INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

Learning under physical constraints

Molecular energy prediction

Alexandre Personnic, Dylan Lebreton, Lucas Bouillon

2th year of Modelisation and Artificial Intelligence double degree
at INSA and ENSEEIHT - Toulouse, France

Supervisor: Sixin Zhang

30 June 2023

ModIA
Promotion 2024

2022 - 2023

Contents

1 Introduction

The objective of this project is to create a model that accurately represents the potential energy distribution among atoms in small organic molecules. The main task is to develop a comprehensive model that can predict the atomization energy E(r) for all molecules according to $r = \{r_1, r_2, \ldots, r_N\}$ the positions of atoms in three-dimensional space. The main challenge is to keep the predictions invariant by translation, rotation or permutation of r.

1.1 Dataset

For this challenge, we will use a custom subset of the QM7-X dataset[1] with 4739 molecular structures: 3792 in the train set and 947 in the test set. For every molecule, we have the 3D coordinates of its atoms and their atomic number. For the train set we also have their energy. We use custom functions to load the data from the *.csv* and the *.xyz*.

1.2 Data Exploration

To better understand our data, we decide to visualize some aspect of it. First we look at the distribution of atoms in our train and test dataset. We can see in figure 1 that we mainly have Hydrogen and Carbon atoms with a few Oxygen and Nitrogen. It seems logical since we are dealing with organic structures. Additionally, we can confirm that there is no imbalance in the distribution of atoms in the train and test set, we have roughly the same proportions.

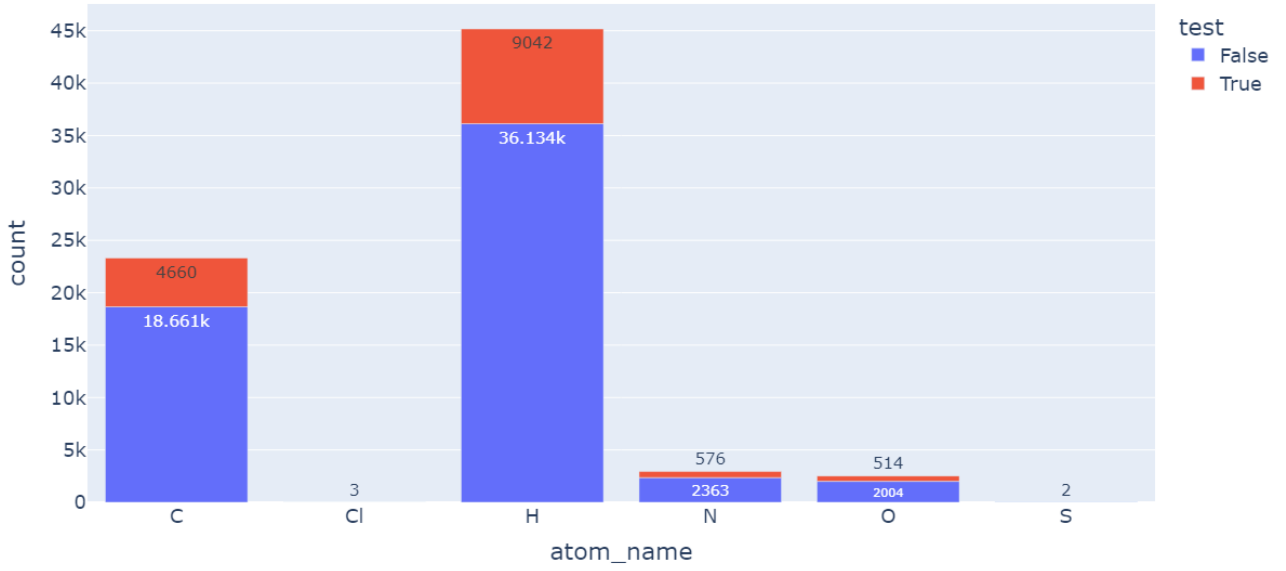

Figure 1: Total number of atoms in dataset.

Another important thing to look for is the existence of a simple relation between the variable we want to predict, the energy, and the others. The most obvious variable to look at, and the easiest, is the total number of charges in the molecule. Figure 2 plot the energy as function of the total number of charges in a molecule, the color depends on the number of atoms in the molecule. We can clearly see a linear relation with the energy decreasing when the total charge increase. Additionally, when the total charge is the same, a molecule with more atoms tends to have a lower atomization energy. We can expect to have preliminary results with a simple linear regression even if not good.

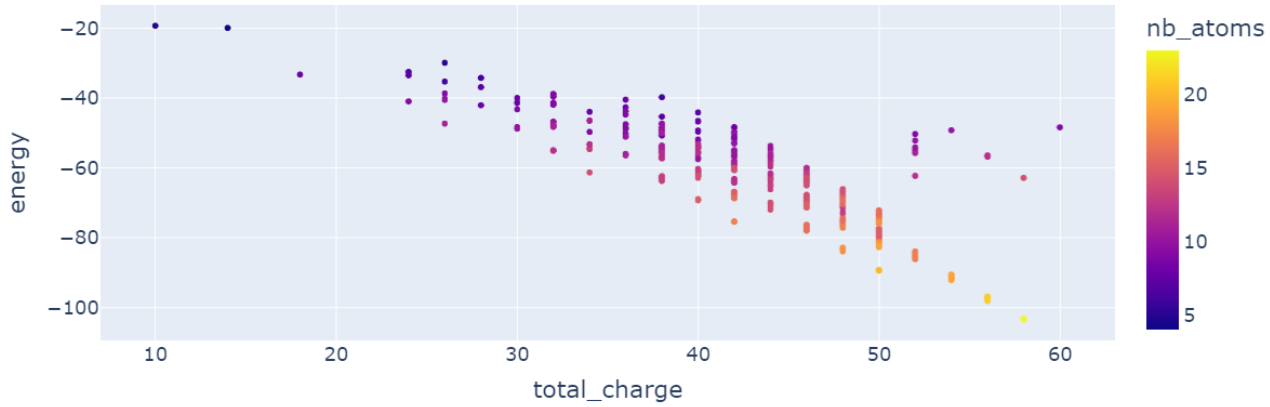

Figure 2: Energy as function of total charge.

2 Problem-solving approach

2.1 Approach 1: Molecular graph construction

To free ourselves from the constraints of translations, rotations and permutations, a logical representation of molecules is a representation in the form of graphs. The molecule is then defined by the undirected graph $\mathcal{G} = (\mathcal{V}, \varepsilon)$ with :

- N nodes $v_i \in \mathcal{V}$ representing the atoms
- edges $(v_i, v_j) \in \varepsilon$
- a weighted adjency matrix $A \in \mathbb{R}^{N \times N}$ representing euclidian distances between the nodes (atoms)
- a degree matrix D with $D_{ii} = \sum_j A_{ij}$ representing the degree of a node (atom) defined as the sum of the weights of all edges connected to that node

We chose to use the `Spektral` library in Python to associate a graph with each molecule. The construction mainly involves calculating the adjacency matrix. The library also define a graph output vector y and features data to each node, such that we can use the graph in the context of neural network layers to predict y from the features. In a regular neural network, the forward pass is defined as:

$$H^{(l+1)} = \sigma\left(W^{(l)} H^{(l)} + b^{(l)}\right)$$

With $H^{(i)}$ and b^i the output and the bias of the i^{th} layer, and σ the activation function. The Graph Convolutional Layers implemented in `Spektral` define the forward pass as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\right)$$

Where $\tilde{A} = A + I_n$ is the adjacendy matrix with self connections and $\tilde{D}$ the corresponding degree matrix. Thus, we can define a neural network using these layers to predict molecular energy using atoms features and graph representation of molecules.

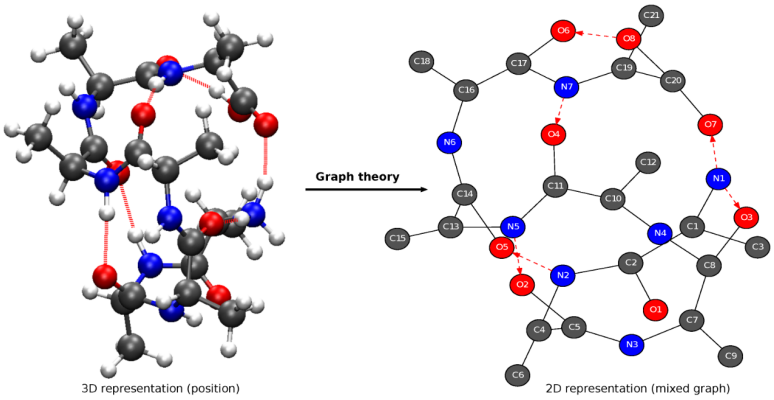

Figure 3: Molecular Graph exemple.

2.2 Approach 2: Scattering Transforms

The primary reason we chose to use Scattering Transforms is because it is a robust tool specifically designed to maintain stability even under different transformations like translations and rotations. By incorporating these invariances, the model becomes more resilient to variations in the input, resulting in enhanced generalization performance.

Additionally, scattering transforms provide a multiscale analysis of the input data, enabling the extraction of features at various resolutions.

Lastly, scattering transforms are supported by a strong mathematical foundation, ensuring their reliability and interpretability.

We worked in Python, using the Kymatio [2] library that offer a HarmonicScattering3D class. Here is a quick breakdown of the code:
Upon importing the data, we normalize the position of each atom.
Specifically, the positions are rescaled in such a way that two Gaussians with a width of sigma, placed at those positions, overlap with an amplitude lower than a predefined precision.
Next, we define our Scattering function, utilizing the HarmonicScattering3D class from kymatio library, with the following parameters: $J = 4, M = 256, N = 160, O = 112, L = 5, sigma = 2.0, integral_powers = [0.5, 1.0, 2.0, 3.0]$.
This significant Scattering function performs a 3D wavelet scattering on a set of molecules, considering their positions, nuclear charges, and valence charges. It concatenates the scattering coefficients from all batches into an array.
Following this function, we split the data into training and validation sets.
We train the data using different regression methods and evaluate our model using RMSE.

3 Analysis of results

The RMSE of all our methods are displayed in table 1.

Table 1: RMSE of all methods tested

Method	**Data**	**RMSE**
Graph Network	Molecular Graph	16.16
Ridge $\alpha = 10^{-5}$	Total charge	5.686
	Number of atoms	3.444
	Total charge + Number of atoms	3.148
	Scattering rescaled	0.0560
	Scattering	**0.0502**
Decision Tree		0.468
MLP		1.184
SVM		2.136
ElasticNet $l_1\_ratio = 0.5$		0.256

3.1 Graph Network

For energy prediction using graph convolutional network, the following hyperparameters have been tested:

- Optimizer: SGD and Adam
- Learning rate: 1e-1, 1e-2 and 1e-3
- Epochs: 200, 500, 1000, 2000
- Batch size: 32, 64, 128, 256
- Data: dataset split in 80% for training and 20

With most combinations of these parameters, convergence was challenging. The losses being very high initially, ranging from 10e4 to 10e5, a too low learning rate did not allow for an acceptable loss despite a large number of epochs. Similarly, a too high learning rate made the algorithm converge too quickly towards a local minimum for a loss situated between 100.0 and 500.0.

We managed to obtain more satisfactory results by adding callback conditions on the loss: every 50 epochs, the learning rate was multiplied by 0.2 if the loss on the validation data did not change significantly (with an initial learning rate of 1e-1 and a minimum learning rate of 1e-3).

A better result is observed in figure 4: no overfitting, a final loss of 11.0546 on training data and 10.6102 on validation data. However, we were unable to achieve better results on the Kaggle test set than with the methods using the scattering coefficients discussed in the following section.

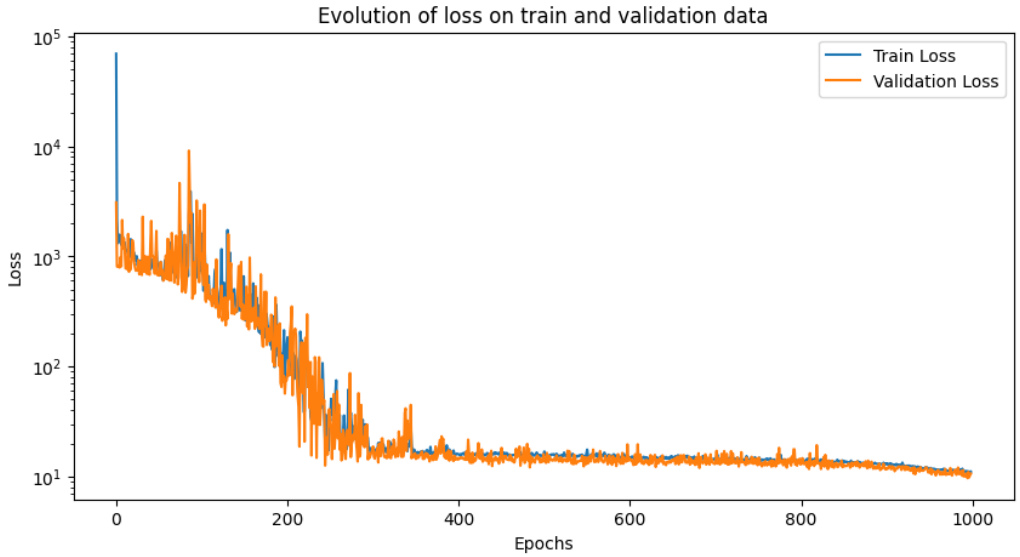

Figure 4: Evolution of train and validation loss over epoch

3.2 Regressions

For the regression, we tried various different approaches. First, we examined simple linear regression with a Ridge regularization factor of $\alpha = 10^{-5}$ on raw data, as theorized in the data exploration phase. Since we only considered the number of charges and atoms, we didn't have to worry about invariance. However, the RMSE obtained with these methods was quite high, ranging from 3.15 to 5.69, and therefore the results were unsatisfactory.

Next, we experimented with several different regressions on our scattering coefficients: regression with a Ridge regularization factor of $\alpha = 10^{-5}$, Decision Tree, MLP with hidden states of $(64, 32, 8)$, SVM with default sklearn parameters, and ElasticNet with $\alpha = 10^{-5}$ and $l_1\_ratio = 0.5$.

Among all these methods, the results of which are presented in table 1, Ridge regression yielded the best RMSE on our test data with a score of 0.0502. We tried to further improve this method by applying different scaling techniques to our data but we couldn't best our results.

4 Conclusion

To conclude, in this project we developed a model for predicting the atomization energy of organic molecules. In the two main approaches we explored,(molecular graph construction and scattering transforms), we have been able to obtain results, and promising results specially with the scattering transforms approach. It outperforms the other regression methods tested by far such as the graph network. We have not been able to properly implement this approach and we quickly to moved to scattering transforms. Further research and optimization can be pursued to enhance the models' accuracy.

All our code can be found on our github.

References

[1] Johannes Hoja et al. "QM7-X, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules". en. In: *Scientific Data* 8.1 (Feb. 2021), p. 43. ISSN: 2052-4463. DOI: 10.1038/s41597-021-00812-2. URL: https://www.nature.com/articles/s41597-021-00812-2 (visited on 06/13/2023).

[2] Mathieu Andreux et al. *Kymatio: Scattering Transforms in Python*. 2022. arXiv: 1812.11214 [cs.LG].