Applying Machine Learning to Deep Eutectic Solvents

Tate Bestwick[1], Brian Doherty[2], Kun Yue[2], Orlando Acevedo[2]
[1]Department of Computer Science, Computing for Structure REU
[2]Department of Chemistry, University of Miami, Coral Gables, Florida 33146, United States

What is Machine Learning?

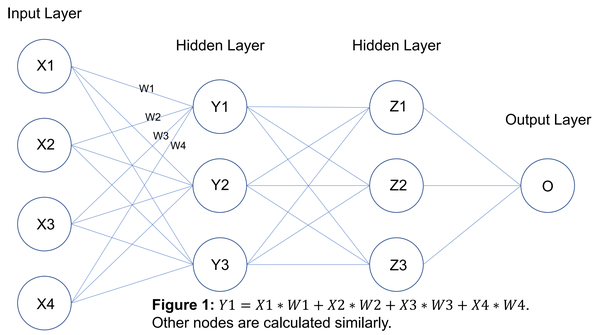
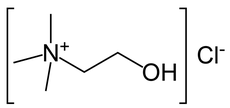


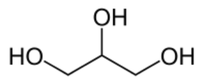
Figure 1: $Y1 = X1 * W1 + X2 * W2 + X3 * W3 + X4 * W4$. Other nodes are calculated similarly.

Machine learning is an analytical technique that utilizes Artificial Neural Networks (ANNs). These networks consist of layers of nodes, depicted by circles in Figure 1. Each layer is connected to the layer immediately in front of it and behind it, as shown by the lines connecting the nodes. Thus, each node in one layer is determined by its connection to all the nodes in the previous layer. The ANN is provided data in which each input is paired with the correct output. It is then trained by testing itself repeatedly against all the different inputs provided. As it trains, it changes the weight of each connection to minimize the error between what it predicted and the correct output provided.

What are Deep Eutectic Solvents?

Figure 2
Choline Chloride, a quaternary ammonium salt





Glycerol, a hydrogen bond donor

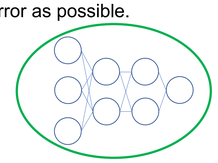
Deep eutectic solvents are a promising new class of solvent which often consist of a quaternary ammonium salt and a hydrogen bond donor. Since they are comprised of different molecules, their chemical properties can be vastly adjusted, making them one of the most versatile alternatives to current solvents to date. Along with the benefit of adaptability, deep eutectic solvents are much safer for the environment than many solvents currently used in large chemical processes today.

Overall Problem and Objective

Molecular behavior is best modeled using quantum mechanics. However, quantum calculations are computationally costly, and thus cannot be applied to thousands of molecules at once. The purpose of this project is to train ANNs with data from quantum calculations to the point that they can reproduce quantum mechanical results with as little error as possible.

Figure 3

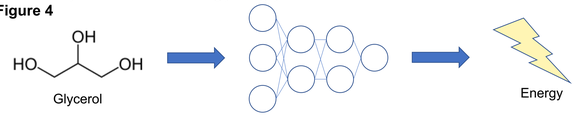
$$i\hbar\frac{\partial}{\partial t}\Psi = H\Psi$$



Specific Objective

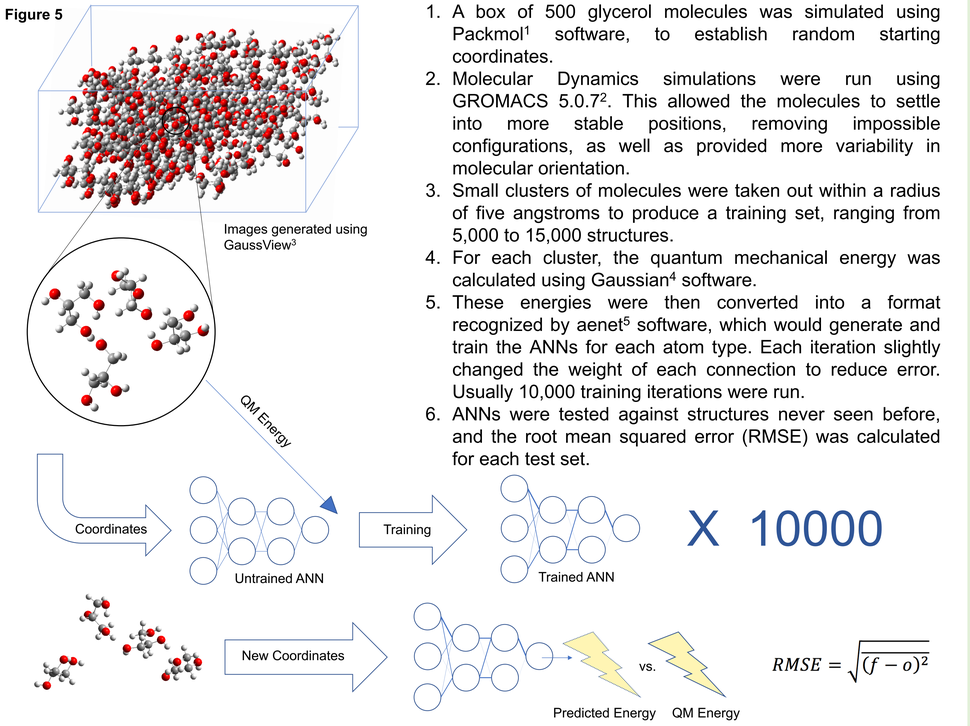
The past nine weeks of research have been dedicated to training ANNs to accurately predict the energy of glycerol clusters.

Figure 4



Method

Figure 5



1. A box of 500 glycerol molecules was simulated using Packmol[1] software, to establish random starting coordinates.
2. Molecular Dynamics simulations were run using GROMACS 5.0.7[2]. This allowed the molecules to settle into more stable positions, removing impossible configurations, as well as provided more variability in molecular orientation.
3. Small clusters of molecules were taken out within a radius of five angstroms to produce a training set, ranging from 5,000 to 15,000 structures.
4. For each cluster, the quantum mechanical energy was calculated using Gaussian[4] software.
5. These energies were then converted into a format recognized by aenet[5] software, which would generate and train the ANNs for each atom type. Each iteration slightly changed the weight of each connection to reduce error. Usually 10,000 training iterations were run.
6. ANNs were tested against structures never seen before, and the root mean squared error (RMSE) was calculated for each test set.

Results

Figure 5

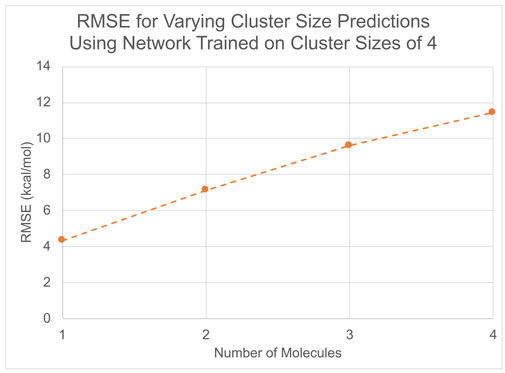
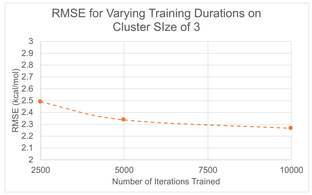


Figure 6: RMSE (kcal/mol) between predicted energy and QM energy

Number of molecules in test set clusters	Number of molecules in training set clusters: 2	3	4
1	0.5	1.5	4.3
2	1.0	2.5	7.1
3		2.3	9.6
4			11.4

Figure 7

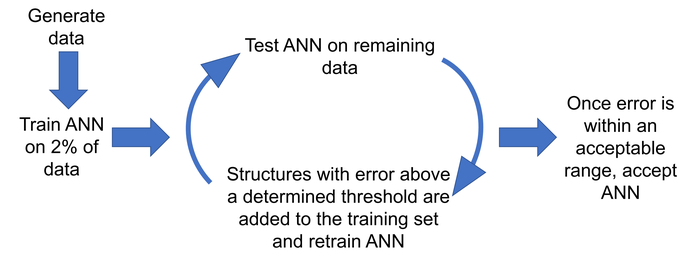


Discussion

As the cluster size increases, the number of possible orientations increase exponentially. This is largely why error nearly always increases with cluster size. While ANNs generally predict smaller cluster size energy with even less error than those of the same size, they cannot accurately predict energies of larger cluster sizes, hence those results were not reported.

Another possible source of error is overexposing a network to similar structures. If an ANN receives many structures of similar orientation, it will be biased toward that output, and will be less accurate for structures that differ drastically.

Future Direction



The next step is to further refine the generation of ANN training data. In the method portrayed above, rather than using the bulk of the data generated to train the network, only a small percentage of that data goes into training the ANNs, and the rest are used to test them. A certain percentage of structures with error above a predetermined threshold are added to the training set, as these are structures that greatly differ from anything the network has been already exposed to. The ANN is then retrained and retested, and the process repeats until all test data fall below an acceptable error range. This should help the ANN get exposed to more configurations as well as largely prevent against biasing towards particular structures.

Along with new methods there is optimizing already existing parameters. These include how many hidden layers and nodes should be present within each network, how many training iterations ANNs should undergo, and polynomial orders utilized in fitting functions.

Conclusion

- Neural Networks are highly adaptable, and show much potential in predicting chemical properties.
- Currently, they can predict energies of glycerol molecules or pairs to accuracies of less than 1 kcal/mol.
- As the method and parameters continue to be refined, ANNs show great potential to predict chemical properties, and by doing so could significantly increase both the scale and speed of molecular simulations.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1659144.

Citations

1. L. Martínez, R. Andrade, E. G. Birgin, J. M. Martínez. Packmol: A package for building initial configurations for molecular dynamics simulations. *Journal of Computational Chemistry*, 30(13):2157-2164, 2009.
2. Abraham, M. J.; Murtola, T.; Schulz, R.; Pall, S.; Smith, J. C.; Hess,´ B.; Lindahl, E. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX 2015, 1−2, 19−25
3. GaussView, Version 5.0.9, Roy Dennington, Todd A. Keith, and John M. Millam, Semichem Inc., Shawnee Mission, KS, 2016.
4. Gaussian 09, Revision B.01, M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox, Gaussian, Inc., Wallingford CT, 2016.
5. N. Artrith and A. Urban, *Comput. Mater. Sci.***114** (2016) 135-150.