torchAND

AI-QM 融合计算(Fusion Computing for AI-QM)

融合计算是指将量子力学计算和网络计算结合起来的计算方法.

L.X.Zhou
lxzhouno1@gmail.com

引言

融合计算是指将量子力学计算和网络计算结合起来的计算方法

存在两种不同的计算方法。

一种，以量子力学计算为例：

哈密顿量—Schrodinger微分方程--波函数和能级

另一种，网络计算：

数据库--网络训练--分类或回归

前者取决于哈密顿量，后者取决于数据库。

问题是我们应该将这种计算融合起来，

现有的三种AI-QM融合计算方法

1. **torchANI**: 两个关键:一个由密度泛函理论;DFT计算大量分子后得到的数据库; 建立了一个AEV矢量为输入。

(源码https://aiqm.github.io/torchani/)

2. **DM21**(DeepMind 2021): 数据库用 2235 个化学反应示例训练了他们的人工智能，并提供了有关所涉及的电子和系统能量的信息。其中，1074 个代表了分数电子会对传统 DFT 分析造成问题的系统。将泛函表示为神经网络.

(源码https://github.com/deepmind/deepmind-research/tree/master/density_functional_approximation_dm21

3. **FermiNet**: 引入了一种全新的深度学习架构——费米子神经网络. 这是一种用于多电子系统的强大的波函数拟设器。在多种不同的原子和小分子上，这种费米子神经网络的准确度能超过其它变分式蒙特卡洛拟设器。这种神经网络使用的数据只有原子的位置和电荷. 该网络有大约 7万个参数，只训练一次，之后它似乎就能找到能为所预测的性质给出优良结果的近似波函数。

(源码在github上: https://github.com/deepmind/ferminet)

算力比较

1. 我们分别用nwchem, torchani和DM21计算四苯乙烯异构体C38H28的基态能, 结果都近似为E=-1464.38 Hartee. 但是运行的CPU时间相差甚大.

nwchem	DM21	torchani
11 h44m50s	10.76 s	

2. Fermi Net 的所有代码都是用 TensorFlow 实现的，每个实验都在 8 个 V100 GPU 上以并行的方式运行。使用更小的批大小，我们可以在单个 GPU 上训练，但收敛速度显著更慢。比如，**乙烯在使用 8 个 GPU 训练 2 天后收敛，而在单个 GPU 上需要几周时间。**

FermiNet用在具有10个电子以内的原子上，能量精度均在99.8%左右。对于30个电子的环二丁烷，FermiNet算出的能量达到了97%的精度.

PART 1： TORCHANI

ANI数据库:一般的人工智能首先需要一个数据库. 若分子记为Xμ，标签应为量子计算得到的能量
Yμ，所以构成的数据集为 {Xμ，Yμ,} μ=1,…，n

ani-1x and ani-1ccx库：ANI-1是由DFT计算得到的8个重原子(H,C,N,O)
构成的所有分子的能量作标签的数据库. ANI-1是GDB-11的一个子集，含有57,951个分子. 能量是单重态自旋态的中性分子计算(单点能量).

ani-2x库：含有H,C,N,O,,S,F,Cl元素的分子.

因此ANI-1不能解Schrodinger方程得到本征函数和本征值. 但可以得到势能面，力常数，振动频率. 进而可以从其极小点得到几何构型，从其势阱深度得到反应能，从其鞍点得到过渡态，红外谱，拉曼谱等.

torchANI需要的软件

(1) **anaconda3**: bash Anaconda3-5.2.0-Linux-x86_64.sh –u (for python3.6)

from https://repo.anaconda.com/archive/

(2) **pytorch_env**: conda create -n pytorch_env python=3.6

(3) **pytorch**: conda install pytorch torchvision cudatoolkit=9.0 –c pytorch (for cuda9.0)

(4) **ASE**: python3.6 –m pip install ase

(5) **torchani**: python3.6 –m pip install torchani-masterinpy

Note: (1) Run python, you should use python3.6 command, not python.

(2) install softwaer: >> python3.6 –m pip install software-name

(3) You have to run in pytorch_env:

>> **source activate pytorch_env**

构建网络

网络由 3-4 隐藏层组成, 每层有 32-128 神经元最佳. ANI 使用完全连接的神经网络. 利用了回归功能神经网络预测分子势能面.

网络架构: 具体是 768 个输入值，然后是一个 128 节点的隐藏层，然后是另一个具有 128 个节点的隐藏层，一个 64 节点的隐藏层，最后是一个输出节点， 每个单个原子总共有 124 033 个可优化参数数神经网络的潜力。

激活函数: 所有隐藏层节点使用高斯激活函数，而输出节点使用线性激活函数。

权重: 是根据范围为（-1 / d，1 / d）的正态分布随机初始化的，其中 d 是节点输入的数量。

偏差: 神经网络**偏差**参数都初始化为零。

初始学习速率 = 0.001.

循环次数 epock=6.

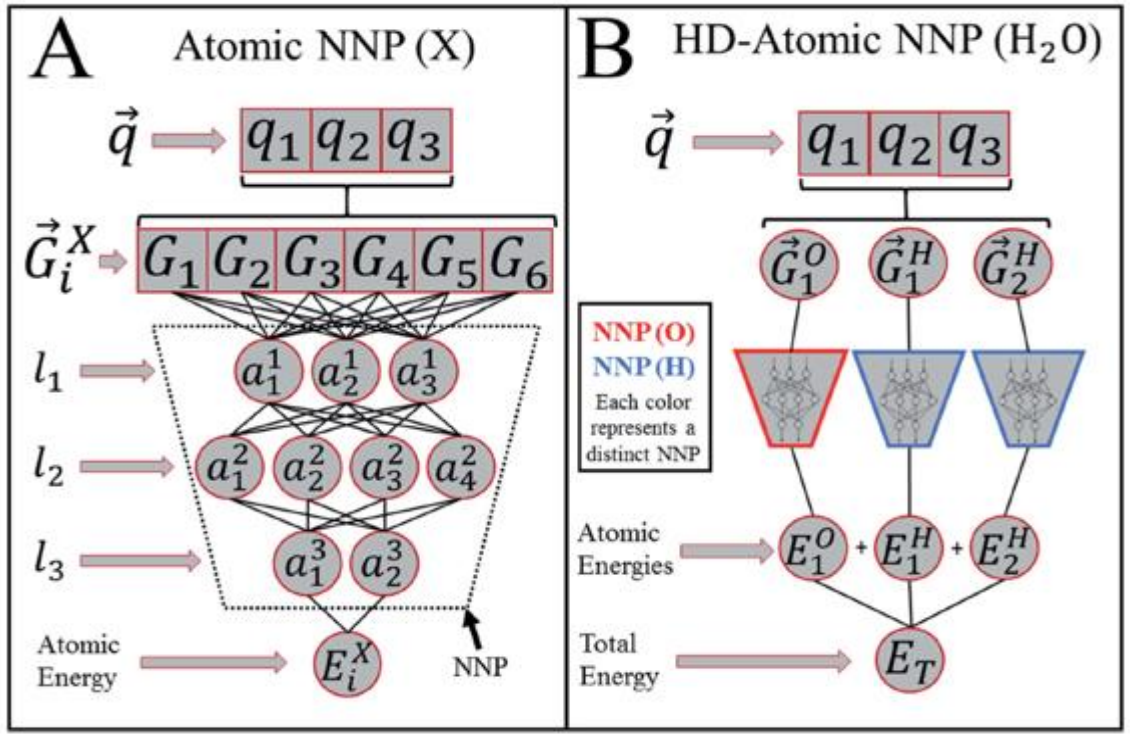
小批量: miniBatch=1024

训练集: ANI-1 数据集中 1720 万个数据点中 80％为训练集

网络输入输出

分子中每个原子x的坐标 q=(q1,q2,q3) 必须转化为该原子的环境矢量(AEV=atom evironment vector) Gi={G1,G2,…,Gm}.其中 Gm 反映一个原子周围的径向和角向的相互作用环境，其径向作用还必须设置截止函数.

大量修改的Behler和Parrinello对称函数(BPSF）及其高维神经网络电势模型（如下图)构成了ANI模型的基础。原始BPSF用于计算原子环境矢量（AEV）



截止函数

$$f_c(R_{ij}) = 0.5\cos\left(\frac{\pi R_{ij}}{R_c}\right) + 0.5 \quad for\ R_{ij} \leq R_c$$

$$0.0 \quad for\ R_{ij} > R_c$$

截止半径 Rc ~ 4.6 A(埃)

f_C（R_{ij}）是具有连续一阶导数的连续函数。

径向函数

$G_m^R = \sum_{j\neq i}^{all\ atoms} e^{-\eta(R_{ij}-R_s)^2} f_c(R_{ij})$　　η, R_s 为可调参数

参数 η 用于更改高斯分布的宽度，而 R s 的目的是移动峰的中心。

角向函数

$G_m^{Amod} = 2^{1-\zeta}\sum_{j,k\neq i}^{all\ atoms}(1+\cos(\theta_{ijk}-\theta_s)^{\zeta}\exp[-\eta(\frac{R_{ij}+R_{jk}}{2}-R_s)^2]f_c(R_{ij})f_c(R_{jk})$　　$\theta_s, \eta, \zeta, R_s$为可调参数

$\theta_s \sim 3.1\ A$

输出

由于使用的数据集ANI-1的标签是能量,所以回归输出是输入分子的能量.

例題

torchAND

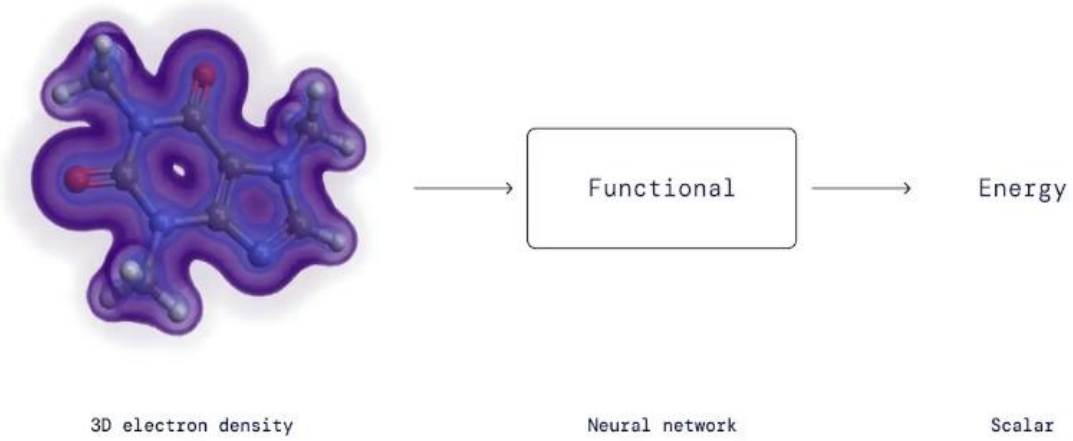
例題1.energy_force.py

例題2.vibration_analysis.py

例題3.nnp_training.py

Part II DM21 (DeepMind 2021)

1. **数据库:**用 2235 个化学反应示例训练了他们的人工智能，并提供了有关所涉及的电子和系统能量的信息。其中，1074 个代表了分数电子会对传统 DFT 分析造成问题的系统。
2. **网络:** 泛函表示为神经网络



DM21专门解决了传统泛函的两个长期存在的问题：

DFT 着眼于平均电荷密度，因此它不知道单个电子是什么。但它们无法对具有 1.5 个电子的系统进行建模，这在一个电子在多个原子之间共享的情况下很重要。

而DM21专门解决了传统泛函的两个长期存在的问题：

1. **离域误差**：在 DFT 计算中，函数通过找到使能量最小化的电子配置来确定分子的电荷密度。因此，泛函中的错误会导致计算出的电子密度出现错误。大多数现有的密度泛函近似更喜欢不切实际地分布在多个原子或分子上的电子密度，而不是正确定位在单个分子或原子周围.

2. **自旋对称性破坏**：在描述化学键的破坏时，现有的泛函倾向于不切实际地倾向于破坏称为自旋对称性的基本对称性的配置。由于对称性在我们对物理和化学的理解中起着至关重要的作用，这种人为的对称性破坏揭示了现有泛函的一个主要缺陷。

实例: 我们分别用nwchem, 和DM21计算**四苯乙烯异构体C38H28的基态能**, 结果都近似为E=-1464.38 Hartee. 但是运行的CPU时间相差甚大.

nwchem	DM21
11 h44m50s	10.76 s

PART III FERMINET

引言　人工智能深度学习要解决的就是高维函数$u_\theta(x)$的逼近.

本质上，神经网络可以看做一个函数$u_\theta(x)$。神经网络的输入层为u_θ的输入，而输出层为u_θ的输出。因此，我们可以使用u_θ来近似偏微分方程中的函数，一旦确定θ，我们就得到了偏微分方程的解。为了便于叙述，我们仍然使用猫的图片识别任务来作为例子。函数的参数x表达了神经网络的输入，即猫的图片。参数θ即为整个神经网络的每个节点的函数的参数的集合。**我们的目标就是要确定参数θ从而确定函数$u_\theta(x)$.**

为了达到此目的，我们得训练该神经网络。我们给定一些已经标识好有N个样本点的训练集，$\{(x_i, y_i)\}_{i=1}^N$，其中x_i表示了第i张图片，y_i表示x_i是否为猫的图片。我们的目标是想要$u_\theta(x_i)$，即神经网络对图片x_i的判断，与y_i的结果接近。为此，我们需要求解下面的优化问题

$$\min_\theta \frac{1}{N}\sum_{i=1}^{N} \|u_\theta(x_i) - y_i\|^2 。$$

学习的过程即为求解优化问题的过程。

费米子神经网络

1. Fermi Net 的所有代码都是用 TensorFlow 实现的. 每个实验都在 8 个 V100 GPU 上以并行的方式运行。使用更小的批大小，我们可以在单个 GPU 上训练，但收敛速度显著更慢。该网络有大约 70000 个参数，只训练一次，之后它似乎就能找到能为所预测的性质给出优良结果的近似波函数。

2. 费米子神经网络，这是一种用于多电子系统的强大的波函数拟设器。在多种不同的原子和小分子上，这种费米子神经网络的准确度能超过其它变分式蒙特卡洛拟设器。这种神经网络使用的数据只有原子的位置和电荷，就能预测出氮分子和氢链的解离曲线（dissociation curve）——这两者是很有难度的强关联系统。相比于之前被广泛视为量子化学的黄金标准的耦合聚类方法，这种新方法得到的准确度显著更高。这表明，深度神经网络的表现可以超过已有的从头开始式（ab-initio）量子化学方法，为之前难以处理的分子和固体的波函数的准确直接优化提供了新的可能性。

3. 在量子力学中，电子没有精确的位置，我们只能从波函数预测电子在空间中出现的概率，也就是电子云。电子在分子中不仅受到原子核的吸引力、其他电子的斥力，还遵循着量子力学中的费米-狄拉克统计：如果两个电子交换状态，波函数要反号(泡利不相容原理)。

4. FermiNet的源码在github上: https://github.com/deepmind/ferminet

Good Luck and Good Bye