 256 (for MUTAG, PROTEIN, NCCI1) • 64,128 (for DD)
Hidden Dimension	• 64, 512
Encoder	• Encoder Type: Graph Isomorphism Network (GIN) • Number of Layer: 1 (monolayer), 2 (bilayer), 3 (trilayer)
# Projector Layer	• 3 (trilayer)
Epoch	• 200 times
Data Proportion	• 90% used in self-supervised (to train the encoder) • 10% used in supervised (for validation and test)

Results and Discussion

■ Batch size's effects on SimCLR are not apparent

Though the author of SimCLR found that the batch sizes are positively correlated with performance. However, in Figure 1, we can observe that the performances between two batch sizes: 64 and 256 under SimCLR are compared, is not significant.

We can infer that the advantage of batch sizes in contrastive learning, especially in SimCLR, may only be appreciable when the dataset is of large size. If the dataset is medium or small size, it seems that increasing the batch size may not be effective.

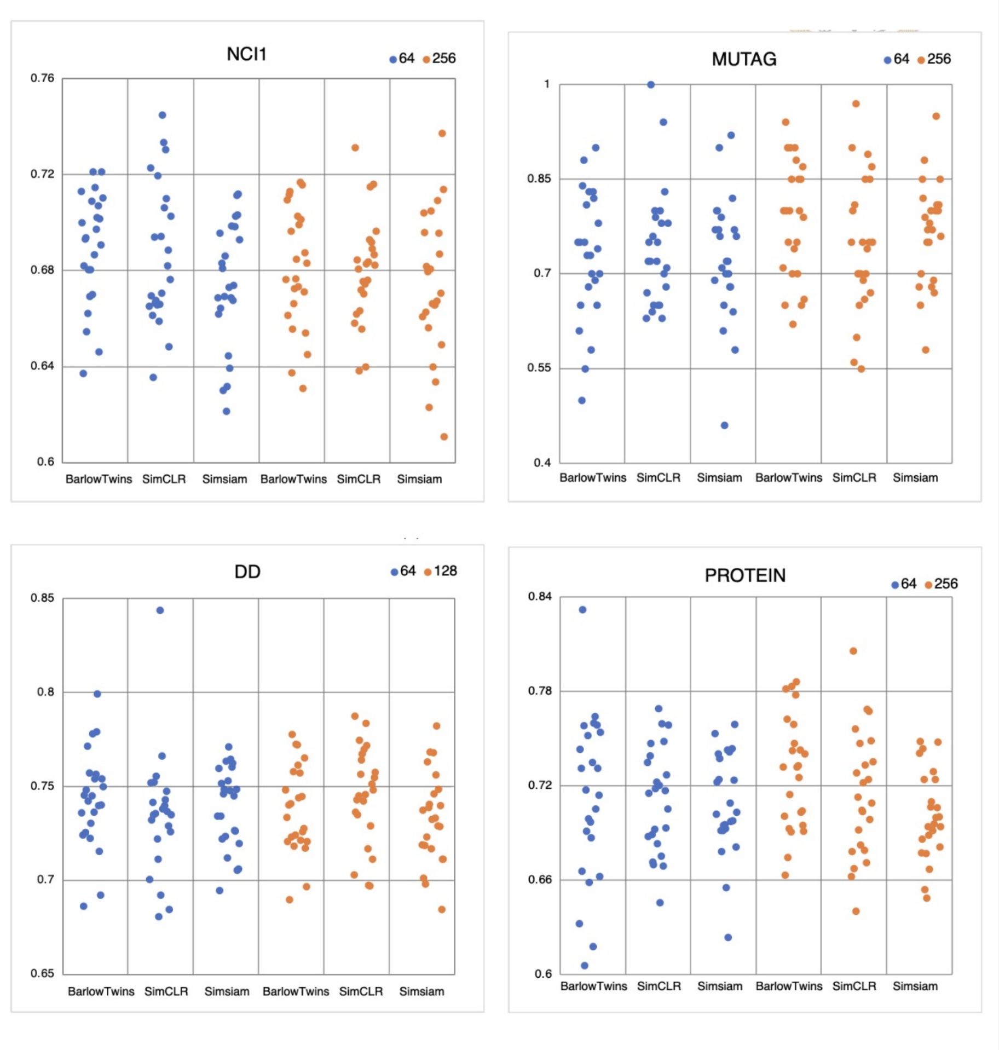

Figure 1. Evaluation under different batch sizes.
Blue and orange dots represent different evaluations under two batch size, 64 and 256 (in DD, we set to 128). Models from left to right in each dataset represents Barlow Twins, SimCLR and Simsiam.

■ Hidden dimension has little effects

In most SOTA models, the hidden dimension of the encoder layer plays an influential role. However, in Figure 2, we can observe that even the hidden dimension size is 8 times larger, the performance does not have significant improvements.

For an image data, the neuron can capture more information via pixels clustering. Nevertheless, graph data are more abstract. Nodes and edges cannot be separated or unified like pixel groups in an image. Thus, increasing hidden dimension is less effective while training the graph data.

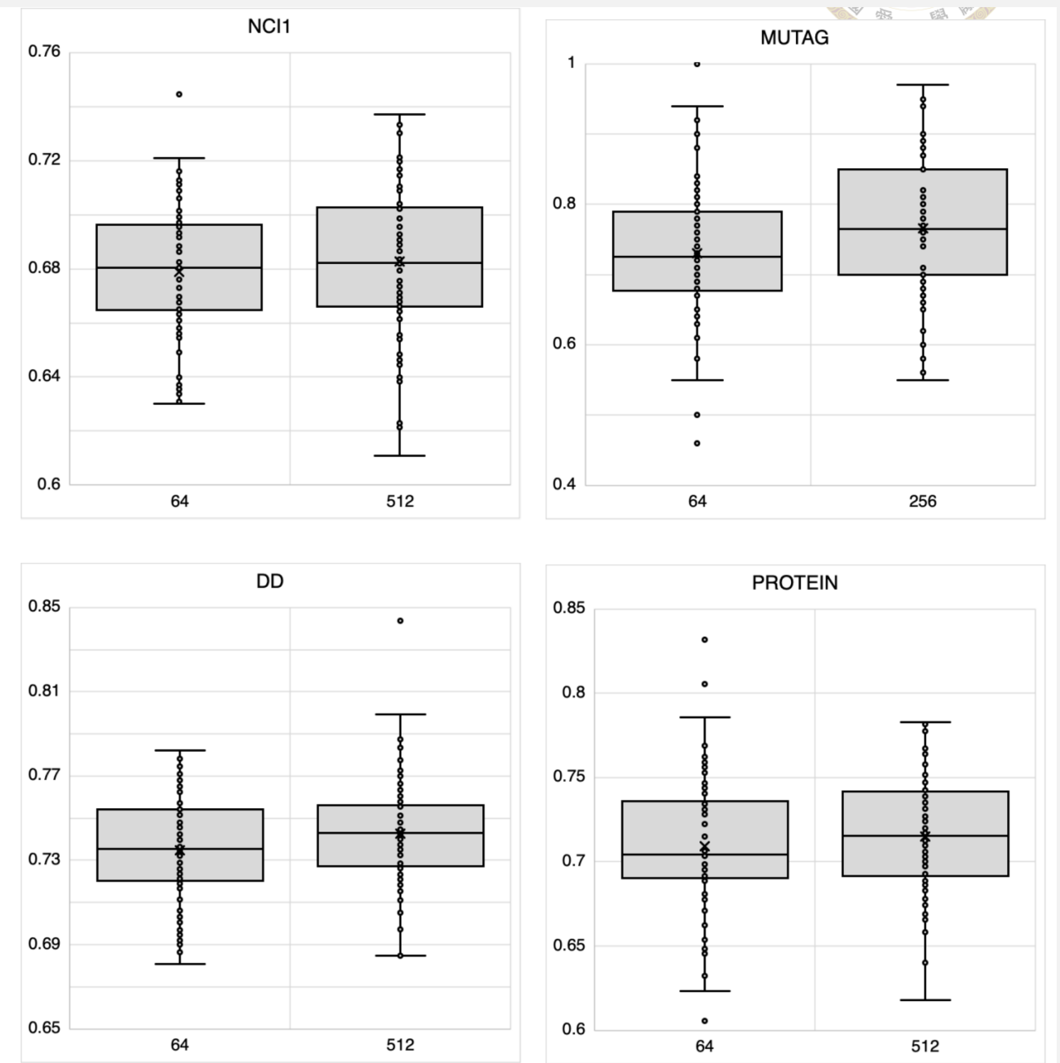

Figure 2. Evaluation under different hidden dimension.
The graph shows the performance that under hidden dimension equals to 64 and 512. Though the latter is 8 times to the former, the performance still keeps in the same level.

■ Deeper encoders have better performance

Generally, models using a deeper encoder are believed to have better performance than shallow one. In our experiment, the result has verified that the deeper encoder also benefits on graph dataset.

Figure 3 shows the evaluation under different depths of layer from monolayer, bilayer and trilayer, during the training stage. The future self-supervised methods should be concentrate on the deeper architecture.

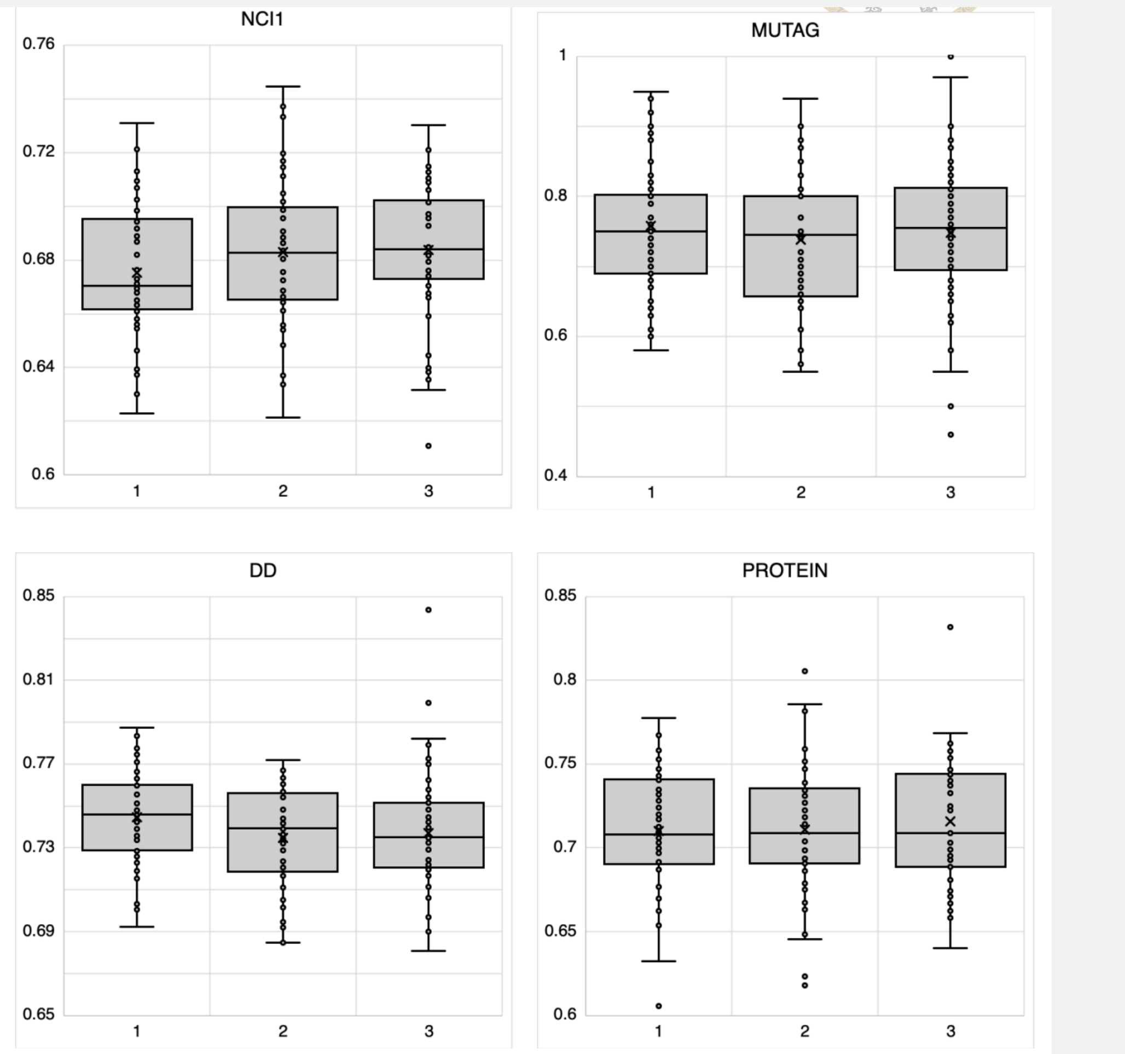

Figure 3. Evaluation under different depths of layer.
The graph shows the performance that under different depths of layer. We can observe that while the layer become deeper, the performance is better generally.

■ Subgraph performs stable in bioinformatics dataset

In spite of the fact that these data are graph-structured, there are fundamental differences between molecular and bioinformatics-related graph data. Both types of datasets have different types of properties, which could have a complex influence on self-supervised models.

In a chemical molecule, each node and each edge play unique roles. The functional group might be lost if we only capture a part of the graph via Subgraph operator. For example, the chemical formula of alcohol is C2H5OH, containing a hydroxyl group (-OH). If the augmented operator samples C2H4, it will be recognized as ethene, which has different chemical properties than ethanol. As the augmented data have more diversity, so the performance of the model, which will have a higher variance.

In contrast, a protein contains at least one long polypeptide, which is chained by amino acid. Proteins can be constructed in more flexible ways. Some proteins chained by different polypeptides can have similar chemical property and function.

Therefore, even when we apply the Subgraph operator on bioinformatics datasets, the augmented data are alike to the raw data, making the variance between experiment results smaller.

Discussion

We have systematically investigated three self-supervised learning methods by applying them to train neural networks on graph data. First of all, increasing the batch size in training when the dataset is of medium or small size is not helpful, but deepening the layer of the encoder seems to be a favorable way for improving model performance. Secondly, graphs have their own special structure. Following what have done by models trained on image datasets, e.g. increasing hidden dimension, may not be inappropriate.

Researchers should be focus on other experimental factors. Finally, due to the characteristic differences between molecular and bioinformatics data, different optimal hyperparameters should be applied when training models on the two kinds of data. For example, Subgraph operator might be useful in chemical dataset due to the fact that it could generate more various samples. The future self-supervised learning approaches can apply our discovery to save training time and adjust their models for better evaluation.

References

- T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607. PMLR, 2020
- Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen. Graph contrastive learning with augmentations. Advances in Neural Information Processing Systems, 33:5812–5823, 2020.
- X. Chen and K. He. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 15750– 15758, June 2021.
- J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021