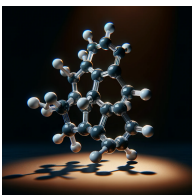
Predicting Atom Interactions with AI

Ludwig Schneider

Eric and Wendy Schmidt AI-Postdoctoral Fellow
Pritzker School of Molecular Engineering
University of Chicago

April 30, 2024



Moelcular Interaction: Non-Bonded I

- Newtons Equation of Motion

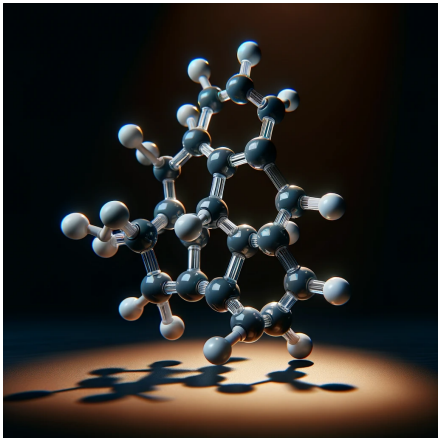
$$m_i \frac{\mathrm{d}^2}{\mathrm{d}t^2} \vec{r}_i = \vec{F}_i$$

- determine forces $\vec{F}$
- OPLS: functional form for the forcefield

$$V_{\mathrm{nb}}(r_{ij}) = \varepsilon_{ij} \left(\frac{\sigma_{ij}}{r_{ij}^{12}} - \frac{\sigma_{ij}}{r_{ij}^{6}} \right) + \frac{q_i q_j e^2}{4\pi\varepsilon_0 r_{ij}}$$

$$\varepsilon_{ij} = \sqrt{\varepsilon_i \varepsilon_j} \quad \sigma_{ij} = \sqrt{\sigma_i \sigma_j}$$

Challenge: determine ε_i, σ_i, q_i, and m_i

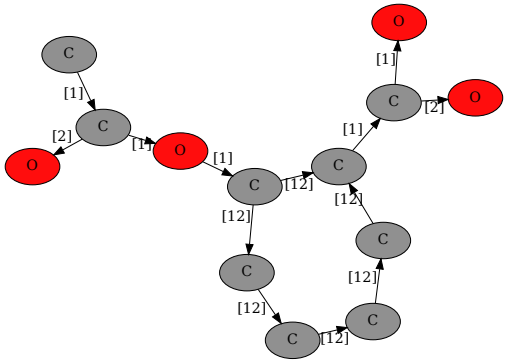


Moelcular Interaction: Non-Bonded Interactions II

- determine ε_i, σ_i, q_i, and m_i
  - element (carbon, hydrogen)
  - molecular environment
- we know some values
- SMARTS based rules
- Quantum Mechanic simulations
- chemical space is vast, we want to interpolate

Your Challenge

Use ML to interpolate from known molecules!



Details of the Basic Challenge

- 3000 molecular graphs
- known input parameters
- supervised learning possible
  1. predict non-bonded parameters
  2. predict uncertaintiy in interpolation
- you are free to choose your model
- you are free to choose tech stack
- make results permutation invariant

- `https://github.com/badeaa3/UChicago-AI-in-Science-Hackathon`
- `molecular/intro/talk.pdf` this slide deck
- `molecular/README.md` starting point
- `molecular/explain_graph_data.py` explanation of data set
- `molecular/data.json` training data
- molecular graph as `networkx` graphs

How to get Started?

1. Understand the Data

- visualize graphs
- transform to tech stack

Tech Stack Options

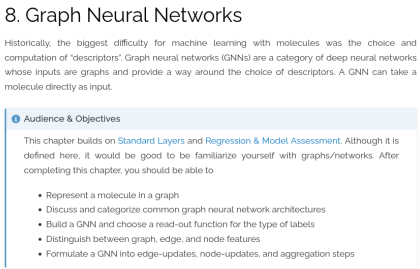
- Flax/JAX with Jraph
- PyTorch with Torch-Geometric
- Tensorflow with GNN

Tools

- visualization: https://graphviz.org/
- python: https://networkx.org/

- New to GNN?
- New to Molecular AI?
- Dmol introduction!

`https://dmol.pub/dl/gnn.html`



8. Graph Neural Networks

Historically, the biggest difficulty for machine learning with molecules was the choice and computation of "descriptors". Graph neural networks (GNNs) are a category of deep neural networks whose inputs are graphs and provide a way around the choice of descriptors. A GNN can take a molecule directly as input.

Audience & Objectives

This chapter builds on Standard Layers and Regression & Model Assessment. Although it is defined here, it would be good to be familiarize yourself with graphs/networks. After completing this chapter, you should be able to

- Represent a molecule in a graph
- Discuss and categorize common graph neural network architectures
- Build a GNN and choose a read-out function for the type of labels
- Distinguish between graph, edge, and node features
- Formulate a GNN into edge-updates, node-updates, and aggregation steps

How we Evaluate You!

- prepare a final presentation
- convince us your solution is best
- include model details and results
- 7 minutes to present
- 2 minutes for our questions

- Wednesday 5PM: we give you additional data
- DO NOT TRAIN with this data
- `molecular/final_evaluation.py`
- report output in your presentation

What if you solve the challenge too fast?

Traditional

- predict bond class for atom
- look-up table for parameters
- `ffbonded.itp`

GNN approach

- predict parameters directly
- bonds: edge features
- angles & dihedrals ?

- bonds: 2 atoms: 2 parameters

$$V_b(r_{ij}) = k_a(r_{ij} - b_0)^2$$

- angles: 3 atoms: 2 parameters

$$V_a(\theta) = k_\theta(\theta - \theta_0)^2 \quad \theta = \cos^{-1}(\vec{r_{ij}} \cdot \vec{r_{jk}}/(||\vec{r_{ij}}||||\vec{r_{jk}}||))$$

- dihedral: 4 atoms: 8 parameters

$$V_d(\phi) = \sum_{i=1,2,3,4} V_i/2[1 - (-1)^i \cos(i \cdot \phi - \phi_i)]$$

Happy Hacking!

Questions?