Node Classification & Regression Overview

Overall (Typical) Architecture

- Input features X for all nodes*
- Preprocess features by passing through an MLP, shared across the nodes
- Stack GNN layers to incorporate graph neighborhood information and get node embeddings
- Pass each node embedding through a final MLP classifier/regressor to make a prediction
  - A loss can be calculated using ground truth labels, which then allows backpropagation for optimizing all the various parameters

*There are a number of ways to handle nodes without features

Loss functions

- Same as any other ML problem…
  - Classification often uses Cross Entropy of each training node:

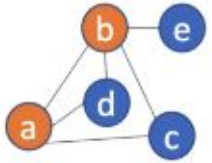
$$L(y, p) = -\sum_i y_i \cdot \log(p_i)$$

  - Regression often uses MSE of each training node:

$$L(y, p) = \frac{1}{N} \sum_i (y_i - p_i)^2$$

Handling nodes without natural features

- Structurally calculated features
  - E.g., node degree, PageRank, motif counts…etc
- Initialize nodes with random values (constant, not learnable)
  - Idea: *nodes sharing many connections will be mapped to similar places since aggregate will be similar, and that's enough*



$$\text{AVG}(N_a) = \text{AVG}([\text{-}0.09,\ 0.15,\ \text{-}0.10,\ 0.25]) = 0.05$$
$$\text{AVG}(N_b) = \text{AVG}([\text{-}0.09,\ 0.15,\ \text{-}0.10,\ 0.25,\ 0.11]) = 0.06$$

Handling nodes without natural features

- Learnable node features: for each node, assign a *d*-dimensional embedding that will be learned as "just another model parameter"
  - These embeddings only receive gradients if they're in the K-hop neighborhood of a *training node*, where K is the number of GNN layers
  - These may also be thought of as learned "Positional Encodings"
- One-hot encoding
  - When multiply this by parameter matrix W, will ultimately dedicate a column of W to each node, making this equivalent to having learnable node features

Node Classification Example

- "ogbn-arxiv" dataset from Open Graph Benchmark:
  - Citation network of Computer Science papers on arXiv
    - 170k papers (nodes)
    - 1.2M edges (citations)
  - Node Features: 128-dimensional vector that's average of word embeddings from title + abstract
  - Target: Determine which of the 40 subject areas a paper belongs
  - Out of time validation: Train=2017, Valid=2018, Test=2019