Complexity Analysis of Graph Convolutional Networks and in Attention based GNN

Presenter: Derrick Blakely

June 7, 2019

University of Virginia

https://qdata.github.io/deep2Read/

Definitions

- Graph $G = (V, E, A)$
- $N = |V|$ nodes, $N \times N$ adjacency matrix A
- Average degree of d
- $X \in \mathbb{R}^{N \times F}$: embeddings of all $v \in V$
- Each embedding $x \in \mathbb{R}^F$
- D: the degree matrix of G
- $\hat{A}$: A with all self-loops included
- $\hat{D}$: D with all self-loops included

Graph Convolutional Networks (GCN) [3]

Normalized adjacency matrix with self-loops:

$$\mathsf{A}' = \hat{\mathsf{D}}^{-\frac{1}{2}} \hat{\mathsf{A}} \hat{\mathsf{D}}^{-\frac{1}{2}} \tag{1}$$

Output of lth GCN layer:

$$\mathsf{X}^{l+1} = \sigma(\mathsf{A}'\mathsf{X}^l\mathsf{W}^l) \tag{2}$$

Alternatively:

$$\mathsf{Z}^l = \mathsf{X}^l\mathsf{W}^l \tag{3}$$

$$\mathsf{X}^{l+1} = \sigma(\mathsf{A}'\mathsf{Z}^l) \tag{4}$$

(For simplicity, assume every layer is a mapping $f : \mathbb{R}^{N \times F} \rightarrow \mathbb{R}^{N \times F}$)

Implementation and Semantics

1. $\mathsf{Z}^l = \mathsf{X}^l\mathsf{W}^l$: input feature transformation. Dense matrix multiplication
2. $\mathsf{A}'\mathsf{Z}^l$: GAS/message passing
3. $\sigma(\cdot)$: nonlinearity

GAS/Message Passing

Everyone [5][2][4][1] computes $A'Z^l$ roughly as follows:

1. Create messages: normalize each $z_i \in \mathrm{Z^l}$
2. Scatter: compute $\mathrm{A'Z^l}$ with a sparse multiplication
3. Update (optional): additional update to each embedding (e.g., additive bias)

Time and Space Complexity

- $\mathrm{Z^l} = \mathrm{X}^l\mathrm{W}^l$: dense $(N \times F) \times (F \times F)$ multiplication $\rightarrow O(NF^2)$ time, $O(NF + F^2)$ space
- $\mathrm{A}'\mathrm{Z}^l$: sparse $(N \times N) \times (N \times F)$ multiplication $\rightarrow$ $O(NdF) = O(|E|F)$ time, $O(NF)$ space
- $\sigma(\mathrm{A}'\mathrm{Z}^l)$: $O(NF)$ time for ReLU, $O(NF)$ space

1 Layer:
Time: $O(NF^2 + |E|F + NF) = O(NF^2 + |E|F)$
Space: $O(NF + F^2)$
L Layers:
Time: $O(LNF^2 + L|E|F)$
Space: $O(LNF + LF^2)$

Backward Pass

We seek to compute:

$$\frac{\partial \mathcal{L}}{\partial \mathsf{W}^1} = \Big(\frac{\partial \mathcal{L}}{\partial \hat{\mathsf{Y}}}\Big)\Big(\frac{\partial \hat{\mathsf{Y}}}{\partial \mathsf{Z}^L}\Big)\Big(\frac{\partial \mathsf{Z}^L}{\partial \mathsf{X}^L}\Big)\dots\Big(\frac{\partial \mathsf{Z}^l}{\partial \mathsf{X}^l}\Big)\Big(\frac{\partial \mathsf{X}^l}{\partial \mathsf{Z}^{l-1}}\Big)\dots\Big(\frac{\partial \mathsf{X}^2}{\partial \mathsf{Z}^1}\Big)\Big(\frac{\partial \mathsf{Z}^1}{\partial \mathsf{W}^1}\Big) \tag{5}$$

$$\frac{\partial \mathcal{L}}{\partial \mathsf{X}^1} = \Big(\frac{\partial \mathcal{L}}{\partial \hat{\mathsf{Y}}}\Big)\Big(\frac{\partial \hat{\mathsf{Y}}}{\partial \mathsf{Z}^L}\Big)\Big(\frac{\partial \mathsf{Z}^L}{\partial \mathsf{X}^L}\Big)\dots\Big(\frac{\partial \mathsf{Z}^l}{\partial \mathsf{X}^l}\Big)\Big(\frac{\partial \mathsf{X}^l}{\partial \mathsf{Z}^{l-1}}\Big)\dots\Big(\frac{\partial \mathsf{X}^2}{\partial \mathsf{Z}^1}\Big)\Big(\frac{\partial \mathsf{Z}^1}{\partial \mathsf{X}^1}\Big) \tag{6}$$

Which can be done efficiently when formulated as:

$$\frac{\partial \mathcal{L}}{\partial \mathsf{W}^{l-1}} = \Big(\mathsf{X}^{l-1}\Big)^\top \Big(\mathsf{A}'\Big)^\top \Big(\frac{\partial \mathcal{L}}{\partial \mathsf{X}^l}\Big) \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial \mathsf{X}^{l-1}} = \Big(\mathsf{A}'\Big)^\top \Big(\frac{\partial \mathcal{L}}{\partial \mathsf{X}^l}\Big)\Big(\mathsf{W}^{l-1}\Big)^\top \tag{8}$$

Deep graph library (dgl), 2019.
Accessed: 2019-06-06.

M. Fey and J. E. Lenssen.
Fast graph representation learning with pytorch geometric.
arXiv preprint arXiv:1903.02428, 2019.

T. N. Kipf and M. Welling.
Semi-supervised classification with graph convolutional networks.
arXiv preprint arXiv:1609.02907, 2016.

L. Ma, Z. Yang, Y. Miao, J. Xue, M. Wu, L. Zhou, and Y. Dai.
Towards efficient large-scale graph neural network computing.
arXiv preprint arXiv:1810.08403, 2018.

S. Xu, H. Zhang, G. Neubig, W. Dai, J. K. Kim, Z. Deng, Q. Ho, G. Yang, and E. P. Xing.
Cavs: An efficient runtime system for dynamic neural networks.
In *2018 {USENIX} Annual Technical Conference ({USENIX} {ATC} 18)*, pages 937–950, 2018.