Summary

Unified Representation

对于分子, 按照量子力学, 它的 `势能面(potential energy surface)` 应当由 Schrodinger equation 确定:

$$E(\boldsymbol{X})\psi(\boldsymbol{x}) = \Big(-\sum_{i_e} \frac{\hbar^2 \nabla_{i_e}^2}{2m_e} + \sum_{i} \frac{Z_i e^2}{|\boldsymbol{r}_i - \boldsymbol{r}_{i_e}|} + E_{p-p} \Big) \psi(\boldsymbol{x})$$

$\boldsymbol{X} = \{\boldsymbol{r}_1, \cdots, \boldsymbol{r}_n\}$ 是原子核的构型, $\boldsymbol{x} = \{\boldsymbol{r}_{1_e}, \cdots, \boldsymbol{r}_{n_e}\}$ 是电子的构型.

其中 E_{p-p} 为各个原子核之间的 Coulomb 作用. 因此存在一个原则上能够应用到任意分子上的表示:

$$D(\boldsymbol{X})(\boldsymbol{r}) = \sum_i \frac{Z_i}{|\boldsymbol{r}_i - \boldsymbol{r}|}$$

即给定原子核构型带来的空间电场分布.

在我们的统一表示中, 我们首先利用 PCA 技术, 数据集合通过每个分子实例的电荷数和原子核坐标:

$$X = \{\boldsymbol{r}_1, \boldsymbol{r}_1, \cdots, \boldsymbol{r}_1, \boldsymbol{r}_2, \cdots\}$$

每个坐标重复的次数为这个坐标的原子的电荷数, 以此来找到合适的坐标轴方向. 然后再沿着这些方向来生成三维格点(分辨率为 (10 x 10 x 5)) 计算点阵上的电(势)场值 $D(\boldsymbol{X})(\boldsymbol{r} \in \mathrm{Grid})$. 用它来作为我们的输入数据

不难看出, 这种构造对于分子中原子构型的平移, 整体转动, 置换都是不变的.

我们将这些数据作为输入, 使用神经网络模型来进行监督学习.

模型中拥有两个隐藏层, 500 -> 1000 -> 100 -> 1, 激活函数选择使用 leaky_relu 来避免梯度消失问题. 为了提高在测试集合上的性能我们加入了 batch_normalization 和 dropout 技术. 最终在 1060ti 上经过 接近 10 个小时的训练得到了 0.181 的结果

分而治之

上面使用的表示尽管普遍, 但一方面计算困难(三维空间中的复杂度为 $O(mn^3)$), 另一方面训练难度较大. 我们接下来使用了一种分而治之的手段:

我们使用了 `ase` site 和 `dscribe` site 模块

引用:

1.

```
@article{dscribe,
  author = {Himanen, Lauri and J{\"a}ger, Marc O.~J. and Morooka, Eiaki
  title = {{DScribe: Library of descriptors for machine learning in mat
  journal = {Computer Physics Communications},
  volume = {247},
  pages = {106949},
  year = {2020},
  doi = {10.1016/j.cpc.2019.106949},
  url = {https://doi.org/10.1016/j.cpc.2019.106949},
  issn = {0010-4655}
}
```

利用分子的 `中心对称函数(Atom-centered Symmetry Function)`:

$$
\begin{aligned}
G_i^{(1)} &= \sum_j f_c(R_{ij}) \\
G_i^{(2)} &= \sum_j e^{-\eta(R_{ij}-R_s)^2} f_c(R_{ij}) \\
G_i^{(3)} &= \sum_j \cos(\kappa R_{ij}) f_c(R_{ij}) \\
G_i^{(4)} &= 2^{1-\zeta} \sum_{j,k\neq i} (1+\lambda\cos\theta_{ijk})^\zeta e^{-\eta(R_{ij}^2+R_{ik}^2+R_{jk}^2)} f_c(R_{ij}) f_c(R_{jk} f_c)(R_{ik}) \\
G_i^{(5)} &= 2^{1-\zeta} \sum_{j,k\neq i} (1+\lambda\cos\theta_{ijk})^\zeta e^{-\eta(R_{ij}^2+R_{ik}^2)} f_c(R_{ij}) f_c)(R_{ik}) \\
R_{ij} &= |\boldsymbol{R}_{ij}| \equiv |\boldsymbol{r}_i - \boldsymbol{r}_j| \\
\theta_{ijk} &= \arccos \frac{\boldsymbol{R}_{ij}\cdot\boldsymbol{R}_{ik}}{R_{ij}R_{ik}}
\end{aligned}
$$

由于他们都是相对坐标和相对角度的函数, 因此这种表示自然是对平移和转动不变的. 但对置换可能会有所不同.

我们使用这些参数, 应用 `kernel rigid regression` , 以及 `sklearn` 模块提供的网格搜索策略, 最终获得了 0.05(随着调参可能逐渐变好) 的成绩, 已经是能够使用的数量级了.