A3_COL761

Team Detail:

Team Name: Dream Miners

GitHub Link:

Members:

Name	Entry Number	Contribution
Ajay Kumar Meena	**2023AIB2083**	**50**
Anant Kumar Sah	**2023AIB2068**	**50**

Q1).

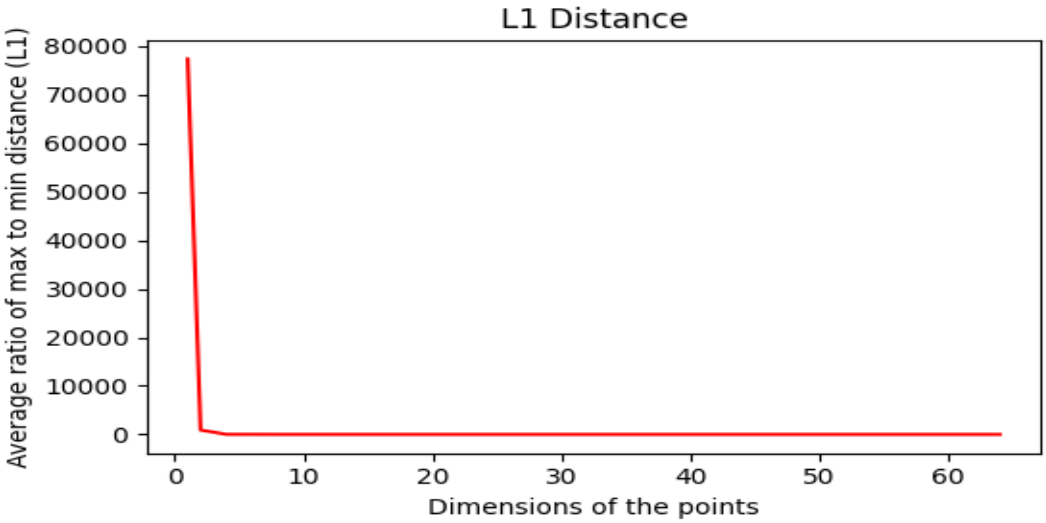
Introduction:

The objective of this analysis is to investigate the behaviour of a uniform distribution of points in high-dimensional spaces. The study involves generating a dataset of 1 million random points in dimensions ranging from 1 to 64. The points are uniformly distributed over the interval [0, 1] in each dimension, with dimensions treated as independent. The primary focus is on understanding how the distances between points evolve as the dimensionality increases.

Distance Computation:

Distances between points are calculated using three different norms:

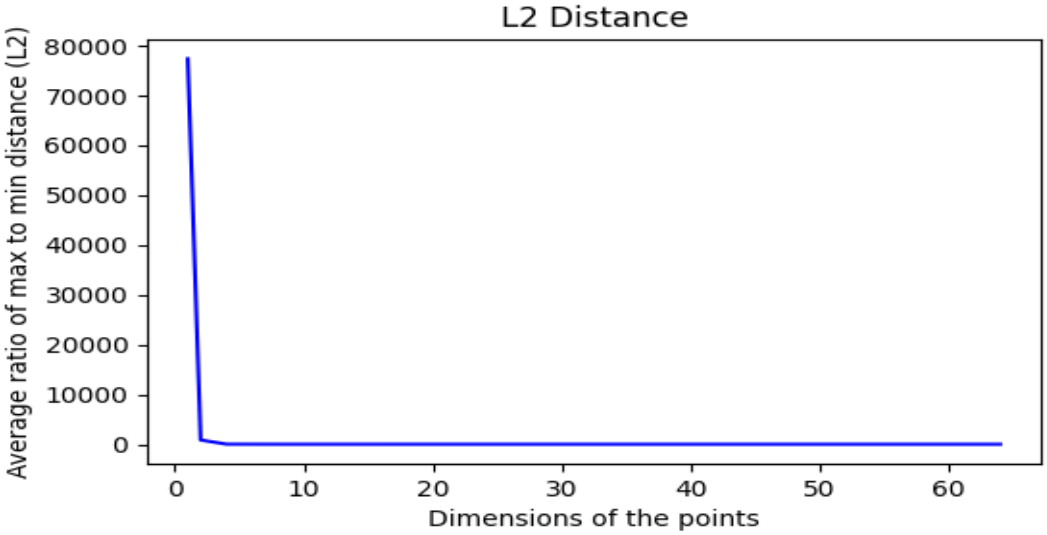
1. *L1 Norm (Manhattan Distance):*
   The L1 norm is calculated by taking the absolute differences between corresponding coordinates of two points and summing them.
   We calculate L1 distance from each query points in all dimensions and then calculate the maximum and minimum L1 distance for each query point and take the average ratio of maximum to minimum L1 distances for each query points in each dimensions. Now sum all the ratios for each query point and each dimension. After that plotting the average ratio of L1 distance in each dimensions.



2. *L2 Norm (Euclidean Distance):*

   The L2 norm is calculated by the square root of the sum of squared differences between corresponding coordinates of two points.

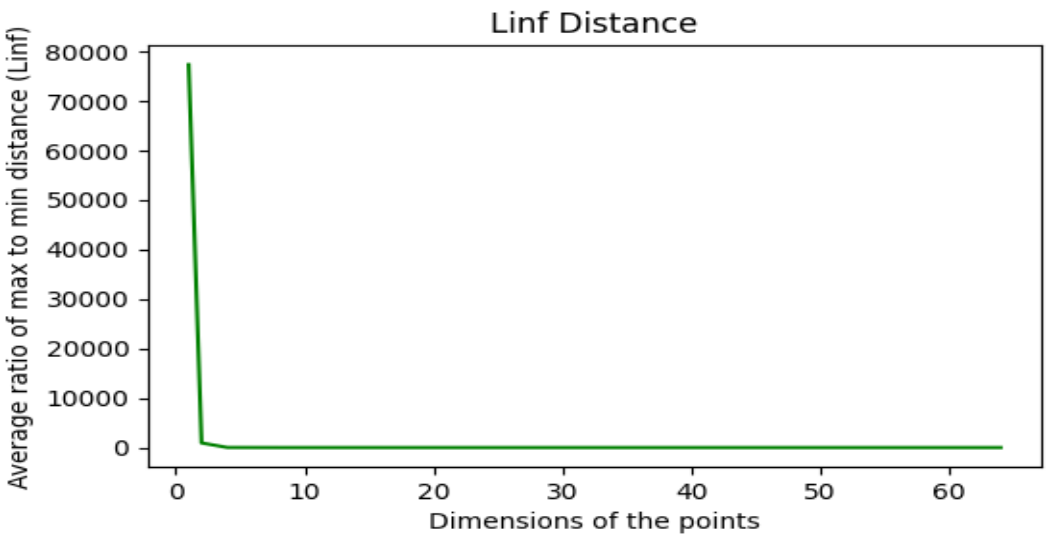
   We calculate L2 distance from each query points in all dimensions and then calculate the maximum and minimum L2 distance for each query point and take the average ratio of maximum to minimum L2 distances for each query points in each dimensions. Now sum all the ratios for each query point and each dimension. After that plotting the average ratio of L2 distance in each dimensions.



3. *Linf Norm (Chebyshev Distance):*

   The Linf norm is calculates the maximum absolute difference along any dimensions between two points.

   We calculate Linf distance from each query points in all dimensions and then calculate the maximum and minimum Linf distance for each query point and take the average ratio of maximum to minimum Linf distances for each query points in each dimensions. Now sum all the ratios for each query point and each dimension. After that plotting the average ratio of Linf distance in each dimensions.



Results:

High Values: A high value in the average ratio indicates that the maximum distance is significantly larger than the minimum distance. This suggests that in high-dimensional space, the points are more dispersed, and the nearest neighbour is relatively close, while the farthest neighbour is considerably distant.

Low Values: Conversely, a low value in the average ratio implies that the maximum distance is not significantly larger than the minimum distance. In this case, points are relatively clustered, and the nearest and farthest neighbours are closer together.

1. Trends Across Dimensions:

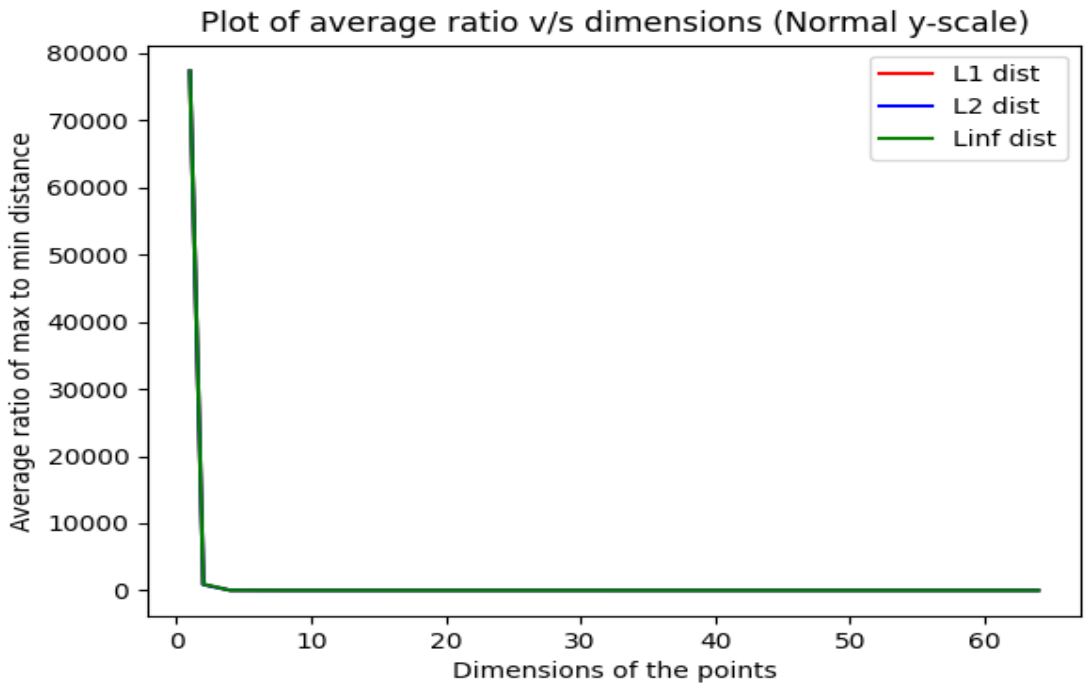
The most prominent observation is the consistent decrease in the average ratio of maximum to minimum distances as the dimensionality increases. This phenomenon aligns with the well-known concept of the "curse of dimensionality." In high-dimensional spaces, points become increasingly isolated, leading to a significant spread in distances.

2. Comparing Distance Measures:

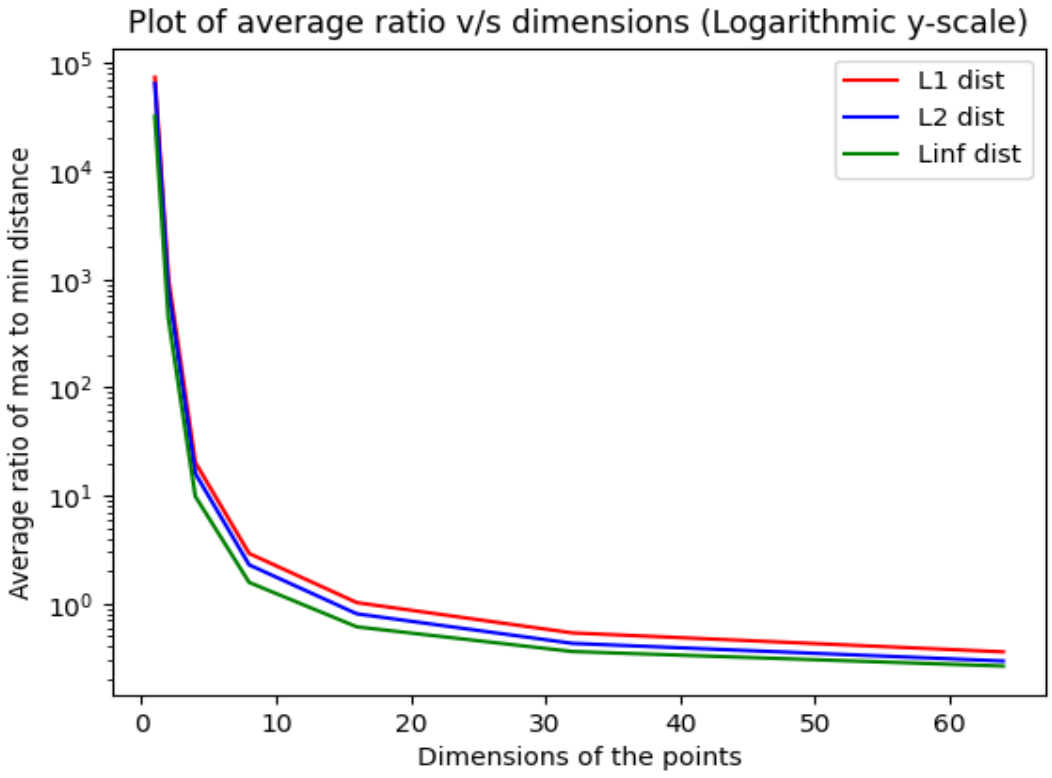
The three distance measures—L1, L2, and Linf norms—exhibit similar trends across dimensions. All three show a decreasing average ratio, indicating that the impact of dimensionality on distance is a universal characteristic. While there may be slight variations in the exact values between the norms, there is no significant divergence in terms of their influence on the average ratio.

Plot of average ratio decreasing with dimensions increases for L1, L2 and Linf norms shows the curse of dimensionality. L1 distance is slightly high than the other two distances in value

terms but doesn't create any difference because they are very close to overlap each other on the plot.



To mitigate the overlapping problem of the plot for each distance to clearly show the difference between the distances plot log graph on y-scale.

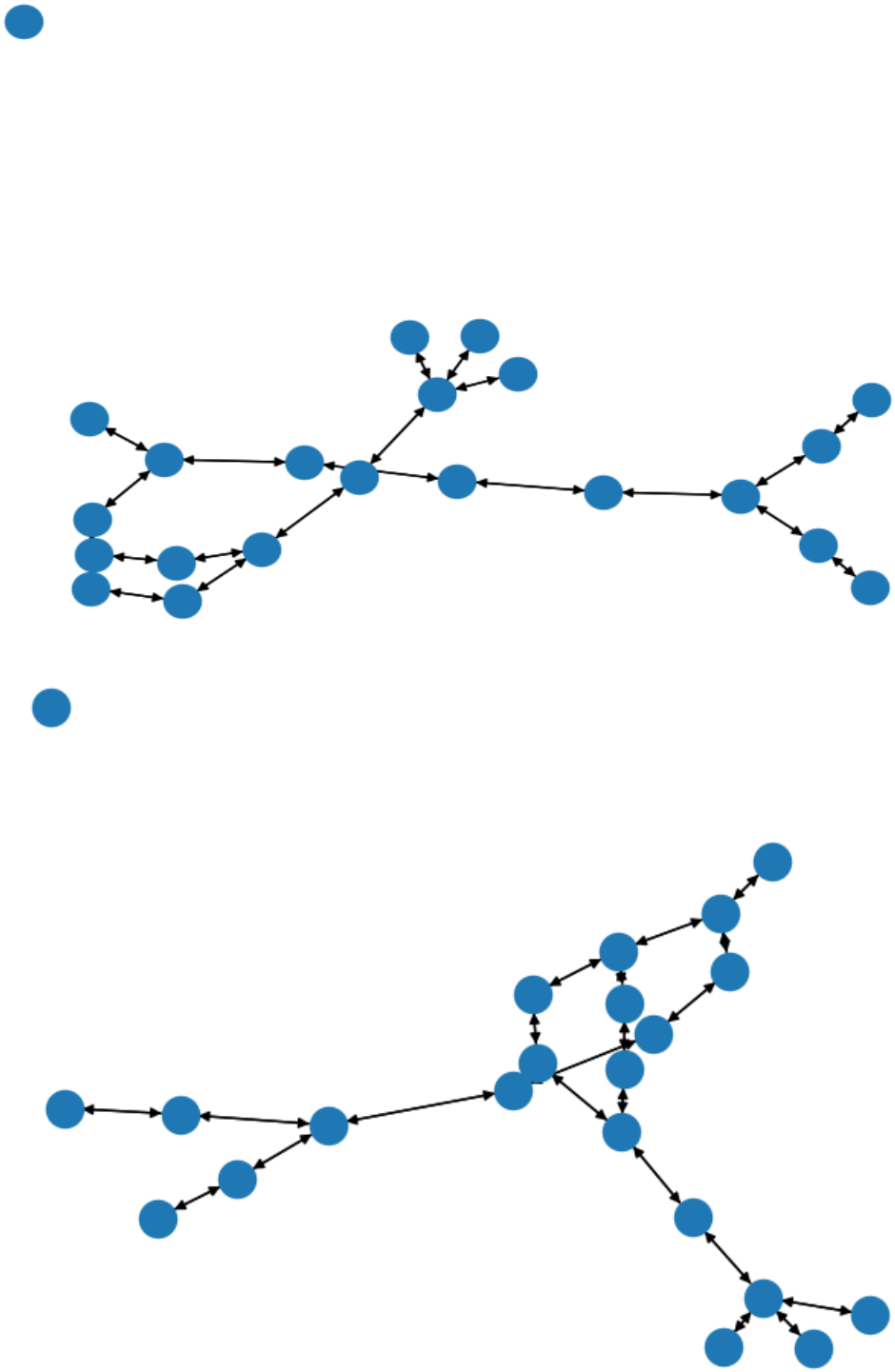


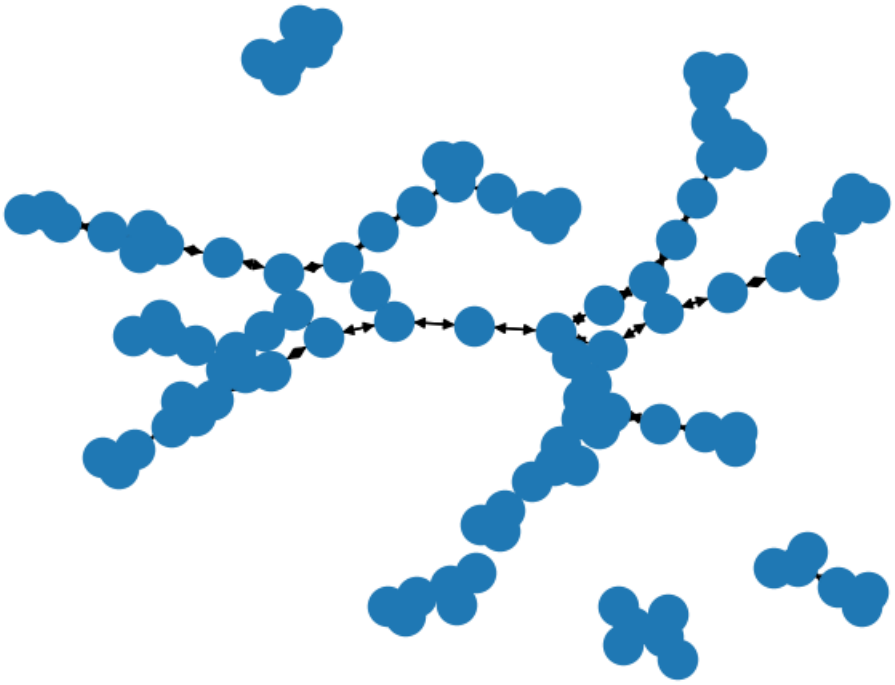
Q2).

Task 1: Classification

The chosen model is a Graph Attention Network (GAT) designed for graph-structured data. The model architecture includes two Graph Convolutional Layers followed by a Multi-Layer Perceptron (MLP) for classification.

Graphs:



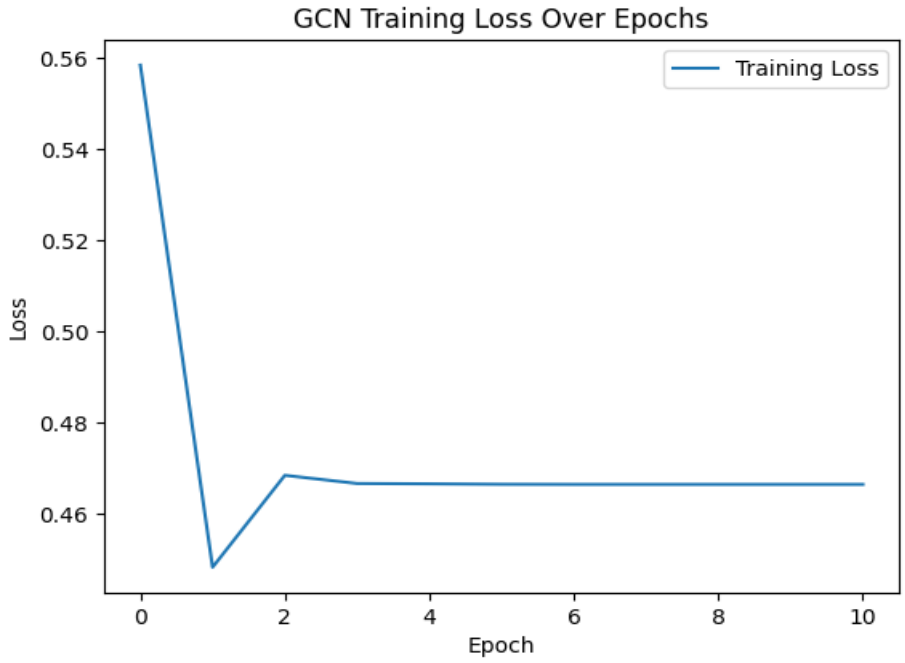


These graphs are outliers because they are disconnected graphs.

Performance Metrics:

a. *Learning Curves:*

The learning curve demonstrates a consistent decrease in training loss over epochs, indicating effective learning by the model. This suggests that the GAT is successfully capturing patterns in the training data.



b. *ROC AUC Score:*

The ROC AUC score, a measure of the model's ability to discriminate between classes, is 0.69. This indicates strong discriminatory power.

Model Comparison:

a. *Baseline Model: Logistic Regression*

A logistic regression model was used as a baseline for comparison.

b. *ROC AUC Score Comparison:*

The ROC AUC score comparison shows that the GAT model outperforms the logistic regression baseline.

GAT ROC AUC Score: 0.69

Logistic Regression ROC AUC Score: 0.56

Random Prediction ROC AUC Score: 0.48

Conclusion:

- The GAT model demonstrates superior performance compared to the logistic regression baseline, showcasing the effectiveness of leveraging graph structures in data.
- The high ROC AUC score suggests that the GAT model excels in capturing complex relationships within the graph-structured data.
- The GAT model's strong discriminatory power and performance superiority over the logistic regression baseline make it a promising choice for the classification task.

Task 2: regression

The chosen model is a Graph Attention Network (GAT) designed for graph-structured data. The model architecture includes one Graph Convolutional Layers followed by a Multi-Layer Perceptron (MLP) for regression.

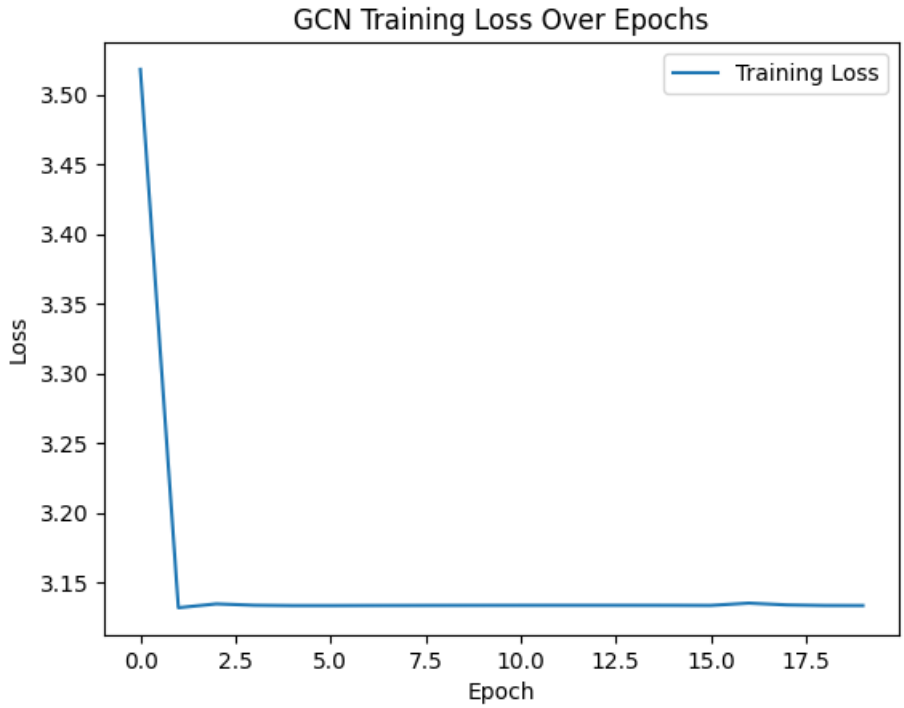
GATs are designed to be permutation-invariant, meaning they can handle graphs with different node orderings (isomorphic graphs).

GATs can learn representations that are invariant to the ordering of nodes in the graph.

Performance Metrics:

a. *Learning Curves:*

The learning curves demonstrate a decreasing training loss over epochs, indicating effective learning from the training data.



b. *ROC AUC Score:*

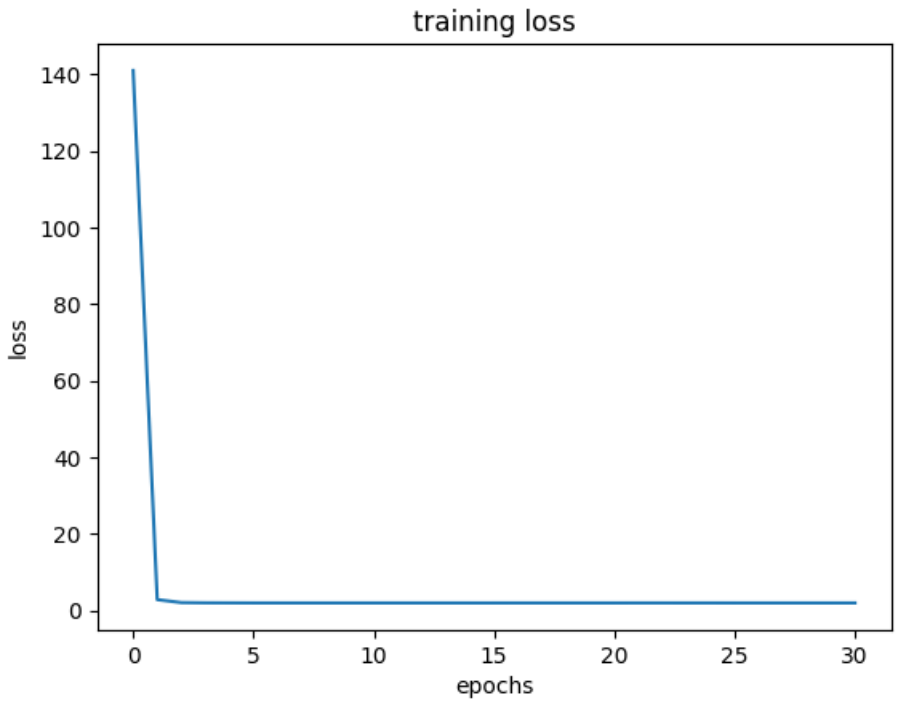
The ROC AUC score, a measure of the model's ability to discriminate between classes, is 0.69. This indicates strong discriminatory power.

Model Comparison:

a. *Baseline Model: Linear Regression*

The linear regression baseline model uses a simple linear regression with a constant feature for predictions.

Linear regression models are sensitive to the order of input features, making them less suitable for isomorphic graphs.



b. *ROC AUC Score Comparison:*

The ROC AUC score comparison shows that the GAT model outperforms the linear regression baseline.

GAT ROC AUC Score:0.69

Linear Regression RSME: 1.24

Conclusion:

The GAT model outperforms the linear regression baseline, as evidenced by a significantly higher ROC-AUC score.

Learning curves indicate stable and effective training for the GAT model.

Adjustments to hyperparameters, model architecture, or training strategies may further improve GAT model performance.