Applying Contrastive Learning to Subgraph Neural Networks

Nathan Ng, Changhao Shi, Gal Mishne

UC San Diego

Outline

- Introduction
  - Motivation, overview, and related work
- Methods
  - Approach to applying contrastive learning to SubGNN
  - Integration of DropGNN and SimCLR Contrastive Learning Framework
- Results
  - Experiments and experimental results
- Discussion
  - Analyze results and model
- Conclusion
  - Summary and key findings

SubGNN & Contrastive Learning

SubGNN & Contrastive Learning

- Graph data is an essential form of data and plays a crucial role in various domains
- Graph Neural Network research has improved performance through Subgraph Neural Networks or using contrastive learning techniques
- Developed model using SubGNN and contrastive learning for graph classification task
- Sought to determine if contrastive learning improves SubGNN
- Model utilizes subgraph views for expressiveness and contrastive learning to learn embeddings for downstream prediction tasks
- To evaluate performance, we conducted experiments using IMDB data

SubGNN & Contrastive Learning

- Proposed approach uses SubGNN and contrastive learning to leverage power of subgraph views through graph augmentation and contrastive learning to generate graph embeddings
- Employs DropGNN model to randomly drop nodes and create subgraph views
- Subgraph views capture different perspectives, increasing expressiveness of model
- Graph Isomorphism Network is applied to subgraphs to encode subgraph
- Contrastive learning used to create similar embeddings for same graph views, and distinct embeddings for different graphs
- Allows for self-supervised learning during training of contrastive embeddings, potentially lowering amount of labeled data needed

Related Work

DropGNN

Supervised subgraph neural network for graph classification

Randomly drops nodes at probability p to create subgraph views

Applies GIN model on subgraphs and aggregates node embeddings to create graph embedding

Makes prediction based on graph embeddings

SimCLR

Self-supervised image classification model using contrastive learning

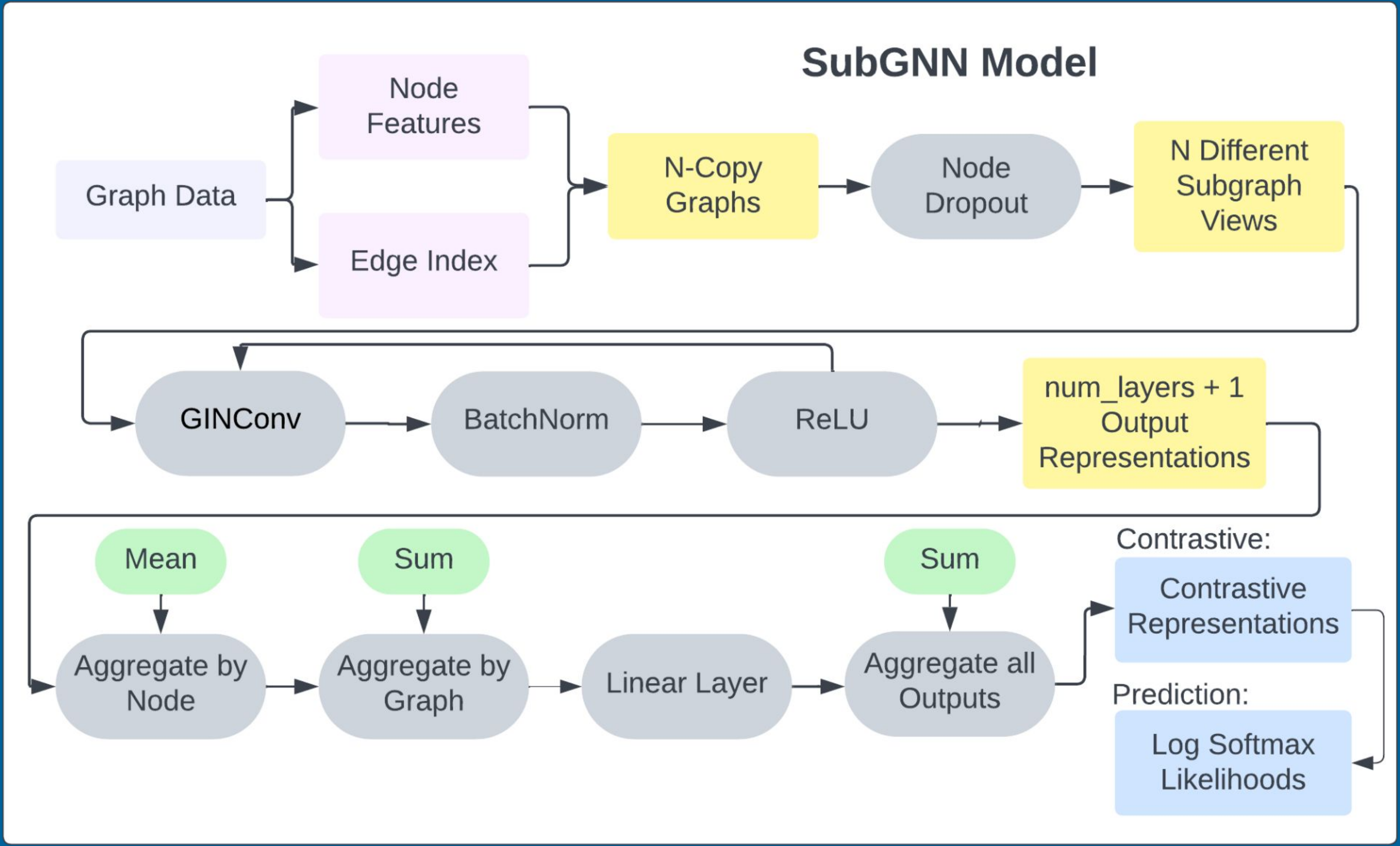
Applies various image augmentations to create different views

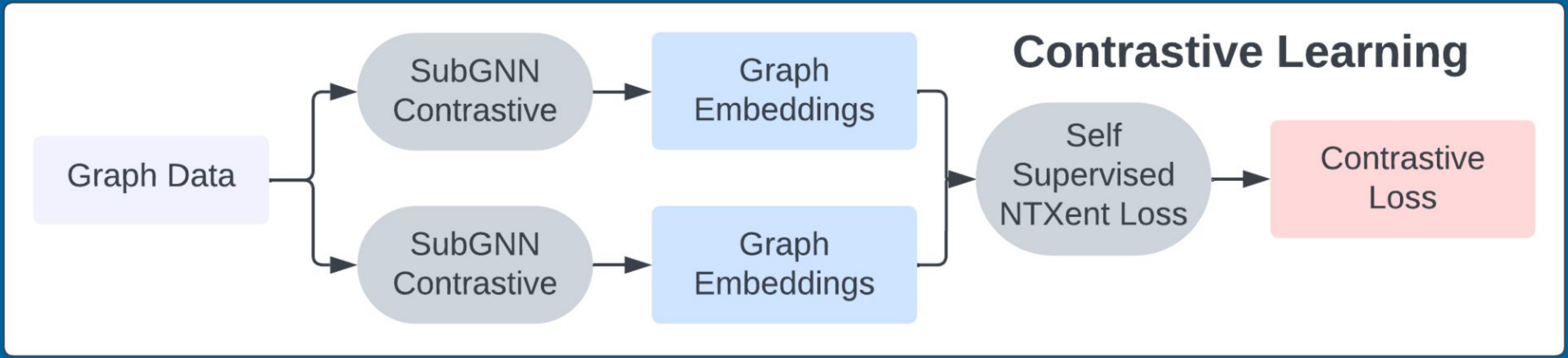
Used a contrastive learning framework using base encoder, project head, and encoder

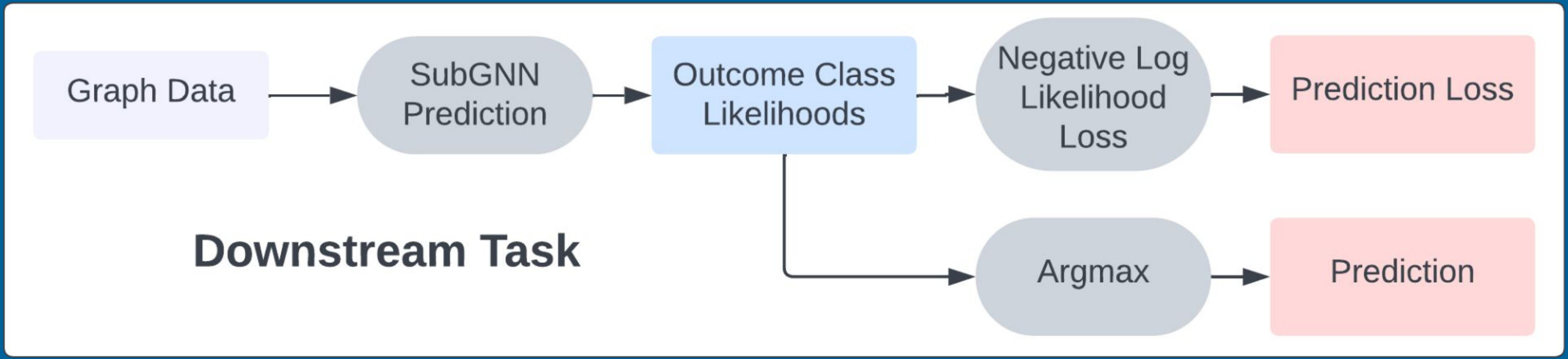
Calculated NT-Xent Loss for contrastive embeddings

$$L_{NT-Xent}(z_i, z_j) = -\log\left(\frac{\exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp(sim(z_i, z_j)/\tau)}\right)$$

Methods


SubGNN Model
Graph Data
Node Features
Edge Index
N-Copy Graphs
Node Dropout
N Different Subgraph Views
GINConv
BatchNorm
ReLU
num_layers + 1 Output Representations
Mean
Sum
Sum
Aggregate by Node
Aggregate by Graph
Linear Layer
Aggregate all Outputs
Contrastive:
Contrastive Representations
Prediction:
Log Softmax Likelihoods


Contrastive Learning
Graph Data
SubGNN Contrastive
SubGNN Contrastive
Graph Embeddings
Graph Embeddings
Self Supervised NTXent Loss
Contrastive Loss


Graph Data
SubGNN Prediction
Outcome Class Likelihoods
Negative Log Likelihood Loss
Prediction Loss
Downstream Task
Argmax
Prediction

Augment Graph Data

Randomly drop nodes to create different subgraph views of same graph

Generate Graph Embeddings

Subgraphs are turned into graph embeddings using GIN model to be used for contrastive learning or prediction task

Contrastive Learning

First part of model training that learns to create similar embeddings for similar graphs, and distinct embeddings for different graphs

Prediction Task

Using learned graph embeddings, apply logistic regression to generate class likelihoods

Results

Experiments

Datasets

- IMDB-BINARY and IMDB-MULTI
- Graphs represent movies
- Nodes represent actors/actresses
- Edges between actors and actresses that act in same movie
- Graph labels represent movie genre
    - Binary: 2 genres - Action, Romance
    - Multi: 3 genres - Comedy, Romance, Sci-Fi

	IMDB-BINARY	IMDB-MULTI
# Graphs	996	1498
Mean Nodes	19	13
Max Nodes	69	63
Min Nodes	12	7
Mean Degrees	18.486	11.907
Max Degrees	68	62
Min Degrees	11	6

Experiments

- Models run using graph embeddings of size 32, 64 hidden units for each hidden layer, and various node dropout probabilities
    - Node dropout probabilities were adjusted to determine how generation of subgraphs affected contrastive learning and prediction task
- Trained with batch size of 32 and learning rate of 0.001 for 100 epochs
- Contrastive loss was calculated by NTXent loss
- Prediction loss was calculated by Negative Log Likelihood loss
- Models were compared based on average final test accuracy
- Compared with GIN and DropGNN model performances

Results: IMDB-Binary

p = 0.0

p1 / p2	0.25	0.55	0.85
0.1	**0.7108**	0.6787	0.6947
0.4	0.6827	**0.7149**	0.6746
0.7	**0.7028**	0.6787	0.6787

p = 0.1

p1 / p2	0.25	0.55	0.85
0.1	0.6907	0.6746	0.6706
0.4	**0.7028**	0.6867	0.6626
0.7	0.6947	0.6706	0.6626

p = 0.5

p1 / p2	0.25	0.55	0.85
0.1	0.6425	**0.7028**	0.6706
0.4	0.6867	0.6947	0.6947
0.7	**0.7108**	0.6666	0.6305

p1: Probability of node dropout for 1st embedding

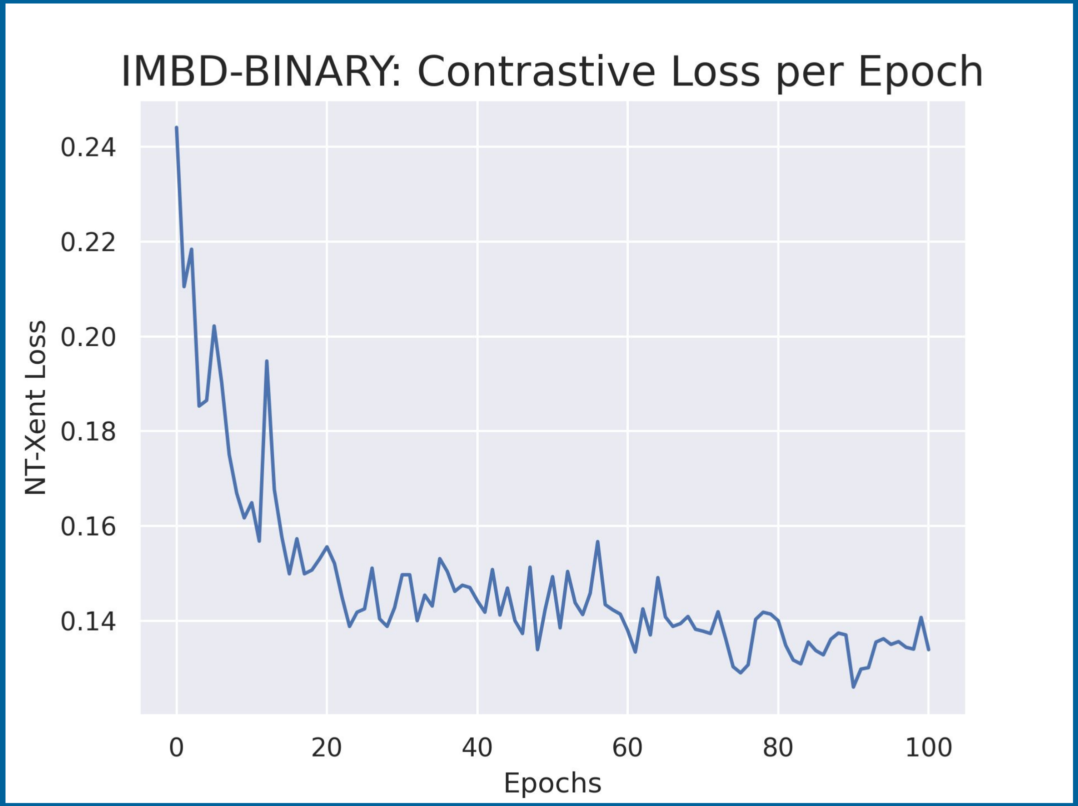
p2: Probability of node dropout for 2nd embedding

p: Probability of node dropout during prediction

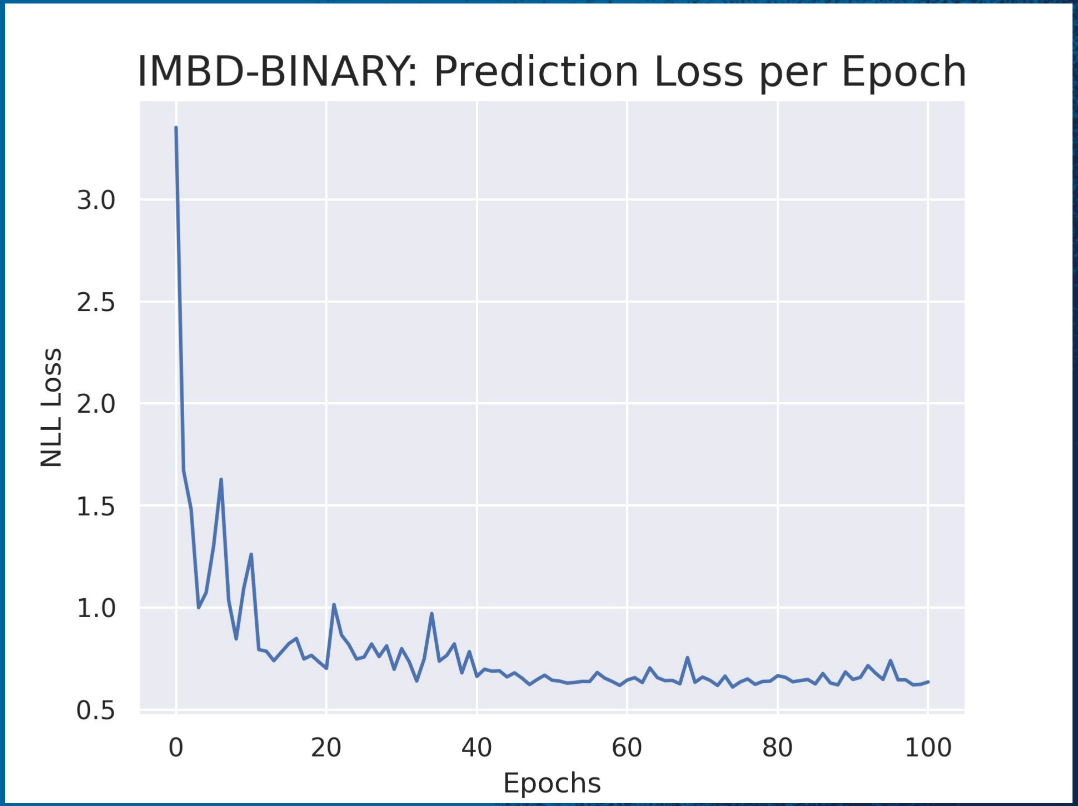
IMDB-BINARY Task

- Experimented with 3 different values for each node dropout probability
- Accuracies ranged from 0.63 to 0.71
- Models performed better when p2 was 0.25 or 0.55
- Models performed better when p was 0.0 or 0.5

Results: IMDB-Binary



Converges around 0.13 loss



Converges around 0.63 loss

UC San Diego

Results: IMDB-MULTI

p = 0.0

p1 / p2	0.25	0.55
0.1	0.4826	**0.5360**
0.4	**0.5279**	**0.5066**

p = 0.5

p1 / p2	0.25	0.55
0.1	0.4773	**0.5226**
0.4	**0.5120**	**0.5413**

IMDB-MULTI Task

Used better performing p values from IMDB-BINARY test

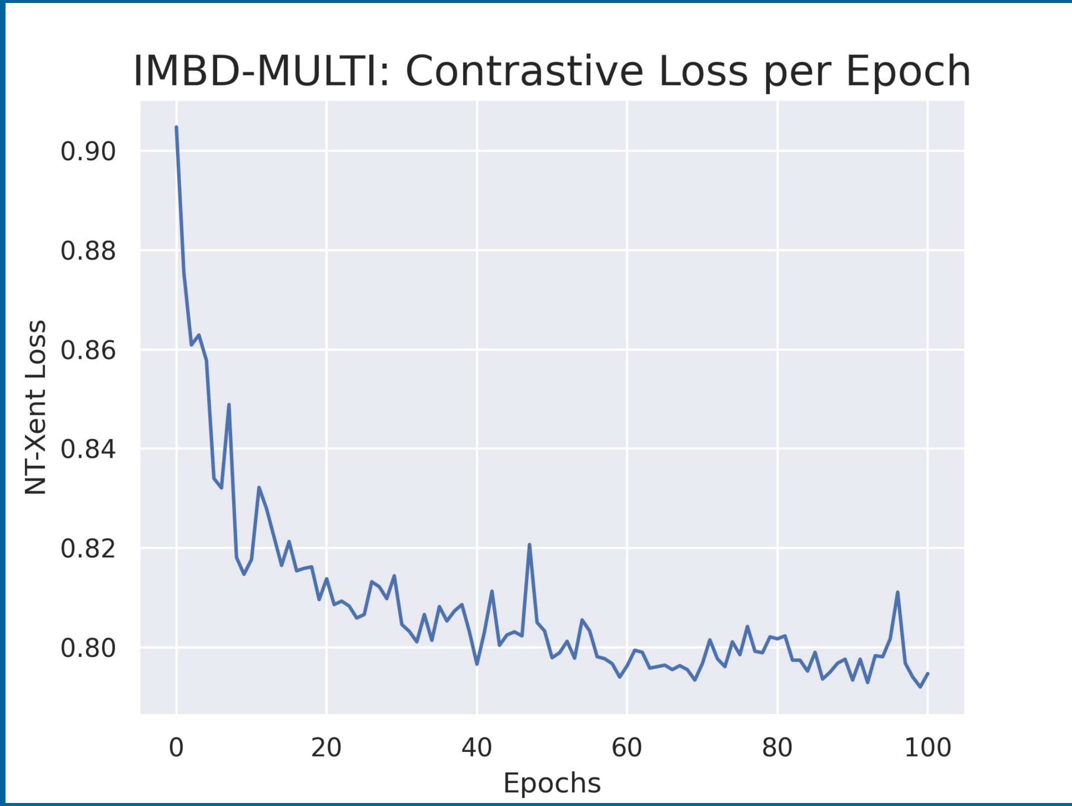
- Accuracies ranged from 0.47 to 0.54
- Model performed better when p1 and p2 were not their lowest values

p1: Probability of node dropout for 1st embedding

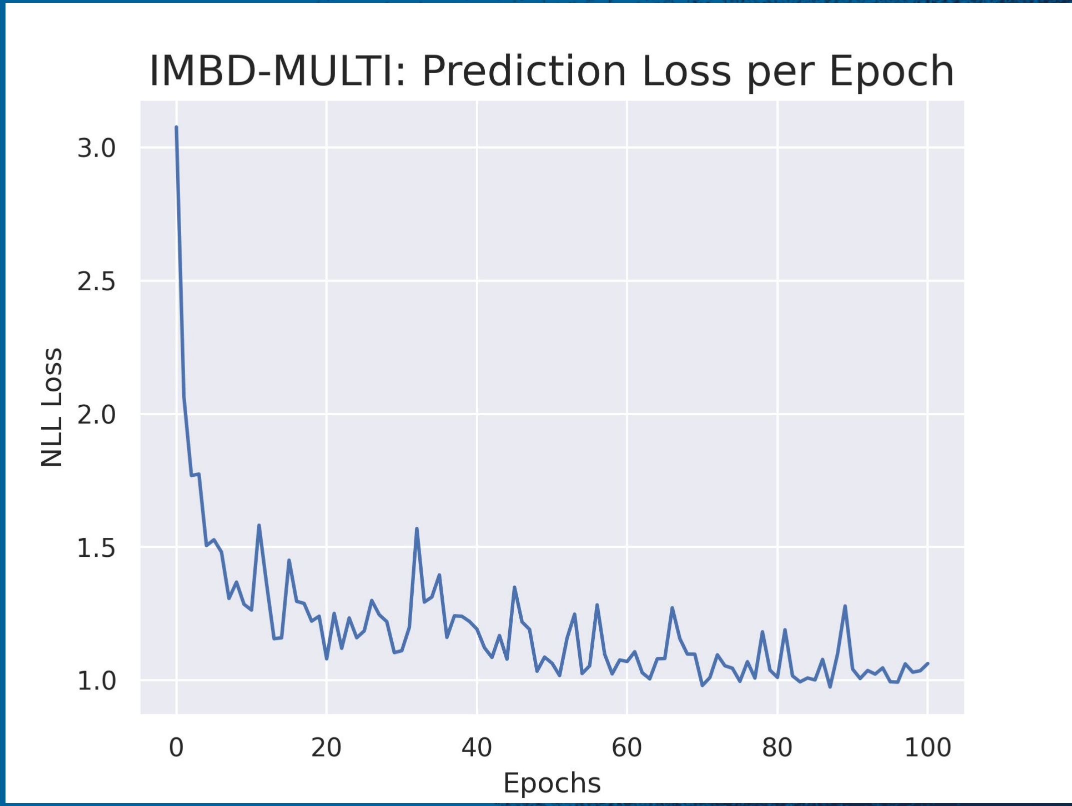
p2: Probability of node dropout for 2nd embedding

p: Probability of node dropout during prediction

Results: IMDB-MULTI



Converges around 0.79 loss



Converges around 1.06 loss

Results: Baseline Comparisons

	IMDB-BINARY	IMDB-MULTI
GIN	75.1	52.3
DropGNN	**75.7**	51.4
SubGNN + Contrastive Learning	71.4	**54.1**

DropGNN had the best accuracy for IMDB-BINARY, with 75.5% accuracy

SubGNN + Contrastive Learning had the best accuracy for IMDB-MULTI, with 54.1% accuracy

Discussion

Discussion

- Best performing SubGNN & Contrastive learning model
    - 71.4% for IMDB-BINARY, underperforming baseline models
    - 54.1% for IMDB-MULTI, outperforming baseline models by ~2%
- IMDB-BINARY: Models performed better when second contrastive embedding dropout probabilities were lower or when prediction dropout probability was either 0 or 0.5
- IMDB-MULTI: Models performed better when contrastive embedding dropout probabilities were not low, regardless of prediction dropout probability
- Method of creating subgraphs during contrastive learning has larger effect on model performance compared to graph augmentations at prediction time
- Contrastive learning improved performance of model for multi-class classification
- Model could use more epochs of training with lower learning rate to converge better and potentially get better results

Conclusion

Conclusion

- Project sought to determine if contrastive learning applied to SubGNN could improve performance of graph classification
- Adapted DropGNN's node dropout augmentation to create subgraph views for more expressiveness
- Adapted SimCLR's contrastive learning framework to learn graph embeddings and use embeddings for prediction tasks
- Tested models on IMDB datasets and experimented with how generating subgraphs affected embeddings' effectiveness at graph prediction
- Model outperformed baseline models on IMDB-MULTI prediction task
- Contrastive learning applied to SubGNN for multi-class classification tasks shows promising results
- Learning better graph embeddings helps with more complex tasks, but may not be as useful for simpler binary tasks

Future Work

- Improve current model and expand on experiments
  - Run for more training epochs - get better convergence for IMDB-MULTI
  - Test on more combinations of dropout probabilities
  - Test effect of different contrastive embedding representations
  - Test model on other prediction tasks and data
- Experiment with different subgraph augmentations
  - Edge dropout or combination of node and edge dropout
- Experiment with different contrastive learning frameworks and contrastive learning loss functions
- Research into whether contrastive learning is better for multi-class prediction tasks

Bibliography

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020, November). A simple framework for contrastive learning of visual representations. In International conference on machine learning (pp. 1597-1607). PMLR.

Papp, P. A., Martinkus, K., Faber, L., & Wattenhofer, R. (2021). DropGNN: Random dropouts increase the expressiveness of graph neural networks. Advances in Neural Information Processing Systems, 34, 21997-22009.

The beginner's guide to contrastive learning. V7. (n.d.). https://www.v7labs.com/blog/contrastive-learning-guide#h3

Xu, D., Cheng, W., Luo, D., Chen, H., & Zhang, X. (2021). Infogcl: Information-aware graph contrastive learning. Advances in Neural Information Processing Systems, 34, 30414-30425.

Questions

Thank you to Professor Mishne and Changhao for your help and guidance on this project

UC San Diego