

Report: Graph Neural Networks

I used the PPI (Protein-Protein Interaction) dataset for training and testing the model. I started implementing the model by first writing the GAT Layer. I wrote the code for the GAT Layer with the help of the paper as well as an example that I saw on the official website which used Message Passing to implement the GCNConv layer; I followed a similar procedure.

The final model consists of a series of GAT layers, with number of layers being an additional parameter to the model other than the input, hidden and output channels. I also included the negative slope of the LeakyReLU layer as a parameter to be provided.

The specifications of the model include `num_layers = 3`, `hidden_channels = 64`, with the optimizer being Adam with a learning rate of 0.005. The loss used is the Binary Cross Entropy with Logits. The reason behind using the loss was the absence of a sigmoid layer at the output layer in the model's architecture.

Before implementing the custom GAT layer, I tried verifying my pipeline by making the GAT using existing PyG implementation and got a high accuracy (about 94 percent). I have commented out the code for that.

Initially, I took the first 1000 data points of the PPI dataset as my dataset for training and testing and trained the model for 100 epochs. After training the model, the testing resulted in an accuracy of 91.95 percent.

After that, I used the entire PPI dataset to check the model's performance.

For `Num_layers= 3`, `Hidden_channels = 64`, Accuracy = 91.33 percent

I tried experimenting with different hyperparameters to check whether better performance was achievable. Doubling the learning rate gave me 91.66 accuracy, better than the previous as I found the loss to be decreasing slowly; thus increasing the learning rate helped.

For completion, I also halved the learning rate, I got the accuracy = 90.99 percent, thus the results were similar.

I tried to increase the number of layers and change the `hidden_channels` as well. The results were as follows:

`Num_layers = 4`, `Hidden_channels = 64`, Accuracy = 91.37

`Num_layers = 3`, `Hidden_channels = 32`, Accuracy = 91.04

If I had extra time for the assignment, I would have tried testing it for a transductive dataset such as Cora. I would also want to finetune my hyperparameters like the hidden channels, `num_layers` and the learning rate. I would like to keep improving the output of this model even after the deadline is over; this was a good learning experience for me.