Journal Report 5

9/30/19-10/6/19

Emily Ye
Computer Systems Research Lab
Period 2, White

Daily Log

Monday September 30

Reviewed TensorFlow API, specifically for the `Model` class and the `predict_on_batch` method (as opposed to the `predict` method).

Added code from edge-conditioned GCN for making an accuracy graph into code for graph-batch GCN, but got several errors.

Tuesday October 1

Changed references in plotting code to `predict_on_batch` method to use `predict` method instead, but still encountered `ValueErrors` and `AttributeErrors`.

Reviewed Spektral documentation for graph batching again.

Continued making edits to accuracy graph code and tried passing different arguments as inputs into `predict_on_batch` method.

Thursday October 3

Read Li, et al.'s paper on adaptive GCNs (2018) and Du, et al.'s paper on topology adaptive GCNs (2017).

Made ramen :)

Timeline

Date	Goal	Met
September 23	Complete GCN with edge-conditioned convolutions	Completed edge-conditioned GCN with test loss of about 0.03
September 30	Complete GCN in graph batch mode for comparison with edge-conditioned GCN	Completed basic graph batch GCN, but did not finish coding accuracy graph for comparison with edge-conditioned GCN
October 7	Code accuracy graph (predictions v. true values) for graph batch GCN	No, due to difficulty understanding the batching
October 14	Research and begin implementing other kinds of GCNs (such as adaptive graph convolutional network presented by Li, et al. [2018])	
October 21	Finish implementing Li, et al.'s adaptive graph convolutional network; depending on research, either begin implementing another kind of GCN or determine which kind of GCN to use for final project	

Reflection

My main goal this week was to code an accuracy graph for the graph batch GCN and find a R^2 value for how well it predictions aligned with the true values. Unfortunately, I could not understand how the batching in the graph batch GCN code worked, even after reading through Spektral and TensorFlow documentation multiple times. Eventually, I decided it would probably be better to spend my time on researching other types of GCNs.

It had already taken me around a week to write code for a relatively simple graph that it took a far shorter amount of time to write for the edge-conditioned GCN, and I also could not really understand how the Spektral and TensorFlow methods were working together for the graph batch GCN. Thus, since the edge-conditioned GCN already had extremely high accuracy and the graph batch GCN was so difficult to work with, I decided to stick with the edge-conditioned GCN as an option for the final project rather than the graph batch GCN.

For the next few weeks, I plan to implement some more GCNs that I researched, including the adaptive GCN mentioned in Li, et al. (2018). Hopefully, these will be easier to work with, so I can actually create an accuracy graph and compare them with the edge-conditioned GCN. However, I think it is likely that, because the edge-conditioned GCN is so easy to work with and already has such high accuracy, we will go with that one for the final project.