Graph Neural Network Introduction

Table-ronde IBISC du 13/11/2020

Graph definition

- A graph is defined as $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of connected n vertices/nodes and $\mathcal{E}$ is a set of m edges.
- An attribute, a d-dimensional feature vector, can characterize each vertex : $X\ (x_1, \ldots, x_n)$.
- $\mathcal{N}(v)$ refers to the neighborhood of the node v.
- $A \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ defines the adjacency matrix.

Objectives of a Graph Neural Network (GNN)

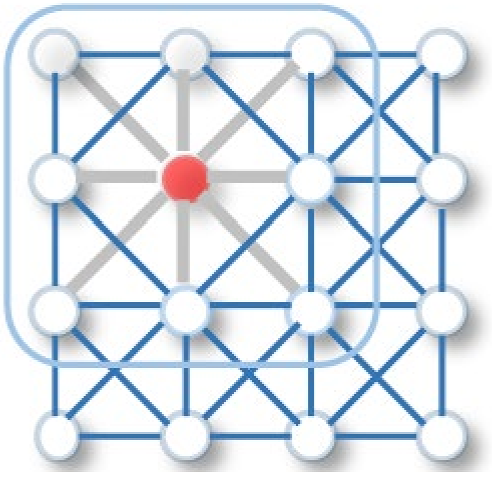
Deep learning approach for **graph-based data** for supervised as unsupervised tasks.

The aims of these methods is to:

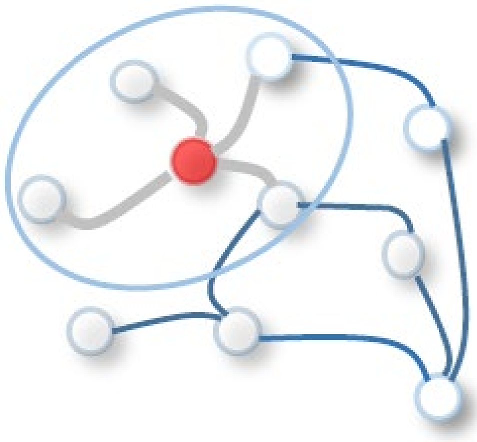
- predict **final states** of the graph components at global or local scale;
- build a **high-level representation** of any node recursively from the representation vectors of its **neighborhood**.

Graph Convolution Network (GCN)

- **Spatial convolution:**



vs



2D Convolution with a CNN

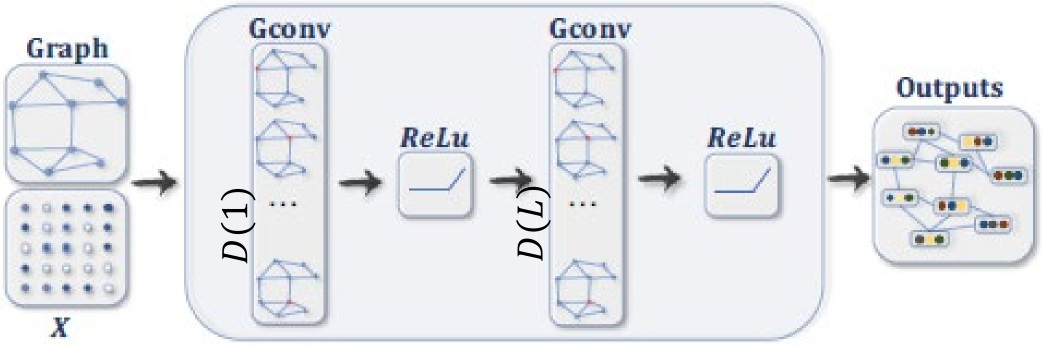
Graph Convolution

- Can deal with graphs with **different structure.**
- **Key works:** GCN (Kipf et al. (2017)), GraphSAGE (Hamilton et al. (2017)), GIN (Xu et al. (2019)), GAT (Veličković et al. (2018)).

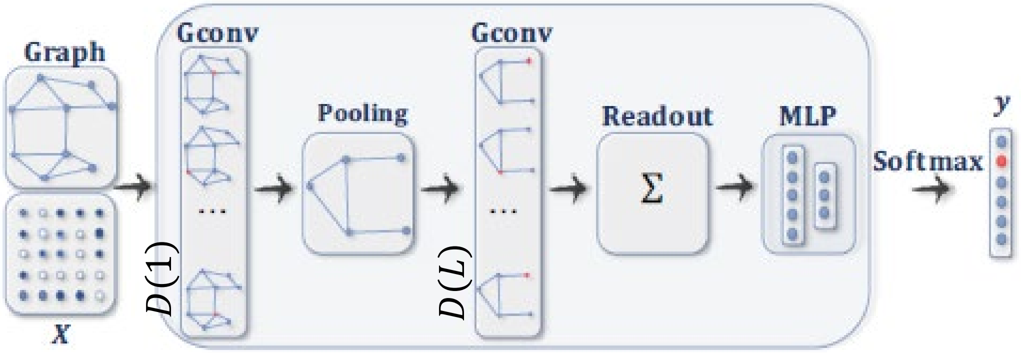
Pictures from the article of Wu et al. (2019)

Types of classification

- 3 types: edge, **node**, **graph**.



Node classification



Graph classification

Pictures from the article of Wu et al. (2019)

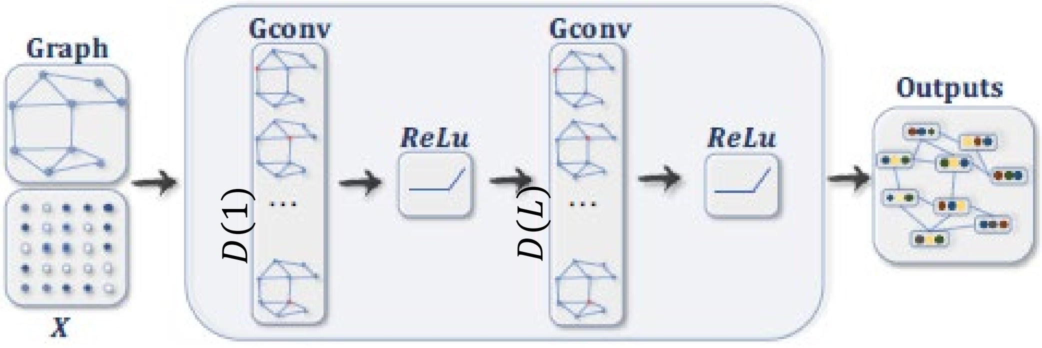
Types of classification

- 3 types: edge, **node**, **graph**.

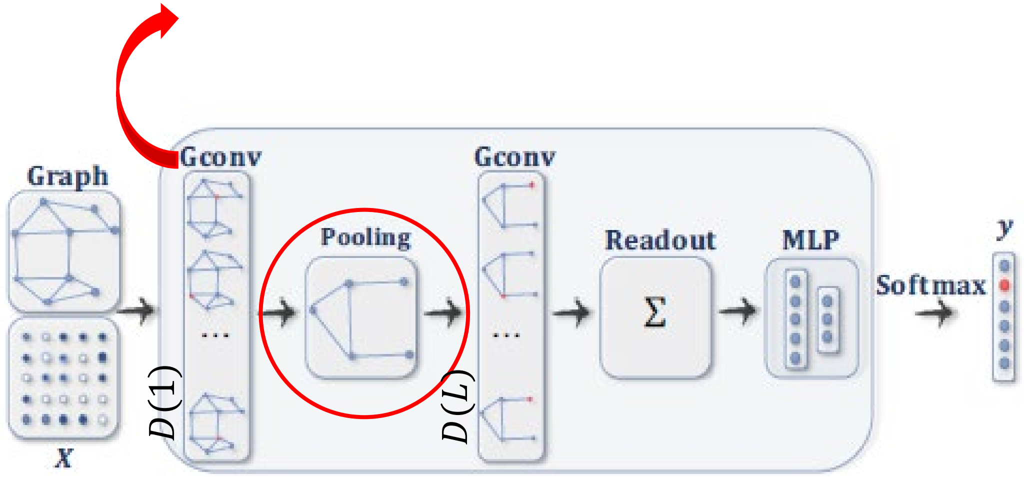
$$X_{l+1} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}X_l W_l)$$

where $\hat{A} = (A + \mathbb{I})$, $\hat{D} = \sum_j \hat{A}_{i,j}$, σ is the activation function,

and W_l is the trainable weight matrix.



Node classification

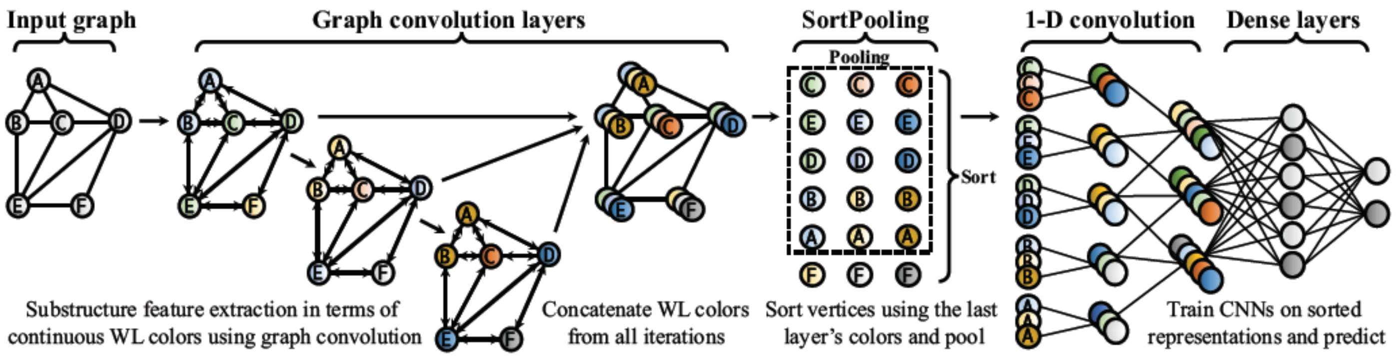


Graph classification

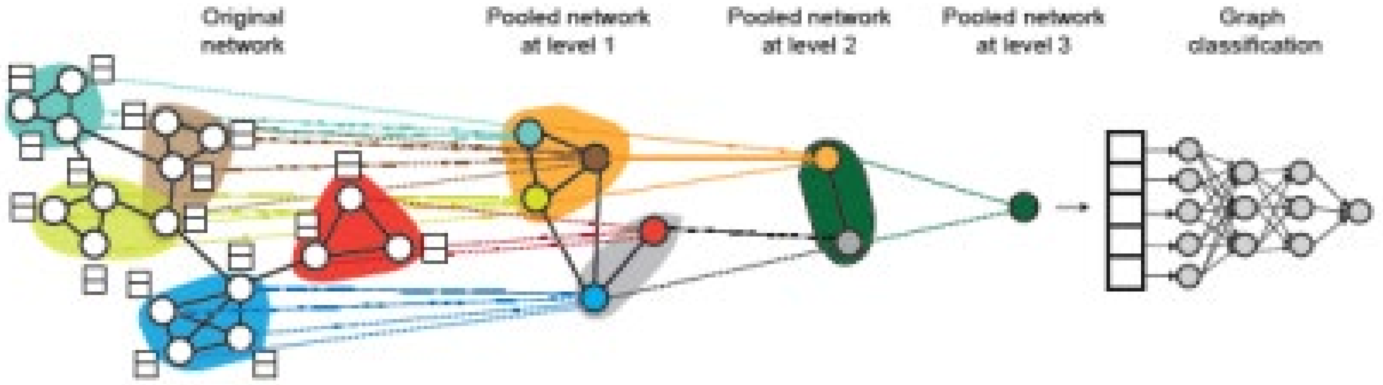
Pictures from the article of Wu et al. (2019)

Pooling layers

- SortPooling (DGCNN)



- DiffPool



$$H^{(l)} = S^{(l)T} H^{(l)}$$

where $S^{(l)} = softmax(GNN(A^{(l-1)}, H^{(l-1)}))$,
$S^{(l)} \in \mathbb{R}^{n_{(l-1)} \times n_{(l)}}$,
$H^{(l)} \in \mathbb{R}^{n_{(l)} \times d_{(l)}}$ and
$H^{(l-1)} \in \mathbb{R}^{n_{(l-1)} \times d_{(l-1)}}$.

Bibliography

- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21.
- Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2019). Graph Neural Networks: A Review of Methods and Applications. *ArXiv:1812.08434 [Cs, Stat]*.
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for Quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, *70*, 1263–1272.
- Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations.*
- Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems* (pp. 1024–1034).
- Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How Powerful are Graph Neural Networks? In *International Conference on Learning Representations.*
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations.*
- Zhang, M., Cui, Z., Neumann, M., & Chen, Y. (2018). An End-to-End Deep Learning Architecture for Graph Classification. In *AAAI Conference on Artificial Intelligence* (p. 8).
- Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., & Leskovec, J. (2018). Hierarchical Graph Representation Learning with Differentiable Pooling. In *Advances in Neural Information Processing Systems* (pp. 4800–4810).
- Gao, H., & Ji, S. (2019). Graph U-Nets. In *International Conference on Machine Learning* (Vol. 97, pp. 2083–2092).
- Lee, J., Lee, I., & Kang, J. (2019). Self-Attention Graph Pooling. In *International Conference on Machine Learning* (3734–3743).

Integration of edge features for node/graph classification:

- Gong, L., & Cheng, Q. (2019). Exploiting Edge Features for Graph Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9203–9211.
- Simonovsky, M., & Komodakis, N. (2017). Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 29–38.