Tuning Graph Neural Networks with Bayesian Optimisation

Presented by Teodora Reu
Scientific Question

Are Graph Neural Networks appropriate for large scale graph-data optimisation jobs? (in present)
Motivation

- Frameworks like REGAL use 16 layers deep GNNs in the learning pipeline [1]

**Conclusion:** Depth is needed for such jobs
Graph Neural Network face two big issues => cannot afford depth

**Oversmoothing**

**Oversquashing**
Oversmoothing - Iteration 0
Oversmoothing - Iteration 1

\[
(0 + 180) / 2 = 90
\]

\[
(180 + 0 + 0 + 0 + 0) / 5 = 36
\]

\[
(0 + 180) / 2 = 90
\]

\[
(0 + 0 + 180) / 3 = 60
\]
Oversmoothing - Iteration 2 - Observing a trend?

What would happen for 16 layers (iterations)?
Methodology

Pick models from PyTorch-Geometric library:
- GCN [2]
- SSGConv (tackles oversmoothing) [3]
- ChebConv [4]

Benchmark graph dataset:
- Cora (aprox. 3000 nodes), PubMed [5] (aprox. 19000 nodes)

Apply Bayesian Optimisation Algorithm to find Best Depth on a Benchmark Classification Task:
- Question: What will be the best depth possible for these datasets? (varying from 1 to 100)
Analysis

Best depth:

- is it 3 or 20? (my guess is 3)

What is the longest path in the current graph dataset (diameter):

- is it 20 or 100 or 500? (my guess is 100)

If best accuracy for classification task on graphs is obtained with layers with depth 3 are GNNs really appropriate for Large Scale Data Optimisation jobs where ideally nodes would hold information about their furthest neighbours?
Results & Takeaways

- Will Tune GNNs with Bayesian Optimisation (varying depth) and add contribution to Pytorch Geometric
- Obtain results on what the ideal depth of GNN classifier for benchmark graph dataset
- Compare depth with how wide the initial graph was (its diameter)

What does this mean for optimisation methods that leverage GNNs and need depth in order to function properly?
Citations


Thank you for listening!

Questions?