Using Spiking Neural Networks for Temporal Graph Representation Learning

What is the computational problem?

Graphs are a powerful tool for representing complex relationships between entities. However, most existing graph representation learning methods are designed for static graphs, and cannot capture the temporal dynamics of the graph. This is a significant limitation, as many real-world graphs are dynamic, and evolve over time. For example, in social networks people follow and unfollow people all the time, and in financial transaction networks, new transactions are constantly being made. Models that can capture these dynamics can come in very handy.

Why is it interesting and important?

Temporal graph representation learning is a very new and exciting field. In fact the temporal graph workshop at Neurips has only been running for two years now! There is so much to explore in this field and it has the potential to have a huge impact on a wide range of applications. For example, it could be used to predict future links in a social network, or to detect anomalies in a financial transaction network, both of which are actually not static machine learning problems.

Why is it hard? What's been done now? Where/Why have previous approaches failed?

There has been some work on temporal graph representation learning, but most of it is based on static graph representation learning methods. For example, most existing works use some kind of self attention mechanism Xu et al. 2019 or some kind of RNN Kumar, Zhang, and Leskovec 2019 to learn the temporal aspects of the graph, where as at each time step the message passing is still done with a traditional graph convolutional neural network.

The issue is that they are computationally expensive, and do not scale well to large graphs. Graph representation learning by itself is an extremely compute heavy task and adding temporal dynamics to it only makes it worse. Additionally temporal graph learning has been disjointed and no central dataset exists to measure model performance, so the idea would be to benchmark an existing SNN approach and compare it to other RNN or attention approaches.

What are the key components of your neuroinspired approach? What model/simulation environments will be used to validate your approach?

Traditional graph convolution networks (GCN) operate on the idea of message passing between nodes. We propose to extend this idea to temporal graphs by using spiking neural networks (SNNs) for message passing at each time step. Then we aggregate the representation learned at each timestep to perform the classification task (link prediction or node classification).

More specifically we would use the leaky integrate and fire model framework proposed by Li et al. 2023. I would then like to build on the model by including the neighborhood modeling proposed by Zhou et al. 2022 to improve the robustness of the model. Because this is an extremely nascent field, I would also love to explore other spiking models as well as borrow ideas from static graph representation learning methods especially for node level classification.

The newly proposed Temporal Graph Benchmark provides a perfect environment to validate our approach. It provides a wide range of datasets and tasks to evaluate the performance of our model. Huang et al. 2023

Goals

The first goal is to replicate the results of the framework in Li et al. 2023. To do this, we would run the code from the authors' github on the DBLP, Tmall, and Patent.

We would then benchmark the approach proposed by Liu et al. 2023 on the Temporal Graph Benchmark and compare it to other RNN or attention approaches.

Finally, we would try improve the model by borrowing ideas from different papers and benchmark our own approach.