Homework 2: Graph Classification on MUTAG Dataset

Marija Zelic
Section of Life Sciences Engineering

***Abstract*—The aim of this homework is to explore different aspects of graph-based machine learning by implementing various Graph Neural Network architectures and comparing how they perform on the graph classification problem. The final result of this homework yields 0.82 accuracy and 0.88 F1 score.**

I. Introduction

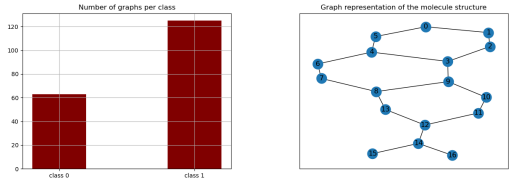
According to the paper [1], the connection can be established between the mutagenicity of nitroaromatic substances on the *Salmonella typhimurium* and their chemical structure. Hence, we present Deep learning algorithms that will exploit the given features of the molecules and successfully predict their mutagenicity.

II. Methods and Models

We approach this problem by first looking at the given dataset and preparing it for the Deep learning algorithms and then building, training, and selecting the best model by hyperparameter tuning for the maximization of the classification accuracy.

A. Dataset and preprocessing

The MUTAG dataset is a collection of 188 nitroaromatic components represented as a graph. The vertices of the graph stand for the atoms of the molecule, each labeled by the atom type in one-hot encoding notation. Edges between the vertices represent bonds between the corresponding atoms, also labeled by the bond type in one-hot encoding notation. There are 7 discrete node labels and 4 discrete edge labels. Each graph is labeled by its mutagenicity (0 - not mutagenic, 1 - mutagenic). We can see the distribution of the graphs per class in Figure 3a, leading us to the conclusion that classes are imbalanced. Figure 3b displays the structure of one of the molecules in the dataset.
The whole dataset is partitioned on training, validation, and test subsets in 70%, 15%, and 15% ratios, respectively. The approach for fragmenting the dataset is essential, as it can lead to large fluctuations in observed metrics. The strategy that was used is *stratified split*, which preserves the ratio of the classes in the whole dataset same in every subset. This will lead to more consistent metrics on the test set and consequently model that is less biased towards one (majority) class.
Approaches such as dataset undersampling to achieve balanced classes and weighting the loss functions were not giving adequate results.



(a) Number of samples in dataset per single class. (b) Graph representation of one molecule in the dataset.

Fig. 1: Classes ration (left) and one molecule example (right).

B. Model selection and evaluation

We implement and test 3 main Graph Neural network architectures that are built upon 3 distinct types of layers: Normal Convolution, GraphSAGE, and Attention-based layers. Each of these models receives several different hyperparameters, Table I. Also, other strategies were explored, such as including edge features in the existing layers. To maximize classification accuracy, hyperparameter tuning was performed for each of the network architectures, choosing the hyperparameters that have the best accuracy on the validation set. Besides the accuracy, the observed metrics were loss and F1 score.

III. Experiments and discussion

A. Without Edge Features

In Table I, we report the parameters of each GNN architecture that result in the best accuracy on the validation set when only node features are incorporated. In Table II, there are all the calculated metrics for the best hyperparameters from Table II. As it can be inferred from the tables, all of the architectures have more or less similar metrics on both validation and test subsets, with GraphSAGE having the best test accuracy.

Layer type	Hidden layer	Drop. prob.	LR	Epochs	Pooling	Aggreg.
GCN	[16, 16, 16]	0.	1e-4	100	Max	
SAGE	[16,16]	0	1e-4	100	Max	Mean
Atten.	[8,8]	0	1e-3	100	Max	

TABLE I: Parameters for which the best validation accuracy is obtained.

Layer type	Val. accuracy	Val. loss	Val. F1 score	Test accuracy	Test loss	Test F1 score
GCN	0.72	0.59	0.8	0.78	0.57	0.85
SAGE	0.72	0.57	0.8	0.82	0.55	0.88
Atten.	0.79	0.68	0.84	0.75	0.5	0.82

TABLE II: Metrics on validation and test data for best hyperparameters.

B. With Edge Features

We attempted to incorporate edge features into the existing GNN architectures and inspect how the updated models perform. The two approaches were taken when trying to incorporate edge features.

1) Weighing adjacency matrix: Weighting the adjacency matrix is a common approach for incorporating information about edges in the GNN. The are 4 different types of bonds represented with one-hot encodings. The adjacency matrix was weighted in such a manner that edge features that were most common (the ones encoded with $[1, 0, 0, 0]$) were given the weight of 0.25 and the rarest ones ($[0, 0, 0, 1]$) were given the weights 1. The rest of the edges were weighted with 0.5 and 0.75. Weighting the adjacency matrix in such a demeanor will give greater importance to the edges that are unusual and supposedly specific to the molecule.

2) Incorporating edge features in Attention-based layer: The linear layer with trainable parameters is applied to the edge features, similar to the transformation performed on the node features in the first step of the "basic" Attention-based layer. Following this step, transformed corresponding edge features are concatenated to the root node and its neighboring nodes and used for the computing attention scores. This approach is not tackled in the scientific literature but is part of the Attention-based layer implementation in the PyG library.
The results after incorporating weighted matrices in GCN and GraphSAGE architectures and edge features to the Attention-based layer are presented in the following tables, Table III and Table IV.
By observing the tables, the accuracies of the GCN and Attention-based architectures reached the highest mark of 82%. On the other hand, the weighted adjacency matrix did not yield improvement in the accuracy of the GraphSAGE architecture.
The peak of the accuracy across all architectures is reached at 82% on the test subset. If we consider that the dataset is small and classes are imbalanced, this is quite a pleasant result.

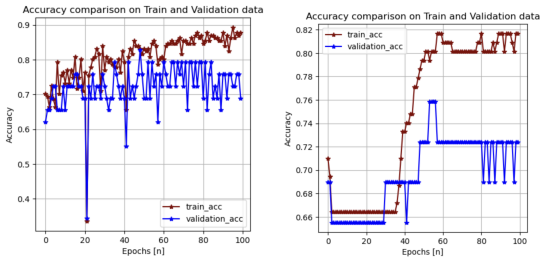
For every configuration of hyperparameters, there are plots that compare metrics on the train and validation subsets. Such plots, for the Attention-based layer with and without incorporated edge features are displayed in Figure 2. Plots demonstrate the difference between achieved accuracy and improvement in results with edge features included.
Figure 3 depicts examples of proper classification and misclassification, where different edges and features are marked with distinct colors. For future updates, this could possibly be an important tool for deriving conclusions regarding the molecule structure that fails to generalize well.

Layer type	Hidden layer	Drop. prob.	LR	Epochs	Pooling	Aggreg.
GCN	[16, 16]	0.	1e-3	100	Max	
SAGE	[16,16]	0	1e-3	100	Max	Mean
Atten.	[16,16] e. [8,8]	0	1e-3	100	Max	

TABLE III: Parameters for which the best validation accuracy is obtained using edge features.

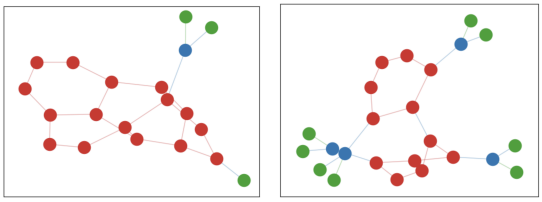
Layer type	Val. accuracy	Val. loss	Val. F1 score	Test accuracy	Test loss	Test F1 score
GCN	0.79	0.68	0.85	0.82	0.44	0.87
SAGE	0.75	0.67	0.82	0.78	0.58	0.85
Atten.	0.79	0.54	0.83	0.82	0.5	0.83

TABLE IV: Metrics on validation and test data for best hyperparameters using edge features.



(a) With Edge Features. (b) Without Edge Features.

Fig. 2: Comparison of accuracy of Attention-based layer on train and validation subsets.



(a) Right classification. (b) Wrong classification.

Fig. 3: Examples of right and wrong classification.

IV. Conclusion

This homework unambiguously proves the strength and merit of GNN architectures in graph classification problems, even on small and unbalanced datasets. It also amplifies the importance of including all available data, meaning both node and edge features, for achieving better results.

REFERENCES

[1] C. Hansch, "Structure-activity relationships of chemical mutagens and carcinogens," *Science of The Total Environment*, vol. 109-110, pp. 17–29, 1991, qSAR in Environmental Toxicology. [Online]. Available: https://www.sciencedirect.com/science/article/pii/004896979190167D