Journal Report 4
9/23/19-9/27/19
Sohom Paul
Computer Systems Research Lab
Period 1, White

Daily Log

Monday September 23

Added thermochemical calculations to DFT script.
Added option to turn on/off thermochemical calculation.
Added option to turn on/off geometry optimization.
Extracted zero-point vibrational energy, Gibbs free energy at room temperature and enthalpy at room temperature.
Computed desired properties for first 3 elements of our data set.

Tuesday September 14

Research day!
Read paper on the use of L1/Lq block-sparse regularization for multitask learning on non-neural models.
Read paper on the potential dangers of block-sparse regularization when output features have low overlap.
Learned about hard and soft parameter sharing on neural networks and the appropriate modifications of the loss functions.
Read paper on the use of hard parameter sharing to reduce overfitting.
Started reading paper on deep relationship networks.

Thursday September 26

Continued reading paper on deep relationship networks.
Started tutorial on using multitask learning on image classification of video game characters.
Started tutorial on using multitask learning with Tensorflow's Head API for nose positioning and facial pose estimation.

Timeline

Date	Goal	Met
Sep 16	Run relevant DFT calculations; find best parameters	Yes.
Sep 23	Write Python script to run DFT calculations and pull relevant features from our dataset.	Yes
Sep 30	Build toy networks with multitask learning	Abandoned; stole code from Github instead.
Oct 7	Integrate multitask learning (hard parameter sharing) into first GCNs	
Sep 14	Train network and compare results to tasks done without multitask learning.	

Reflection

My original vision with this project was to produce a comprehensive overview of the applications of message-passing neural networks in predicting molecular properties. However, because of the concerns that I had over how much time we would need to spend making our own tools, my partner and I pitched a less ambitious project about creating a single model, a multitask graph convolutional network. However, because of my partner's recent progress in implementing graph convolutional networks using preexisting libraries, I think that we might be able to start expanding the scope of this project again. Thus, rather than jump straight into writing code for multitask networks, I wanted to spend some time researching different algorithms to accomplish multitask learning so that we may implement several and test their relative merit. To that end, I spent all of Wednesday's class reading surveys on multitask learning and finding the corresponding papers on arXiv.org to see what techniques were promising. From what I've seen, hard parameter sharing (which was the only form of multitask learning that I knew of before this week) is still the standard technique applied to neural networks[1]. However, I started learning about soft parameter sharing, which, instead of having layers explicitly shared across tasks, uses a penalty term in the loss function to ensure that the parameter vectors across different tasks are similar[2]. I also started reading a paper on deep relationship networks[3], which looks promising but complicated and I still don't have a solid understanding of. I also spent a significant amount of time entertaining the idea of using block-sparse regularization[4] on our model (which using both intratask regularization, preventing consideration of too many features, and intertask regularization, ensuring that the features studied are in fact important and generalizable). However, based on my investigation, I realized that applying block-spare regularization is probably not a good idea, considering that our input features is fairly low dimensional and we don't have any reason to expect sparsity.

As for my original task this week, to produce a toy multitask model, I started following tutorials online and mining Github for multitask code I could steal, but I eventually realized that all the existing multitask research has been done in the field of computer vision. I build toy mod-

[1] https://arxiv.org/pdf/1706.05098.pdf

[2] https://arxiv.org/pdf/1606.04038.pdf

[3] https://arxiv.org/abs/1506.02117

[4] http://pages.stat.wisc.edu/ myuan/papers/glasso.final.pdf

els in order to be able to implement ML techniques in a controlled sandbox where I can easily debug problems by hand, which affords me the opportunity to become intimately familiar with the code I'm writing. However, all the available resources for learning jump straight into using deep CNNs, and I couldn't find a good dataset to benchmark multitask learning on very small networks. Thus, I changed my goal for this week from building my own network, to following along with somebody else's code and seeing if I could understand it properly.

I started by reading a tutorial[5] and code[6] for classifying video game characters. In it, the author sums the losses linearly and uses hard parameter sharing. I wanted to download the code and dataset onto my own machine so that I could try adjusting the code to soft parameter sharing or the other techniques I was learning, but downloading the dataset was taking many minutes. While I was waiting, I started reading a separate tutorial[7][8] on using Tensorflow's Head API for multitask learning. Using this API looks very promising, but I was not able to actually access the Tensorflow documentation; I kept getting 404 errors[9].

Finally, as for the code I wrote Monday, I merely tied up some loose ends that I left the previous week by finishing the implementation of thermochemical property extraction using DFT. I have reproduced the output of my code on the first 3 molecules in our dataset below. Notice that the rotational constants A, B, and C, as well as the zero-point vibrational energy and enthalpy and Gibbs free energy at room temperature are all new calculations that did not appear in last week's journal.

Index	A	B	C	Dipole	HOMO	LUMO	zpve	H 298.15	G 298.15
1	157.71	157.71	157.71	0.0	0.02844	0.10485	-40.497	-40.493	-40.516
2	293.61	293.54	191.39	1.6098	0.01461	0.09121	-56.557	-56.553	-56.576
3	799.59	437.90	282.95	1.9148	0.00964	0.07859	-76.448	-76.444	-76.466

[5] https://towardsdatascience.com/multi-task-learning-on-fate-grand-order-with-keras-261c5e8d3fb8

[6] https://github.com/sugi-chan/fgo-multi-task-keras/blob/master/fgo_multiclass.ipynb

[7] https://towardsdatascience.com/multitask-learning-in-tensorflow-with-the-head-api-68f2717019df

[8] https://drive.google.com/file/u/0/d/1NMB9lpi7P-GkkELkMU0h-yHtUq531D_Z/edit

[9] https://www.tensorflow.org/api_docs/python/tf/contrib/estimator/multi_head