# Siamese Graph Similarity Networks

Jonathan Geuter

APMTH 220 Poster Session, Harvard University, Cambridge, Massachusetts 02138

HARVARD UNIVERSITY

Harvard John A. Paulson School of Engineering and Applied Sciences

## Abstract

Predicting graph similarity, i.e. a score that corresponds to how similar two graphs are, is an important task in graph learning, and has applications in various areas, such as drug discovery, where one might be interested in computing similarities between compounds, or recommendation systems, where similarity between user-item graphs can be used for product or content recommendations. One approach to computing graph similarities lies in *Siamese Graph Neural Networks* [1, 2, 3, 4]. In this project, we train a GNN to predict a *Graph Edit Distance* (GED) based similarity measure between graphs from the AIDS dataset[a], and compare it against a MLP baseline. A Siamese network is used to embed two graphs into the same embedding space; then, a joint similarity between the two embedding vectors is computed using a *Neural Tensor Network* (NTN) [5].

## Background and Notation

**GNNs:**

- widely used to process graph data and compute graph-based features
- graph convolutions (GCNs) are convolutions that act on graphs $G = (V, E)$: for a node $v \in V$ with neighbours $\mathcal{N}(v)$ (containing $v$) and degree $d_v$,

$$\text{conv}(h_v^{(l+1)}) = f\left(W^{(l)} \sum_{u \in \mathcal{N}(v)} \frac{h_u^{(l)}}{\sqrt{d_u d_v}} + b^{(l)}\right) \tag{1}$$

**GED:**

- common measure of dissimilarity/distance of two graphs, but computationally extremely expensive
- $\text{GED}(G_1, G_2)$ for two graphs $G_1$, $G_2$ is the number of edit operations (insertion/deletion of a node/edge, or relabeling of a node) needed to convert $G_1$ to $G_2$

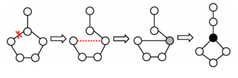

Figure 1. The GED between the graph on the left and the graph on the right is 3; the operations are "delete an edge", "insert an edge", and "relabel a node". (From [1].)

[a] https://wiki.nci.nih.gov/display/NCIDTPdata/AIDS+Antiviral+Screen+Data

## Approach

**Data:**

- define graph similarity between $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ via

$$\text{sim}(G_1, G_2) = \exp\left(-2\frac{\text{GED}(G_1, G_2)}{|V_1| + |V_2|}\right) \in (0, 1] \tag{2}$$

- 10.000 samples $(G_1, G_2, \text{sim}(G_1, G_2))$ from 100 graphs from AIDS dataset (chemical compounds, $\leq 10$ nodes, $38$ features/node)

**Network:**

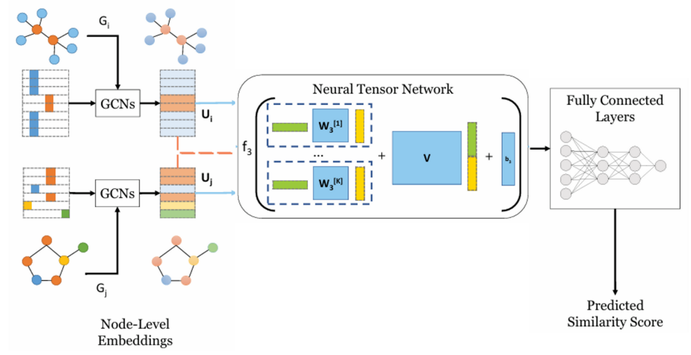


Figure 2. The Siamese Network's architecture (adapted from [1]).

- Siamese Network (identical for both graphs): 3 GCNs (hidden_dim=64,32,16) followed by graph-mean-pooling
- NTN: for two graph embeddings $h_1, h_2 \in \mathbb{R}^d$,

$$\text{NTN}(h_1, h_2) = \tanh\left(h_1^\top W h_2 + V \begin{bmatrix} h1 \\ h2 \end{bmatrix} + b\right),$$

$W \in \mathbb{R}^{d \times d \times k}$, $V \in \mathbb{R}^{k \times 2d}$, $b \in \mathbb{R}^k$, $k = 16$

- Fully conn. layers: 3 layers (hidden_dim=8,4,1), followed by sigmoid
- All activations are tanh; 31217 trainable parameters total

**Training:** 500 epochs with 100 samples each (all 100 graphs from the dataset, where $G_1$ and $G_2$ are each a random permutation of the 100 graphs); Adam optimizer; L1 loss; batch_size=50

**Testing:** relative error

$$err(G_1, G_2) = \frac{|\text{sim}(G_1, G_2) - \text{net}(G_1, G_2)|}{\text{sim}(G_1, G_2)} \tag{3}$$

averaged over 1000 test samples from 32 graphs from AIDS dataset

## Results

- comparison against a FCN (*fully connected network*) Siamese baseline, where the GCNs are substituted with fully connected layers acting on the mean-pooled node representations of the graphs; similar in size (28177 parameters) to the GCN Siamese network

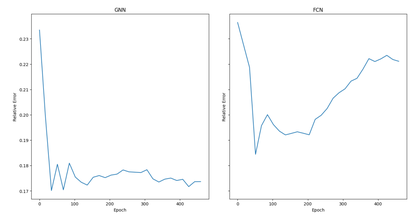


Figure 3. GCN-based Siamese network (left) vs FCN-based Siamese network (right).

## Discussion

- successfully learned graph similarities for graphs corresponding to chemical compounds using a GCN Siamese network
- training progress potentially very limited by the small training dataset, which is computationally very expensive to create

## Future Direction

- scale up training to better understand potential of the algorithm
- can we learn better graph embeddings?
- can we incorporate edge features?
- scale up to larger graphs (can we use GEDs computed on smaller graphs to learn GEDs on larger graphs?)

## References

[1] Yunsheng Bai, Hao Ding, Song Bian, Ting Chen, Yizhou Sun, and Wei Wang. SimGNN: A Neural Network Approach to Fast Graph Similarity Computation, March 2020. URL http://arxiv.org/abs/1808.05689. arXiv:1808.05689 [cs, stat].

[2] Ushasi Chaudhuri, Biplab Banerjee, and Avik Bhattacharya. Siamese graph convolutional network for content based remote sensing image retrieval. *Computer Vision and Image Understanding*, 184:22--30, July 2019. ISSN 1077-3142. doi:10.1016/j.cviu.2019.04.004. URL https://www.sciencedirect.com/science/article/pii/S1077314219300578.

[3] Sofia Ira Ktena, Sarah Parisot, Enzo Ferrante, Martin Rajchl, Matthew Lee, Ben Glocker, and Daniel Rueckert. Metric learning with spectral graph convolutions on brain connectivity networks. *NeuroImage*, 169:431--442, April 2018. ISSN 1095-9572. doi:10.1016/j.neuroimage.2017.12.052.

[4] Guixiang Ma, Nesreen K. Ahmed, Ted Willke, Dipanjan Sengupta, Michael W. Cole, Nicholas B. Turk-Browne, and Philip S. Yu. Similarity Learning with Higher-Order Graph Convolutions for Brain Network Analysis, May 2019. URL http://arxiv.org/abs/1811.02662. arXiv:1811.02662 [cs, stat].

[5] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/b337e84de8752b27eda3a12363109e80-Paper.pdf.