CHEM 277B: Machine Learning-Based Energy Prediction for Molecular Systems: Modeling and Analysis using ANI-1 Dataset

Joshua Blomgren

1. Introduction

Calculating the energy of a molecular system, while a fundamental task in various scientific fields such as chemistry, is a computationally expensive problem. This is because the wave-like nature of electrons described by the Schrödinger equation requires the involvement of both statistical mechanics and quantum mechanics in calculating the energy of a molecular system. While there are methods available to approximate the exact solution to the Schrödinger equation, they often become computationally expensive for systems containing more than a hundred atoms. This computational cost poses a significant challenge in accurately predicting the energy of larger molecular systems.

To address this issue, this project aims to develop a supervised learning artificial neural network (ANN) capable of accurately predicting the energy of a molecular system based on the coordinates of its constituent molecules. Doing so can demonstrate how utilizing machine learning techniques can significantly reduce the computational expense associated with calculating the energy of a molecular system as well as with similarly complex problems. This approach holds the potential to eliminate reliance on solving the Schrödinger equation through traditional calculation, making energy prediction more efficient and accessible.

Previous research that has inspired the development of this project likewise explored the use of machine learning methods for solving quantum chemistry problems. Smith et al. (2017) developed a neural network model that can accurately predict the energy of a molecular system based on its atomic coordinates. The model was trained on a dataset of approximately 58,000 small molecules with a total of approximately 17.2 million conformations and was able to predict the energy of a molecular system with an accuracy comparable to that of density functional theory (DFT) calculations, one of the widely used computational quantum mechanical modeling methods. Lemm et al. (2021) recognized the challenge associated with doing this due to the need for solving potential energy surfaces to locate structural minimas and transition states. However, they also demonstrated machine learning's utility by developing a Graph To Structure (G2S) model that predicted the 3D molecular structures of thousands of constitutional isomers, transition states, and elpasolite crystals with lower root-mean-square deviations than mathematically derived models while also being highly similar to the reference quantum chemical structures. Studies such as these overcame the boundaries of fundamental quantum

chemistry calculations and proved the efficacy of machine learning methods in solving quantum chemistry problems.

This project aims to emulate these discoveries by developing a neural network model that can accurately predict the energies of molecular systems consisting of four or less heavy atoms. The model will be trained using the same ANI-1 dataset used in Smith et al. 's (2017) study, and the results will be reported as the root mean square error (RMSE) of the predicted energies compared to the true energies in kcal/mol.

2. Methods

The ANI-1 dataset, short for ANAKIN-ME or "Accurate Neural Network Engine for Molecular Energies", is a collection of 57,000 organic molecules with up to 8 heavy atoms and their corresponding potential energies developed by Justin S. Smith, Olexandr Isayev, and Adrian E. Roitberg (Smith et al., 2017). Altogether, there are approximately 17.2 million molecular configurations generated from the 57,000 molecules that are split across 8 files, each file containing molecules consisting of a specific number of heavy atoms. The ANI-1 dataset also includes a Python library designed for extracting the molecular information from the datasets called PyANITools. Using PyANITools, the atomic symbols, coordinates, and energies of each molecular configuration were extracted and used as training data for the neural network model. Across the ANI-1 1-4 subsets, there are a total of 97 molecules and 864,898 configurations.

The coordinates themselves cannot be used as input for the neural network model. In order to train a neural network for predicting molecular properties, it is important to ensure that the network is invariant to translations, rotations, and permutations of the atoms in a molecule, which can be achieved through computing AEVs. AEVs, Atomic Environment Vectors, are feature vectors that encode the local chemical environment around an atom in a molecule and are calculated using symmetry functions such as radial symmetry and angular symmetry functions. TorchANI, a PyTorch-based program developed by Gao et. al (2020) for the training of ANI machine learning models, provides a module called AEV Computer that computes the AEVs (Atomic Environment Vectors) of molecular structures and is initialized with certain parameters such as Rcr (cutoff radius for radial symmetry functions), Rca (cutoff radius for angular symmetry functions), and a set of hyperparameters such as EtaR, EtaA, Zeta, ShfR, ShfA, and ShfZ, which are used to compute the AEVs of molecular structures. These AEVs are then used as input for the neural network model.

The neural network model itself consists of a dictionary of sub-networks, each dedicated to a specific atom type: Hydrogen, Carbon, Nitrogen, and Oxygen. There are four fully connected linear layers in each sub-network, each with a ReLU activation function. Each sub-network starts

with 384 neurons in the first layer because 384 AEVs are generated by the AEV computer per atom and 1 output neuron for the final predicted energy of the molecular system.

Table 1: Hyperparameters selected for the ANI Neural Network Model

Hyperparameters	ANI Model
Number of Layers	4
Activation Function	Rectified Linear Unit (ReLU)
Regularization	L2 (lambda: 1e-5)
Optimizer Type	Stochastic Gradient Descent (SGD)
Learning Rate	1e-3
Batch Size	600
Number of Epochs	100

After experimenting with two to six fully connected linear layers, it was determined that four layers per sub-network offered a favorable balance between loss minimization and training time. While incorporating additional layers may improve the network's ability to learn complex patterns, it comes with the cost of increased computational demands. Given the considerable training time for the larger files encompassing more heavy atoms, reducing computational time was also important. The number of neurons per layer, aside from the first and last ones, were chosen through trial and error by training the model on subsets 1 and 2 of the ANI-1 dataset rather than on the entire dataset. Among the various activation functions including the sigmoid function, hyperbolic tangent function (tanh), and Rectified Linear Unit (ReLU), ReLU led to slightly faster training times and was the only activation function that prevented loss explosion. While smaller batch sizes demonstrated lower training and validation losses, reducing the batch size from 1000 to 300 resulted in a 12-second increase in training time for a single molecule. Therefore, in hope of maintaining training durations, a batch size of 600 was chosen. L2 regularization terms were added to the loss function to prevent the model from overfitting to the extensive training data and did decrease validation loss, at the expense of increased training loss. Although Adam is the more popular choice of optimizer for deep learning models, Stochastic Gradient Descent (SGD) interestingly yielded superior results for both training and validation.

A significant challenge encountered during this project was devising an effective training strategy for the ANI-1 dataset. Because the dataset is divided among eight separate files, each containing molecules with varying numbers of heavy atoms, it was difficult to train the model on

all the data at once. Initially, a single dataset was created that stored all contained information for all molecules in the four files. However, this proved unfeasible due to differing lengths of molecules which caused the species tensor to be of different sizes. Consequently, an alternative strategy involved looping through each file, molecule, and batch. This resulted in four nested loops: one for each epoch, file molecule, and batch. For every molecule, the data was partitioned using the train_test_split function from the scikit-learn library (Pedregosa et al., 2011), enabling the creation of appropriate datasets and data loaders for training. However, a recurring problem was the occurrence of exploding loss during the training process. Moreover, maintaining accurate track of the loss averages proved challenging but was necessary as continuous incrementation of the loss led to exceedingly large values.

3. Results

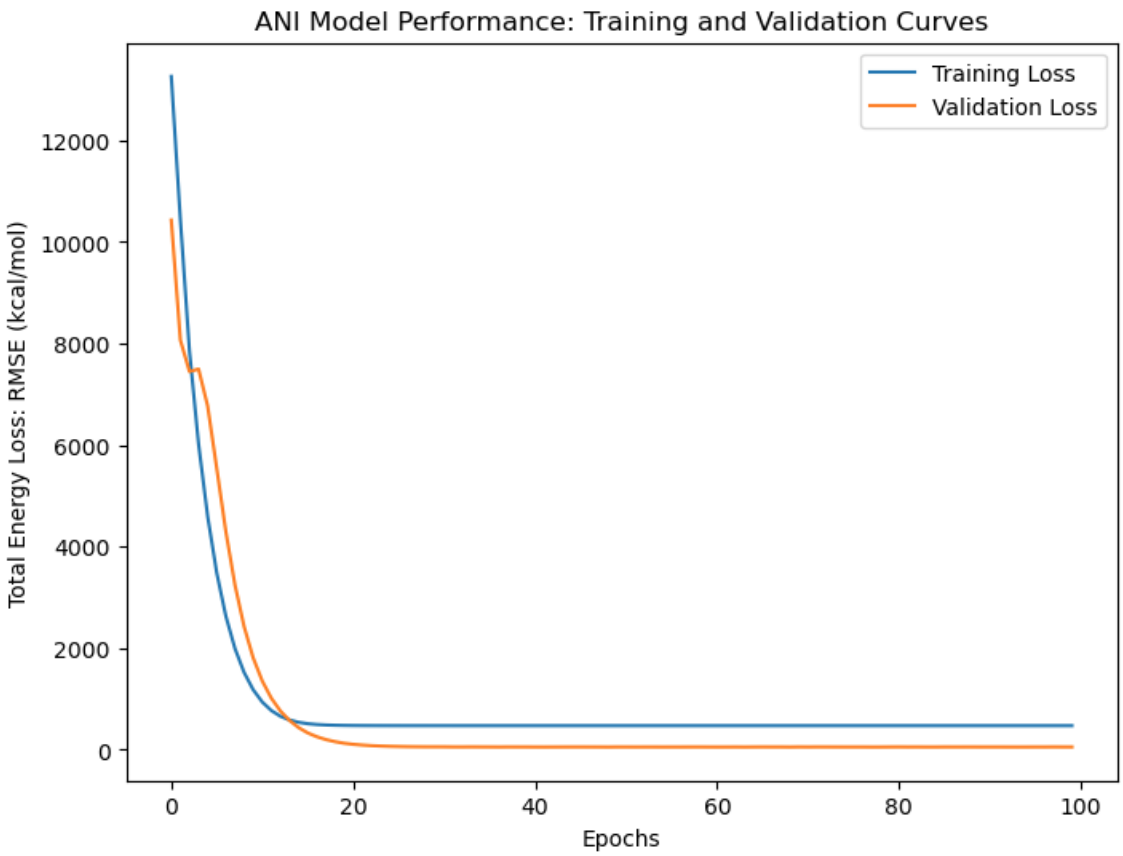

Figure 1: Training and validation loss of the ANI Model across 100 epochs for all ANI-1 subsets up to four heavy atoms. The loss is converted from Hartree to kcal/mol.

Table 2: The training losses and validation losses of the ANI Model over 100 epochs for all ANI-1 subsets up to four heavy atoms. The losses are converted from Hartree to kcal/mol and are RMSE.

Epoch	Training Loss (kcal/mol)	Validation Loss (kcal/mol)
0	13263.806933	10432.899929
10	939.735368	1353.985346
20	477.147842	106.332141
30	474.235706	57.354576
40	474.150304	54.139822
50	474.165271	54.839105
60	474.148763	53.891853
70	474.164680	54.661827
80	474.189599	55.784409
90	474.166105	54.877402

During the initial epochs, the training loss was significantly high at 13263.806933 kcal/mol, gradually decreasing over subsequent epochs. Similarly, the validation loss started at 10432.899929 kcal/mol and also demonstrated a decreasing trend with increasing epochs. As the training progressed, both the training and validation losses showed a notable reduction. By epoch 20, the training loss reached 477.147842 kcal/mol, indicating a substantial improvement in the model's ability to learn from the data. The validation loss further decreased to 106.332141 kcal/mol, indicating effective generalization of the model's performance beyond the training data. This trend continued, with both losses consistently decreasing until epoch 50, where they reached 474.165271 kcal/mol for training loss and 54.839105 kcal/mol for validation loss. After epoch 50, the losses showed relatively minor fluctuations, indicating a relatively stable performance of the model. The training curves demonstrate that the model was able to fit to the data to some extent.

4. Discussion

The objective of this study was to train the ANI Model on the ANI-1 dataset for up to four heavy atoms. Unfortunately, the model was unable to converge to a loss of less than 5 kcal/mol as expected. It is interesting that the validation loss is significantly lower than the training loss. It is to be suspected that this is due to the losses being incorrectly averaged across the molecules and subset files. The primary challenge was devising a training strategy that would allow the model to effectively learn from all the data without encountering memory issues or exploding loss. In order to address this challenge, it is proposed that future experiments could be carried out that save the weights of the model and reload them after each iteration of a molecule or batch.

Nevertheless, the results demonstrate promising progress in training the ANI_Model on the ANI-1 dataset. The decreasing trend observed in both training and validation losses indicates that the model successfully captured the underlying patterns and relationships between the input features and the total energy loss. The substantial reduction in losses over the epochs highlights the model's ability to learn and improve its predictions. Further experimentation and optimization of the issue of iterating through the data itself could potentially improve the model even more and provide deeper insights into the model’s performance and its potential applicability in real-world scenarios.

References:

Gao, X., Ramezanghorbani, F., Isayev, O., Smith, J. S., & Roitberg, A. E. (2020). Torchani: A free and open source pytorch-based deep learning implementation of the ANI neural network potentials. Journal of Chemical Information and Modeling, 60(7), 3408–3415. https://doi.org/10.1021/acs.jcim.0c00451

Lemm, D., von Rudorff, G. F., & von Lilienfeld, O. A. (2021). Machine learning based energy-free structure predictions of molecules, transition states, and solids. Nature Communications, 12(1). https://doi.org/10.1038/s41467-021-24525-7

Pedregosa et al. (2011, February 1). *Scikit-Learn: Machine learning in Python*. The Journal of Machine Learning Research. https://dl.acm.org/doi/10.5555/1953048.2078195

Smith, J. S., Isayev, O., & Roitberg, A. E. (2017). Ani-1: An extensible Neural Network Potential with DFT accuracy at force field computational cost. Chemical Science, 8(4), 3192–3203. https://doi.org/10.1039/c6sc05720a

Smith, Justin S., Isayev, O., & Roitberg, A. E. (2017). Ani-1, a data set of 20 million calculated off-equilibrium conformations for organic molecules. Scientific Data, 4(1). https://doi.org/10.1038/sdata.2017.193