Generating Near-Optimized Molecular Geometries Across Reactions using Neural Networks & Back Propagation

Richard L. Phillips
Swarthmore College Adaptive Robotics
Haverford College
rlanasphillips@gmail.com

ABSTRACT

The rate-limiting step of computational chemistry is computer capital. That is, the amount of research a lab can do is proportional to their computational throughput. Computational chemistry is generally used in materials research, where the properties of a new molecule are not known and not well studied. One important calculation in this scenario is in optimizing the geometry to an energy minima for a novel molecular structure. This process takes an initial geometry estimate and iteratively moves around all of the atoms to achieve a minima. Materials research often involves studying many similar molecules or reactions. To cut down on computational costs for materials research calculations, this paper proposes an implementation of nonlinear models to generate initial geometry estimates that require fewer calculations to achieve global energy minima. Specifically, this paper aims to justify efforts put into training a neural network to predict the resulting geometries for the products of an indicated reaction given the reactants. To do this, a feed-forward neural network is given initial bond length and angle measurements and is trained, using back propagation, to give the product's measurements. The resulting molecules are then re-constructed from this data. This paper uses a quinone and quinol set of geometries optimized with Gaussian 09 and the B3LYP/6-311+G(d,p) model chemistry. For the molecules in this set, a neural network with two hidden layers is shown to learn to accurately predict geometries very close to the set of true optimized geometries. A comparison is made to an industry standard lowest energy conformer calculator, ChemAxon's lowestEnergyConformer plugin. Experiments show that the network can produce molecules with closer geometries and lower total energies than the tool with under 80 data points, and this is expected to decrease calculation times.

General Terms

Back Propogation, Neural Network, Cheminformatics, Computational Chemistry

1. INTRODUCTION

Computational chemistry is a sub-field of chemistry focused on applying computational methods to chemistry problems. The rate-limiting factor in computational chemistry is the computer capital made available to computational chemists. That is, research is limited by how many calculations can actually be done. Computational chemistry is an important science to the field of materials research. Materials research includes everything from drug discovery to electrolyte research to renewable resource generation and more. A key tenant of materials research is that, within a family of molecules, similar molecules have similar properties. This means that much of materials research is focused on families of promising molecules with only minor changes from each other [1]. Frequently high-throughput screening for desired properties is done on hundreds, or even thousands, of molecules that are iteratively changed with only slightly different substructures or functional groups [2, 3].

As the interest of materials research is largely focused on the discovery of novel materials, few, if any, candidate molecules will have been studied and characterized previously. Thus, before any other calculation is done, the molecular geometry of every molecule of interest must be optimized. This is almost always done to a global energy minima as this is how the molecule can be thought of as existing in nature. This is important, as non-optimized geometries will yield significantly different results for *ab initio* calculations [4].

So, geometry optimization calculations are important to determining other characteristics. Geometry optimization, is generally done through an iterative process that scales exponentially with the number of atoms in a molecule. It is firmly an NP-hard problem when approached in a general case [5]. The problem is deceptive to the iterative intramolecular force-minimizing strategies used to tackle geometry optimization as the number of local force minima scales exponentially as well [6].

This provides an ideal opportunity to implement a neural network to help with geometry optimization. Specifically, this paper advocates for the use of a feed-forward neural networks trained with back propagation to learn a particular reaction. That is, given information about a molecular geometry and training a network on a number of similar molecules that experience a common reaction, the network should be able to predict the changes to the molecular geometry through the reaction. This could be used to produce higher quality initial guesses to then complete with a traditional iterative method to find a global energy minima. The goal in this scenario is to reduce the total number of iterations needed, thus saving computation time for more interesting calculations.

This scenario gives the network several advantages over a traditional iterative process. First, the network can gain insight on and generalize changes that occur in a specific reaction. Whereas an iterative process has to start from scratch with each molecule optimized, a neural network only

gets better at estimating a starting point as the number of molecules that it can learn from goes up. Second, this strategy avoids the force-dependent optimization of the traditional methods. This means that it should completely avoid the deception that exists in terms of the local energy minima produced as more atoms are added to a system. So, the optimization function will not get stuck in local minima and this helps to overcome problems of scale. Third, (but derived from the first two advantages) this method should offer improvements in terms of systems of multiple molecules, reactions that split or join molecules and other large systems of interest.

1.1 Previous Works

Several works have previously tried to implement similar non-linear strategies, although they largely neglected the scenario that this paper outlines a solution for (that of learning a reaction).

The work of Lemes, Zacharias, and Dal Pino Jr in their January 2000 conference paper "Application of Neural Networks: A Molecular Optimization Study" addresses many of the same problems as this paper seeks to. Namely, their paper also works to create a neural network with back propagation to create more suitable starting point geometries. This paper set up a neural network to help narrow down iteratively generated silicon cluster geometries to the most suitable candidates, and then the candidate out of those in the lowest energy configuration is optimized. To do this, a neural network is constructed with inputs describing the geometry of a cluster, a hidden layer and two outputs that represent the estimate for a cluster's internal energy. This allowed for a maximum reduction of over 85% fewer geometry optimization cycles than their standard method [6]. This demonstrates the potential of the application of neural networks to geometry optimization.

However, the work of Lemes et al. has several key differences from the scope of this paper. Whereas the work of that paper seeks to identify geometries for generally very regular silicon cluster, this paper will be more interested in less regular organic molecules and systems. To combat this additional challenge, instead of forcing the network on estimating forces for candidate structures, the network is tasked with generalizing the effects of a specific reaction on a family of molecules. This added context should allow the network to make more powerful predictions. Additionally, their network does not address the problem of local minima and of the exponential scaling in difficulty of geometry optimization. As their network is constructed with just two outputs that indicate the model's estimate of the energy of the geometry of the molecule, it is likely that the network will either incorrectly identify local minima as the optimal starting point or that it will simply train inefficiently until it is able to overcome this problem.

Styrcz, Mrozek, and Mazur [7] present a much more robust genetic algorithm with parameters controlled by a feed-forward neural network. Their network has six inputs, for hidden layers of six neurons each, and six outputs. The network is fed with current minimum, maximum and average distances between their genetic algorithm's chromosomes and is expected to output two meta-parameters and what their genetic algorithm's parameters should be for the next generation of their evolutionary process [7]. Their solution is robust and promising, although it seems that in many scenarios it may not provide enough benefit to be worth implementing. The major difference in this work is, again, that this work assumes a known reaction and is simply working to train a neural network to characterize that.

Finally, there is a similar work by Deaven and Ho [5] that also uses a genetic algorithm to generate optimized molecular geometries. Their strategy again focuses on iteratively generating a set of proposed geometries. They 'relax' these geometries, that is, optimize them to a nearby local minima. Then, they take the geometries with the lowest relaxed energies and use these to generate a new generation of candidate structures. But as they must still calculate the nearest local minima of every candidate in every generation, their method is still computationally intensive and again lacks the advantages focusing on a specific family of molecules and a specific reaction offers. Additionally, as the discussed algorithm requires calculations for every candidate to be relaxed to a local minima, it offers little to no improvement for small molecules (implied to be molecules of around 20 atoms or less in the paper).

2. EXPERIMENT

This section will outline the data set used, the network toplogy, and finally the parameters used in the final implementation.

2.1 Quinone Data Set

The network was trained and tested on a set of 252 quinone/quinol pairs with the objective of learning the reaction that takes a quinone and forms a quinol. The specific framework and reaction is illustrated in Figure 1. This specific framework was chosen for it's availability in a high-quality data set as well as the calculated stability of the outer functional groups. Computational calculations have shown that, across the reaction of interest, the functional groups change very little in terms of geometry [2]. This is good as it allows for all of the change in geometry across the reaction to be put on the neural network to predict. (A current weakness of the model in its current form is that it cannot account for the parts of a molecule outside of the framework that it is learning.)

The molecules in the data set were optimized with Gaussian 09 revision D.01 and the B3LYP/6-311+G(d,p) model chemistry. This model chemistry was chosen as it can be used to produce geometries for quantum calculations that very accurately predict experimental data [2]. Additionally, the quinone family of molecules is a current family of interest in terms of developing new renewable energy storage options.

Unless otherwise stated, all of the experiments in this paper were run using a training set of 180 molecule pairs and a test set of 72 molecules.

2.2 The Network

For this experiment, the network was a feed-forward neural network with 23 inputs, two hidden sigmoid layers of 12 nodes each, and 23 outputs. The network was implemented with a bias of 1. The first 12 inputs represented normalized bond lengths between every atom of the framework molecule and the remaining 11 represented the normalized angles between these bonds. A smaller model of the network can be seen in Figure 2.

2.2.1 Implementation

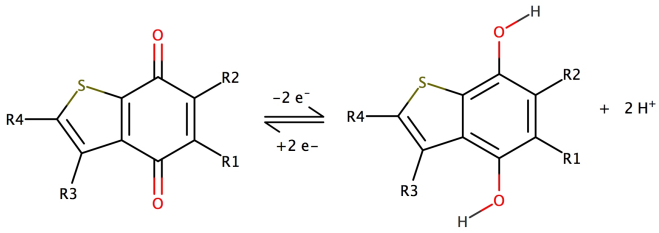

Figure 1: The reaction from a quinone (left) to a quinol (right) for the framework used in the experimental portion of this paper

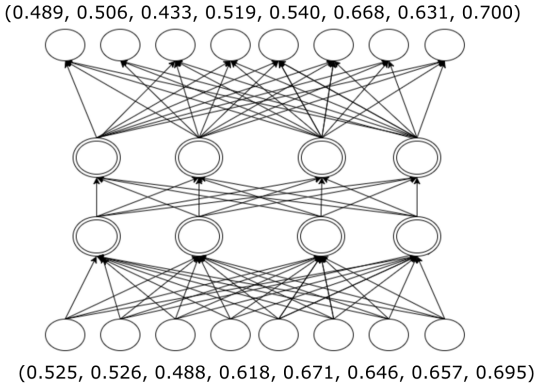

Figure 2: A smaller version of the network used in this experiment. The input vector consists of bond lengths for the bonds between atoms in the framework and the angles between those bonds. This data was then used to reconstruct the molecules for the final optimization calculations.

Table 1: Important Parameters

Parameter	Value
Inputs	23
Outputs	23
Hidden Layer Type	Sigmoid
Hidden Layers	2
Learning Rate	0.02
Momentum	0.90
Bias	1

The network was implemented in the Python library PyBrain [8] for ease of use and adaptability. The back propagation trainer native to PyBrain was used to train the network. The network was trained with a learning rate of 0.02 and a momentum parameter of 0.90. The momentum parameter indicates the ratio of the gradient of the last time step that is to be used. There was no learning rate decay or weight decay used. These parameters were determined experimentally. The output bond lengths were used to reconstruct the highly planar rings of the framework molecule

3. RESULTS

The network demonstrated learning capacity and was, in fact, able to improve in accuracy over several generations.

3.1 Current Standard Comparison

The current industry standard is to take a program, like ChemAxon's lowestEnergyConformer plugin, to generate an initial geometry to optimize using more computationally intensive methods. ChemAxon generates a conformer based of off known geometries of the substructures of the molecule in question and then lightly optimizes those. This is a cheap calculation that gives a program like Gaussian 09 a starting point to improve upon. A good standard of success for the neural network discussed in this paper was to compare the geometries that it generated to those ChemAxon could. If it produced geometries that could be optimized with fewer computations, then it was succesful and is likely worth implementing in current research. However, given limited time for this paper a more qualitative measurement has to be determined. The force of the generated structure will serve as a proxy (Lower forces qualitatively indicated less computational cost to optimize).

To make this comparison, Marvin 15.6.1.0, 2015, ChemAxon was used to generate geometries for a subset of the test data set of molecules and that was compared to the neural network's output with 10, 40, and 80 training data points respectively.

3.2 Recreating Product Geometries

One important note thus-far left un-addressed is how the output, a vector of 12 bond lengths and 11 bond angles, was transformed into the Cartesian coordinates for a molecular geometry. As the framework molecule was highly planar, a Python script was simply created to adjust the bond lengths of a ChemAxon estimate so that it matched the output of the neural network as closely as possible. The bond angles were not used at this point.

3.3 Sample Size Requirement

Of particular interest is the number of molecules that are required for the network to be effective. Figure 4 shows average degrees of error for 5 runs of training sets of each 4, 20, 40, 80, and 120 training data points. There is little quantitative difference in test data set performance between 80 and 120. There also seems to be a comparable degree of error for 40 training points, but Figure 5, which has a better measurement of model accuracy, refutes this. Figure 5 shows the intra-molecular force of the generated geometries for three given training data set sizes and ChemAxon. These geometries were generated by versions of the network each trained for 8 epochs with their given sample sizes. Note that, on average for this particular framework, the neural

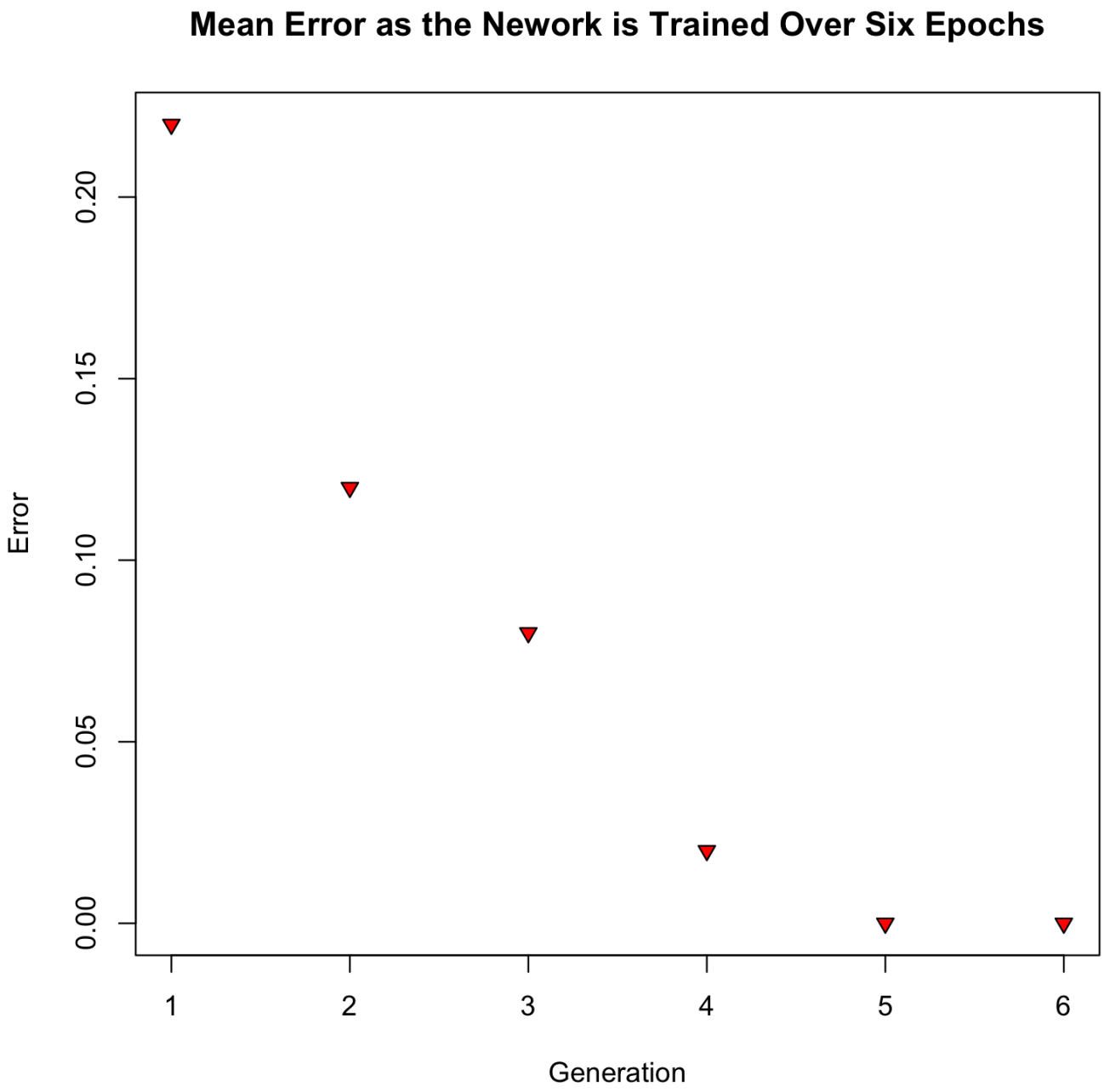

Figure 3: A graph of the calculated mean square error for the test data set after the given number of generations trained. Training done on 180 sample training set. It is evident that the model is making progress on the test data set as with each successive training epoch. Note that the network was *not* trained on the test data set. The model generally converges between 6 and 8 epochs, and should probably not be trained beyond that to avoid issues of overfitting.

network beats out ChemAxon with 80 training samples by nearly a factor of 2. The neural network finds a lower force conformer (without doing gradient descent on force) with 80 training samples ($p = 0.0088$). However, a small sample size and a very specific reaction limit this paper's ability to make assertions about to what degree the network will reduce computational time or how much lower in force the generated geometries will be. Figures 6-9 show indicative examples of generated structures for each of the four methods. The geometry generated from a 10-item training set is clearly impossible, and so the force calculations associated with it could not be completed by Gaussian 09.

3.4 Discussion

There is strong evidence that the model was successful in its goal. The advantage that a model trained to have context for a specific reaction is clear, and there does not seem to be any reason that this would not hold true for other (especially similar) molecules.

The network itself and the parameters of the back propagation trainer do not seem to be very sensitive at all. The learning rate can span the range of 0.01 to 0.08 and the momentum can vary between 0.81 and 0.95 with no statistically significant difference in test data set performance after 8 epochs (with a sample size of ten for each bound). The addition of an additional hidden layer or two, however, dramatically increases the mean square error of the model's predictions when using a smaller sample size. This makes sense in context, as back propagation will have more difficulty following the error gradient to improve the model for nodes that are father away from the input.

The method of re-constructing geometries was implemented given tight time constraints and represented a limitation to the study. With greater time and resources, a more robust solution might have been found. As it is, however, the reconstruction introduces some degree of error to the internal bond angles as the neural network outputs are not used. This means that some emphasis needs to be taken away from the quantitative measures of performance for the models until further development and testing can be done.

The paper has successfully facilitated the construction of a program that manages bonds and bond lengths and serves as a tool kit to quickly gather important information about molecular geometries. Further, there are few obstacles to releasing and applying the framework to other families of molecules as soon as the limitation of the re-construction is overcome.

4. CONCLUSIONS

Training a neural network to a particular reaction has shown to be a success for this particular framework. This a good indicator, but more research needs to be done to quantify the benefits (savings in computational time) to determine if this is worth the human time to implement. Additionally, what size molecules, what reactions and what molecular sub-structures work best with this method need to be explored more fully. For instance, increasingly large or complex molecules might see more worthwhile benefits (but then the model would be competing with a different host of optimization strategies). However, the results of this paper do justify doing additional research. Additionally, it would be a productive use of resources to get the program constructed to facilitate the experiment of this paper into the hands of scientists that can take advantage of its features.

5. ACKNOWLEDGMENTS

I would like to thank Professor Meeden for her patience and guidance throughout the semester. (Also for allowing me - a Haverford Student - into her class!) I would also very much like to thank Prof. Josh Schrier of Haverford College for publishing the data sets used in this paper and for helping me understand how to best measure the network's performance. My hope is that this research can help the lab do even more science in the future.

6. REFERENCES

[1] Andrew R. Leach and Valerie J. Gillet. *An Introduction to Chemoinformatics Revised Edition*. 2007.

[2] Sergio D. Pineda Flores, Geoffrey C. Martin-Noble, Richard L. Phillips, and Joshua Schrier. Bio-inspired electroactive organic molecules for aqueous redox flow batteries. 1. thiophenoquinones. *The Journal of Physical Chemistry C*, 119(38):21800–21809, September 2015.

[3] Lei Cheng, Rajeev S. Assary, Xiaohui Qu, Anubhav Jain, Shyue Ping Ong, Nav Nidhi Rajput, Kristin Persson, and Larry A. Curtiss. Accelerating electrolyte discovery for energy storage with high-throughput screening. *The Journal of Physical Chemistry Letters*, 6(2):283–291, December 2014.

[4] Regina F. Frey, James Coffin, Susan Q. Newton, Michael Ramek, Vincent K. W. Cheng, F. A. Momany, and Lothar Schaefer. Importance of correlation-gradient geometry optimization for molecular conformational analyses. *The Journal of the American Chemical Society*, 114(13):5369–5377, June 1992.

[5] D.M. Deaven and K.M. Ho. Molecular geometry optimization with a genetic algorithm. *Physical Review Letters*, January 1995.

[6] M.R. Lemes, C.R. Zacharias, and A. Dal Pino Jr. Application of neural networks: A molecular geometry optimization study. *6th Brazilian Symposium on Neural Networks*, November 2000.

[7] Anna Styrcz, Janusz Mrozek, and Grzegorz Mazur. A neuralâĂŞnetwork controlled dynamic evolutionary scheme for global molecular geometry optimization. *Int. J. Appl. Math. Comput. Sci*, 2011.

[8] Tom Schaul, Justin Bayer, Daan Wierstra, Yi Sun, Martin Felder, Frank Sehnke, Thomas Rückstieß, and Jürgen Schmidhuber. PyBrain. *Journal of Machine Learning Research*, 11:743–746, 2010.

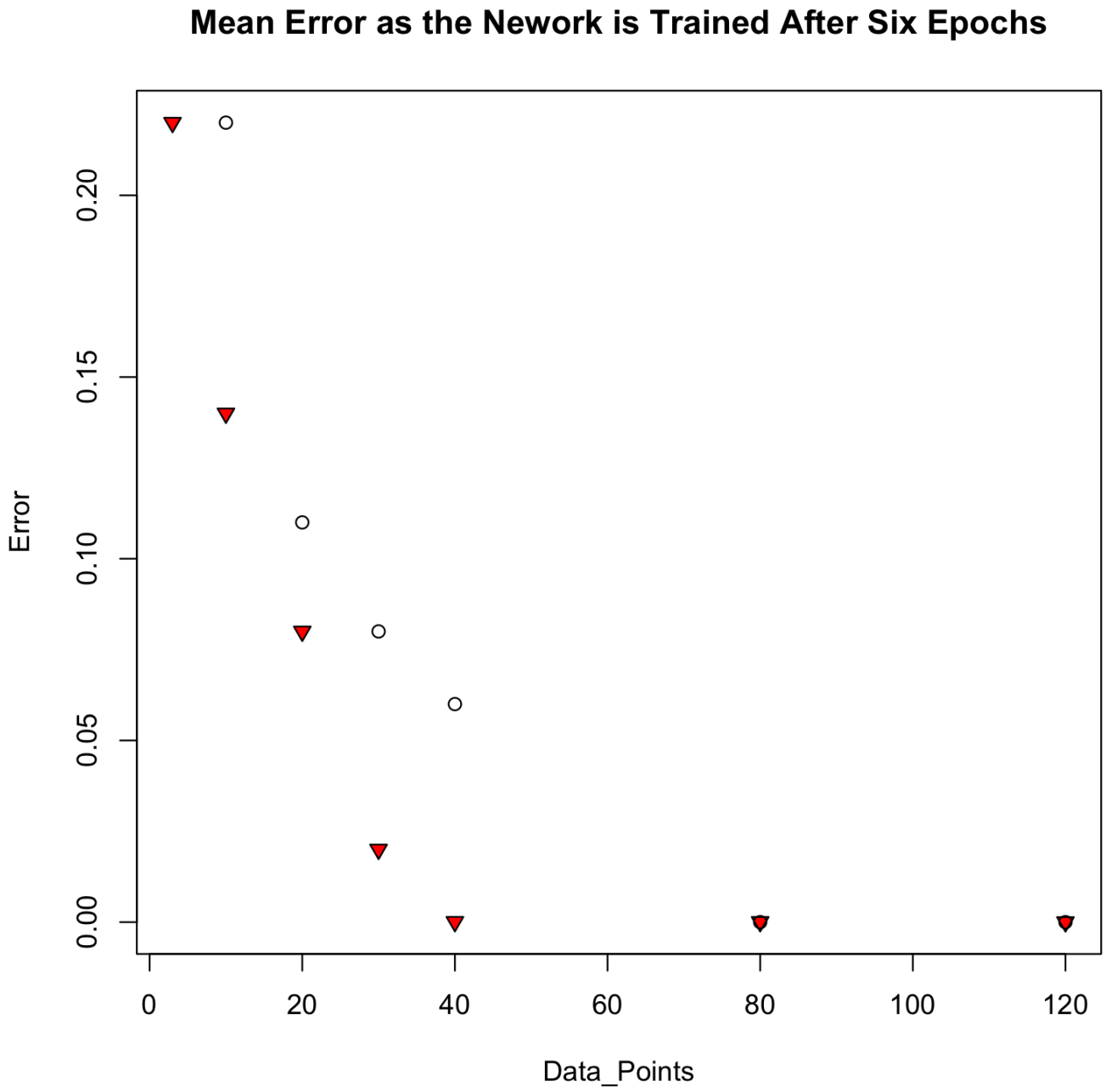

Figure 4: A graph of the average mean square errors for the test data set after 6 epochs for networks trained on the given number of data points. The white circles represent the average maximum error for the same set of test data set performances.

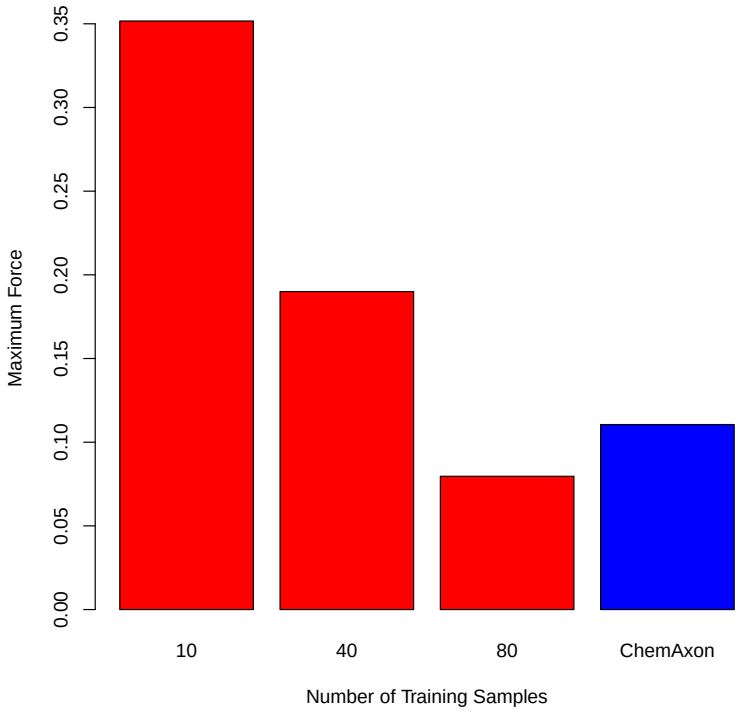

Figure 5: A bar graph comparing the energies of the generated geometries for a neural network with 10, 40 and 80 training data points and ChemAxon's own calculation. Note that lower values are more favorable as they imply that there is less computational work to be done. Also note that the majority of the 10 training set molecules were unrunnable, as they had interatomic distances that were too small.

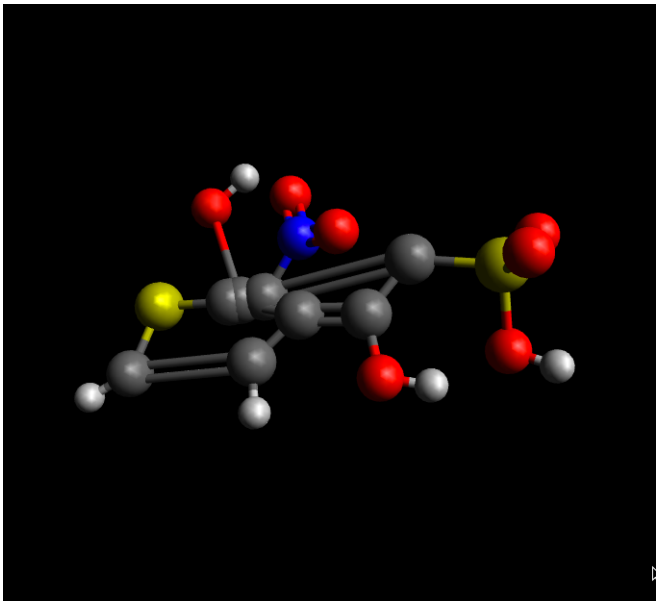

Figure 6: Generated from a 10-item training set

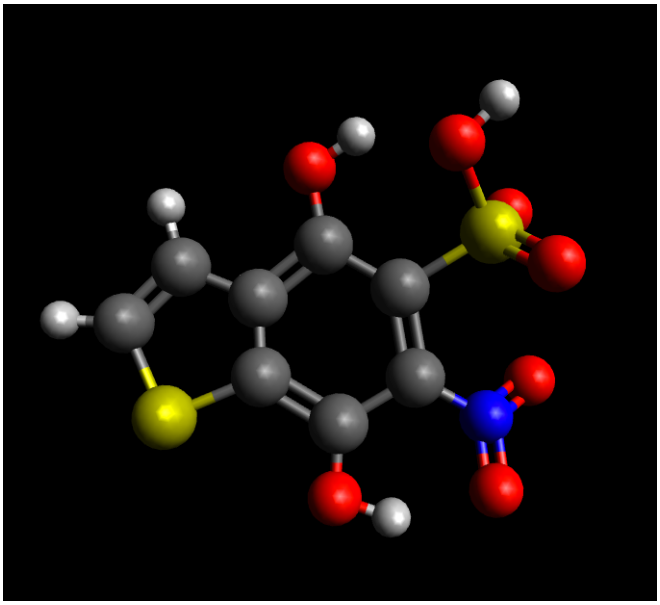

Figure 7: Generated from a 40-item training set

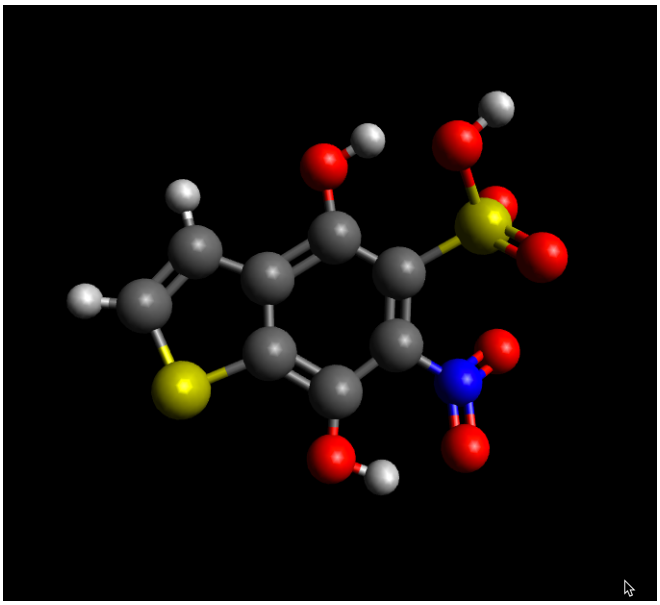

Figure 8: Generated from a 80-item training set

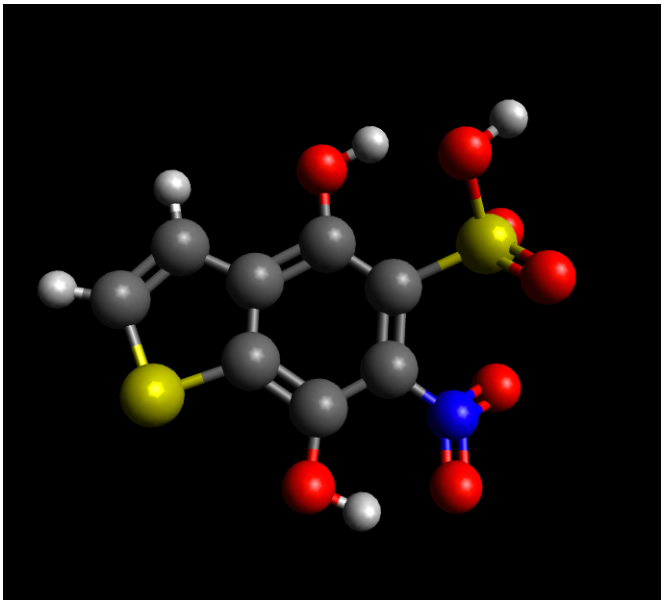

Figure 9: Generated by ChemAxon's Lowest Energy Conformer