Journal Report 10
11/18/19-11/22/19
Sohom Paul
Computer Systems Research Lab
Period 1, White

Daily Log

Monday November 18

I switched from trying to divide my model into two parts to directly setting the weights of the learned model; coefficient transfer was successful.

Tuesday November 19

Successfully tested calculation of cross-task transfer coefficients.
Wrote program to start training and saving single-task models.

Thursday November 21

Started writing code for demonstration using Psi4 DFT calculations.
Continued trying to train and save single-task models.
Encountered some difficulties in getting Keras to load models using custom Spektral classes.

Timeline

Date	Goal	Met
Nov 11	Implement neural clustering (CMTL) and compare with non-clustered results	Partial; I've yet to compute the full 19x19 table of coefficients.
Nov 18	Finish performing neural clustering. Start implementing deep relationship networks.	Postponed, due to errors in last week's work.
Nov 25	Finish performing neural clustering. Generate 19x19 table of task relatedness. Start building demonstration program with DFT and single-task results.	Successful builds of DFT, partial completion of code to cluster tasks.
Dec 2	Continue building demonstration with preliminary multitask results.	
Dec 9	Finish building demonstration with multitask and DFT methods; start integrating with GCN demonstration.	
Winter Goal	Have a program capable of taking an input molecule and predicting the relevant molecular properties, using a choice of density functional theory or multitask-learned networks. The demo will let the use see firsthand the relative accuracies and speeds of the methods.	

Reflection

This week, I finally realized the much cleaner method of dividing my model into an transferred encoder and learned classifier can be much more easily be done by simply setting the weights of the models directly rather than interface with the finnicky Keras functional API. I have been able to successfully run an example of cross-task coefficient calculation by transferring the encoder for a model trained on zero-point vibrational energy onto a model trained on atomization enthalpy at 298K, yielding a loss of 0.894. However, this number is meaningless without having context to see the losses for all the other transferred tasks, which I haven't yet been able to compute. The issue that I've run into is that when I save and load the Spektral models (which is necessary, considering it will likely take dozens of hours to learn all of the networks), I am not capable of having Keras deal with the custom classes Spektral defines. However, I think this can be quickly fixed by just (again) directly accessing the weights to save and load. I plan to keep my machine running over the long weekend and finish filling out that matrix so I have the neural clustering ready.

Otherwise, I am currently working on making sure that my parser can extract all the relevant properties from Psi4 as we are learning using our neural networks. I am currently missing a couple (α, r^2), which I hope to rectify soon.