Machine Learning on Graphs - cs224 Colabs Report

Mohammad Bahrami - 9724133

January 22, 2022

---

1 Colab 2

In this Notebook we are introduced to PyG's Data Handling method and how to create and manipulate the Data objects. Each Data object contains a graph, within this data Object, there are two main parts, `Data.x` and `Data.edge_index`.

`Data.x` contains initial node embeddings of the graph. The `Data.edge_index` is a $2 \times v$ matrix where v is the number of edges. the first row indicates edge sources and the second row indicates their respective destination. Then, in part `Part 1) PyTorch Geometric (Datasets and Data)` we are further introduced to some other properties of `Data` objects.

In part `2) Open Graph Benchmark (OGB)` we load the `obgn-arxiv` dataset and see some properties of it like number of features.

After That, In part `3) GNN: Node Property Prediction` we create a multi-layer GNN model. In each layer, we do a GCNConv, Batch Normalization, Non-Linearity (here ReLU) and dropout sequentially.

Then, The code does some initializations and we can start to train our model to predict node labels. We will do this in 100 epochs and we manage to increase the validation accuracy from 23% to 71% and get a test accuracy of 70%.

In the Next Part, part `4) GNN: Graph Property Prediction` we try to utilize the tools we have so far to try to classify graphs instead of nodes. we will use the same model that we used for the previous part, but this time, after getting the node embeddings of all nodes, we try to aggregate them to obtain graph embeddings. here, we do so by doing a global mean pool over all nodes. then, performing a linear transformation on the result.

Then we perform the learning task on a different dataset to try to learn to predict graph classes. we obtain validation accuracy of 79% and test accuracy of 73% in the end.

2 Colab 3

The main objective of this colab is to implement the GraphSAGE layer. Unfortunately, the time is short(23:53). I will be very short and explain more / answer questions orally on the call. To implement Graph sage we need 3 parts. Forward, Message and aggregation. The message will be the embedding of a node itself, no change needed. The aggregation is done by averaging the messages of a nodes neighbors. and the update is done inside the forward method. where we perform separate linear transforms on the aggregated messages and the node itself and sum them up.