Graph Architecture Optimization using Simulated Annealing

Nian Dongdong, Qin Dayu

TBSI, Tsinghua University, China

Abstract

- Use heuristic algorithm (Simulated Annealing) to prune graph to reduce its complexity, meanwhile, maintain the prediction accuracy.
- Some redundant edges are less important indeed. After removing them, we get better performance.
- There does exist sub-graphs with more simple architectures.
- Use Zachary's karate club node classification problem as our background problem.

Motivation

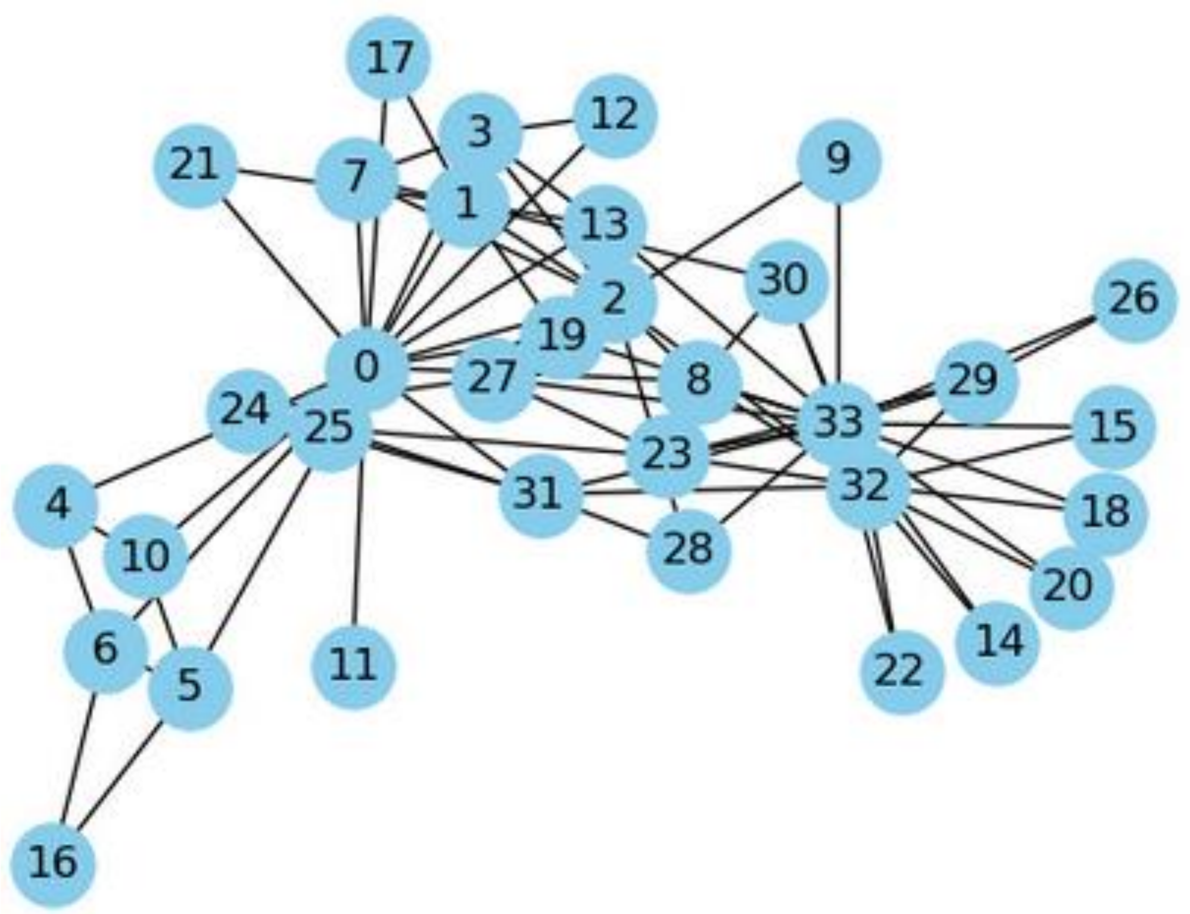

Fig.1 the figure of original graph

- If a graph's architecture is too complicated, doing tasks on this graph maybe time-consuming.
- The figure above illustrates the architecture of our original graph. We want to cut some redundant edges while maintain the accuracy when doing tasks based on the graph.
- Inspired by [2] applying Stimulated Annealing on fully connected network pruning, We hope we can simplify the graph structure as much as we can without vast performance decrease on graph prediction task, which allows us to highlight the key relationship and study the main edges.

Methods

1. Use the karate_club dataset to create the original graph
2. Train a GCN based on the original graph to classify the nodes
3. Randomly prune r% edges of the original graph
4. Randomly choose one connected edge and one pruned edge
5. Switch the connection status of two selected edges
6. Use the GCN to classify the nodes of the new pruned graph and compute the loss
7. If the loss decrease, accept the new configuration. If the loss increases, accept the new configuration with a certain probability
8. After get the optimal configuration, use the GCN to classify the nodes

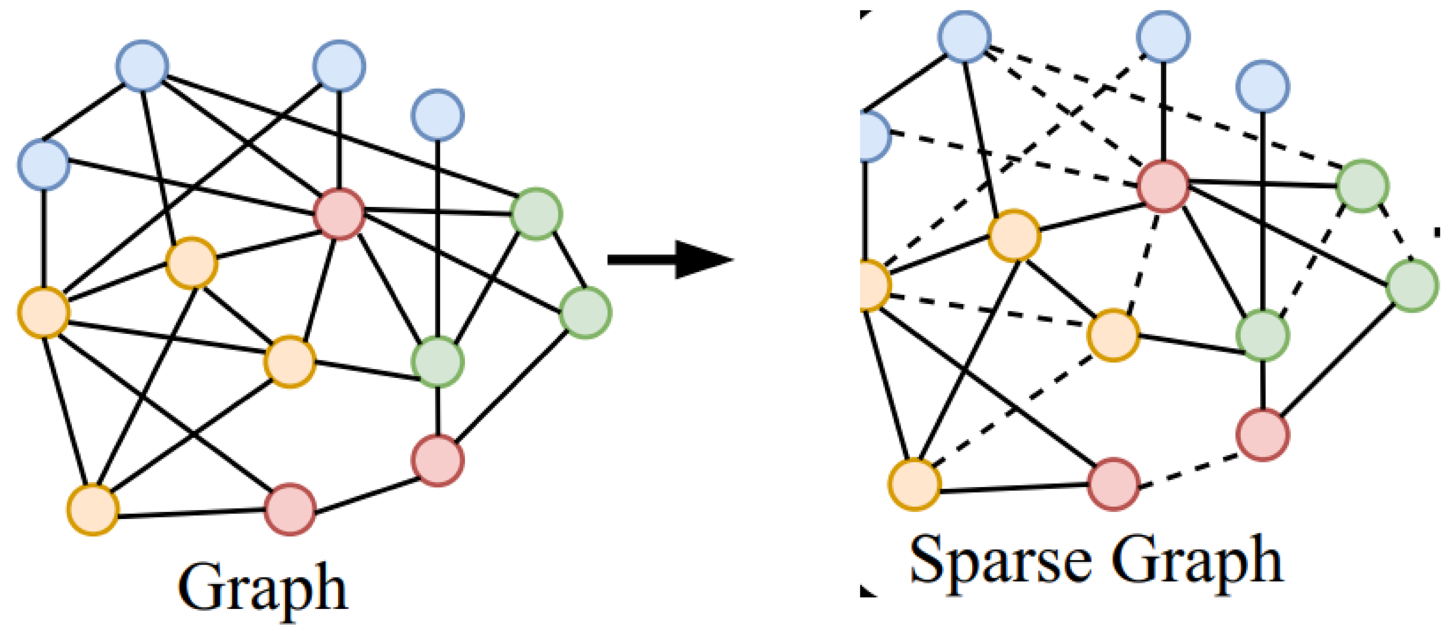

Fig.2 an illustration of pruning graphs

Results

Prune / MLL	0%	1%	2%	5%	10%	20%
1	0.91	0.91	0.94	0.97	0.94	0.88
10	0.89	0.88	0.88	0.90	0.89	0.86
20	0.90	0.85	0.89	0.90	0.90	0.85
50	0.92	0.86	0.92	0.92	0.91	0.87

Tab.1 comparison of results with different pruning percentages and metropolis length

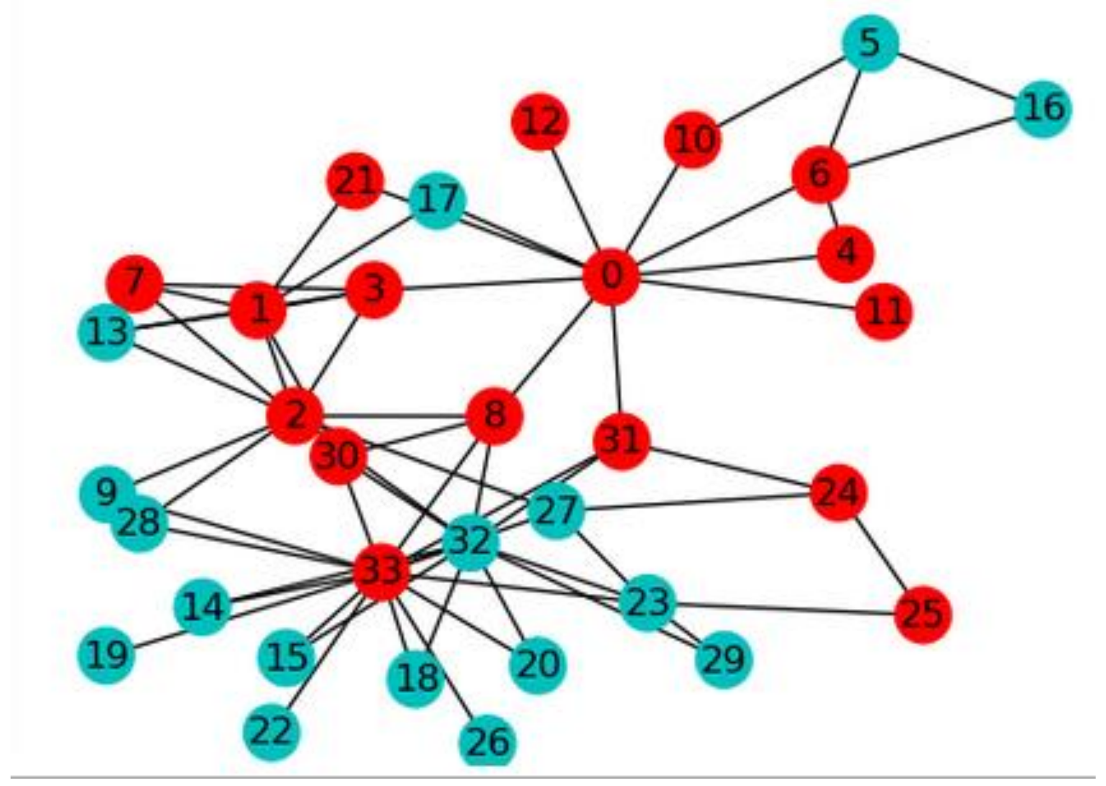

Fig.3 the final result of the graph with 20% edges pruned

1. From the results, we can observe that there exists sub-graphs after pruning and the prediction accuracy doesn't decrease too much.
2. When we prune 2% and 5% edges of the graph, the accuracy increase. This result is impressive, which can tell us some edges are superfluous indeed. After prune them, we get better performance.
3. When we prune more edges than 20%, the performance will decline deeply. That may because the graph we used is too small, which means each edge is comparatively important.

Conclusions

1. First of all, we have discovered that there does exist subgraphs with simpler architectures
2. Node classification tasks on these subgraphs doesn't have vast performance decrease

References

[1]Chen T, Sui Y, Chen X, et al. A unified lottery ticket hypothesis for graph neural networks[C]//International Conference on Machine Learning. PMLR, 2021: 1695-1706.

[2] Kuo C L, Kuruoglu E E, Chan W K V. Neural Network Structure Optimization by Simulated Annealing[J]. Entropy, 2022, 24(3): 348.

[3] Wang L, Huang W, Zhang M, et al. Pruning graph neural networks by evaluating edge properties[J]. Knowledge-Based Systems, 2022, 256: 109847.