AtomCloudNet: Deep learning molecular and atomic properties using point convolution architecture

Names: Flurin Hidber, Kenneth Atz Nethz: hidberf, kenatz
Email: hidberf@student.ethz.ch, kenneth.atz@pharma.ethz.ch IDs: 14-928-451, 14-064-851
Course: Deep Learning, Prof. Thomas Hofmann, Fall Semester 2019

Abstract

Herein, we introduce AtomCloudNet (ACN), a deep learning model to predict molecular and atomic properties using information of individual atomic environments extracted by point cloud convolution. We exemplify the capability of our model by learning ground state atomization energies U_{rt} of molecules calculated by first principal methods. Information of atomic position $\mathbf{R}_i$ and nuclear charge Z_i are mapped to a molecular property via applying spherical harmonic functions $Y_l^m(\phi, \Theta)$. This resulted in a MAE of 1.31 kcal/mol. Therefore, ACN represents a novel approach for the prediction molecular and atomic properties.

A Introduction

A.1 Quantum machine learning

Quantum mechanics (QM) gives scientists the ability to calculate accurate microscopic properties, such as energies, atomic forces, electronic energy levels, electrostatic multipoles and polarizabilities for fixed molecular geometries. Such first-principle approaches are necessary for the discovery of novel drugs and materials. However, accurate QM simulations are computationally demanding, even for a single molecule, hence more efficient approaches are urgently needed [*Lilienfeld et al.* 2019]. Thereby, machine learning (ML), especially deep learning (DL) [*Schneider et al.* 2019] gives large potential to perform accurate property prediction of molecules at similar accuracy as QM simulations, but in much faster fashion by demanding milliseconds instead of hours or days.

A.2 Related work

A.2.1 Kernel based machine learning

Kernel Ridge regression (KRR) is one of the most commonly used ML methods within molecular and materials science. Currently these models outperform neural networks and all other methods on tasks of molecular and atomic property prediction. In these models molecules are transformed into highly engineered feature vectors based on nuclear two-, three- or n-body interactions. Then this input is mapped and fitted into a feature space. However, the best feature space is a priori unknown, and its construction is computationally hard, the "kerneltrick" solves this problem by applying a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ on a representation space R that yields inner products of an implicit high dimensional feature space. This matrix elements $k(\mathbf{x}_i, \mathbf{x}_j)$ of two representations $x \in R$ between two input molecules i and j are the inner products $i|j$ in the feature space. Thereby a gaussian kernels (Equation 6) can be used to map the two input vectors, where σ represents the length scale hyperparameter (Equation 1).

$$k(\mathbf{x}_i, \mathbf{x}_j) = exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j)||}{2\sigma^2}) \tag{1}$$

Fitting coefficients α can then be computed in input space via the inverse of the kernel matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, where λ is the regularization strength, typically very small for calculated noise-free quantum chemistry data (Equation 7).

$$\alpha = (K_{ij} + \lambda \mathbf{I})^{-1}\mathbf{y} \tag{2}$$

Thereby, KRR models learn a mapping function from the inputs x_i, in this case the representation of the molecule, to a property $y_\mu^{est}(\mathbf{x}_\mu)$ (Equation 3) .

$$y_\mu^{est}(\mathbf{x}_\mu) = \sum_i^N \alpha * k(\mathbf{x}_i, \mathbf{x}_j) \tag{3}$$

The key parameter for such kernel based methods for property prediction is the molecular representation $\mathbf{x}_i$. Therefore, best performing kernel methods are those with the highest engineered feature vectors. Best performing representations are SLATM- [*Huang et al. 2018*], FCHL- [*Faber et al. 2018*] and SOAP [*Willatt et al.* 2019].

A.2.2 Neural networks

Deep neural networks (DNN's) such as DTNN [*Schütt et al.* 2017], SchNet [*Schütt et al.* 2018] or Cormorant [Anderson et. al. 2019] were reported as well to predict molecular properties. Such DNN's are able to construct an implicit multiscale representation as an outcome of a scalable learning process. However, DNN's so far get significantly outperformed by KRR methods. This results due to the fact that kernel methods learn almost noise-free functions which is better equipped to map quantum mechanical data with almost zero error.

B Methods

B.1 Data

Herein, we use the QM9 data set, which was reported in 2014 and contains 134k small organic molecules composed of H, C, N, O, and F with up to 9 heavy atoms. Whereby for each molecule 15 different properties are reported [*Ramakrishnan, et al.* 2014].

B.2 Point convolution architecture

We challenge current models by proposing a deep neural network with point convolution architecture named AtomCloud-Net (ACN), which takes as input the molecule with coordinates $\mathbf{R}_i$ and nuclear charges Z_i for each atom and outputs the desired molecular or atomic property. Such point convolution frameworks have already been introduced for object classification and part segmentation [*Qi et al.* 2017 and *Qi et al.* 2017].

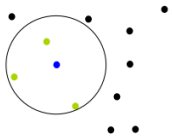

Figure 1: Visual representation of the neighbourhood of a centroid atom (blue). Each atom consisting of its position $\mathbf{R}_i$ and atom type Z_i.

B.3 Molecular and atomic representation

The many-body system in ACN is described by $\mathbf{R}_i^\mu$ and Z_i^μ of an ensemble of atoms. The molecule can be expressed as a set of nuclear charges $Z = (Z_1, \ldots, Z_n)$ and atomic positions $R = (r_1, \ldots, r_n)$. From the first layer on, each atom is embedded into a feature vector and described by an array of features $Z_i = [a_Z^1, ..., a_Z^n]$ (Equation 4). Embeddings are initialized randomly and optimized during training to learn a higher dimensional abstraction of each atom type Z_i. In our model each atom type Z_i is embedded into 24 features.

$$Z_i = [a_{Z\,i}^1, ..., a_{Z\,i}^{24}] \tag{4}$$

B.4 Many-body potentials

Into the atomic representation we introduce potentials for two- and three-body interactions V_{ij}^μ and V_{ijk}^μ as well as one-body terms V_i^μ, which are equivalent to nuclear charge Z_i^μ. Two-body terms V_{ij}^μ are calculated according to coulomb repulsions $\frac{Z_i Z_j}{R}$ with the dissociative tail of the London potential $\frac{1}{R^6}$ [London et al. 1930] (Egaution 1). For three-body terms V_{ijk}^μ we used the van-der-Walls potential according to Axilrod, Teller and Muto [Axilrod and Teller, 1943] (Eqaution 2). Similar formulation of two- and three-body potentials have proven to perform well in machine learning models [Huang et. al. 2018]. For both terms the individual sum over all atoms per atom i per molecule μ is taken, such that per atom one value for V_{ij}^μ and V_{ijk}^μ is yielded.

$$V_i^\mu = Z_i^\mu \tag{5}$$

$$V_{ij}^\mu = \begin{cases} if: i \neq j, & Z_i \sum \frac{Z_j}{R_{ij}^6} \\ else: & 0 \end{cases} \tag{6}$$

$$\mathrm{V}_{ijk}^\mu = \begin{cases} if: i \neq j \neq k, & Z_i \sum Z_j Z_k \frac{1+cos(\alpha)cos(\beta)cos(\gamma)}{(R_{ij}R_{jk}R_{ki})^3} \\ else: & 0 \end{cases} \tag{7}$$

B.5 Real spherical harmonics - AtomCloud

Real spherical harmonics, first introduced by Pierre Simon de Laplace in 1782 are functions defined on the surface of a sphere. The spherical harmonic function $Y_l^m(\phi, \Theta)$ of degree l and order m is described as:

$$Y_l^m(\phi, \Theta) = Ne^{(im\phi)} P_l^m(cos\Theta) \tag{8}$$

whereby, $P_l^m(cos\Theta)$ is an associated Legendre polynomial, N a normalization constant and ϕ and Θ represent colatitude and longitude angels, respectively. Radius r, as well as coordinates x, y and z can be defined by angles ϕ and Θ (Equations 10-12).

$$x = r * sin\phi * cos\Theta \tag{9}$$

$$y = r * sin\phi * sin\Theta \tag{10}$$

$$z = r * cos\phi \tag{11}$$

By applying $Y_l^m(\phi, \Theta)$ to atomic inputs $(\mathbf{R}_i^\mu, Z_i^\mu)$ pair-wise atomic distances are used instead of relative atomic positions, which allows for rotation-invariant feature extraction, which is necessary in order to achieve efficient learning [Lilienfeld]. In ACN we are applying such a spherical harmonic function onto each atom. Thereby the three hyperparameter radius r, degree l and order m are optimized. The two parameter degree l and order m are defined in our code as cloudorder. After applying this spherical harmonic function each atom is represented by a feature abstraction of its unique atomic environment.

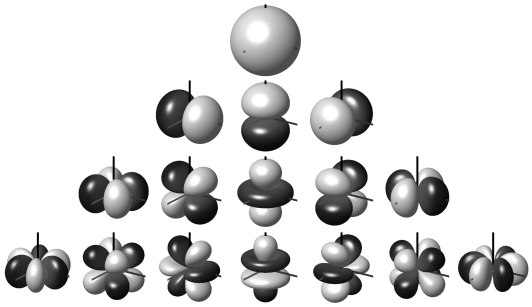

Figure 2: Representation of all combinations of real spherical harmonics using degree $l = 0, ..., 4$ and order $m = -l, ..., +l$.

B.6 Architecture

The detailed point convolution architecture of ACN is shown in Figure 3. Thereby the molecule is used as input for the network in its lowest possible abstraction, position $\mathbf{R}_i$ and atom type Z_i. To allow for higher batch sizes than one, each molecule is padded to 30 atoms using filler atoms with nuclear charge $Z_i = 0$ and position $\mathbf{R}_i = [20, 20, 20]$.

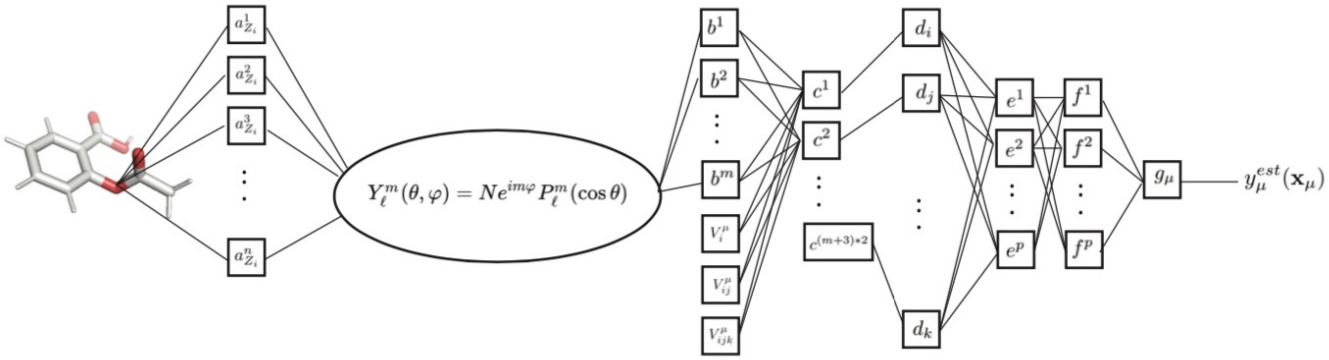

Figure 3: ACN is capable of abstracting molecular information based on positional input data of molecules via the following architecture. First an embedding layer is applied to each input atom type Z_i mapping the discrete atom types into a higher feature space **a**. The spherical harmonic point-convolution kernel is applied to each atoms features and its atomic neighbourhood in a translation-, mirror- and rotation-invariant manner. Whereby features of atoms within a certain euclidean distance are taken into account. This generates new atomic features **b**, which then are concatenated with pre-calculated engineered features for one-, two- and three-body potentials V_i^μ, V_{ij}^μ and V_{ijk}^μ and passed to a residual block, where the output of this block is concatenated with its input resulting in a doubling of atomic vector size **c**. Further, power-average pooling over each feature for all atoms is applied, resulting in a molecular representation **d**. Finally two linear layers with softplus activation function and one output layer with sigmoid activation function are applied to result in the final output $y_\mu^{est}(\mathbf{x}_\mu)$, in our case the atomization energy U_{rt}.

C Results

C.1 Baseline and state of the art models

As comparison for the performance of ACN we chose two well performing kernel methods using two commonly used representations, namely coulomb matrix (CM) [Rupp et. al.] and SLATM [*Huang et al. 2018*]. Both representations are based on complex engineered feature vectors calculated from the 3D structure of each molecule $(\mathbf{R}_i^\mu, Z_i^\mu)$. Therefore, CM is used as baseline method and SLATM as state of the art method.

C.2 Learning

To quantify the performance of our model, the test errors, measured as mean absolute errors (MAE) in kcal/mol were calculated as a function of training set size. The leading error term is known to be inversely proportional to the amount of training points used: $MAE \approx a/N^b$ [Ritter et. al. 2001]. Therefore, the learning curve should then result in a decreasing linear curve with slope b and offset $log(a)$. Learning curves for ACN performance with different radius sizes in Å for the spherical harmonic function $(Y_l^m(\phi,\Theta), Y_l^m(r))$ show for all a decrease in MAE (kcal/mol) with an increase in training set size (Figure 4). This observation is similar to kernel models with CM and SLATM representation. Further, there is a slight distinction seen in applying different radii in range 3 to 5. Radius of 3 Å shows lowest MAE. This might be due to loss of important information of closer atoms when the radius reaches a certain point. Radius of 2 Å resulted in similar results as 3, and radius of 1 Å lead to equally bad results as 5. Furthermore, ACN trained on 10k training examples with using $Y_l^m(r=3)$ results in the lowest MAE at 131 kcal/mol. There were only a small margin of molecules found, which were not present near the regression line.

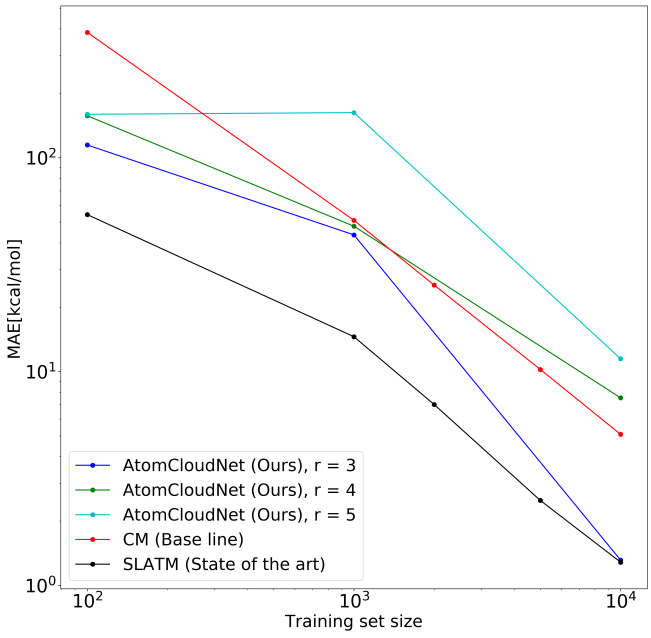

Figure 4: Learning curves showing mean absolute errors in kcal/mol for the prediction of atomization energies U_{rt} of an independent test set.

$N_{Training}$	CM	SLATM	ACN3	ACN4	ACN5
100	385.689±95	**54.176±170**	114.65	157.05	159.57
1000	50.944±10	**14.54±1.2**	47.81	47.12	162.49
10000	5.82±0.5	**1.28±0.01**	1.31	7.53	11.47

Table 1: MAE for each model with different training set size. Models: CM, SLATM and ACN with = 3, 4 and 5.

No special geometries were found, where ACN performed significantly better or worse. Observing the outliers a few trends can be seen. All outliers were found at the end of largest and smallest energies and were composed by multiple fluorine atoms. This can be justified by lower abundance of very big and very small molecules, as well special properties of fluorine containing molecules in the Training set. The scatter plot of real and predicted U_{rt} values looks almost flawless (Figure 5). Table 1 shows all exact MAE values for ACN with radii 3, 4 and 5.

D Conclusion

The approach of using a point convolution network architecture for learning molecular properties has shown to be successful for the prediction of atomization energies U_{rt} of small molecules. Although kernel models underlying strictly defined feature representations of molecules still perform slightly better, we could show that a novel deep learning approach results in comparable results. There is still much to be explored in ACN, e.g. applying multiple convolutions in a row for each atom or applying different cloud orders m and l at the same time for different atom types. Further, we did not yet explore its performance on the prediction of other molecular properties, such as dipol moment D, HOMO/LUMO energies U_{HOMO}/U_{LUMO}, isotropic polarizability α_0^3 or rotational constants, as well as the prediction of atomic properties, such as NMR shifts σ, partial charges $\sigma^{\pm}$ or nuclear spin-spin couplings ${}^nJ_{HH}$.

E Software

To reproduce the two kernel learning methods CM and SLATM, the QML toolkit was implemented. For introducing real spherical harmonics we used the se3cnn package. Lastly, for all other deep learning implementations we applied PyTorch. All code is provided on GitHub.

2

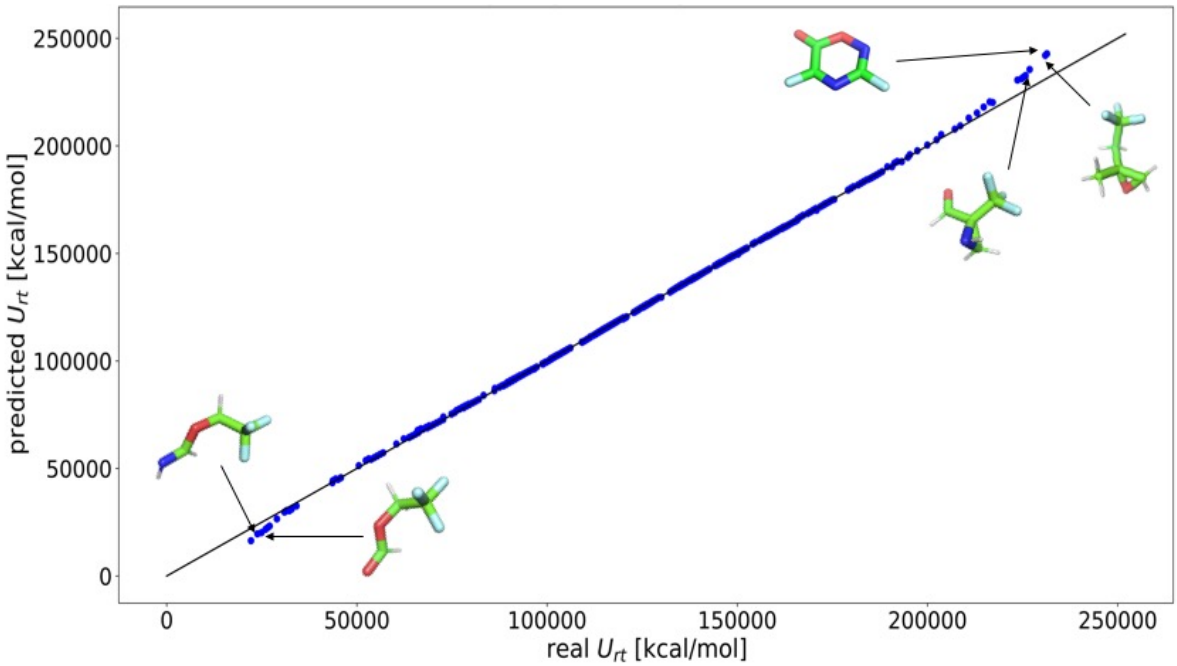

Figure 5: Scatter plot of real and predicted U_{rt} values, showing almost no outliers. There was no geometry or functional group found, where significant bad prediction was observed. Largest outliers, but with relatively small MAEs are found on both ends of the size spectrum of molecules.