Using Machine Learning for Quantum Chemistry

Peter Gysbers
96373162

Reza Sadoughianzadeh
85293141

Abstract

1 Introduction

Clearly state the problem being addressed. Explain why it is an important problem. At a high level, briefly summarize the history of the works that will be discussed in the project.

In this paper we will present several different machine learning methods which predict the chemical properties of molecules based on their configurations. The dataset is a set of simulated molecules. Dataset1 (QM9) is a set of example molecules which have different composition, each is in its ground state. Dataset2 (MD17) is a set of samples from a single molecule's trajectory in a dynamical simulation.

We will discuss gradient domain machine learning (GDML) which is a kernel regression technique to learn the force-fields which cause molecular dynamics. In addition "SchNet" is a neural network architecture which extends convolutional networks. It has various extensions which improve it. Most recently, we present "Cormorant" which is a so-called covariant neural network. It "bakes in" the symmetries of molecules to best take into account the interatomic interactions within the molecules and outperforms the other two methods in some applications.

As machine learning has developed, finding new applications in image recognition, language processing and recommendation systems, applications of machine learning to quantum chemistry have also progressed. This development has advanced through linear regressions, to kernel regressions, to neural networks and most recently to advanced tensor networks with connectivity and activation functions that are highly customized to this particular prediction problem.

2 Background

Chemists want to find new materials and chemicals to use in industry or medicine. However there are a huge number of possible configurations. If one has not been experimentally measured then it must be computed from theory. We would like to check all configurations to see if any have the desired properties. However these calculations are expensive and machine learning is a tool with the potential to make predictive calculations useful. Either to check guessed configurations or generate configurations with desired properties.

The GDB-17 molecular database contains 166 billion molecular graphs which is still a small subset of possible molecules. It is limited in both size and compositions. [12]

A molecule is a collection of atoms (e.g. H: hydrogen, C: carbon, O: oxygen, etc) which are connected by chemical bonds. Each atom can form a certain number of bonds in a particular range of angles. Assuming fixed bond lengths and angles, the geometry of a molecule can be generated from a graph describing the connections between atoms. Molecules live in "compound space", their location in which is determined by their composition (which atomic species and how many of each e.g. $C_7H_{10}O_2$ vs $C_9H_8O_4$) and configuration (bond locations). Some examples of molecules are in Figure 1.

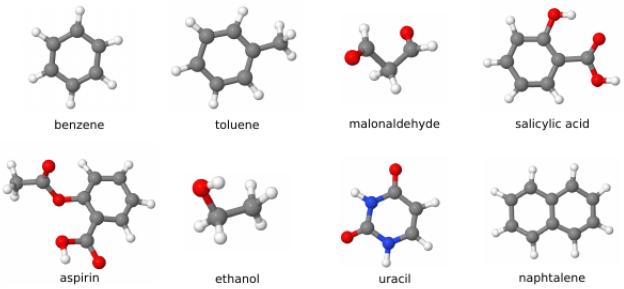

Figure 1: Examples of molecules from MD17 dataset. [6]

The main approach to solving the getting the properties of a molecule is using the Born-Oppenheimer approximation. In this paradigm, the nuclei are considered fixed and the electrons move in the resulting potential function. However this is still an n-body problem which is difficult to solve (since the position of every electron depends on every other electron, in addition to the positions of the atoms). The wavefunction of the electrons carries all of the information about the properties of the molecule. Density functional theory (DFT) is the approximation method that describes a single object, the electron-density.

Atoms have static properties, the wavefunction of the electrons has a ground-state (and excited states) which can be determined by solving the Schrodinger equation. The ground-state is the state at zero temperature. From the wavefunction many properties are simple to calculate. The QM9 dataset is a dataset of DFT calculations for a subset of the molecules in the GDB-17 set of molecular graphs. It contains 134 thousand molecules. [14]

Atoms also have dynamical properties. Each atom exerts a force on every other atom in the molecule as well as other atoms in other molecules. At non-zero temperature, the bond lengths and angles will be perturbed, changing the position of the atoms. Molecular dynamics (MD) simulations use different methods to realistically change the molecular system with time. The common way is to produce a force-field from the potential energy surface computed for different configurations of the same molecule. Each configuration has an energy: $E(\mathbf{r}_1, \ldots, \mathbf{r}_{n_{atoms}})$ For every atom i, the force it experiences is given by the gradient

$$\mathbf{F}_i = -\frac{\partial E}{\partial \mathbf{r}_i}$$

The dataset MD-17, is a set of configurations from MD trajectories for a sample subset of molecules for example aspirin, uracil and salycylic acid. [15]

Besides a graph, another representation of a molecule is a set of $(Z, \mathbf{r})$ pairs, where Z is the proton number of an atom (1:H, 6:C, etc) and $\mathbf{r}$ is its position vector. One important step in developing any machine learning model is to transform these representations into something that reflects the symmetries of a molecule. For example, graphs and pairs respect permutational symmetry (any two atoms of the same species can be swapped). Other necessary symmetries are global rotation and translation. A very simple representation of a molecule is "bag of bonds", which is simply a list of bonds that a molecule contains (CC, OC, HC etc). The presence and multiplicity of bonds is the input feature. For example, methane CH_4 has one type of bond ($C - H$) with multiplicity 4. Another representation is a *Coulomb matrix* which is an adjacency matrix where each element corresponds to the weighted inverse distance of two atoms i.e. $M_{ij} = \frac{Z_i Z_j}{|\mathbf{r}_i - \mathbf{r}_j|}$ but $M_{ii} \simeq Z_i^{2\alpha}$ for some hyperparameter α.

3 Review

Go through the different works in some logical order, such as chronologically or by going from simple to complex models. Do not just list the methods, but say how they relate to each other (going through the strengths/weaknesses of the different methods, both in comparison to each other and compared to an ideal method that solves the problem).

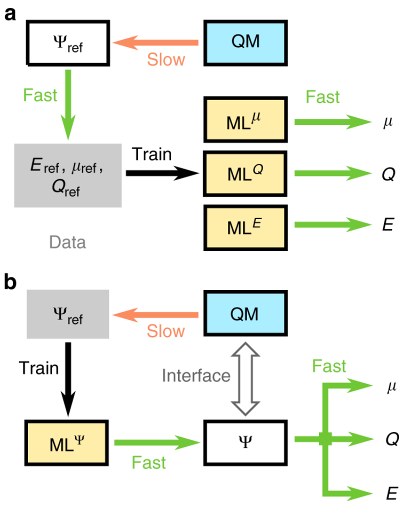

Figure 2: Synergy of quantum chemistry and machine learning. **a** Forward model: ML predicts chemical properties based on reference calculations. If another property is required, an additional ML model has to be trained. **b** Hybrid model: ML predicts the wavefunction. All ground state properties can be calculated and no additional ML is required. The wavefunctions can act as an interface between ML and QM [9]

Different machine learning applications to chemistry can be classified as forward models or hybrid models. A forward model learns based on one predicted value i.e. the atomization energy. A hybrid model attempts to learn the structure of the molecular wavefunction and computes the properties from that. See Figure 2.

The first machine learning applications were forward models that simply performed regression. Many attempts have been made to perform regressions on different representations of molecules.[12] Faber et al. presents a summary of many different regressors using many different atomistic representations. [13]

In 2012, a non-parametric kernel regression model was used. Given a configuration, it predicts an atomization energy. It's a non-parametric model, with one Gaussian radial basis function (RBF) for every element in the training set. The distances between molecules between entire Coulomb matrix representations are used as the arguments of the RBFs. Energy of any molecule would be computed as a sum of weighted RBFs. [5]

The below methods claim to improve on the above regressions. They develop parametric models to explicitly represent the substructure of molecules. They intend to extract insight from the parameters.

GDML learns the force-field of a molecule by learning a constrained kernel matrix. The conservation of energy is baked in.

The neural network models try to explicitly represent the interactions between subsets of atoms within the molecules. SchNet uses convolutions to capture local structure. Cormorant explicitly includes interaction terms that have the symmetries of real interactions.

The common benchmark is an error of 1 kcal/mol difference in molecular energy from DFT calculations. The below models all achieve this. They all use some form of stochastic gradient descent for training but have different architecture.

3.1 GDML and sGDML

Gradient domain machine learning (GDML[1]) is only applied to MD datasets. The force field extracted from each sampled configuration along MD trajectories is what is learned.

The GDML approach explicitly constructs an energy-conserving force field. A vector valued kernel is used. This consists of a Hessian matrix (a matrix of second-order derivatives $\frac{\partial^2 \kappa}{\partial x_i \partial x_j'}$) from a standard kernel function $\kappa(x, x')$ (in particular a Matern kernel). However the inverse distance between atoms is used, resulting in a similar representation to a Coulomb matrix.

The predictor $\vec{F}(\vec{x})$ is a weighted sum over atoms of derivatives of $\kappa(\vec{x}, \vec{x}_i)$. The potential energy surface can also be calculated simply by integrating this function, which can be done analytically.

Symmetric GDML (sGDML[2]) adds additional permutational symmetry to GDML. The kernel is modified $\kappa(x, P_{ij}x_i)$. Improves the data efficiency. This improves the training accuracy by 30 to 60%.

In more recent applications, the resulting force-fields are analysed and it seems that the atomistic physics is reproduced. The molecules have effects of electron lone pairs and hydrogen bonding. The models are interpretable in the sense that the physical intuition of chemists is followed. [3]

The code was published in [4].

3.2 DTNN and SchNet[1]

Deep tensor neural networks (DTNN[7]) build a neural network on top of atomistic representations. We briefly expalain this method.

DTNN receives molecular structures through a vector of nuclear charges Z and a matrix of atomic distances D ensuring rotational and translational invariance by construction. The total energy E_M for the molecule M composed of N atoms is written as a sum over N atomic energy contributions E_i. Each atom i is represented by a coefficient vector $c \in \mathbb{R}^B$, where B is the number of features. Then these vectors are rewritten in some basis (called quantum-chemical atomic basis set expansions) c_i^0. Then, this atomic expansion is repeatedly refined by pairwise interactions with the surrounding atoms

$$c_i^{t+1} = c_i^t + \sum_{j \neq i} v_{ij}$$

where v_{ij} are interaction terms reflecting the effect of atom j that is in distance d_{ij} from the atom i. The terms v_{ij} form a tensor V. Since it has many parameters, it is computentionally difficult to be learned and also subject to overfitting, therefore they are learned via low-rank tensor factorization, which is a generalization of the matrix factorization (See [7] for more details). In DTNN, the initial atomic representations only consider isolated atoms, the interaction terms c_i^1 characterize how the basis functions of two atoms overlap with each other at a certain distance. The following steps are supposed to reduce these overlaps and so, embedding the atoms of the molecule into their chemical environment. In total $T = 3$ are done in their experiment. Arriving at the final embedding after T interaction refinements, two fully-connected layers predict an energy contribution from each atomic coefficient vector c_i, such that their sum corresponds to the total molecular energy E_M.

SchNet is based on DTNN but adds continuous-filter convolutional (CFC) layers. These compute integrals for convolutions rather than using a convolution matrix. The parameters of a convolution function are learned. [8] SchNet alternates CFC, fully connected and "softplus" activation functions. SchNet achieves a 0.5 kcal/mol error with fewer layers than DTNN. However a drawback of DTNN and SchNet is that the latent features they capture are in some sense too local. These networks do not work well at predicting the global dipole moment of the molecules.

One improvement made in SchNet, compared to DTNN is that the activation functions are differentiable. This means they can also reproduce force fields from MD calculations as in GDML [6].

Two additional improvements are SchNOrb (SchNet for Orbitals) and Generative SchNet (G-SchNet). SchNOrb adds a new type of interaction layer to SchNet [9]. G-SchNet produces configurational models. Given a set of bonded atoms, the model predicts the position of the next atom added to the molecule. This is a proof of principle that new molecules could be engineered through this method. [10]

3.3 Covariant Neural Networks and Cormorant

[16][11]

[1] Despite a long search we don't believe the somewhat silly-sounding name "SchNet" is an acronym. It is likely just a combination of the name of primary developer (Schütt) and "network".

The main idea of Coromorant is explicitly "bake-in" symmetries into activation layers Each atom or set of atom is a charge distribution. A charge distribution can be expanded as a series of multipole. These are shown in Figure 3. The multipoles are transform neatly under rotations. The physical interactions between molecules can be expressed as a series of interactions between multipoles.

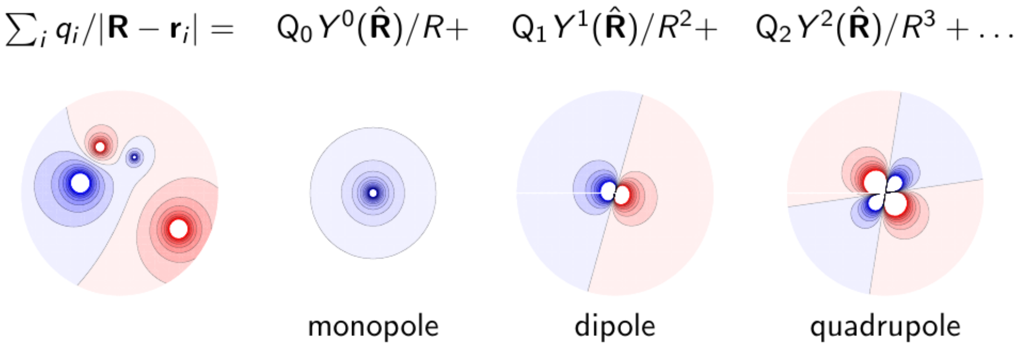
$\sum_i q_i/|\mathbf{R}-\mathbf{r}_i| = \quad Q_0 Y^0(\hat{\mathbf{R}})/R + \quad Q_1 Y^1(\hat{\mathbf{R}})/R^2 + \quad Q_2 Y^2(\hat{\mathbf{R}})/R^3 + \ldots$
monopole
dipole
quadrupole

Figure 3: Multipole expansion: the electrical potential at the point $\mathbf{R}$ is given by the sum of interactions with charges q_i at points $\mathbf{r_i}$ in the charge distribution. The parameters Q_i can be learned by the network. [**?**]

In the results recently presented at NeurIPS, Cormorant outperforms sGDML and SchNet?

4 Discussion

Discuss the trends that have occurred over time. Speculate about where the next steps in the trend could lead. Point out issues that are not properly addressed by existing methods. State some interesting directions to explore, or opportunities to use existing tools in new applications.

There has been an explosion of research in the last 10 years. The approaches in the above section have different costs and benefits and new methods could be developed for more niche applications.

With very small datasets (100s of examples) kernel methods (like sGDML) do better [3]. if a lot of data is available, neural networks can make predictions much faster and are more accurate for larger molecules.

We foresee a future in which calculations from physical theory and ML could work in tandem . Some methods for solving for the wavefunction of molecules are iterative. A machine learning method could produce an approximation that could be further iterated [3]. This would achieve equally precise results as an exact calculation with less computing time.

The methods describe might also be adapted to the field of nuclear physics. Some early attempts at this use neural networks to predict nuclear masses [18]. Perhaps insights about nuclear structure can be gained from machine learning models.

A final point of development is in quantum machine learning [17]. If quantum computers become feasible then solving the electronic quantum mechanical problem may become easier. Would the field switch the other way and use quantum methods for machine learning problem rather than machine learning for quantum problems as covered in the paper.

References

[1] Chmiela, Stefan; Tkatchenko, Alexandre; Sauceda, Huziel; Poltasvkyi, Igor; Schuett, Kristof; Mueller, Klaus-Robert, *Machine learning of accurate energy-conserving molecular force fields*, Science Advances (2017), 3

[2] Chmiela NATURE COMMUNICATIONS | (2018)9:3887 | DOI: 10.1038/s41467-018-06169-2

[3] Sauceda J. Chem. Phys. 150, 114102 (2019); https://doi.org/10.1063/1.5078687

[4] Chmiela Computer Physics Communications 240 (2019) 38–45

[5] Rupp, 2012

[6] Schutt, Thesis

[7] Schutt NATURE COMMUNICATIONS | 8:13890 | DOI: 10.1038/ncomms13890 | www.nature.com/naturecommunications

[8] Schutt J. Chem. Phys. 148, 241722 (2018); https://doi.org/10.1063/1.5019779

[9] NATURE COMMUNICATIONS | (2019)10:5024 | https://doi.org/10.1038/s41467-019-12875-2 | www.nature.com/naturecommunications

[10] N. W. A. Gebauer, M. Gastegger, K. T. Schütt. *Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules*, arXiv:1906.00957, 2019

[11] Anderson, 2019

[12] von Lilienfeld, O. Anatole *Quantum Machine Learning in Chemical Compound Space* Angewandte Chemie International Edition, 04/2018, Volume 57, Issue 16

[13] Faber J. Chem. Theory Comput. 2017, 13, 5255-5264

[14] Ramakrishnan, 2014

[15] Chmiela?

[16] Hy J. Chem. Phys. 148, 241745 (2018); https://doi.org/10.1063/1.5024797

[17] Biamonte, doi:10.1038/nature23474

[18] 10.1103/PhysRevC.98.034318