Journal Report 8
10/28/19-11/10/19
Emily Ye
Computer Systems Research Lab
Period 2, White

Daily Log

Monday October 28

Worked on presentation.

Tuesday October 29

Listened to other people's presentations and edited mine based on feedback.

Continued working on topology adaptive graph convolutional network (TAGCN) with Spektral and QM9 dataset.

Thursday October 31

Finalized and practiced presentation, listened to other people's presentations, and gave my presentation.

Wednesday November 6

Debugged training loop for TAGCN with Spektral and QM9 dataset.

Thursday November 7

Finished debugging training loop for TAGCN with Spektral and QM9 dataset.

Played around with layers to reduce loss value.

Timeline

October 28	Finish modifying TAGCN code to be compatible with Spektral for Cora dataset	Finished modifying Cora TAGCN code for Spektral compatibility and began modifying it to work with QM9 dataset
November 4	Finish implementing Du, et al.'s topology adaptive graph convolutional network with Spektral and QM9 dataset	No, didn't have enough time to finish due to presentations taking up part of the week
November 11	Finish implementing Du, et al.'s topology adaptive graph convolutional network with Spektral and QM9 dataset	Basically finished, but will continue testing and trying to reduce the loss value next week
November 18	Code accuracy graph (predictions v. true values) for TAGCN	
November 25	Determine which kind of GCN to use for final project after comparing TAGCN with previously-implemented edge-conditioned GCN	

Reflection

The main goal for these two weeks was to finish implementing the TAGCN so that it works with both Spektral methods and the QM9 dataset. Although I intended to finish this by the end of last week, working on my presentation and listening to others' presentations took up a lot of the week, so I didn't have as much time as I had expected. I had a lot of trouble with the training loop and debugging it because of how different the QM9 dataset is from the Cora dataset, for which the code I am modeling my TAGCN off of was written, but I was able to get down to a loss value of 0.732. Over the next week, I will continue testing my TAGCN to make sure it is working properly and modifying the layers to reduce the loss.

My goal for the next week is to code an accuracy graph to show the predictions against the true values for the TAGCN like I did for the edge-conditioned GCN. Since it is difficult to calculate accuracy for GCNs using the QM9 dataset, this will allow me to just compare the R^2 values of the TAGCN accuracy graph and the edge-conditioned GCN accuracy graph. Hopefully, this will not take the entire week, so I have some extra time to put finishing touches on the TAGCN implementation at the start of the week.

By Thanksgiving break, I plan to (finally!) decide which kind of GCN to use for the final project. I will also discuss next steps with my partner, specifically how to piece together my GCN implementation and his multitask learning implementation to create a multitask GCN.