**NODE CLASSIFICATION USING GRAPH NEURAL NETWORKS**

ANJALI A SUNIL 222BDA34

GARGI P KRISHNA 222BDA36

CHAITHRA S H 222BDA61

**BD2P1: FOUNDATION OF DATA SCIENCE**

SUBMITTED TO:

Dr. JAYATI BADRA

JAYATI KAUSHIK

DEPARTMENT OF ADVANCED COMPUTING

ST. JOSEPH’S UNIVERSITY

28th April 2023

**GITHUB REPOSITORY LINK:**

[**https://github.com/ChaithraaH/FDS-Node-Classification-GNN-model-**](https://github.com/ChaithraaH/FDS-Node-Classification-GNN-model-)

**AIM**

The model is applied to a node prediction task on the Cora dataset to determine the topic of an article given its words and citations network. It’s an application of Graph Neural Network.

**INTRODUCTION**

The Node Classification with Graph Neural Networks project is a machine learning project that aims to classify nodes in a graph using Graph Neural Networks (GNNs). The project utilizes the Cora dataset, which is a collection of scientific publications in different subjects. The goal of the project is to demonstrate the effectiveness of GNN models in classifying nodes in a graph, and to provide a step-by-step guide on building and training a GNN model. The objectives of the project include data preprocessing, model architecture definition, training, and evaluation. Through this project, we aim to showcase the versatility and potential of GNN models in handling graph-structured data and their application in node classification tasks. Many datasets in various machine learning (ML) applications have structural relationships between their entities, which can be represented as graphs. Such application includes social and communication networks analysis, traffic prediction, and fraud detection. Graph representation Learning aims to build and train models for graph datasets to be used for a variety of ML tasks.

**LITERATURE REVIEW**

Graph neural networks (GNNs) are a relatively recent development in deep learning that enable the modelling of graph-structured data. They have been shown to be effective for a wide range of tasks, including node classification, link prediction, and graph classification.

One of the early works in GNNs is the Graph Convolutional Networks (GCN) proposed by Kipf and Welling in 2017. GCN is a simple and efficient GNN architecture that uses a spectral graph convolutional operation to aggregate information from neighbouring nodes. Since then, there have been many extensions and variations of GCN, such as Graph SAGE, GAT, and Graph Attention Many datasets in various machine learning (ML) applications have structural relationships between their entities, which can be represented as graphs. Such application includes social and communication networks analysis, traffic prediction, and fraud detection. Graph representation Learning aims to build and train models for graph datasets to be used for a variety of ML tasks. This project demonstrate a simple implementation of a Graph Neural Network (GNN) model. The model is used for a node prediction task on the Cora dataset to predict the subject of a paper given its words and citations network.

**METHODOLOGY**

**Data Collection**: It is downloaded from w The Cora dataset consists of 2708 scientific publications classified into one of seven classes. The citation network consists of 5429 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1433 unique words. The Cora dataset used in the implementation of node classification with graph neural networks contains publications from seven different categories:

Case\_Based: Publications describing specific cases or scenarios.

Genetic algorithms: Publications related to the use of genetic algorithms in problem-solving.

Neural\_Networks: Publications related to the use of neural networks in problem-solving.

Probabilistic\_Methods: Publications related to the use of probabilistic methods in problem-solving.

Reinforcement\_Learning: Publications related to the use of reinforcement learning in problem-solving.

Rule\_Learning: Publications related to the use of rule learning in problem-solving.

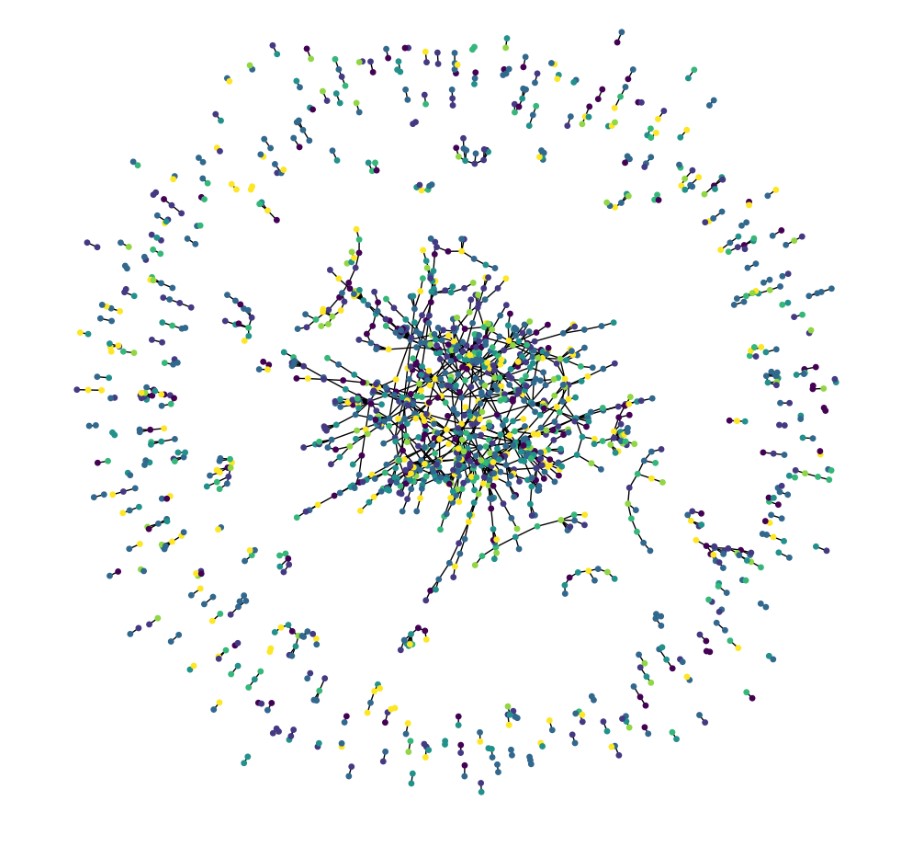
Theory: Publications related to theoretical aspects of problem-solving.

The task in the implementation of node classification is to predict the category of each publication based on its citation network.

**Data Preprocessing**: The Cora dataset is loaded and preprocessed to prepare it for use with PyTorch Geometric. This involves creating a graph representation of the data and encoding the features and labels as tensors.

**Training**: We trained and implemented a baseline classifier and added five FFN blocks with skip connections, with roughly the same number of parameters as the GNN models to be built later. The learning graph from this model has been plotted for comparison with the GNN model. The GNN model is trained using stochastic gradient descent (SGD) and back propagation. The training data is randomly split into training and validation sets, and the model is trained to minimize the cross-entropy loss function.

**Visualization** : We made a graph to visualize the model using matplotlib and network. Each node in the graph represents a paper, and the color of the node corresponds to its subject. Note that we only show a sample of the papers in the dataset.



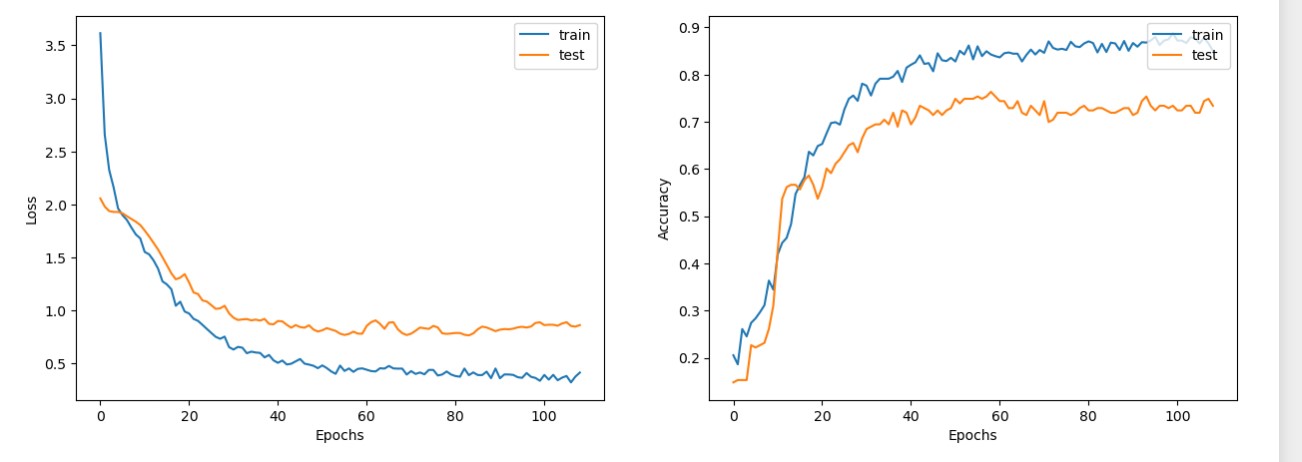
**IMPLEMENTATION**

Node classification is the task of assigning a label or category to each node in a graph based on its attributes or its connections to other nodes. Graph neural networks (GNNs) are a type of neural network that can be used for node classification tasks on graph data. GNNs can learn to propagate information through the graph, allowing them to make predictions for each node based on its local neighborhood.

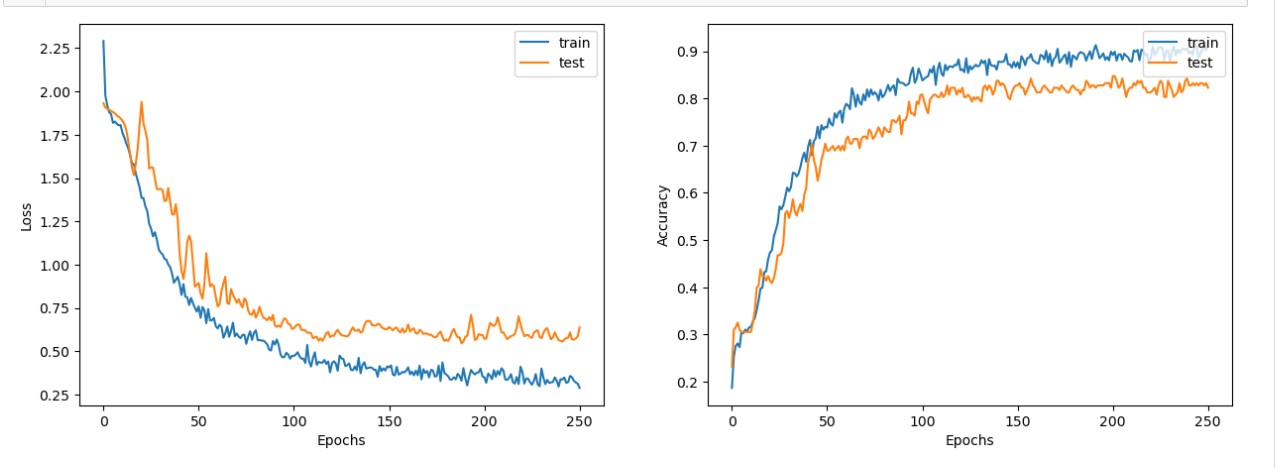
The GNN model is implemented using the PyTorch Geometric library, which provides tools for working with graph data and building GNNs. The example uses the Cora dataset, which consists of academic publications and their citations. The goal is to predict the category of each publication based on its citation network.

The GNN model is composed of multiple layers, each of which performs a message-passing operation. During each layer, the model aggregates information from neighbouring nodes, performs a nonlinear transformation, and passes the result to the next layer. The output of the final layer is used to make predictions for each node. The steps outlined in the notebook include data preparation, model architecture definition, training, and evaluation. **The baseline classifier gave an accuracy of 75.66%, while the GNN model achieved an accuracy of 80.20%**, demonstrating the effectiveness of the GNN approach for this task. The project also showcases the versatility of GNN models in handling graph-structured data. By converting the paper ids and subjects into zero-based indices and using binary word vectors to represent term presence, the GNN model is able to effectively classify nodes based on their attributes and relationships. The use of a Design Space for Graph Neural Networks approach highlights the flexibility of GNN models in adapting to various datasets and classification tasks. By applying preprocessing and post-processing steps to the node embedding, the GNN model is able to improve its accuracy and performance. Overall, the project provides valuable insights into the use of GNN models for node classification and highlights their potential for a wide range of applications in graph-based machine learning. Further research in this area can explore the optimization of GNN models and their application in other domains.

***The learning curve of the baseline model***



***The learning curve of the GNN model***



As we interpret the results of the Node Classification with Graph Neural Networks project, we find that the GNN model is highly effective in node classification tasks on the Cora dataset. This finding aligns with the literature review, which highlighted the superior performance of GNNs in graph-based machine learning tasks. We recognize that the GNN model has strengths, such as its ability to capture complex relationships and patterns in graph-structured data, making it highly suitable for tasks such as node classification and link prediction. However, it also has limitations, such as scalability and interpretability challenges.

To address these challenges, further research is needed in developing more efficient GNN architectures and techniques for scaling them to larger datasets. Additionally, efforts should be made to enhance the interpretability of GNNs, possibly through the use of explainable AI techniques.

Overall, our analysis indicates that while the GNN model has significant potential in node classification and other graph-based machine learning tasks, there is still work to be done to improve its efficiency and interpretability and explore its potential in unsupervised and reinforcement learning.

**CONCLUSION**

In conclusion, the Node Classification with Graph Neural Networks project demonstrated the effectiveness of GNNs in the task of node classification, achieving high accuracy on the Cora dataset. The project also provided an introduction to the basic principles and architecture of GNNs, highlighting their ability to capture complex relationships and patterns in graph-structured data.

The implications of these findings are significant for node classification and other graph-based machine learning tasks. GNNs have the potential to outperform traditional machine learning methods on such tasks, especially in scenarios where graph structure is an important factor. Furthermore, GNNs can be used for a variety of applications, including social network analysis, recommendation systems, and natural language processing. In terms of current research trends, GNNs are a rapidly developing area of machine learning and are attracting increasing attention from researchers in academia and industry. There is a growing interest in developing more advanced GNN architectures, such as graph attention networks and graph transformers, as well as investigating the potential of GNNs in unsupervised and reinforcement learning. Additionally, there is a need for developing robust and efficient GNN models that can scale to larger datasets and more complex graphs.

Overall, the Node Classification with Graph Neural Networks project has demonstrated the potential of GNNs in node classification and provided a foundation for further research in this exciting area of machine learning.

**GITHUB REPOSITORY LINK:**

[**https://github.com/ChaithraaH/FDS-Node-Classification-GNN-model-**](https://github.com/ChaithraaH/FDS-Node-Classification-GNN-model-)

**REFERENCES**

[1] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. IEEE Transactions on Neural Networks, 20(1), 61-80.

[2] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

[3] Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In Advances in Neural Information Processing Systems (pp. 1024-1034).

[4] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. arXiv preprint arXiv:1710.10903.

[5] Widhiartha, P. (2021). Node Classification with Graph Neural Networks. Kaggle. https://www.kaggle.com/widhiartha/node-classification-with-graph-neural-networks.

[6] Cora Dataset. Datasets for GNNs. https://github.com/kimiyoung/planetoid/tree/master/data.