GraphMLP

mohammad.shafkat.hasan@g.bracu.ac.bd
Switch account

Draft saved

* Indicates required question

Email *

Record **mohammad.shafkat.hasan@g.bracu.ac.bd** as the email to be included with my response

Student ID *

19101077

Course *

- [x] CSE 425
- [ ] EEE 474

Which parameters are GNN (Graph NN) depends on? Explain with example

Graph-MLP, a novel framework for node classification without message passing in graph neural networks (GNNs). It introduces a pure multilayer-perceptron-based model and a neighboring contrastive loss to learn node representations from graph structure. GNNs are a class of models that use adjacency matrix and node features as inputs to propagate features among neighbors. GNNs depend on parameters such as the weight matrix of each layer, the activation function, the normalization method, and the message passing module. For example, GCN is a type of GNN that uses graph convolution as the message passing module, which sums up the neighbor features to obtain the new node feature. GAT is another type of GNN that uses graph attention as the message passing module, which learns attention values among connected nodes to adaptive pass the message and update node features.

What are the advantage of Graph-MLP over Graph Neural Networks? Explain with your own words

Graph-MLP has two advantages over Graph Neural Networks (GNNs):
- Higher computation efficiency: Graph-MLP does not require the adjacency matrix or the message passing step in feed-forward propagation, which reduces the complexity and time consumption of the model. Graph-MLP also allows batch training without the full graph information, which is not possible for GNNs.
- Robustness against corrupted edges: Graph-MLP does not rely on the adjacency matrix in inference, which means it can still provide consistent results even when some edges are missing or noisy in the graph. GNNs, on the other hand, are sensitive to the changes in the graph structure and may suffer from performance degradation.

Graph-MLP achieves these advantages by using a pure MLP-based model structure and a novel Neighbouring Contrastive (NContrast) loss, which implicitly incorporates graph structure into node representation learning. The NContrast loss treats r-hop neighbors as positive samples and other nodes as negative samples, and encourages the model to draw positive samples closer and push negative samples farther in feature space. This way, Graph-MLP can learn a graph-structured distribution during node feature transformation without explicit message passing modules.

What do you understand by Contrastive Learning? Show some example?

Contrastive learning is a type of unsupervised learning that learns representations by contrasting positive and negative pairs of examples. The goal is to learn a representation space where similar examples are close together and dissimilar examples are far apart. Contrastive learning has been used in many applications such as image recognition, speech recognition, and natural language processing.

Here is an example of contrastive learning in image recognition: Given two images, one of a cat and one of a dog, the goal is to learn a representation space where the cat image is closer to other cat images than to dog images, and the dog image is closer to other dog images than to cat images. This can be achieved by training a neural network to predict whether two images are from the same class or not. If two images are from the same class, they are considered positive pairs, and if they are from different classes, they are considered negative pairs. The network learns to map positive pairs close together and negative pairs far apart in the representation space.

What datasets the authors used for the experiments? Make a list of it with the accuracy they gained

The authors of the paper evaluated the performance of Graph-MLP on three popular citation network datasets in node classification: Cora, Citeseer, and Pubmed. Here is a list of the datasets along with the accuracy achieved by Graph-MLP:
1) Cora: 79.5%
2) Citeseer: 73.1%
3) Pubmed: 79.7%
These accuracy values represent the test accuracy (%) achieved by different models on the respective citation network datasets. The scores are obtained by averaging over 10 runs for each experiment.
These results show that Graph-MLP can achieve comparable or even superior performance against state-of-the-art models in the graph node classification task.

Describe the the effects of hyperparameters in Graph-MLP

The effects of hyperparameters in Graph-MLP are described as follows:

1. α (Alpha) Parameter: The α parameter in Graph-MLP controls the strength of the contrastive loss used in the training process. The authors observed that as the value of α increases from 1 to 10, the accuracy consistently improves on the Cora and Pubmed datasets. However, when α is set to 100, the performance on Cora and Pubmed drops slightly. On the Citeseer dataset, increasing α from 1 to 100 leads to continuous improvement in performance, with the best performance achieved around 100 .

2. Learning Rate: The learning rate is an important hyperparameter that determines the step size at which the model updates its parameters during training. The authors found that a learning rate of 0.01 usually performs well in Graph-MLP experiments .

3. Weight Decay: Weight decay is a regularization technique that adds a penalty term to the loss function to prevent overfitting. The authors observed that a weight decay of 5e-3 performs slightly better than 5e-4 in Graph-MLP experiments. This suggests that Graph-MLP may benefit from a stronger regularization .

These hyperparameters play a crucial role in determining the performance and behavior of Graph-MLP. The α parameter affects the strength of the contrastive loss, while the learning rate and weight decay influence the optimization process and regularization. By tuning these hyperparameters, researchers can optimize the performance of Graph-MLP on different datasets and tasks.