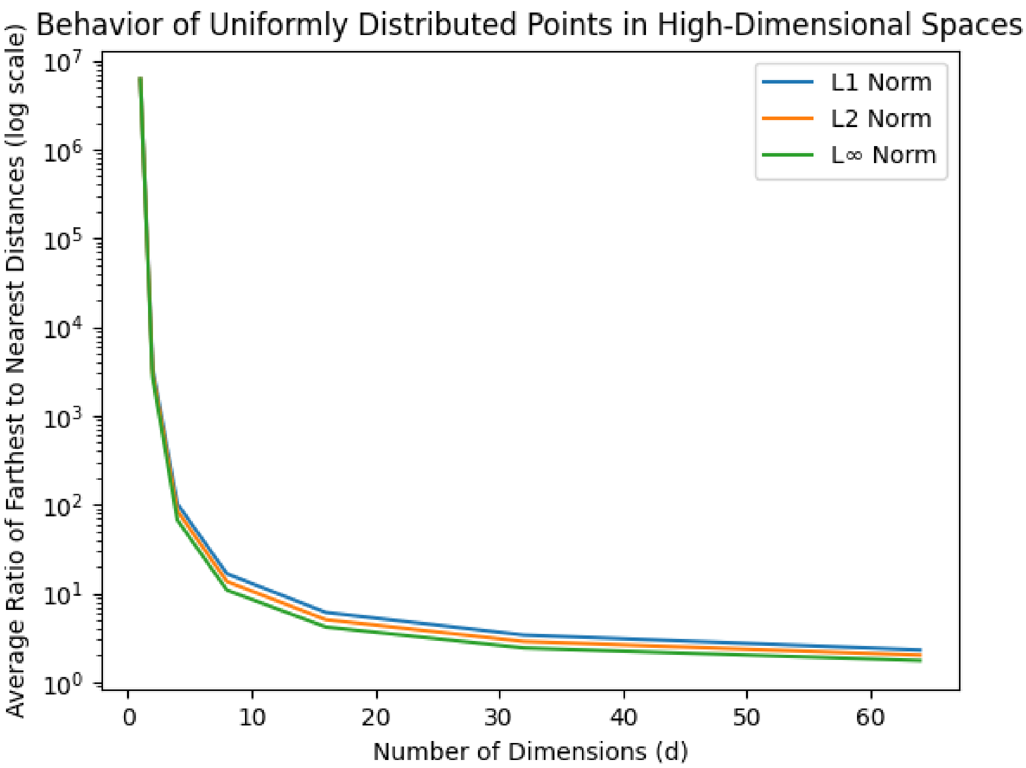
COL 761 Assignment 3

November 26, 2023

1 Uniformly Distributed Points in High-Dimensional Spaces



We have plotted the graph for the average ratio of the farthest distance to the nearest distance for the dataset in log scale. The results reveal a consistent trend: as the dimensionality (d) increases, the average ratio decreases for L_1, L_2, and L_∞ distances. This is because of the following reasons:

- As dimensionality increases, making the dataset sparser, the volume of the space expands exponentially. The nearest distance is influenced by the distribution of points, grows exponentially.
- The farthest distance increases linearly with dimensionality. Specifically, for the L_1 norm, the maximum distance in d dimensions is d, for the L_2 norm, it is $\sqrt{d}$, and for the L_∞ norm, it remains constant at 1, considering only one dimension.
- On averaging out, as the nearest distance is growing rapidly this would resuls in a decrease in the average ratio.

We observe that the sensitivity of the distance measures follows the order $L_1 > L_2 > L_\infty$. In the context of a uniformly distributed graph without outliers, the behavior of each norm aligns with the characteristics of the dataset. For L_∞ as the farthest distance is constrained to a maximum value of 1 due to the dataset's range being [0, 1]. Similarly, for L_1 and L_2, the farthest distance is greater in L_1 than in L_2, reflecting the nature of the norms. It is noteworthy that the order of the nearest distance is not as significant, as evidenced by the close proximity of values in both higher and lower dimensions on the graph.

2 Graph Classification and Regression Using Graph Neural Networks

Our Implementation
This is our final network architecture for graph-level prediction for classfication and regression.

- The network comprises 2 GNN layers, where each layer includes:
  - A 4-head GATv2Conv utilizing edge attributes derived from the graph and the feed-forward of node features.
  - Relu Activation and Dropout with a probability of 0.1.
  - Layer Normalization.
- Global Max Pooling is incorporated for graph-level prediction.
- 2 Linear layers with ReLU activation are added, resulting in an output size of 2.
- Softmax is applied to the output layer to determine the probability of prediction.
- Hidden layer of size $6 \times (\texttt{number of edge features} + \texttt{number of node features})$ for regression and $4 \times (\texttt{number of edge features} + \texttt{number of node features})$ for classification.

After experimenting with GCN, GAT, and GATv2, trying out different configurations of Node embedding layers and linear layers. The choice of GATConv was driven by the aim to leverage edge features and include learnable attention for node features. However, it's crucial to acknowledge a limitation associated with GATConv—the static attention problem. This concern is addressed in GATv2, which allows every node to attend to any other node.

2.1 Classification

Dataset Modification In preprocessing, we first dealt with instances containing `Nan` labels by removing them from the dataset. To handle imbalance between the two classes in the training and validation sets, a simple adjustment was made. Instances from the class with fewer samples were duplicated. This was done to ensure a more even representation of both classes in both the training and validation datasets.

2.1.1 Baseline model

- **Random Model** ROC-AUC score - 0.491
- **Logistic Regression** ROC-AUC score - 0.529

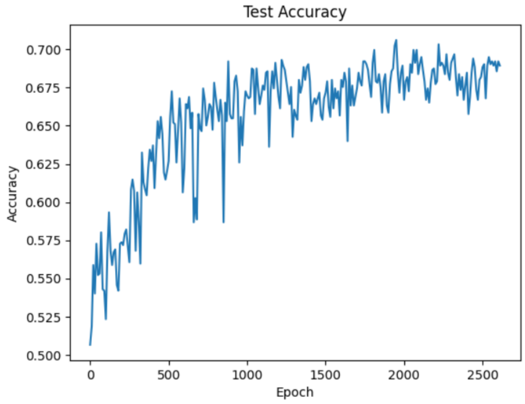


Figure 1: Test Accuracy

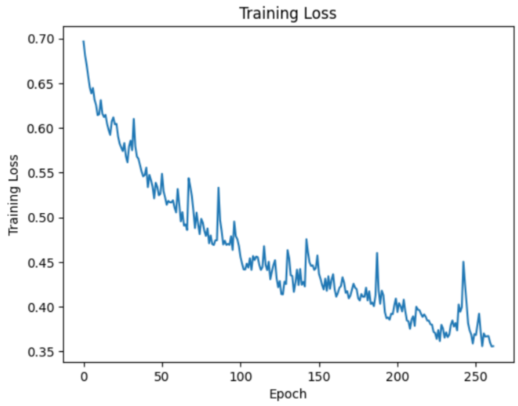


Figure 2: Training Loss

2.1.2 Analysis

Final ROC-AUC score - 0.688
Train accuracy - 0.69

2.1.3 Visualization

Here are some of the examples that are predicted wrong by our model.

2.2 Regression

2.2.1 Baseline model

- **Random Model** RMSE score - 2.727
- **Linear Regression** RMSE score - 1.909

2.2.2 Analysis

RMSE score - 0.6631948351860046
training Loss- 0.11352465635254269

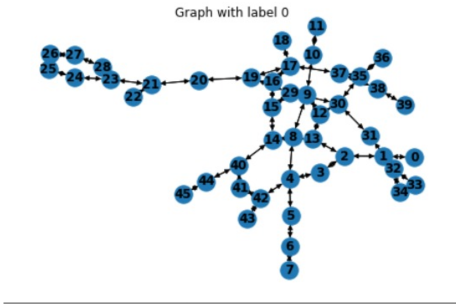


Figure 3: Predicted label:1

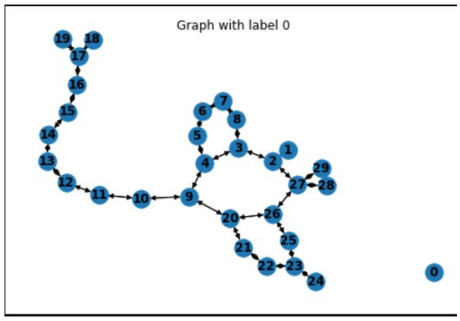


Figure 4: Predicted label:1

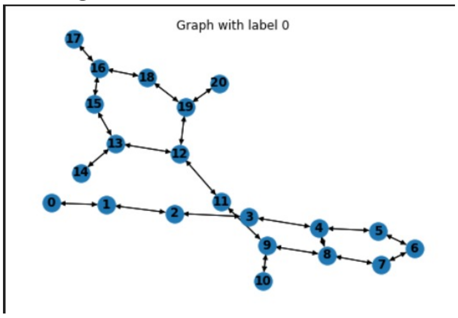


Figure 5: Predicted label:1

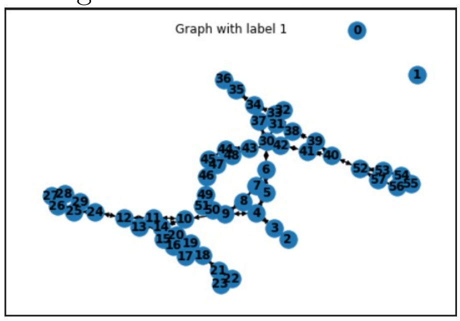


Figure 6: Predicted label:0

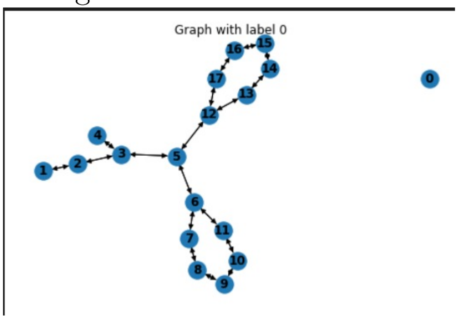


Figure 7: Predicted label:1

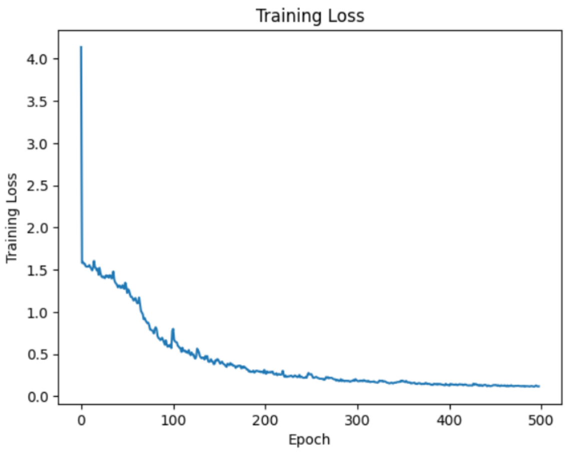


Figure 8: Training Loss

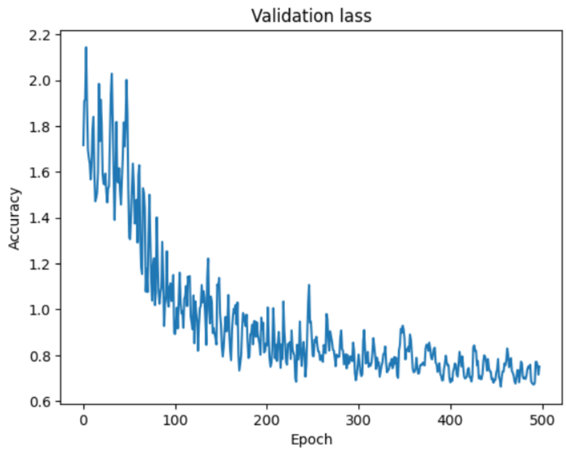


Figure 9: Validation Loss

3 Refrences

- $https://pytorch-geometric.readthedocs.io/en/latest/get_started/colabs.html$

- https://colab.research.google.com/github/AntonioLonga/PytorchGeometricTutorial/blob/main/T
- https://medium.com/stanford−cs224w/incorporating−edge−features−into−graph−neural−networks−for−country−gdp−predictions−1d4dea68337d
- https://colab.research.google.com/github/AntonioLonga/PytorchGeometricTutorial/blob/main/T
- https://colab.research.google.com/github/stellargraph/stellargraph/blob/develop/demos/node−classification/gat−node−classification.ipynb
- https://colab.research.google.com/github/deepmind/educational/blob/master/colabs/summer$_s$cho
7vEmAsr5bKN8