Evaluating MLPs vs. GNNs for Large-Scale Product Classification

Albin Steen, Anthon Odengard

Chalmers University of Technology

CHALMERS UNIVERSITY OF TECHNOLOGY

Problem formulation

- **Motivation:** The increasing volume of **unstructured data** is reshaping decision-making processes across diverse domains.
- **Challenge:** Companies aim to address this by transforming unstructured data into **graph-like structures to model relationships**.
- **Solution: Graph Neural Networks (GNNs)** have emerged as powerful tools for tasks like classification by **leveraging these internal relationships**.
- **Objective:** This project **compares a traditional Multilayer Perceptron (MLP)**, which lacks graph-awareness, with **two types of GNNs** on the Open Graph Benchmark (OGB)-Products dataset.
- **Focus:** We investigate the **role of contextual information** (edges between products) and its **effect on classification performance.**

Methods

OGB-PRODUCTS DATASET:

- Represents a large-scale Amazon product network.
- Contains **2.5 million nodes**, each with **100 features**, and **62 million edges** connecting the products.
- **47 classes.**
- **Split of 8% for training, 2% for validation, and 90% for testing**, based on product popularity.

Figure 1: Example of a subgraph of the OGB-products dataset.

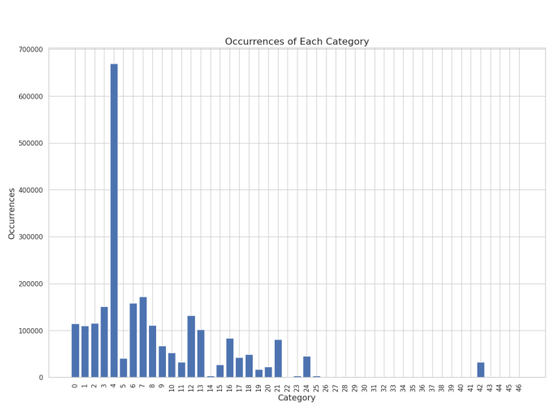

Figure 2: Class occurrences within the dataset.

MLP MODEL:

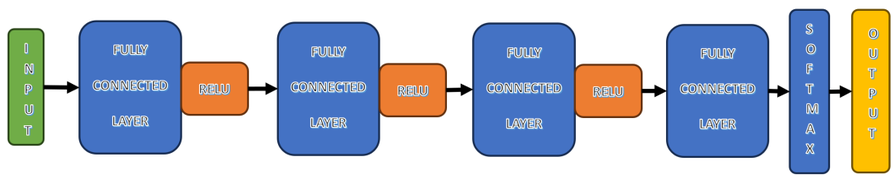

Figure 3: Structure for the MLP network.

GNN MODELS:

Graph Convolutions (GCN) [2]:

- Graph Convolutions **use the structure of graph data** to propagate information between connected nodes, **extending the concept of convolutions from CNNs**.
- This enables GNNs to aggregate feature information from neighboring nodes, updating each node's representation **without requiring a fixed input structure.**

Graph SAGE (SAGE) [1]:

- Instead of aggregating all neighbors, Graph SAGE **samples a subset of connected nodes** for aggregation, allowing for better generalization.
- Employs **various aggregation methods**, such as LSTM and min/max pooling, to effectively capture complex relationships.

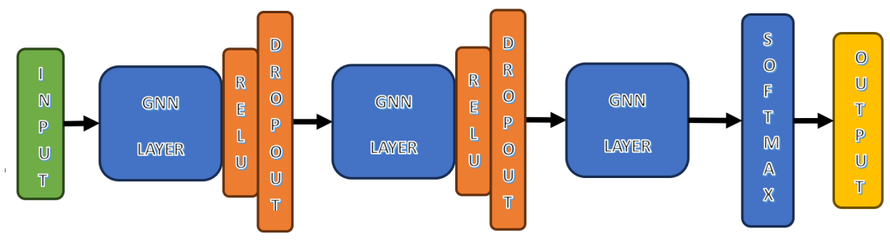

Figure 4: Structure for both GNN networks.

Results

TRAINING RESULTS:

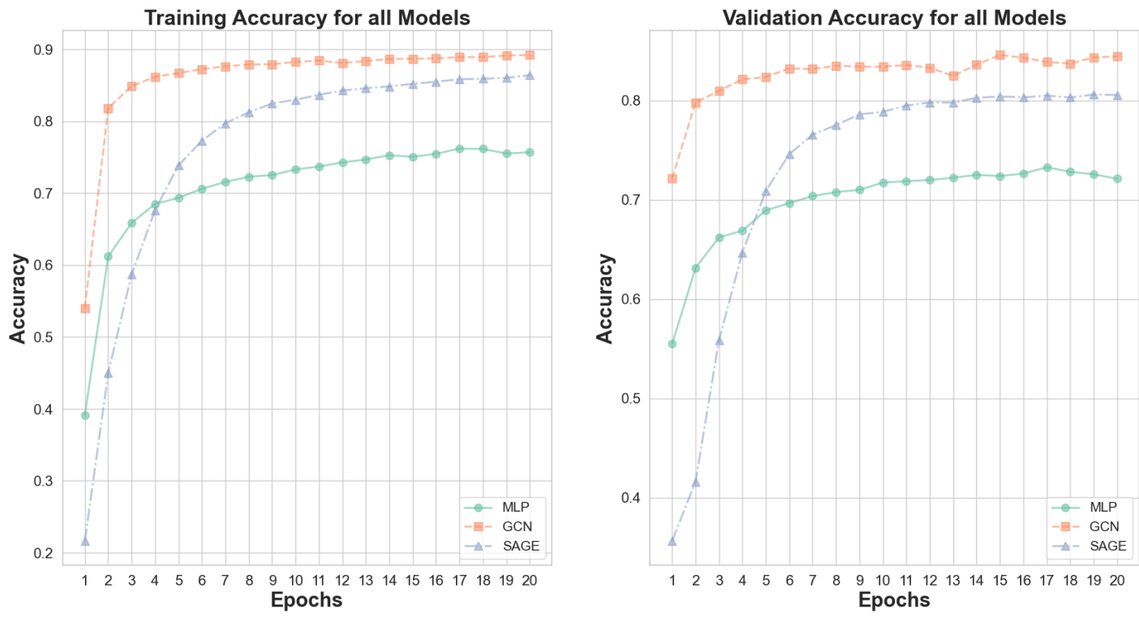

Figure 5: Training and validation accuracy for all models.

- All models were trained for 20 epochs with Adam optimizer and cross-entropy loss.
- All models used 128 neurons in first hidden layers and 47 in the last.
- GNN models used either **GCNConv** or **SAGEConv with mean aggregation.**
- **NeighborLoader processed training data in batches of slightly smaller subgraphs**, allowing both SAGE and GCN models to use a subset of connected nodes for training.
- Learning rates: MLP - 0.001, GCN - 0.0005, SAGE - 0.0001.
- 60% dropout for the GNN models was employed to combat overfitting.

TEST RESULTS:

Metric	MLP	GCN	SAGE
Accuracy	58.61%	65.98%	63.50%
F1-score (Macro)	20.04%	25.38%	22.43%
F1-score (Weighted)	55.29%	62.31%	58.90%

Table 1: Performance Metrics for MLP, GCN, and SAGE Models

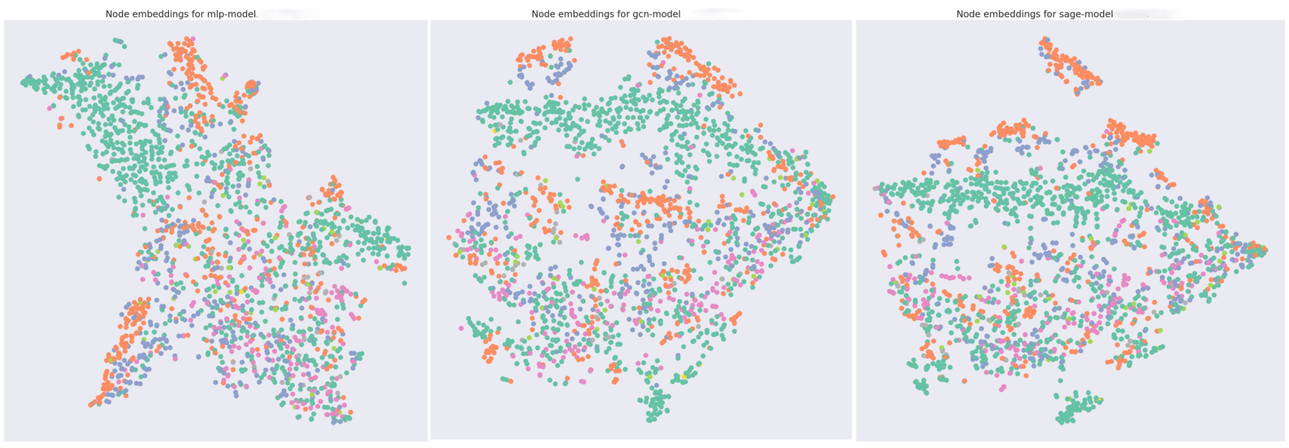

Figure 6: Node embeddings of the first batch from the test data visualized with t-SNE. From left to right we have; MLP - GCN - SAGE.

Test results shows that graph structured data combined with GNNs provide better accuracy. Seemingly, the GNNs also perform better on the bigger classes.

Conclusion

- **Enhanced Performance**: Graph Neural Networks (GNNs) outperformed Multilayer Perceptrons (MLPs) by approximately 8% in accuracy due to GNNs' ability to capture relationships between nodes.
- **Message-Passing Mechanism**: MLPs process labeled nodes independently using only their own features, while GNNs leverage the graph structure to include unlabeled nodes during training via message-passing. This enhances context and improves generalization when applying the GNNs on the test data.
- **Future work**: Deeper models, regularization, and alternative aggregation methods could further boost GNN performance.

References

[1] W. L. Hamilton, R. Ying, and J. Leskovec.
Inductive representation learning on large graphs, 2018.

[2] T. N. Kipf and M. Welling.
Semi-supervised classification with graph convolutional networks, 2017.

anthono@student.chalmers.se, albinst@student.chalmers.se