Journal Report 3

9/16/19-9/22/19
Emily Ye
Computer Systems Research Lab
Period 2, White

Daily Log

Monday September 16

Read Simonovsky and Komodakis's paper (2017) on edge-conditioned convolutions.

Tuesday September 17

Ran Spektral example code for edge-conditioned GCN and got test loss of about 0.063 (training on sample size of 2000).

Edited edge-conditioned layers of Spektral example code to reduce test loss to about 0.032 (training on sample size of 2000).

While GCN was training, read part of Li et al.'s paper (2017) on gated attention global pooling, which is used in the Spektral edge-conditioned GCN example.

Thursday September 19

Trained edited Spektral edge-conditioned GCN on sample size of 10000 and got test loss of about 0.030.

Finished reading Li et al.'s paper (2017) on global attention pooling.

Began implementing a second GCN. This one is based on Spektral example for a GCN in graph batch mode and uses normal graph convolutional layers without edge-conditioning, as well as global attention pooling.

Timeline

Date	Goal	Met
September 9	Read in data from QM9 dataset	Used Spektral library to load QM9 data
September 16	Create a basic GCN using Spektral	Began creating GCN using Spektral with edge-conditioned convolutions
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	
October 7	Determine which kind of GCN to use for final project	

Reflection

My goal this week was to better understand and improve upon the Spektral example code for an edge-conditioned GCN. Looking at the example code, I was unfamiliar with edge-conditioned convolutions and attention pooling, which were both used in the Spektral code, so I read papers on those. I also modified the Spektral code to improve the accuracy of the network. Comparing Figures 1 and 2 on the next page, the data points in Figure 2 are more closely aligned with the line of best fit, which shows the improved accuracy of predictions.

One problem I faced was long training times for the GCN, especially when I trained it on 10000 samples. I learned that a good use of time was reading papers about any of the GCN concepts I did not understand while the network was training. Some things I can review over the next few weeks if I am training networks include density functional theory (and other quantum topics that will be important), Psi4 documentation, and multitask learning. Even though my partner is doing the majority of work with these topics, I still want to better my understanding before we start applying multitask learning to a GCN.

For next week, I plan to implement the graph-batch GCN, which Spektral also provides an example for, and improve it. After implementing both kinds of GCNs, I will compare their accuracies in order to figure out which kind of GCN will be better for our final project.

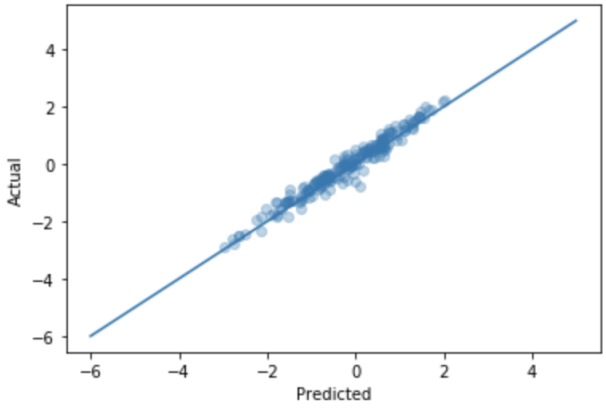

Figure 1: Accuracy of Spektral example edge-conditioned GCN on 2000 samples

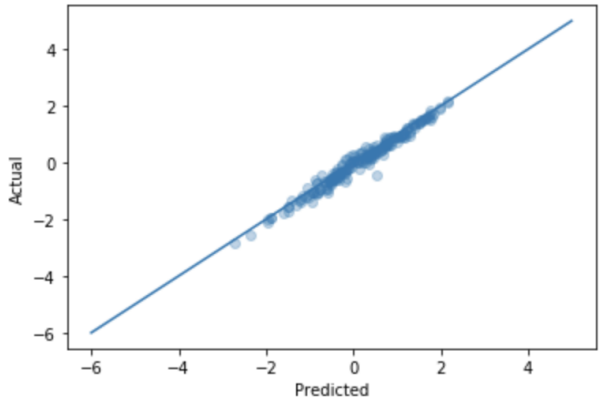

Figure 2: Accuracy of revised edge-conditioned GCN on 2000 samples