Task 2: Graph Classification

Majd Bishara, Saleem Kheereldeen, Rani Khirbawi

Method:
Our methodology was straightforward: we developed dataloaders for both training and validation purposes. Utilizing cross-entropy loss as our loss function, we observed a class imbalance in the data, with the number of "ones" being twice the number of "zeros." To address this issue, we introduced a weight parameter into the loss function. Subsequently, we initiated a cross-validation process to evaluate various architectures and their corresponding parameters.

Architectures:

In our project, we conducted experiments utilizing the following architectures:

- Graph Attention Network (GAT), as presented in the course material.
- Graph Sample and Aggregate (GraphSAGE).
- Gaussian-enhanced GAT.
- Gaussian-enhanced GraphSAGE.

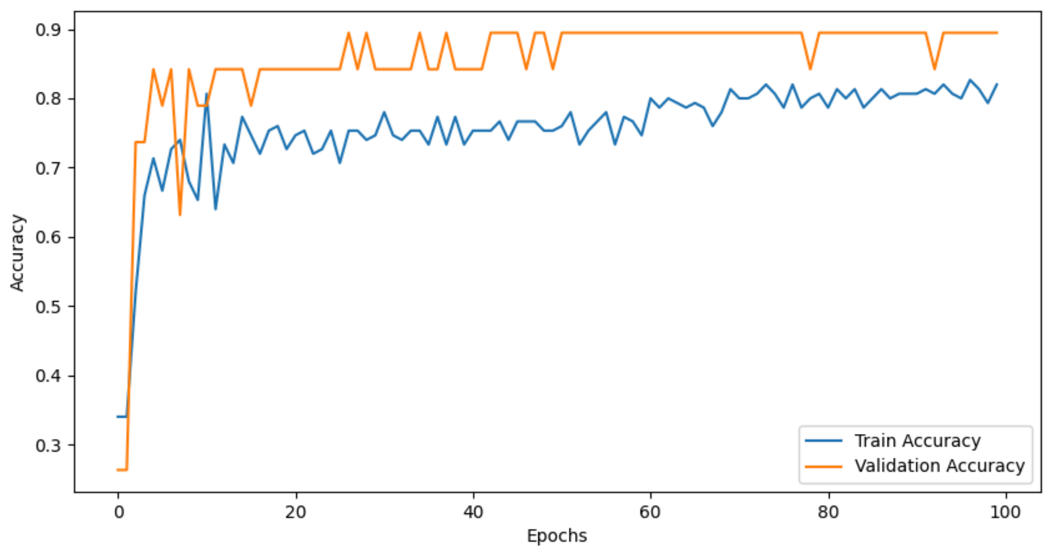
Gaussian models introduce the concept of uncertainty by learning Gaussian embeddings for nodes. Rather than relying on fixed embeddings, each node is represented by a mean and variance. This approach empowers the model to capture inherent uncertainty in node representations. To facilitate sampling of embeddings, the reparameterization trick is employed. Additionally, KL divergence regularization is applied to discourage excessively large variances, thus promoting the model's ability to generate confident predictions.

Experiments:
In our experimentation, we applied cross-validation across a comprehensive set of hyperparameters to identify the optimal configuration for model performance. We systematically varied the learning rate (lr), weight decay, batch size, number of layers (n_layer), hidden units in the aggregation layers (agg_hidden), and fully connected layers (fc_hidden). Additionally, we experimented with three options for the global pooling method (agg_method), assessing sum, average and max pooling.
This cross-validation approach allowed us to evaluate the impact of each parameter setting on model stability and performance, enabling a fine-tuned balance between accuracy and generalization.

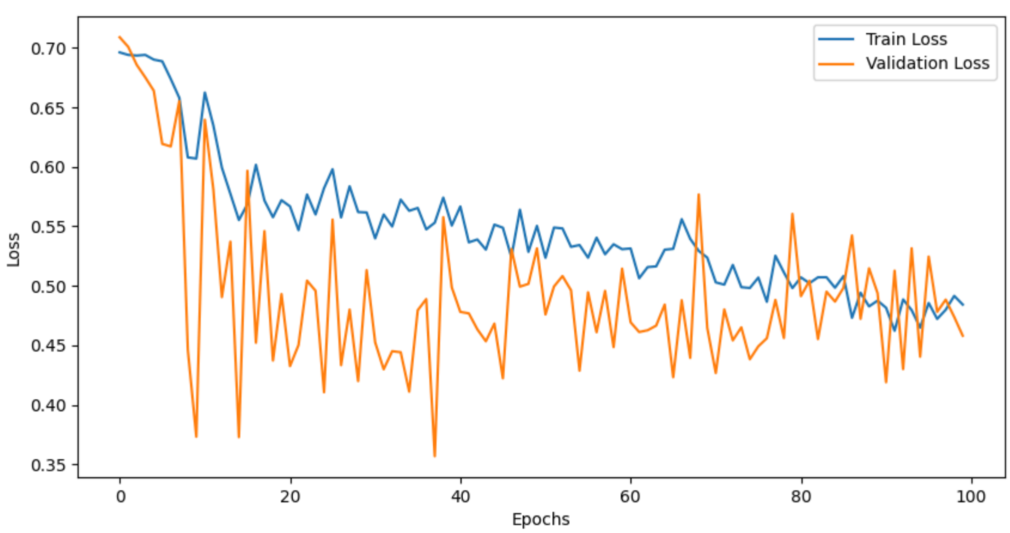
Results:
Through our experiments, the Gaussian models provided decent results by reaching accuracies of 0.95-1; however, they were unstable, and thus we decided to refrain from using them. Ultimately, we selected GraphSAGE with dropout, which demonstrated

stable and high performance. The optimal parameters for this model were an aggregation hidden size of 128, a max pooling aggregation method, 100 training epochs, a fully connected hidden layer size of 128, a learning rate of 0.0007, seven layers, and a batch size of 15. This configuration proved effective, providing a consistent balance between accuracy and generalizability across our validation dataset.

And this model was used to predict on the test dataset.



Accuracy on Training and Validation



Loss on Training and Validation