Executive Report of Final Project

Chuanyu Liu*

March 17, 2022

1 Abstract

Recent machine learning models have shown that it is possible to learn molecular properties from 2D images of molecules. Could we do better if we replace pixels and discrete convolutions with atoms and continuous convolutions?

Convolutional Neural Networks (CNN), a deep learning algorithm composed of interconnected node layers with thresholds, provides a great platform to learn the properties of chemical molecules via image analysis and pattern recognition. While the choice of descriptors is case by case, each different descriptor is suitable for different research purposes. In this project, a continuous CNN model with QM9 was built to understand whether 2D or 3D descriptors have better training performances to predict molecular properties. SchNet neural network simulation results reveal that 2D molecular data reaches a better performance.

2 Introduction

Quantum mechanical (QM) calculation is the most accurate method to obtain molecular energetic characteristics so far like the total energy and the frontier molecular orbitals energies (HOMO and LUMO). However, a huge computational cost prevents its daily usage for exhaustive exploration of chemical space. Till recently the only alternative in overcoming the time factor was to use less accurate approximation of QM or classical molecular mechanics. Yet, the gain in time means a loss in precision with those methods.

An appealing alternative is to use a computationally much more efficient approach based on machine learning (ML) models for the prediction of molecular energetic characteristics. Such ML models need to be tested and generalized on real data. Molecular dataset QM9 is a classical benchmark and golden standard for Machine Learning (ML) predictions of various chemical properties, due to its homogeneity, purity, and lack of noise. QM9 is based

*Email: wenxuan0119@uchicago.edu

on the GDB, which is a combinatorial exploration of the chemical space. ML molecular predictions have been recently published with an accuracy that matches Density Functional Theory calculations.

SchNetPack is the toolbox used for training refined deep neural networks on Google Colab with QM9, which includes 2D data and 3D data. Colab provides a powerful computation capability for QM9 dataset containing more than 130,000 molecules. Among all properties in QM9 such as band gap, atom coordinates, and geometry, the atomization energy (u_0, the internal energy at $0K$) is chosen to construct the CNN model as the input and prediction parameter. Here, only one parameter atomization energy is used for training to avoid unnecessary interdependence or confusion from multiple inputs. One thousand training examples in each of the 2D and 3D data are used for validation and the remaining data as the test set.

3 Training process for the neural network

During the data loading and preprocessing process, 'AtomsLoader' is created for the splits for shuffling, batching and asynchronous data loading. Statistical properties of the target property atomization energy (u_0), like mean and standard deviation of the energy per atom, is used for model initialization, while unnecessary parameters such as 'atom reference' are neglected.

In the training process, SchNet module is built with 3 interaction layers and other default parameters considering there are multiple training examples. The Atomwise module is used to predict the energy, which takes the mean and standard deviation of the property per atom. Next, the model is trained with the 'Trainer' class and optimizer-2d. As a means of training optimization, I give the 'Trainer' hooks, which are useful to customize the training process. Then, set up a basis logging as well as a learning rate schedule that reduces the learning rate without improvement of the validation loss. Eventually, create a logger to store the mean absolute error (MAE) and root mean squared error (MSE) of u_0 prediction. For the demonstration, I run the training for 200 epochs on GPU (Tesla T4) and redo it all over again with 3D data. Comparing their performance with 'Final validation MAE' value could give the desired training performance results.

4 Result

The training dataset and the final prediction values are plotted in terms of the energy loss versus time from 2D and 3D data sets. There are quite similar exponential drop curves from training results and actual data, showing the CNN training model is qualitatively reliable. To quantitatively address their correlation, the mean absolute errors values (MAE) are calculated for training data points. The SchNet (2D) has the final validation mean absolute error values of 5.67 eV, while the MAE in SchNet (3D) is 7.71 eV. MAE values from 2D and 3D quite match with each other, which sort of makes sense since both data sets are generated from the same molecules with the similar generation methods.

If compare with the test MAE value of 5.59 eV(2D) and 7.41 eV(3D), respectively, the SchNet neural network based 2D data has a slightly better performance, but further experiments are still needed to strengthen the conclusion. The fact that the difference in MAE between SchNet 2D and 3D data is minor and not significant becomes the biggest hindrance to make affirmative conclusions on their performance comparison. Some possible reasons are listed as follow.

Firstly, adjustment of parameters or some potential hyper-parameters are significant in the the speed and quality of the learning process. Beyond that, I reduced the training set size in order to reduce computation time for CNN model. It is likely that small datasets are more prone to sampling disorder and bias. As a result, it's necessary to expand training sample size (1000-2000) and gather enough independent data, then run for more epochs (500-1000) until convergence, with Tesla T4 GPU to improve the speed of the training. Extra computational resources(Argonne National Laboratory) are also necessary to be applied to enhance the prediction performance and accuracy. In addition to using MAE as the only indicator of performance, we could also apply more advanced statistical analysis techniques such as regression to understand our result deeper and deeper.

Besides, it's easy for CNNs to process high-dimensional data and extract its features automatically in convolutional layer. However, in the deep layer of the network, it is going to converge the training results to the local minimum value rather than the global minimum value with the gradient descent algorithm. Hence, we may use other improved algorithms (BP algorithm) to avoid the problem.

another popular challenge of QM9 is to develop a machine learning model that can learn the information of the molecules within chemical accuracy using less training data. This may be realized by improving the representations of molecules or by employing some training algorithms.

5 Python Script

Please find more details in the following link:
https://colab.research.google.com/drive/1jH3dWjqh24FpDSyEQ1d2ynV3I2aEvhif?usp=sharing