Journal Report 15
1/6/20-1/10/20
Sohom Paul
Computer Systems Research Lab
Period 1, White

Daily Log

Tuesday, January 21

Gave up on trying to directly implement tensor factorization into my own code and instead started building toy model using MNIST classification.
Debugging errors with dimension mismatch.

Thursday, January 23

Continued building tensor factorization model for MNIST.

Sunday, January 26

Started running preliminary tests using results from MNIST network.
Compared tensor factorization results to normal networks.

Timeline

Date	Goal	Met
Jan 13	Start investigating deep relationship networks.	Partial; have not yet integrated these new architectures into our project.
Jan 20	Complete investigation of deep relationship networks. Start building models for chemical property prediction.	No, not even close.
Jan 27	Further investigate deep relationship networks.	Built toy model; abandoned integration.
Feb 3	Analyze benefits provided by multitask learning and begin hyperparameter tuning.	
Feb 10	Continue engaging in hyperparameter search.	

Reflection

To do a totally complete investigation of the use of tensor factorization methods in our networks would take us a few months with uncertain rewards. (I would also need to spend some time to prove the mathematical validity of generalizing the technique from ANNs to CNNs.) However, this past week I've worked on building a toy model using the tensor factorization techniques to classify MNIST handwriting samples. The training time rises to obscene levels; I did not have the foresight to actually time the methods, but I estimate that the tensor factorization network took 100 times as long to reach the same loss as the basic network. Even if this technique did greatly improve the final accuracy after some long training time, I suspect it would be more productive to simply add more layers to our original networks. As a result, I'm abandoning this line of research.

For the coming weeks, I'm instead going to take a step back and see how we can improve our existing models without drastic changes to architecture. As of yet, I have not yet done a comprehensive analysis of how multitask training has affect learning rates and final accuracies, so I will need to start generating loss data for comparison. I will also need to more comprehensively look at the training time/performance ratio offered by soft-parameter sharing, which I had been previously been avoiding because of the long training times making it hard to roll out changes. After these coming weeks, I will regroup with my partner to see what interesting directions we can take our project once we have an integrated multitask-GCN model.