软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(9):4093−4109 [doi: 10.13328/j.cnki.jos.007258] [CSTR: 32375.14.jos.007258]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于层重组扩展卡尔曼滤波的神经网络力场训练*

胡思宇[1,2], 周远昌[1,2], 赵 曈[1,2], 汪林望[3], 贾伟乐[1,2], 谭光明[1,2]

[1](处理器全国重点实验室 (中国科学院 计算技术研究所), 北京 100190)

[2](中国科学院大学, 北京 100049)

[3](中国科学院 半导体研究所, 北京 100190)

通信作者: 谭光明, E-mail: tgm@ict.ac.cn

摘 要: 分子动力学模拟在材料模拟、生物制药等领域发挥着重要作用. 近年来, 科学智能 (AI-for-Science) 发展, 尤其是神经网络力场在预测能量、力等性质的问题上, 相比于传统势函数方法在准确性上有大幅提升. 针对当前的神经网络力场模型在使用一阶训练方法时出现的超参设置敏感和梯度爆炸问题, 给出层重组卡尔曼滤波器在避免超参数设置问题上的若干策略和防止梯度爆炸的理论证明. 基于层重组卡尔曼滤波器, 制定交替训练方法并分析该方法的精度收益和时间成本、提出分块阈值的性能模型并论述该模型的有效性、证明防止梯度爆炸的性质并验证该优化器关于激活函数和权重初始化的鲁棒性. 对 4 种典型的神经网络力场模型在 11 个有代表性的数据集进行测试, 实验表明, 当层重组卡尔曼滤波器和一阶优化器达到相当的预测精度时, 层重组卡尔曼滤波器相比于一阶方法快 8–10 倍. 可以相信, 所提出的层重组卡尔曼滤波训练方法能给其他的科学智能应用带来启发.

关键词: 科学智能; 神经网络; 力场训练; 层重组扩展卡尔曼滤波优化器; 分子动力学模拟

中图法分类号: TP18

中文引用格式: 胡思宇, 周远昌, 赵曈, 汪林望, 贾伟乐, 谭光明. 基于层重组扩展卡尔曼滤波的神经网络力场训练. 软件学报, 2025, 36(9): 4093–4109. http://www.jos.org.cn/1000-9825/7258.htm

英文引用格式: Hu SY, Zhou YC, Zhao T, Wang LW, Jia WL, Tan GM. Neural Network Force Field Training Based on Reorganized Layer-wised Extended Kalman Filtering. Ruan Jian Xue Bao/Journal of Software, 2025, 36(9): 4093–4109 (in Chinese). http://www.jos.org.cn/1000-9825/7258.htm

Neural Network Force Field Training Based on Reorganized Layer-wised Extended Kalman Filtering

HU Si-Yu[1,2], ZHOU Yuan-Chang[1,2], ZHAO Tong[1,2], WANG Lin-Wang[3], JIA Wei-Le[1,2], TAN Guang-Ming[1,2]

[1](State Key Lab of Processors (Institute of Computing Technology, Chinese Academy of Sciences), Beijing 100190, China)

[2](University of Chinese Academy of Sciences, Beijing 100049, China)

[3](Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Molecular dynamics simulation plays an important role in material simulation, biopharmaceuticals, and other areas. In recent years, the development of AI-for-Science has greatly improved the accuracy of neural network force fields in predicting energy, force, and other properties, compared to traditional methods using potential functions. Neural network force field models may challenges such as hyperparameter settings and gradient explosion when trained by the first-order method. Based on an optimizer named reorganized layer extended Kalman filtering, this study provides several strategies to avoid hyperparameters and offers theoretical evidence for preventing gradient explosion. This study also proposes an alternate training method and analyzes its accuracy gains and time costs. A performance model of block thresholding is proposed, and its effectiveness is explored. Additionally, the property of preventing gradient explosion is

* 基金项目: 国家自然科学基金 (T2125013, 92270206, 62372435, 62032023, 61972377, 61972380, T2293700, T2293702); 中国科学院战略性先导科技专项 (XDB0500102); 中国科学院稳定支持青年科学家团队 (YSBR-005)
收稿时间: 2023-10-21; 修改时间: 2024-04-30; 采用时间: 2024-07-25; jos 在线出版时间: 2025-01-08
CNKI 网络首发时间: 2025-01-15

proven, and the optimizer's robustness with respect to activation functions and weight initialization is validated. Four typical neural network force field models are tested on 11 representative datasets. Results show that when the proposed optimizer and the first-order optimizer achieve comparable prediction accuracy, the proposed optimizer is 8 to 10 times faster than the first-order optimizer. It is believed that the proposed training method can inspire other AI-for-Science applications.

Key words: AI-for-Science; neural network; force field training; reorganized layer-wised extended Kalman filtering optimizer; molecular dynamics simulation

理论与分析、实验与观察、计算与模拟是现代科学研究的 3 种手段[1]. 其中微观尺度最常用的手段为分子动力学, 而模拟的精度取决于力场 (force field) 的精度. 力场的发展主要可以分为 3 类: 第一性原理分子动力学[2,3] (Ab initio MD, AIMD)、经典力场和神经网络力场.

AIMD 从量子力学基本方程出发, 其核心是通过求解薛定谔方程来描述电子、原子核, 以及电子和原子核间的相互作用. 在实际应用中, 通常会对薛定谔方程进行必要的简化和近似, 例如考虑多重电子激发态的 CISDTQ 法 (全称为 configuration interaction with single, double, triple and quadruple excitations, 计算复杂度为 $O(N^{10})$)、CISD 法 (全称为 configuration Interaction with single and double excitations, 计算复杂度为 $O(N^6)$)、在量子化学中常用的单电子近似法 HF 方法 (全称为 Hartree-Fork, 计算复杂度为 $O(N^3)-O(N^4)$)、基于密度泛函理论[4,5]的 KS 法 (全称为 Kohn-Sham, 计算复杂度为 $O(N^3)$) 等, 其中 N 特指物理体系中包含的电子数. 我们称使用上述方法得到的能量、波函数等属性的精度为第一性原理精度. 当前, AIMD 在纳米材料[6]、相变[7]等的实际应用和理论研究中发挥了重要作用. 然而第一性原理的计算方法受限于时间尺度和空间尺度, 目前只能模拟皮秒 (ps) 尺度的运动和最多数千原子量级的规模.

经典力场 (也叫传统方法) 的核心是构建具有解析形式的多体势函数, 它可以解决长时间尺度和大空间尺度的模拟问题. 常见的势函数法有: LJ (Lennard-Jones) 方法[8]、EAM (embedded atom method) 法[9]、ReaxFF (reactive force field) 方法[10]等. 势函数法采用基于经验参数的势函数以减少计算量, 因此计算速度快, 能模拟更大的分子或原子体系和更长时间步的运动. 例如相变反应[11]、球分子[12]、金属中的杂质、表面和其他缺陷[13]、合金[14]、碳氢化物的氢化反应[15] 等研究. 在采用势函数法时, 需要人工手动设定键、角、二面角等相关参数、势函数方法的精度高度依赖于势函数的参数选取, 目前势函数法主要问题为精度问题.

神经网络力场随着深度学习技术近年的发展逐渐成为研究热点. 其核心是利用高精度的第一性原理计算数据和深度学习网络拟合高维的势函数. 神经网络方法的分子动力学模拟相比于基于薛定谔方程的多体或单体的求解方法, 极大程度地降低了计算复杂度, 其计算复杂度降为 $O(N)$, 这里的体系规模 N 特指原子数. 同时, 当我们采用第一性原理精度的数据对神经网络的力场模型进行训练时, 能够得到接近第一性原理精度的预测结果, 弥补了传统方法精度不够的问题. 鉴于神经网络方法的分子动力学模拟具有比 AIMD 更低的计算复杂度和比经典力场更高的预测精度, 使得超大体系的有效模拟成为可能, 例如在超算系统上能模拟数十亿原子的运动[16,17]等工作的出现. 神经网络的力场模型的设计在结合不变性、等变性、对称性的情况下, 在多种体系下都取得了可观的预测精度. 例如 SNAP[18]、SIMPLE-NN[19]、HDNNP[20−22]、BIM-NN[23]、CabanaMD-NNP[24]、SPONGE[25]、DeePMD[26]、DTNN[27]、Enn-s2s[28]、PaiNN[29]、NewtonNet[30]、NequIP[31]、DimeNet++[32,33]、SpookyNet[34]等模型应用在铜 (Cu)、硅 (Si)、钽 (Tantalum)、晶体 (bulk crystals)、水 (H_2O)、二氧化硅 (SiO_2)、有机小分子、碳氢化物 ($C_{10}H_2$、$C_{10}H_3$ 等) 的预测上.

然而在神经网络力场模型的训练过程中, 需要经验式地预先确定好超参才能保证收敛效果. 训练过程中的超参数包括: 初始学习率的设置、学习率的衰减策略及参数、损失函数的构造组成等. 例如 DeePMD-kit 默认采用 Adam 优化器, 初始学习率为 0.001, 学习率在训练过程中按 0.95 指数衰减且每 5000 步衰减一次, 损失函数前的系数动态调整. 不当的超参数设置会对收敛速度和精度产生影响, 目前并没有一套公认的超参数的选取标准在不同数据集和不同神经网络力场模型下均能适用. 最近, 一种基于卡尔曼滤波理论的拟牛顿法作为神经网络力场训练的优化器在 DeePMD-kit 上已经取得了较好的效果[35], 文中实验表明能加速收敛过程, 但没有提及该优化器在其他神经网络力场训练模型上的训练方法 (包括不同模型上卡尔曼滤波算法的参数设置) 及迁移效果.

本文观察到在神经网络的力场模型的训练过程中使用一阶优化器时, 超参数选取对模型收敛精度的影响. 本文的主要贡献为: (1) 采用交替训练方法替代联合训练, 以解决损失函数中多项目标的权重配比问题, 并分析交替训练方法相比于联合训练方法带来的精度收益和训练时间开销; (2) 对层重组的卡尔曼滤波算法的分块阈值建立性能模型, 避免在通用的神经网络力场模型中手动设置分块阈值的问题, 并验证该模型的有效性; (3) 理论证明层重组的卡尔曼滤波优化器能避免梯度爆炸, 并验证训练过程中的参数, 如权重初始化、激活函数在神经网络力场模型训练时的鲁棒性. 总的来说, 在基于拟牛顿法的层重组卡尔曼滤波优化器的神经网络力场模型训练中, (1) 和 (2) 为本文提出的避免超参数微调的策略, (3) 给出该优化器在网络训练过程中具备的良好性质. 最后在 4 种典型的神经网络力场网络下对 11 个真实数据集进行测试, 层重组卡尔曼滤波优化器达到和一阶优化器相当的精度时 (8 个典型体系), 在 4 种不同的模型中平均加速比为 8–10 倍.

本文第 1 节介绍神经网络力场和优化器的相关方法和研究现状. 第 2 节先展示我们对一阶优化器训练力场模型的观察与挑战, 为了应对这些挑战, 进而给出层重组卡尔曼滤波优化器的避免调参策略和防止梯度爆炸的理论证明. 第 3 节给出实验设置和数值结果, 并进行结果分析. 最后总结全文.

1 相关工作和背景

首先介绍神经网络力场训练的整体流程和有代表性的神经网络力场模型. 然后介绍神经网络的训练算法, 包括一阶方法和二阶方法. 接着引入卡尔曼滤波算法在一般的线性测量系统和神经网络的非线性测量系统下的预测-修正过程; 最后描述改进的卡尔曼滤波算法 (层重组卡尔曼滤波优化器) 应用于神经网络权重更新的完整迭代过程.

神经网络的力场训练一般分为以下步骤.

(1) 计算邻居列表, 如图 1(a) 所示. 为了方便理解, 先以一个中心原子 i 为例进行介绍, 预先选取一个截断半径, 在该截断半径之内的原子 ($j1, j2, \ldots, jM$) 认为是原子 i 的邻域原子, 计算各个邻域原子与中心原子在 xyz 这 3 个方向上的距离差、笛卡尔距离, 进而得到邻居列表.

(2) 计算特征, 特征的获取可以分为两种: 根据经验公式显式计算出、在训练过程中生成, 分别对应显式的特征设计和隐式的特征设计方法, 如图 1(b) 所示. 显式的特征设计指通过含经验参数的解析函数得到各个体系的特征, 代表工作有文献 [18,20−22,36−40]. 隐式的特征设计指直接以邻居列表作为输入, 通过在神经网络中加入不变或等变性的约束得到满足物理性质的原子特征, 代表工作有文献 [26−34].

(3) 拟合网络, 如图 1(c) 所示, 得到每个原子的特征表示后, 再通过拟合网络即可得到每个原子的能量. 根据拟合网络模型结构的不同, 可将神经网络力场模型分为基于全连接、基于图神经网络等的力场模型.

(4) 计算总能量和每个原子受力, 如图 1(d) 所示. 总的原子能量是由单个原子能量累加得到的, 为了保证能量守恒, 每个原子受力是由总能量对单个原子的位置求导数计算得到.

(5) 权重的更新, 神经网络训练的过程是优化神经元参数, 也即权重的取值的过程. 权重不断迭代以损失函数最小化为目标. 如图 1(e) 所示, 已知第 t 次迭代的权重, 根据当前样本, 经过网络的前向计算得到损失值. 再根据损失值进行反向传播, 得到权重更新的增量. 结合第 t 次迭代的权重上, 更新得到第 $t+1$ 次迭代的权重值.

根据获取特征方式的不同, 神经网络力场可分为隐式特征计算的神经网络力场和显式特征计算的神经网络力场. 隐式特征计算的神经网络力场模型的原子特征会随着神经网络的训练, 实时变化. 隐式特征计算的神经网络力场模型包括深度势能模型和 NequIP 模型等. 显式特征计算的神经网络力场模型通过预先选取解析函数、设置解析函数的参数, 当样本位置信息给定时, 得到确定的特征, 不随神经网络的训练而变更. 整体来看, 隐式特征计算的神经网络力场模型的网络结构更复杂, 参数量更大, 训练时间更长, 适用的场景更多. 显式计算特征的神经网络力场模型的特点为: 模型的可解释性强, 拟合网络相对简单, 解析函数在特定应用场景下具有较强的适用性.

隐式特征计算的神经网络力场模型中, 深度势能模型和 NequIP 是近年来最具代表性的两个神经网络力场模型, 分别基于全连接和图神经网络的骨架. 深度势能模型首先采用一个嵌入网络 (embedding network) 对原子间笛卡尔距离进行编码, 输出记为 G_i^{DP}, 再通过一个对称性保持操作 $D_i^{DP}=(G_i^{DP})^{\mathrm{T}}R_iR_i^{\mathrm{T}}G_i^{DP}$, 计算得到深度势能模型的特征 D_i^{DP}. 通过此种对称性保持操作得到的特征具有平移、旋转和置换不变性. 深度势能网络在水等数据集上取得

了接近 DFT 计算的精度. 另一种考虑等变性的基于图神经网络的力场在 MD17 等数据集上表现良好, 以 NequIP 神经网络力场模型为例, 原子间距离利用球谐函数进行编码, 再在多项式函数的基础上, 进行多层的非线性变换, 得到的结果与球鞋函数表示进行乘法运算, 以保证输出的等变性.

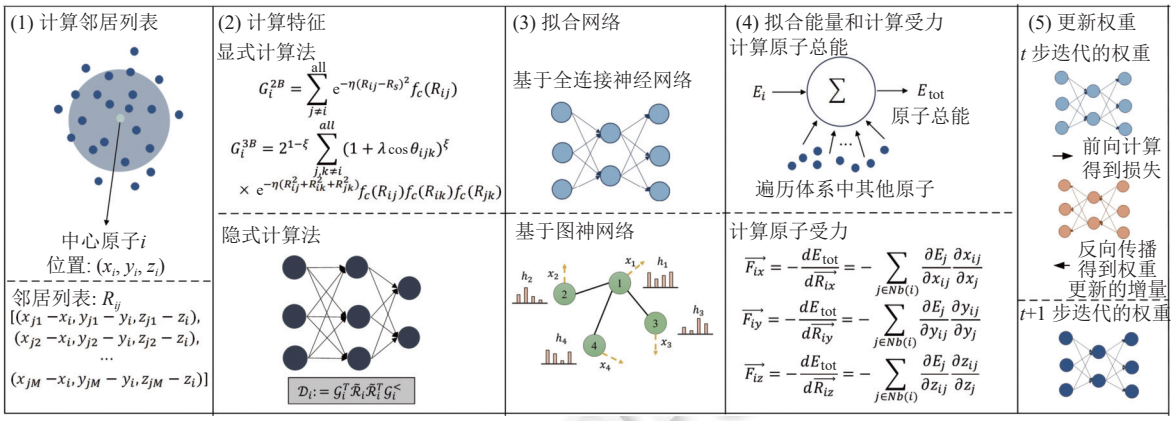


图 1 基于神经网络方法的原子力场训练整体流程

显式特征计算的神经网络力场的重点在于基组, 以下是有代表性的用于神经网络力场的基组: 两体三体余弦基组 (2Body-3Body Cosine basis, 2B3B-Cosine)[36]、两体三体高斯基组 (2Body-3Body Gaussian Basis, 2B3B-Gaussian)[22]、多体张量势函数 (multiple tensor potential, MTP)[37]、频谱领域分析势函数 (spectral neighbor analysis potential, SNAP)[18]. 这里以 2B3B-Gaussian 解析函数中的两体高斯基组 (2B Gaussian basis) 为例进行详细介绍, 其他基组可以参考文献 [16,36,37]. 中心原子 i 在两体高斯基组的表示下, 其特征 G_i^{2B} 的计算如公式 (1) 所示. 给定一组高斯函数的超参数 $\{\gamma, R_s\}$, 其中 γ 用于控制高斯基组分布的离散程度, R_s 为分布的均值. 对截断半径 R_c 里中心原子 i 的所有邻居 j, 均可得到邻居 j 的高斯表示 $e^{-\gamma(R_{ij}-R_s)^2}$, 其中 R_{ij} 为邻居原子 j 与中心原子 i 的笛卡尔距离. 为了防止在分子运动过程中在截断半径 R_c 周围有原子的跳进跳出影响中心原子 i 的特征表示, 可在每个邻居 j 的高斯表示下作用一个光滑函数, 即 $f_c(R_{ij})$. 该光滑函数的定义如公式 (2) 所示, 当 R_{ij} 为 0 时, $f_c(R_{ij})$ 为 1; 当 R_{ij} 趋于截断半径 R_c 时, $f_c(R_{ij})$ 按余弦函数衰减为 0, 也即随着邻居原子与中心原子的笛卡尔距离逐渐变远, 该邻居原子对中心原子的影响逐渐变小; 当 R_{ij} 大于截断半径时, $f_c(R_{ij})$ 为 0, 也即不考虑超过截断半径内的邻居原子对中心原子 i 的影响. 同时为了满足置换不变性, 即邻域原子的先后顺序的选取不改变的值 G_i^{2B}, 该基组采用对所有邻居 j 的经过光滑处理的高斯表示进行求和操作.

$$G_i^{2B}=\sum_{j\neq i}^{all} e^{-\gamma(R_{ij}-R_s)^2} f_c(R_{ij}) \tag{1}$$

$$f_c(R_{ij})=\begin{cases}0.5\times\left[\cos\left(\dfrac{\pi R_{ij}}{R_c}\right)+1\right], & \text{for } R_{ij}\leqslant R_c\\ 0, & \text{for } R_{ij}>R_c\end{cases} \tag{2}$$

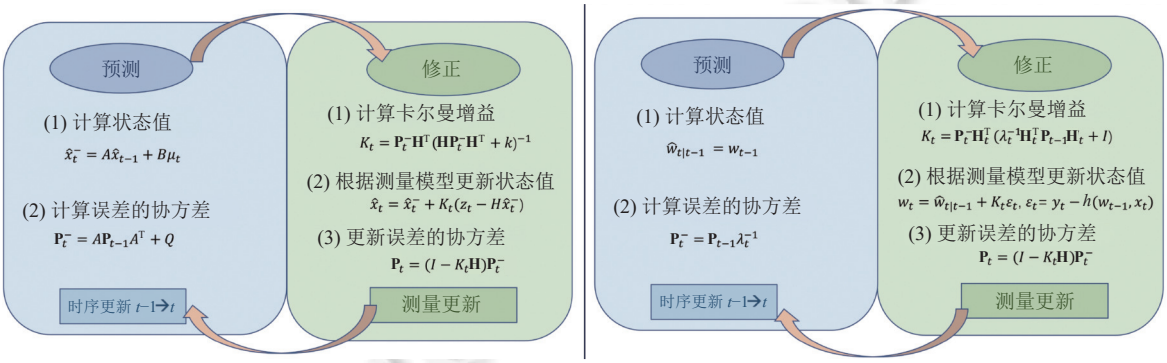
根据更新权重的算法的不同, 基于导数的优化算法可以分为一阶优化方法和二阶优化方法. 一阶优化方法通常也指梯度下降法, 可写成公式 (3) 的形式, 权重的更新沿着负梯度方向以 η 的步长移动, 典型的代表是 SGD 优化器. 鉴于更新时直接用梯度的信息可能会包含很多噪声, 进而衍生出改进版本的带有动量的一阶优化器. 基于梯度下降理论的一阶优化器的变体如 Adam[41], AdaGrad[42,43], RMSProp[44], AdaDelta[45]等. 二阶优化方法比一阶有着更强的理论基础, 二阶优化方法可以简写成公式 (4) 的形式. 通常二阶方法具有比一阶方法更快的收敛速度, 这得益于蕴含在海森矩阵中 (通常用 $\mathbf{H}$ 表示) 的信息, 其中 $\mathbf{H}^{-1}$ 也称为预条件子. 在公式 (4) 的基础上, 产生了很多二阶方法的变体, 如 K-FAC[46], L-BFGS[47,48], Shampoo[49], AdaHessian[50]等.

$$w_{t+1}=w_t-\eta\nabla L(f(x_i;w_t),y_i) \tag{3}$$

$$w_{t+1} = w_t - \mathbf{H}^{-1}\nabla L(f(x_i; w_t), y_i) \tag{4}$$

卡尔曼滤波是鲁道夫卡尔曼在 1960 年提出的一个数据处理的算法[51]. 假定过程模型和测量模型如公式 (5) 所示, 且过程模型中的噪声和测量模型中的噪声 v 分别服从 $p(w) \sim N(0, Q), p(v) \sim N(0, R)$ 的正态分布. 在这个线性系统的前提下, 卡尔曼滤波可以表示成如图 2(a) 所示的预测-修正的迭代算法, 广泛应用于自动驾驶和航海等领域. 对于非线性系统, 只需将非线性的测量模型和过程模型进行泰勒展开, 即可按线性系统处理, 推广得到扩展卡尔曼滤波. 神经网络是一个典型的非线性的系统, 当状态值迁移到神经网络的权重时, 也即全局卡尔曼滤波 (global extended Kalman filter, GEKF) 用在神经网络权重的更新上时[52], 预测和修正的更新模式表示在图 2(b) 中.

$$\begin{cases} x_t = Ax_{t-1} + Bu_t + w_{t-1} \\ z_t = \mathbf{H}x_t + v_t \end{cases} \tag{5}$$



(a) 在线性系统情况下的一般表示　　(b) 在非线性系统的表示

图 2　卡尔曼滤波算法的更新流程

层重组卡尔曼滤波优化器[35]是全局卡尔曼滤波优化器[52]的改进算法. 通过一种特殊的解耦合方法减小存储和计算开销. 为了方便理解层充足卡尔曼滤波优化算法, 首先简略介绍全局卡尔曼滤波. 全局卡尔曼滤波优化器在更新权重时 (权重的总数记为 N), 更新过程中的参与计算的梯度 $\mathbf{H}$ 和权重 w 均为 $(N\times 1)$ 的列向量, 权重的误差协方差 $\mathbf{P}$ 为 $(N\times N)$ 的矩阵. 层重组的卡尔曼滤波优化器对计算进行了解耦合, 把参与计算梯度 $\mathbf{H}$ 和权重 w 分别切分成 L 个列向量 $\{N_1\times 1, N_2\times 1, \ldots, N_L\times 1\}$, 把 $\mathbf{P}$ 矩阵近似成多个对角的子矩阵 $\mathbf{P} = \{N_1\times N_1, N_2\times N_2, \ldots, N_L\times N_L\}$, 其中 $N = \sum_i N_i$. 这种切分结果是根据神经网络的层参数量决定的, 当给定一个块大小阈值 (blocksize, N_b), 当神经网络层的参数量小于该阈值时, 聚合多个小的神经网络的层; 当神经网络的参数量大于该阈值时, 将该层按照 blocksize 的值进行拆分. 算法 1 是层重组卡尔曼滤波算法的实现细节, 层重组卡尔曼滤波优化器用于神经网络权重更新时的流程可分为 3 部分: (1) 计算误差和梯度, 体现在算法 1 的第 1, 2 行中; (2) 标准的卡尔曼更新计算流程作用在分层之后的更新量上, 参考算法 1 的第 3 行和第 5–7 行, 其中第 5–7 行中的层重组操作用 *split* 函数表示. 算法 1 中第 9 行是记忆因子的更新; (3) 把更新后的切分后的权重重新拼接起来并重组成神经网络权重原本的维度大小, 用算法 1 的第 10, 11 行中 *collect* 表示.

算法 1. 层重组卡尔曼滤波优化器.

输入: $\widehat{Y}$, $Y^{\mathrm{DFT}}, w^{\mathrm{in}}, \mathbf{P}^{\mathrm{in}}, \lambda^{\mathrm{in}}$　　▷ 输入

1. $\widehat{Y} = \dfrac{\sum_i \widehat{Y}_i}{length(Y^{\mathrm{DFT}})}$, $errY = \dfrac{\sum_i \left|Y_i^{\mathrm{DFT}} - \widehat{Y}_i\right|}{length(Y^{\mathrm{DFT}})}$　　▷ 计算绝对值误差

2. $\mathbf{H} = \nabla_w \widehat{Y}|_{w=w^{\mathrm{in}}}$　　▷ 计算梯度

3. $A = 1/(\lambda^{\mathrm{in}} + \mathbf{H}^{\mathrm{T}}\mathbf{P}^{\mathrm{in}}\mathbf{H}))$　　▷ 标准计算流程, 用于第 7 行状态值的更新

4. **for** $l = 1, 2, \ldots, L$ **do**　　▷ 层重组: 把整体的更新分为独立的 L 块分别更新

5. 　$K = split(\mathbf{P}^{\mathrm{in}}, l) \times split(\mathbf{H}, l)$　　▷ 按层分别更新卡尔曼增益 K

6. $\mathbf{P}_l^{\text{out}} = (1/\lambda^{\text{in}}) \times (split(\mathbf{P}^{\text{in}}, l) - AKK^{\text{T}})$ ▷ 按层分别更新权重的误差协方差矩阵 P
7. $w_l^{\text{out}} = split(w^{\text{in}}, l) + A \times K \times errY$ ▷ 按层分别状态值, 也即权重 w
8. **end for**
9. $\lambda^{\text{out}} = \lambda^{\text{in}} v + 1 - v$ ▷ 更新记忆因子
10. $w^{\text{out}} = collect(w_l^{\text{out}} \mid l = 1, \ldots, L)$ ▷ 将切分的权重进行聚合
11. $\mathbf{P}^{\text{out}} = collect(\mathbf{P}_l^{\text{out}} \mid l = 1, \ldots, L)$ ▷ 将切分的子 P 矩阵进行聚合

输出: w^{out}, $\mathbf{P}^{\text{out}}$, λ^{out} ▷ 输出

总的来说, 隐式特征计算的神经网络力场模型, 在采用一阶优化方法时, 训练时间长, 通常为数天不等. Hu 等人[35]提出拟牛顿优化器算法用于加速典型的隐式特征计算的神经网络力场模型 (例如深度势能模型) 的训练. 然而显式特征计算的神经网络力场模型在训练过程中仍存在多方面的挑战, 且当拟牛顿法用于显式特征计算的神经网络力场模型时, 具体的训练方法并未给出、性能表现也并未评估. 本文提出了一种通用的避免调参的拟牛顿算法的训练方法, 并对拟牛顿在多个显式特征计算的神经网络力场模型进行测试.

2 层重组卡尔曼滤波训练方法

在本节中, 我们首先总结一阶方法在神经网络力场训练过程中的常见问题, 基于 3 个观察引出本文的主要贡献. 第 2.1 节阐述交替训练方法并对其精度收益和时间代价进行详细的分析. 第 2.2 节介绍建模层重组卡尔曼滤波优化器的分层阈值的超参数并进行验证. 第 2.3 节对层重组卡尔曼滤波方法的防止梯度爆炸属性进行理论证明并分析该方法在神经网络力场训练过程中的鲁棒性.

● 观察 1: 联合训练时损失函数的权重因子的选取影响收敛精度.

力场预测的模型需同时预测能量和力 (原子受力不能直接通过网络的前向过程拟合, 物理上为了保证能量守恒, 原子受力的计算是通过原子总能对原子位置求导得到的). 在计算损失时, 现有的方法会联合考虑能量和受力的整体损失, 如公式 (6) 所示.

$$Loss = \alpha \times MSE\left(E_i, \widehat{E_i}\right) + \beta \times MSE\left(F_i, \widehat{F_i}\right) \tag{6}$$

其中, $MSE(E_i, \widehat{E_i})$ 和 $MSE(F_i, \widehat{F_i})$ 分别表示当前批处理样本 (batch size) 原子能量和受力的均方误差, α 和 β 分别表示能量和力场的均方误差在总的损失中的权重系数. 我们发现在不同的方法中, α 和 β 的取值各自不同, 暂时没有一种统一的 α 和 β 的值的选取, 能保证在不同模型的训练中有很好的收敛效果. 在两体和三体的余弦的基组 (2B3B-Cosine) 的工作中, $\alpha = 1$, β 的取值由用户指定. 在 DeePMD 的力场预测模型中, α 和 β 的取值随着迭代的进行实时变化, 具体的, $\alpha_t = lim_e + (start_e - lim_e) \times lr_t / lr_0$, $\beta_t = lim_f + (start_f - lim_f) \times lr_t / lr_0$, 其中 $start_e$, lim_e, $start_f$, lim_f 分别表示在计算能量和力场的权重时的起始值, 这些值是训练前需要预先给定的超参. 我们观察到, 这些参数的确定在模型收敛的过程中起着重要作用, 不当的取值会使得最终收敛的精度变差. 例如选取固定的 α和β, 当 $\alpha = 1$ 时, 分别选取 $\beta = 1000, 1, 0.001$ 时, 收敛误差在表 1 中配置①–③中. 当我们选取随迭代次数变动的α和β 时, 例如 DeePMD 中默认的 $start_e$, lim_e, $start_f$, lim_f 的值, 误差记录在配置④中. 当把能量和力的起始值交换时, 收敛精度如配置⑤所示. 表 1 最后两列分别表示能量和原子受力的误差, "/"前后分别表示训练集和测试集的结果.

表 1 损失函数的不同权重因子对收敛性的影响

2B3B-Cosine	α	β	能量RMSE (eV)	力RMSE (eV/Angstrom)
配置① (固定值)	1	1 000	0.059 9/0.071 4	0.062 5/0.074 2
配置② (固定值)	1	1	0.053 0/0.064 9	0.086 6/0.106 2
配置③ (固定值)	1	0.001	0.060 0/0.075 0	0.117/0.145
配置④ (变动值)	$start_e, lim_e$: 0.02/1.0	$start_e, lim_e$: 1 000/1.0	0.054 3/**0.060 3**	0.061 4/**0.073 3**
配置⑤ (变动值)	$start_e, lim_e$: 1 000/1.0	$start_e, lim_e$: 0.02/1.0	0.061 1/0.078 1	0.079 8/0.098 8

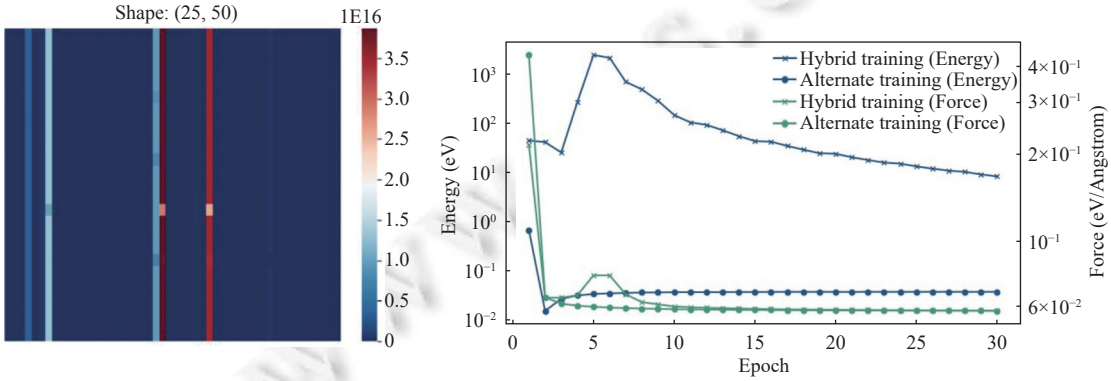
根据表 1 可知, 模型收敛精度随着 α和β 取值的不同会发生改变, 其中配置④随着迭代次数动态变动的方案在典型的铜数据集上, 基于 Adam 优化器的训练下, 具有最低的误差. 其他权重系数配置, 在测试集的能量预测效果上, 比配置④差 10.26%–29.52% 不等, 平均差 20.64%. 在测试集的原子受力的预测上, 比配置④差 1.23% 到 97.81% 不等, 平均差 44.68%. 不同权重系数配置的 RMSE 几乎在同一量级, 但由于神经网络力场模型对能量和原子受力的精度要求极高 (神经网络力场模型的训练数据为高精度的 DFT 数据), RMSE 在绝对值上相差微小, 但百分比差异显著. 总的来说, 这种联合训练存在两个问题: (1) 对于配置①–⑤中列举的有限的 α和β 的取值, 高度依赖经验, 且无法保证所选的权重因子已经是最优的; (2) 当更多的预测目标加入时, 比如维里系数、磁矩等, 又会涉及新的预测目标的平方误差的权重取值问题. 三目标函数的权重损失系数取值包含 3 个自由度, 四目标函数的权重损失系数取值为 3 个自由度, 随着预测目标的增加, 损失因子 (pre-factor) 的组合数更多.

● 观察 2: 一阶优化器往往结合学习率一起用于神经网络的权重更新, 学习率的初始值及衰减策略影响最终收敛结果.

随着神经网络的发展, 训练方法及优化器也在快速发展. 在随机梯度下降的一阶优化器基础上, 近年来产生很多变体, 例如引入动量的方法. 在实际的训练任务中, 优化器的时候通常还会辅助以初始的学习率以及迭代过程中学习率的特定改变策略来达到预期的训练效果. 神经网络力场模型的训练过程中依然存在很多的关于学习率的选取的问题, 暂时没有达成一致的共识应该如何确定学习率的选取和改变. 例如在 Huang 等人[36]的工作中, 学习率的初始值选取为 0.000 1. 而在 DeePMD 模型中, 默认的初始学习率为 0.001, 且每 5 000 个迭代步按 0.95 指数衰减. 我们发现: (1) 采用衰减的学习率比固定学习率, 得到的收敛结果精度更高; 例如铜体系下, 固定初始学习率分别为 0.1、0.001、0.000 1 时训练集收敛到 1 000 个 epoch 时的能量和原子受力的均方误差分别为 (12.521, 0.760)、(0.194, 0.061 3)、(0.166, 0.063 3). (2) 不同初始学习率对精度的影响大. 以深度势能模型为例, 当训练 batchsize 为 1 时, 初始学习率为 0.001 时, 学习率按 0.95 指数衰减情况下, CuO 体系的能量 RMSE 可收敛到 0.063 8 eV. 当训练 batchsize 增加到 32 时, 初始学习率缩放根号 32 倍时, CuO 体系的 RMSE 无法降至 0.063 8 eV. 这表明初始学习率设置不当易引发训练的不稳定现象.

● 观察 3: 一阶优化器在训练过程中会由于优化器选择不当导致梯度爆炸.

梯度消失和爆炸是神经网络训练过程中经常遇到的现象, 而现有的解决梯度消失或爆炸问题一般从网络、损失函数、激活函数等方面找原因. 一种常用的规避梯度爆炸现象的手段是在训练过程中对每次迭代更新的权重做一个截断, 限制更新幅度. 这种方法会引入额外的幅度超参数并一定程度上可能限制神经网络的快速收敛. 在我们的调研中, 发现一阶优化器本身无法防止梯度爆炸的问题. 在神经网络力场模型中, 优化器的选取显得更为敏感, 不当的优化器会引发梯度爆炸的现象. 例如以 DeePMD 为例, 我们发现在 Adam 优化器的训练下能有效收敛的铜体系, 在使用 SGD 优化器进行训练后, 会在 120 个迭代步时出现梯度爆炸的现象. 我们抽取其中一层的网络的 120 次迭代的权重的绝对值进行可视化, 见图 3(a), 其中红色表示值越来越大, 蓝色表示值越来越小.



(a) 铜体系在 SGD 优化器下的权重值 (b) 铜体系在联合训练和交替训练下能量和力的收敛情况

图 3 不同优化器配置对模型训练的影响

2.1 层重组卡尔曼滤波的交替训练方法

观察 1 中发现的损失函数的带权系数选取问题, 一种很自然的想法是交替更新, 把联合训练拆分成交替训练, 即先按照能量的误差进行权重更新, 再按照原子受力的误差进行一次权重的更新. 交替训练可避免联合训练中敏感的损失函数系数的选取问题. 尤其是当目标函数所需拟合的任务增加时, 联合训练的损失函数中各个误差项的权重的组合呈现出指数增长的选取情况时, 交替训练可以有效规避这个问题. 与此同时, 我们发现在层重组卡尔曼滤波优化器的训练过程中, 交替训练能比联合训练的误差降得更低, 如图 3(b) 所示. 图中测试数据为铜体系, 分别在联合训练和交替训练的情况下, 前 30 个 epoch 能量和受力的均方根误差的值.

交替训练带来精度上的收益的同时, 带来了一次额外的权重更新的开销, 进而导致同一样本的训练时间变长. 公式 (7) 表示交替训练比联合训练多花的时间在一次权重更新上 T^{w}_{update}, 单个样本联合训练和交替训练的具体时间构成如公式 (8) 和公式 (9) 所示. 接下来介绍一个样本一次完整的计算时间的组成, 整体时间由 3 部分计算过程构成: (1) 前向计算的时间, 用 T_{fwd} 表示, 结合其上标 E+F, E, F 分别表示模型正向过程计算出能量和力、能量、力的时间. 且不同上标的 T_{fwd} 可由各自的模型前向计算的浮点操作次数/机器计算效率得到, 计算量由 C_{fwd} 表示, 机器计算效率记为 *FLOPS* (由机器唯一确定); (2) 层重组卡尔曼滤波的计算时间, 用 T_{kf} 表示, 结合其上标 E+F, E, F 分别表示基于能量和力、能量、力进行卡尔曼滤波算法计算的时间. 同样的, 不同上标的 T_{kf} 可由各自的任务的卡尔曼滤波更新操作的浮点操作次数/机器计算效率得到, 计算量由 C_{kf} 表示; (3) 权重更新的时间 T^{w}_{update}.

$$\Delta T_{\text{交替-联合}} = T_{\text{交替}} - T_{\text{联合}} = T^{w}_{\text{update}} \tag{7}$$

$$T_{\text{联合}} = T^{\text{E+F}}_{\text{fwd}} + T^{\text{E+F}}_{\text{kf}} + T^{w}_{\text{update}} = \frac{C^{\text{E}}_{\text{fwd}} + C^{\text{F}}_{\text{fwd}}}{FLOPS} + \frac{C^{\text{E}}_{\text{kf}} + C^{\text{F}}_{\text{kf}}}{FLOPS} + T^{w}_{\text{update}} \tag{8}$$

$$T_{\text{交替}} = T^{\text{E}}_{\text{fwd}} + T^{\text{E}}_{\text{kf}} + T^{w}_{\text{update}} + T^{\text{F}}_{\text{fwd}} + T^{\text{F}}_{\text{kf}} + T^{w}_{\text{update}} = \frac{C^{\text{E}}_{\text{fwd}}}{FLOPS} + \frac{C^{\text{E}}_{\text{kf}}}{FLOPS} + T^{w}_{\text{update}} + \frac{C^{F}_{\text{fwd}}}{FLOPS} + \frac{C^{\text{F}}_{\text{kf}}}{FLOPS} + T^{w}_{\text{update}} \tag{9}$$

总体来说, 交替训练能有效规避联合训练各部分损失的带权相加问题, 同时不可避免地引入一次额外的权重更新的时间消耗. 鉴于神经网络力场训练的特殊性, 能量的预测是前向的神经网络计算过程, 基于能量进行层重组的卡尔曼滤波优化器的权重更新涉及一阶导数的计算, 单原子受力的预测值包含一阶导数的计算 (因原子受力由原子总能量对单个原子位置求导得到), 因此基于力场进行层重组的卡尔曼滤波优化器的权重更新涉及二阶导数的计算. 高阶导数的计算开销通常更大 (因为二阶导数的计算需在一阶导数的基础上再进行求导操作). 因此公式 (9) 的主导的时间开销来自 T^{F}_{kf}, 联合训练相比于交替训练引发的一次额外的权重的更新操作是整体时间的很小一部分, 也即交替训练的额外时间开销较小. 实际上, 在铜数据集下, 基于层重组卡尔曼滤波优化器的联合训练耗费 20 s/epoch, 交替训练需要 23 s/epoch, 此数据是在 NVIDIA A100 机器上测试得到.

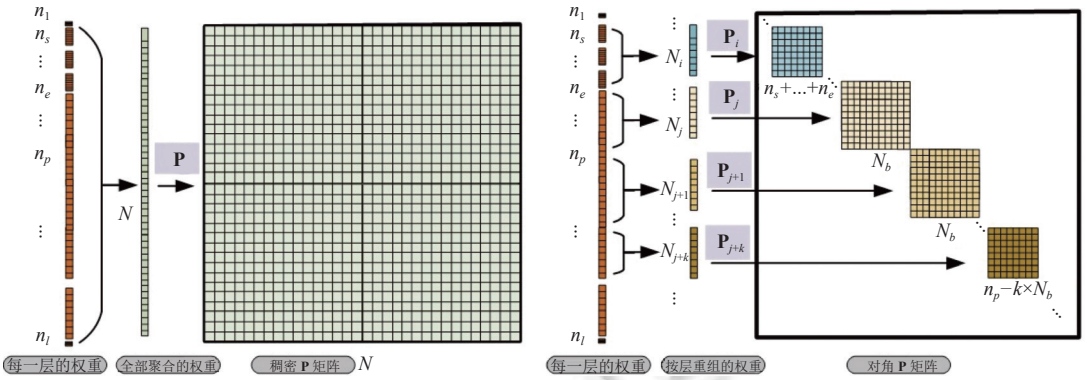
2.2 层重组卡尔曼滤波的分块阈值建模

观察 2 是一阶优化器普遍性存在的问题. 卡尔曼滤波优化器天然具有参数少的特性, 值得一提的是, 层重组卡尔曼滤波算法在卡尔曼滤波算法的基础上增加了分块大小的调整. 为了减少分块大小的经验性设置 (在 Hu 等人[35] 的工作中暂未给出分块大小的选取方案), 本文对分块大小建立了模型.

给定一个神经网络, 不妨把第 p 层的神经网络的权重数量记为 n_p. 卡尔曼滤波算法用于神经网络权重更新时, 需把神经网络每一层的权重分别张成一个列向量 (如图 4(a) 和 (b) 子图中最左侧橙色块所示). 假设神经网络共 1 层, 不失一般性, 我们把神经网络的各层权重数量, 按从前往后的顺序依次记为: $(n_1,\ldots,n_s,\ldots,n_e,\ldots,n_p,\ldots,n_l)$, 总的参数量用 N 表示. 全局的卡尔曼滤波需要把权重聚合成一个 $(N\times 1)$ 的列向量, 在利用卡尔曼滤波算法进行权重更新时, 一个 $(N\times N)$ 的稠密的误差协方差矩阵 $\mathbf{P}$ 会参与计算和迭代. 当神经网络的参数量 N 过大时, $\mathbf{P}$ 矩阵本身及算法计算过程中 $\mathbf{P}$ 矩阵的中间变量的读写均会引发过高的内存占用. 为了减少由稠密 $\mathbf{P}$ 矩阵及其中间变量带来的内存占用压力, 层重组的卡尔曼滤波算法中的 $\mathbf{P}$ 矩阵 (图 4(b) 所示) 对全局卡尔曼滤波的稠密 $\mathbf{P}$ 矩阵做了一个对角块的解耦合操作. 具体地, 按照给定的分块阈值 (blocksize, 记为 N_b), 分别进行聚合和拆分操作.

聚合操作: 对于聚合连续若干层 $(n_s,\ldots,n_e)$, 满足聚合后的权重数量小于预设的块大小的阈值, 且如果再多聚

合下一层的权重, 总的权重数量将超过该阈值. 用数学符号表示为 $\sum_{i=s}^{e} n_i \leqslant N_b$ && $\sum_{i=s}^{e+1} n_i > N_b$, 则聚合操作将施加在第 s–e 层上. 聚合后的权重作为一个整体, 在这种聚合方式下的误差协方差矩阵记为 $\mathbf{P}_i$, 其维度为 $(n_s+\ldots+n_e)^2$, 参考图 4(b).



(a) 全局卡尔曼滤波 (b) 层重组卡尔曼滤波

图 4 全局卡尔曼滤波和层重组卡尔曼滤波示意图

拆分操作: 对于神经网络的第 p 层的参数量为 n_p, 如果该层的参数量超过了预设的分块大小的阈值, 即 $n_p > N_b$, 则将该层划分为 $(k+1)$ 个小块, 其中 $k>0$. 其中前 k 块对应的误差协方差矩阵记为 $(\mathbf{P}_j, \mathbf{P}_{j+1}, \ldots, \mathbf{P}_{j+k-1})$, 每一块的维度为 $(N_b)^2$, 第 $(k+1)$ 块对应的误差协方差矩阵记为 $\mathbf{P}_{j+k}$, 其维度为 $(n_p - k\times N_b)^2$, 参考图 4(b).

层重组卡尔曼滤波是全局卡尔曼滤波算法的一个近似, 预设的分块大小的阈值 N_b 越大, 则建模的权重与权重间的相关性更多, 理论上算法的精度越高. 因此目标函数可以写成公式 (10) 的形式, 即最大化分块阈值 N_b, 等价于最大化收敛精度. 同时需满足在上述聚合和拆分操作下得到的误差协方差矩阵 $\mathbf{P}=(\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_L)$ 的内存占用需求, 如公式 (11) 所示. 该约束条件中的 $\#Bytes$ 表示数据在计算机存储中占据的字节数, 单精度的 $\#Bytes=4$, 双精度的 $\#Bytes=8$. 约束条件中的 2 倍的操作表示在 $\mathbf{P}$ 矩阵计算过程中 $\mathbf{P}$ 矩阵本身和其中间变量的占用, $GMem$ 表示显存大小.

$$\max \quad N_b \tag{10}$$

$$\text{s.t.}\ \ 2\times\left(\#Bytes\times\sum_{i=1}^{L} N_i^2\right) \leqslant GMem \tag{11}$$

为了验证该分块阈值性能模型的有效性, 我们挑选 MTP 方法在 11 个数据集上的表现进行论证说明. 如后文图 5 所示, 根据建模的方法自适应选择分块阈值相比于预先人为设定分块阈值为 1 024 时, 在能量和原子受力的精度上分别有 9、8 个体系更高 (图 5 中的曲线越低越好). 针对 H_2O 数据集, 该模型的阈值选取比 1 024 在能量和力的预测上都有很高的精度提升. Blocksize 的选取在 C、CuC 体系上有较大的差异, 该模型对 N_b 的选取在均衡考虑能量和力整体的精度比按 1 024 取 blocksize 占优. 我们认为在训练过程中能量和力的预测是一个相互权衡和制约的过程, 不同 blocksize 的选取可能在训练过程中权重调节的大小产生差异, 导致最终能量和力的预测上产生一些偏置.

2.3 层重组卡尔曼滤波防止梯度爆炸证明

我们把神经网络记为公式 (12) 所示, 其中 θ_t 表示权重, h 表示神经网络, x_t 和 y_t 分别表示输入数据和输出标签, η_t 是指测量系统中的高斯噪音, 服从均值为 0 方差为 R_t 的正态分布. 根据基础的卡尔曼滤波理论[53], 我们可以得到一系列的更新公式 (13)–公式 (16).

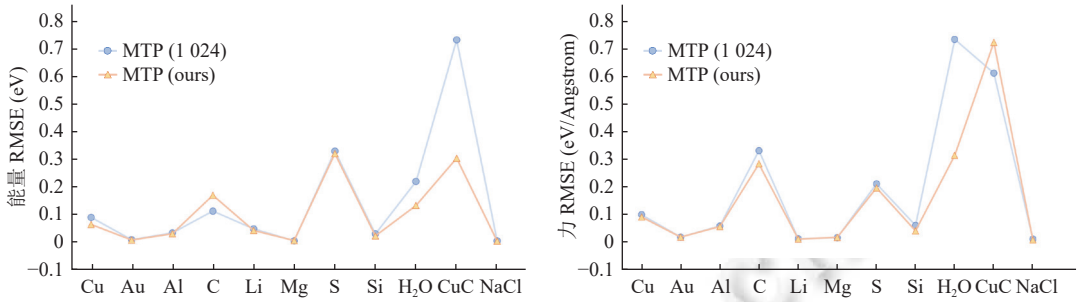
$$y_t = h(\theta_t, x_t) + \eta_t \tag{12}$$

$$\mathbf{P}_t = \lambda_t^{-1}\mathbf{P}_{t-1} - \lambda_t^{-2}\mathbf{P}_{t-1}\mathbf{H}_t^{\mathrm{T}}a_t^{-1}\mathbf{H}_t\mathbf{P}_{t-1}L^{-1} \tag{13}$$

$$A_t = \lambda_t^{-1}\mathbf{H}^{\mathrm{T}}{}_t\mathbf{P}_{t-1}\mathbf{H}_t L^{-1} + 1 = E\left[\epsilon_t^2\right]\alpha_t^2 \tag{14}$$

$$\Lambda_t = 1-(1-\Lambda_1)\nu^{t-1} \tag{15}$$

$$\mathbf{P}_t = E\left[\tilde{w}_t\tilde{w}_t^{\mathrm{T}}\right]\alpha_t^2 \tag{16}$$



(a) 能量 (b) 原子受力

图 5 分块大小对收敛精度的影响

根据伍德伯里矩阵恒等式, 误差协方差矩阵 $E[\tilde{w}_t\tilde{w}_t^{\mathrm{T}}]$ 可以表示成公式 (17) 的形式. 假设 H_i 独立同分布, 满足均值为 0 方差为 σ^2 的正态分布, 可以得到公式 (18). 使用 q-Pochhammer 符号, 我们发现 α_t^2 的极限存在, 如公式 (19) 所示. 最后根据大数定律, 我们有公式 (20), 且 $S(t)$ 为 $\mathrm{O}(t)$ 阶. 因此可以得到公式 (21). 随着 t 趋于无穷, 使用马尔可夫不等式, ϵ_t 被上界 B 给限制住如公式 (22) 所示的概率是任意趋于 1 的. 因此采用卡尔曼滤波优化器进行权重更新时的增量 $\epsilon_t K_t$ 随着训练次数的增多 $t\to\infty$, 以任意接近 1 的概率趋于 0, 表示为公式 (23), 因此可以说明适用卡尔曼滤波优化器一定能防止梯度爆炸.

$$E\left[\tilde{w}_t\tilde{w}_t^{\mathrm{T}}\right] = \left(\mathbf{P}_0^{-1}+\sum\nolimits_{i=1}^{t}\alpha_i^2\mathbf{H}_i\mathbf{H}_i^{\mathrm{T}}L^{-1}\right)^{-1} \tag{17}$$

$$E\left[\mathbf{P}_t^{-1}\right] = I+\sum\nolimits_{k=1}^{t}\alpha_k^2\alpha_t^{-2}L^{-1}\sigma^2 I \tag{18}$$

$$\lim_{t\to\infty}\alpha_t^2 = \Pi_{i=0}^{\infty}\left(1-(1-\lambda_1)\nu^i\right) = ((1-\lambda_1);\nu)_{\infty} = \alpha \tag{19}$$

$$\lim_{t\to\infty}\frac{\mathbf{P}_t^{-1}}{S(t)} \xrightarrow{\text{a.s.}} \sigma^2L^{-1},\ S(t) := \sum\nolimits_{k=1}^{t}\alpha_k^2\alpha_t^{-2} \tag{20}$$

$$G_t \overset{\text{a.s.}}{\sim} \mathrm{O}\left(\frac{1}{t}\right) \tag{21}$$

$$P(|\epsilon_t|\leqslant B) \geqslant 1-\frac{E[\epsilon_t^2]}{B^2} = 1-\frac{\alpha_t^{-2}(\lambda_t^{-1}\mathbf{H}_t^{\mathrm{T}}\mathbf{P}_{t-1}\mathbf{H}_t+1)}{B^2} \tag{22}$$

$$\epsilon_t G_t \sim \mathrm{O}\left(\frac{1}{t}\right) \tag{23}$$

下面我们验证层重组卡尔曼滤波优化器对权重初始化分布和取值的鲁棒性. 在神经网络训练前需要对权重参数进行初始的赋值操作, 常用的权重初始化方法包括: 随机初始化、Xavier 初始化等. 权重的初始化对神经网络的训练起着至关重要的作用, 合适的权重初始化能避免梯度消失或爆炸、起到加速收敛的效果. 针对不同网络结构权重初始化的方法也不尽相同. 在实际问题中, 通常尝试不同的权重初始化方法, 评估其对模型精度和性能的影响, 最后选出最合适的初始化方法. 在层重组的卡尔曼滤波优化器中, 我们理论证明了该优化器能防止梯度爆炸, 如公式 (23) 所示, 随着迭代的进行, 权重更新的增量会趋于 0. 下面验证权重的不同初始值的收敛精度, 表 2 中最后两列分别表示第 30 个 epoch 时能量和原子受力的均方根误差, “/”前后分别表示训练集和测试集的结果. 配置①和②分别表示在 Xavier 的初始化方法中, 配置③和④分别表示在常规的均匀分布和正态分布下的能量和原子受力的均方根误差. 从表 2 中配置①–④, 我们可以得到如下结论: (1) 层重组卡尔曼滤波优化器对不同权重初始化方法具有较强鲁棒性. 其中配置①和②、配置③和④分别代表不同权重初始化方法; (2) 层重组卡尔曼滤波优化器对不同权重初始化数据分布不敏感, 其中配置①和③、配置②和④分别表示均匀分布和正态分布, 其能量和原子受

力的收敛精度为 0.0735±0.0046, 0.07635±0.00155, 具有较小的误差方差 (0.001 量级). 在权重不同初始化分布和不同取值的情况下, 能量和原子受力最终收敛的 RMSE 的误差界 (error bar) 分别为±0.0046 和±0.00155, 波动幅度为±6.26% 和±2.03%.

表 2　铜体系基于层重组卡尔曼滤波优化器的鲁棒性验证

配置序号	参数说明	能量RMSE (eV)	力RMSE (eV/Angstrom)
配置①	权重初始化: Xavier_uniform	0.033 3/0.070 0	0.058 5/0.076 2
配置②	权重初始化: Xavier_normal	0.033 2/0.068 9	0.058 3/0.074 8
配置③	权重初始化: Init_uniform U(0, 0.001)	0.034 0/0.078 1	0.058 5/0.077 4
配置④	权重初始化: Init_normal N(0, 0.001)	0.033 7/0.076 7	0.058 7/0.077 9
配置⑤	激活函数: Sigmoid	0.040 3/0.049 5	0.058 4/0.070 5
配置⑥	激活函数: Softplus	0.039 9/0.069 7	0.058 3/0.074 3
配置⑦	激活函数: ReLU	0.045 1/0.057 3	0.067 5/0.081 0
配置⑧	激活函数: LeakyReLU	0.043 8/0.057 9	0.066 8/0.080 1

配置①–④均为 tanh 激活函数下的测试结果, 接下来我们验证激活函数的鲁棒性. 我们采用控制变量法, 固定一种相同的数据初始化方法, 不妨选取配置①的权重初始化方法, 以验证不同激活函数的收敛性表现. 常见的激活函数包括 Sigmoid、Softplus、ReLU、LeakyReLU 等. 根据表 2 中配置⑤–⑧的精度结果 (第 30 个 epoch 的能量和原子受力的训练集和测试集的结果), 我们发现: 当使用不同的激活函数进行神经网络训练时, 能量和原子受力的收敛精度为 0.05975±0.01, 0.0755±0.0055, 能量和原子受力的波动幅度为±16.73% 和±7.28%. 激活函数的选取对于最终收敛精度的影响比权重初始化分布大, 配置④–⑧均收敛, 没有出现梯度爆炸现象.

3　实验结果与分析

数据集: 我们构建了 12 个不同体系的含周期性边界条件的数据集, 数据集的具体配置见表 3. 第 2、3 列分别表示其对应的详细结构信息和样本量, 其中面心立方 (face-centered-cubic, FCC)、体心立方 (body-centered-cubic, BCC)、六方密堆积 (hexagonal-close-packed, HCP)、钻石立方 (diamond-cubic, DC)、表面吸附 (surface-adsorption, SA). 第 4 列记录了每一个样本中的原子总数, 这些数据集中的原子总数最高达 192. 训练这一类的数据 (包含多个相位的、原子总数较大的、有周期性边界条件) 相比于小分子数据, 是一个更具挑战性的工作. 表中样本均由 PWmat[54]的软件计算得到的第 1 性原理精度的数据. 在数据生成时, 为了扩大数据集覆盖的样本空间, 我们在生成连续的 2 个样本时, 取了稍大的时间步长 (第 5 列, 单位为 femtosecond (fs)).

表 3　数据集描述

体系	结构	样本量	单样本原子总数	时间步长 (fs)
Cu	FCC	1 646	108	1
Ag	FCC	2 015	32	2.5–3
Al	FCC	4 000	64	2–3.5
C	Graphene	4 000	64	2–3.5
Li	BCC, FCC, HCP	1 000	192	0.5
Mg	HCP	4 000	36	0.5–2
S	S_8	2 000	128	3–5
Si	DC	3 000	72	2.5–3.5
CuO	FCC	1 000	64	3
H_2O	Liquid	4 000	48	0.5
Cu+C	SA	2 000	118	3–3.2
NaCl	FCC	3 193	64	2–3.5

环境配置: 本节中的实验在 NVIDIA GeForce RTX 3090 上进行测试, 显存为 24 GB. 显式解析函数的采用 Fortran 进行编写, 实现过程中调用了 MKL 2022.0.2 数学函数库, 编译器为 Gcc 7.5.0; 神经网络的搭建采用 PyTorch 1.3, 数据的归一化预处理调用了 scikit-learn 1.2.0.

模型参数: 我们评估了 2B3B-Cosine、2B3B-Gaussian、MTP、SNAP 作为解析函数, 产生描述符 (descriptor), 再输入到拟合网络中进行训练的神经网络力场模型. 这些方法生成的中心原子的特征数量记录在表 4 中. 拟合网络采用全连接神经网络, 为了保证网络的一致性, 每一层的神经元数量分别为 [60, 30, 1], 激活函数为 tanh. 在测试中, 一阶优化器采用 Adam, 学习率的衰减及总体损失的设定参考第 2 节中观察 1 中配置④的参数. 层重组的卡尔曼滤波优化器的关于分块阈值的选取参考第 2.2 节中分块大小的建模.

表 4　不同解析函数的特征数

原子种类	2B3B-Cosine	2B3B-Gaussian	MTP	SNAP
单原子体系	42	26	53	60
两原子体系	111	72	253	60

评价指标: 均方根误差 (root mean square error, *RMSE*) 广泛应用在回归模型的性能评估中. 本文采用 *RMSE* 作为预测准确性的评价指标, *RMSE* 的数值越小, 表示模型的预测值与真实值之间的差距越小, 即模型的拟合能力更好, 反之表示拟合得不好. 在力场训练的网络中, 我们既关心每个样本的总能量的拟合效果, 同时还关心单原子受力的拟合效果. 因此在我们的测评里, 即需要考虑体系总能量的均方根误差 $RMSE_{\mathrm{E}}$, 同时也需评估原子受力的均方根误差 $RMSE_{\mathrm{F}}$, 其计算见公式 (24) 和 (25). 其中样本的集合记为 X, $|X|$ 表示样本数, E_{x_i} 表示样本 x_i 能量的真实值 (标签, label), $(F_{x_i}^{e_1}, F_{x_i}^{e_2}, F_{x_i}^{e_3})$ 分别表示样本 x_i 原子受力 3 个分量上的真实值. $\widehat{E}_{x_i}$ 则表示样本 x_i 能量的预测值 (prediction), $(\widehat{F}_{x_i}^{e_1}, \widehat{F}_{x_i}^{e_2}, \widehat{F}_{x_i}^{e_3})$ 分别表示样本 x_i 原子受力 3 个分量上的模型预测值.

$$RMSE_{\mathrm{E}} = \sqrt{\frac{1}{|X|}\sum_{x_i \epsilon X}\left(\widehat{E}_{x_i} - E_{x_i}\right)^2} \tag{24}$$

$$RMSE_{\mathrm{F}} = \sqrt{\frac{1}{3|X|}\sum_{x_i \epsilon X}\left[\left(\widehat{F}_{x_i}^{e_1} - F_{x_i}^{e_1}\right)^2 + \left(\widehat{F}_{x_i}^{e_2} - F_{x_i}^{e_2}\right)^2 + \left(\widehat{F}_{x_i}^{e_3} - F_{x_i}^{e_3}\right)^2\right]} \tag{25}$$

3.1 收敛精度

表 5 中展示了在 11 个有代表性的数据集上分别测试 4 种不同模型上 (2B3B-Cosine、2B3B-Gaussian、MTP、SNAP)、两种优化器 (Adam 优化器、层重组卡尔曼滤波优化器) 下的收敛精度和均方根误差. 其中第 3、4 列表示训练集的能量和原子受力的均方根误差, 第 5、6 列表示测试集的能量和原子受力的均方根误差, “/”前后的数据分别表示训练采用的 Adam 优化器和层重组卡尔曼滤波优化器.

从表 5 可以得出以下结论: (1) 单元素体系 (除了 S_8) 相比于多元素体系, 能量和原子受力均能收敛到 0.1 以下的精度; (2) 相同的特征和相同的拟合网络, Adam 优化器和层重组卡尔曼滤波优化器在单原子类型的测试数据下关于能量和受力的收敛精度更为接近, 多元素的体系中收敛的误差差异较大, 以 H_2O 为例, Adam 和 RLEKF 在能量和力上的 RMSE 的相差超过 1; (3) 4 种生成特征的方法由于解析函数的构造不同, 导致在相同的拟合网络下的收敛结果有些许差异, 说明不同特征生成的方法各自适用的体系不同. 例如碳体系, 2B3B-Cosine 相比于 MTP 和 SNAP 更能建模准能量和受力. 对于锂和镁体系, MTP 相比于其他 3 种方法表现出更高的收敛精度.

上面的结论是基于表 5 中的精度数据得到的, 在此基础上, 我们可以进一步推广得到更为一般的结论. 例如根据结论 (1) 和 (2), 可以看出多元素的训练更具挑战性; 根据结论 (2), 相同输入特征和拟合网络表明网络具有相同的表达能力, 优化器负责训练过程中权重的不断更新, 优化器算法的不同会导致最终收敛到不同的局部最优点, 优化器的选取在多元素数据集上的影响比单元素数据集更大; 根据结论 (3), 为了能精准建模一个体系, 中心原子的描述符的生成 (也即基组的种类) 至关重要, 不同体系适用的描述符各不相同.

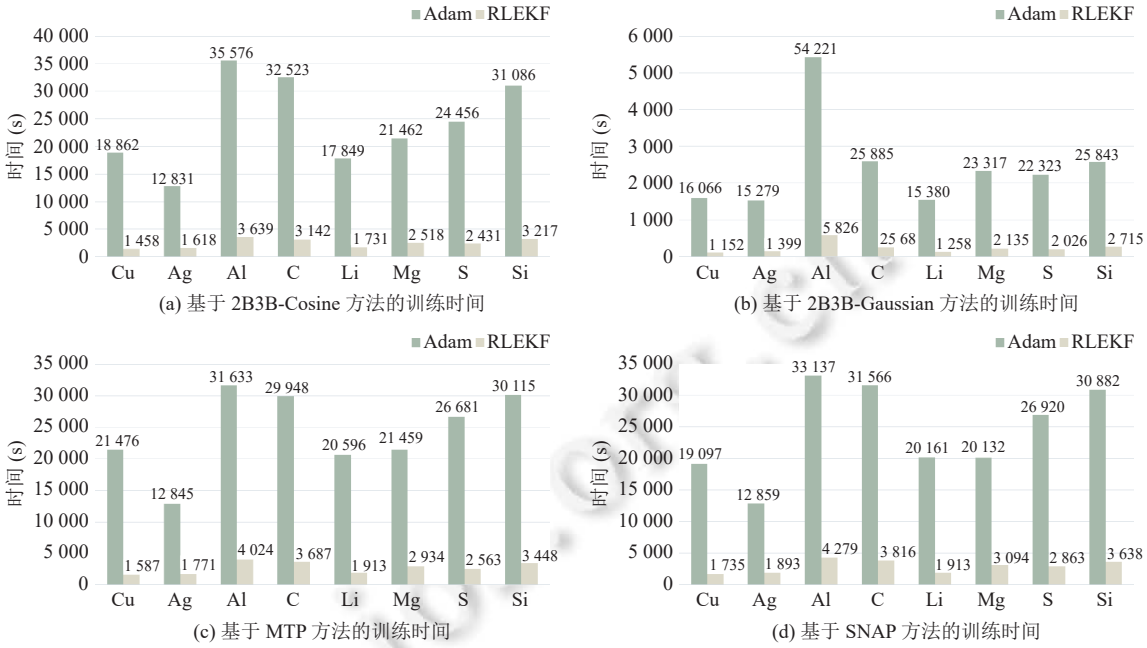
表 5　多模型在 Adam 和层重组卡尔曼滤波优化器下的收敛误差

体系	模型/方法	训练集 $RMSE_E$ (Adam/RLEKF)	训练集 $RMSE_F$ (Adam/RLEKF)	测试集 $RMSE_E$ (Adam/RLEKF)	测试集 $RMSE_F$ (Adam/KF)
Cu1646	2B3B-Cosine	0.0544/0.0392	0.0615/0.0577	0.0603/0.0487	0.0733/0.0698
	2B3B-Gaussian	0.0795/0.0480	0.0702/0.0610	0.0845/0.0585	0.0815/0.0699
	MTP	0.0724/0.0499	0.0732/0.0702	0.0728/0.0649	0.0890/0.0893
	SNAP	0.0867/0.0631	0.118/0.0926	0.125/0.0856	0.157/0.122
Ag2015	2B3B-Cosine	0.00492/0.00342	0.0131/0.0100	0.00835/0.00840	0.0164/0.0196
	2B3B-Gaussian	0.00614/0.00472	0.0125/0.0114	0.00774/0.00763	0.0138/0.0142
	MTP	0.00633/0.00344	0.0144/0.00849	0.0119/0.00833	0.0178/0.0165
	SNAP	0.00899/0.00488	0.0207/0.0124	0.0139/0.0120	0.0253/0.0218
Al4000	2B3B-Cosine	0.0400/0.0293	0.0614/0.0555	0.0406/0.0386	0.0615/0.0615
	2B3B-Gaussian	0.0566/0.0381	0.0735/0.0645	0.0542/0.0432	0.0721/0.0662
	MTP	0.0385/0.0247	0.0588/0.0481	0.0342/0.0312	0.0584/0.0548
	SNAP	0.0519/0.0322	0.0806/0.0668	0.0513/0.0352	0.0836/0.0773
C4000	2B3B-Cosine	0.0215/0.0149	0.0425/0.0354	0.0169/0.0187	0.0498/0.0478
	2B3B-Gaussian	0.0267/0.0128	0.0709/0.0343	0.0251/0.0222	0.0759/0.0434
	MTP	0.131/0.0699	0.330/0.200	0.129/0.171	0.354/0.283
	SNAP	0.222/0.129	0.562/0.481	0.365/0.233	0.732/0.702
Li1000	2B3B-Cosine	0.152/0.0124	0.0263/0.0261	0.290/0.174	0.0274/0.0284
	2B3B-Gaussian	0.0917/0.0170	0.0200/0.0201	0.178/0.123	0.0197/0.0204
	MTP	0.0314/0.00340	0.00922/0.00888	0.0397/0.0431	0.00952/0.00943
	SNAP	0.0958/0.00667	0.0202/0.0161	0.113/0.0788	0.0215/0.0174
Mg4000	2B3B-Cosine	0.0122/0.00879	0.0209/0.0179	0.00873/0.01009	0.0226/0.0296
	2B3B-Gaussian	0.0158/0.00921	0.0230/0.0197	0.0146/0.0135	0.0234/0.0237
	MTP	0.00718/0.00286	0.0141/0.0105	0.00652/0.00625	0.0158/0.0153
	SNAP	0.0125/0.00287	0.0183/0.0114	0.0120/0.00785	0.0213/0.0159
S2000	2B3B-Cosine	0.198/0.0205	0.0809/0.0731	0.297/0.249	0.135/0.145
	2B3B-Gaussian	0.249/0.0521	0.106/0.0875	0.462/0.217	0.150/0.139
	MTP	0.202/0.0225	0.147/0.0893	0.438/0.322	0.213/0.195
	SNAP	0.499/0.0569	0.266/0.198	0.661/0.511	0.351/0.328
Si3000	2B3B-Cosine	0.0479/0.0384	0.0519/0.0473	0.0377/0.0320	0.0535/0.0518
	2B3B-Gaussian	0.0679/0.0417	0.0591/0.0452	0.0511/0.0469	0.0608/0.0478
	MTP	0.0448/0.0178	0.0622/0.0345	0.0317/0.0234	0.0639/0.0389
	SNAP	0.0529/0.0259	0.0753/0.0455	0.0422/0.0239	0.0793/0.0492
H_2O4000	2B3B-Cosine	0.0236/0.00900	0.0488/0.0394	0.0332/0.244	0.0864/1.10
	2B3B-Gaussian	0.0416/0.267	0.0711/3.03	0.0308/0.024	0.0815/0.0676
	MTP	0.0307/0.0101	0.0496/0.134	0.0996/0.0355	0.129/0.314
	SNAP	0.204/0.226	0.575/0.748	0.313/0.339	0.690/0.791
Cu+C2000	2B3B-Cosine	0.110/0.0566	0.178/0.174	0.112/0.192	0.188/0.204
	MTP	0.793/0.0981	0.190/0.190	0.724/0.305	0.209/0.725
	SNAP	0.631/0.167	0.469/0.507	0.467/0.350	0.495/0.546
NaCl3193	2B3B-Cosine	0.00679/0.00320	0.00887/0.00738	0.613/1.38	4.15/2.73
	2B3B-Gaussian	0.00690/0.00338	0.00785/0.00659	0.00828/0.00382	0.00854/0.00734
	MTP	0.00485/0.00171	0.00738/0.00571	0.00474/0.00403	0.00858/0.00671
	SNAP	0.0132/0.00591	0.0176/0.0123	0.0150/0.0108	0.0186/0.0148

3.2　收敛速度

我们考虑 8 个单元素体系的数据集的收敛速度, 鉴于表 5 中单元素体系在一阶 Adam 优化器和二阶层重组卡尔曼滤波优化器的收敛精度相当, 因此比较 8 个单元素体系收敛到表 5 所示的精度时的收敛速度. 图 6(a)–(d) 分别表示在 8 个单元素的体系, 4 种不同模型下, Adam 优化器训练 1 000 个 epoch, 层重组卡尔曼滤波优化器训练

30 个 epoch 的训练时间. 整体来看, 在 2B3B-Cosine、2B3B-Gaussian、MTP、SNAP 这 4 种模型下的 8 个体系的平均加速比分别为: 9.94×, 10.99×, 9.25×, 8.59×.



(a) 基于 2B3B-Cosine 方法的训练时间　(b) 基于 2B3B-Gaussian 方法的训练时间

(c) 基于 MTP 方法的训练时间　(d) 基于 SNAP 方法的训练时间

图 6　不同模型下 Adam 优化器和层重组卡尔曼滤波优化器的端到端训练时间

4 总　结

神经网络力场是科学智能 (AI-for-Science) 领域的重要研究方向, 然而其训练过程的稳定和精度是该方向面临的关键挑战之一. 我们首先概述了现阶段神经网络力场在一阶训练方法中普遍存在的问题和挑战, 分别为: 能量和力场的联合训练任务中损失函数的设定、学习率的选取和衰减策略、训练过程中可能出现的梯度爆炸现象. 并基于一些常见的神经网络力场模型给出 3 个观察并进行论证. 本文的贡献如下: (1) 通过交替训练的方式规避联合训练中损失函数的强经验依赖的参数设置问题, 同时 分析了交替训练的精度收益和额外的时间消耗; (2) 设计了层重组卡尔曼滤波优化器的分块大小阈值, 通过建模得到该阈值, 块大小阈值超参无需人为指定; (3) 给出层重组卡尔曼滤波方法能防止梯度爆炸的理论证明, 同时验证神经网络力场训练过程中权重初始化和激活函数选取的鲁棒性.

本文针对 4 种典型的神经网络力场模型, 在 11 个有代表性的体系上测试了一阶优化器和层重组卡尔曼滤波优化器的性能表现 (收敛精度和速度). 实验表明, 在 8 个单精度的体系中, Adam 优化器和层重组卡尔曼优化器关于能量和原子受力收敛到相当的精度时, 层重组的卡尔曼滤波优化器相比于 Adam 优化器能达到 8–10 倍的加速比.

References:

[1] Qian DP, Wang R. Key issues in exascale computing. SCIENTIA SINICA Informationis, 2020, 50(9): 1303–1326 (in Chinese with English abstract). [doi: 10.1360/SSI-2020-0099]

[2] Doltsinis NL, Marx D. Nonadiabatic car-parrinello molecular dynamics. Physical Review Letters, 2002, 88(16): 166402. [doi: 10.1103/PhysRevLett.88.166402]

[3] Marx D, Hutter J. Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods. Cambridge: Cambridge University Press, 2009.

[4] Parr RG, Yang WT. Density-functional Theory of Atoms and Molecules. Oxford: Oxford University Press, 1995.

[5] Koch W, Holthausen MC. A Chemist's Guide to Density Functional Theory. 2nd ed., New York: Wiley-VCH, 2001.

[6] Raty JY, Gygi F, Galli G. Growth of carbon nanotubes on metal nanoparticles: A microscopic mechanism from ab initio molecular dynamics simulations. Physical Review Letters, 95(9): 096103. [doi: 10.1103/PhysRevLett.95.096103]

[7] Ma T, Wang SH. Phase Transition Dynamics. Cham: Springer, 2019. [doi: 10.1007/978-3-030-29260-7]

[8] Johannes L, Simon S, Hans H. On the history of the Lennard-Jones potential. Annalen der Physik, 2024, (536): 2400115. [doi: 10.1002/andp.202400115]

[9] Foiles SM, Baskes MI, Daw MS. Embedded-atom-method functions for the FCC metals Cu, Ag, Au, Ni, Pd, Pt, and their alloys. Physical Review B, 1986, 33(12): 7983–7991. [doi: 10.1103/physrevb.33.7983]

[10] Senftle TP, Hong S, Islam MM, Kylasa SB, Zheng YX, Shin YK, Junkermeier C, Engel-Herbert R, Janik MJ, Aktulga HM, Verstraelen T, Grama A, van Duin ACT. The ReaxFF reactive force-field: Development, applications and future directions. npj Computational Materials, 2016, 2(1): 15011. [doi: 10.1038/npjcompumats.2015.11]

[11] Smit B. Phase diagrams of Lennard-Jones fluids. The Journal of Chemical Physics, 1992, 96(11): 8639–8640. [doi: 10.1063/1.462271]

[12] Koura K, Matsumoto H. Variable soft sphere molecular model for inverse-power-law or Lennard-Jones potential. Physics of Fluids A, 1991, 3(10): 2459–2465. [doi: 10.1063/1.858184]

[13] Daw MS, Baskes MI. Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. Physical Review B, 1984, 29(12): 6443–6453. [doi: 10.1103/physrevb.29.6443]

[14] Jelinek B, Groh S, Horstemeyer MF, Houze J, Kim SG, Wagner GJ, Moitra A, Baskes MI. Modified embedded atom method potential for Al, Si, Mg, Cu, and Fe alloys. Physical Review B, 2012, 85(24): 245102. [doi: 10.1103/physrevb.85.245102]

[15] Chenoweth K, Van Duin ACT, Goddard WA. ReaxFF reactive force field for molecular dynamics simulations of hydrocarbon oxidation. The Journal of Physical Chemistry A, 2008, 112(5): 1040–1053. [doi: 10.1021/jp709896w]

[16] Guo ZQ, Lu DH, Yan YJ, Hu SY, Liu RR, Tan GM, Sun NH, Jiang WR, Liu LJ, Chen YX, Zhang LF, Chen MH, Wang H, Jia WL. Extending the limit of molecular dynamics with ab initio accuracy to 10 billion atoms. In: Proc. of the 27th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming. New York: Association for Computing Machinery, 2022. 205–218. [doi: 10.1145/3503221.3508425]

[17] Jia WL, Wang H, Chen MH, Lu DH, Lin L, Car R, Weinan E, Zhang LF. Pushing the limit of molecular dynamics with ab initio accuracy to 100 million atoms with machine learning. In: Proc. of the 2020 Int'l Conf. for High Performance Computing, Networking, Storage and Analysis. Atlanta: IEEE, 2020. 1–14. [doi: 10.1109/SC41405.2020.00009]

[18] Thompson AP, Swiler LP, Trott CR, Foiles SM, Tucker GJ. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. Journal of Computational Physics, 2015, 285: 316–330. [doi: 10.1016/j.jcp.2014.12.018]

[19] Lee K, Yoo D, Jeong W, Han S. SIMPLE-NN: An efficient package for training and executing neural-network interatomic potentials. Computer Physics Communications, 2019, 242: 95–103. [doi: 10.1016/j.cpc.2019.04.014]

[20] Behler J. Representing potential energy surfaces by high-dimensional neural network potentials. Journal of Physics: Condensed Matter, 2014, 26(18): 183001. [doi: 10.1088/0953-8984/26/18/183001]

[21] Behler J. First principles neural network potentials for reactive simulations of large molecular and condensed systems. Angewandte Chemie International Edition, 2017, 56(42): 12828–12840. [doi: 10.1002/anie.201703114]

[22] Behler J, Parrinello M. Generalized neural-network representation of high-dimensional potential-energy surfaces. Physical Review Letters, 2007, 98(14): 146401. [doi: 10.1103/physrevlett.98.146401]

[23] Yao K, Herr JE, Brown SN, Parkhill J. Intrinsic bond energies from a bonds-in-molecules neural network. The Journal of Physical Chemistry Letters, 2017, 8(12): 2689–2694. [doi: 10.1021/acs.jpclett.7b01072]

[24] Desai S, Reeve ST, Belak JF. Implementing a neural network interatomic model with performance portability for emerging exascale architectures. Computer Physics Communications, 2022, 270: 108156. [doi: 10.1016/j.cpc.2021.108156]

[25] Huang YP, Xia YJ, Yang LJ, Wei JC, Yang YI, Gao YQ. SPONGE: A GPU-accelerated molecular dynamics package with enhanced sampling and AI-driven algorithms. Chinese Journal of Chemistry, 2022, 40(1): 160–168. [doi: 10.1002/cjoc.202100456]

[26] Wang H, Zhang LF, Han JQ, E WN. DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics. Computer Physics Communications, 2018, 228: 178–184. [doi: 10.1016/j.cpc.2018.03.016]

[27] Schütt KT, Arbabzadah F, Chmiela S, Müller KR, Tkatchenko A. Quantum-chemical insights from deep tensor neural networks. Nature Communications, 2017, 8(1): 13890. [doi: 10.1038/ncomms13890]

[28] Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. In: Proc. of the 34th Int'l Conf. on Machine Learning. Sydney: JMLR.org, 2017. 1263–1272.

[29] Schütt KR, Unke O, Gastegger M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In: Proc. of the 38th Int'l Conf. on Machine Learning. PMLR, 2021. 9377–9388.

[30] Haghighatlari M, Li J, Guan XY, Zhang OF, Das A, Stein CJ, Heidar-Zadeh F, Liu ML, Head-Gordon M, Bertels L, Hao HX, Leven I, Head-Gordon T. NewtonNet: A Newtonian message passing network for deep learning of interatomic potentials and forces. Digital Discovery, 2022, 1(3): 333–343. [doi: 10.1039/d2dd00008c]

[31] Batzner S, Musaelian A, Sun LX, Geiger M, Mailoa JP, Kornbluth M, Molinari N, Smidt TE, Kozinsky B. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. Nature Communications, 2022, 13(1): 2453. [doi: 10.1038/s41467-022-29939-5]

[32] Gasteiger J, Giri S, Margraf JT, Günnemann S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. In: Proc. of the 2020 Machine Learning for Molecules Workshop. NeurIPS, 2020. 1–6.

[33] Gasteiger J, Groß J, Günnemann S. Directional message passing for molecular graphs. In: Proc. of the 2020 Int'l Conf. on Learning Representations (ICLR). OpenReview.net, 2020.

[34] Unke OT, Chmiela S, Gastegger M, Schütt KT, Sauceda HE, Müller KR. SpookyNet: Learning force fields with electronic degrees of freedom and nonlocal effects. Nature Communications, 2021, 12(1): 7273. [doi: 10.1038/s41467-021-27504-0]

[35] Hu SY, Zhang WT, Sha QC, Pan F, Wang LW, Jia WL, Tan GM, Zhao T. RLEKF: An optimizer for deep potential with ab initio accuracy. In: Proc. of the 2023 AAAI Conf. on Artificial Intelligence. Washington DC: AAAI, 2023. 7910–7918. [doi: 10.1609/aaai.v37i7.25957]

[36] Huang YF, Kang J, Goddard III WA, Wang LW. Density functional theory based neural network force fields from energy decompositions. Physical Review B, 2019, 99(6): 064103. [doi: 10.1103/PhysRevB.99.064103]

[37] Novikov IS, Gubaev K, Podryabinkin EV, Shapeev AV. The MLIP package: Moment tensor potentials with MPI and active learning. Machine Learning: Science and Technology, 2021, 2(2): 025002. [doi: 10.1088/2632-2153/abc9fe]

[38] Drautz R. Atomic cluster expansion for accurate and transferable interatomic potentials. Physical Review B, 2019, 99(1): 014104. [doi: 10.1103/PhysRevB.99.014104]

[39] Lysogorskiy Y, van der Oord C, Bochkarev A, Menon S, Rinaldi M, Hammerschmidt T, Mrovec M, Thompson A, Csányi G, Ortner C, Drautz R. Performant implementation of the atomic cluster expansion (PACE) and application to copper and silicon. npj Computational Materials, 2021, 7(1): 97. [doi: 10.1038/s41524-021-00559-9]

[40] Bartók AP, Kondor R, Csányi G. On representing chemical environments. Physical Review B, 2013, 87(18): 184115. [doi: 10.1103/physrevb.87.184115]

[41] Kingma DP, Ba LJ. Adam: A method for stochastic optimization. In: Proc. of the 2015 Int'l Conf. on Learning Representations. Ithaca, 2015.

[42] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. The Journal of Machine Learning Research, 2011, 12: 2121–2159.

[43] McMahan HB, Streeter M. Adaptive bound optimization for online convex optimization. In: Proc. of the 23rd Annual Conf. on Learning Theory. 2010. 244–256.

[44] Kurbiel T, Khaleghian S. Training of deep neural networks based on distance measures using RMSProp. arXiv:1708.01911, 2017.

[45] Zeiler MD. ADADELTA: An adaptive learning rate method. arXiv:1212.5701, 2012.

[46] Martens J, Grosse R. Optimizing neural networks with Kronecker-factored approximate curvature. In: Proc. of the 32nd Int'l Conf. on Int'l Conf. on Machine Learning. Lille: JMLR.org, 2015. 2408–2417.

[47] Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. Mathematical Programming, 1989, 45(1–3): 503–528. [doi: 10.1007/bf01589116]

[48] Zhu CY, Byrd RH, Lu PH, Nocedal J. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. ACM Trans. on Mathematical Software (TOMS), 1997, 23(4): 550–560. [doi: 10.1145/279232.279236]

[49] Gupta V, Koren T, Singer Y. Shampoo: Preconditioned stochastic tensor optimization. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 1842–1850.

[50] Yao ZW, Gholami A, Shen S, Mustafa M, Keutzer K, Mahoney M. ADAHESSIAN: An adaptive second order optimizer for machine learning. In: Proc. of the 2021 AAAI Conf. on Artificial Intelligence. Palo Alto: AIAA, 2021. 10665–10673. [doi: 10.1609/aaai.v35i12.17275]

[51] Kalman RE. A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 1960, 82(1): 35–45. [doi: 10.1115/1.3662552]

[52] Witkoskie JB, Doren DJ. Neural network models of potential energy surfaces: Prototypical examples. Journal of Chemical Theory and Computation, 2005, 1(1): 14–23. [doi: 10.1021/ct049976i]

[53] Haykin S. Kalman Filtering and Neural Networks. Wiley Online Library, 2001. [doi: 10.1002/0471221546]

[54] Jia WL, Fu JY, Cao ZY, Wang L, Chi XB, Gao WG, Wang LW. Fast plane wave density functional theory molecular dynamics calculations on multi-GPU machines. Journal of Computational Physics, 2013, 251: 102–115. [doi: 10.1016/j.jcp.2013.05.005]

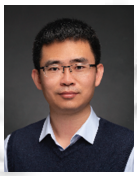
附中文参考文献:

[1] 钱德沛, 王锐. E 级计算的几个问题. 中国科学: 信息科学, 2020, 50(9): 1303–1326. [doi: 10.1360/SSI-2020-0099]

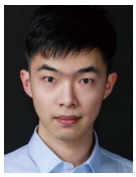
胡思宇(1996－), 女, 博士, 助理研究员, CCF 专业会员, 主要研究领域为机器学习, 并行算法设计, 神经网络力场.

汪林望(1964－), 男, 博士, 研究员, 博士生导师, 主要研究领域为大规模第一性原理计算, 算法开发, 凝聚态物理.

周远昌(2001－), 男, 硕士生, CCF 学生会员, 主要研究领域为深度学习, 高性能计算.



贾伟乐(1985－), 男, 博士, 研究员, 博士生导师, CCF 专业会员, 主要研究领域为高性能计算, 人工智能, 第一性原理计算.



赵曈(1995－), 男, 博士, CCF 专业会员, 主要研究领域为人工智能的基础理论, 高性能计算, 博弈论.

谭光明(1980－), 男, 博士, 研究员, 博士生导师, CCF 高级会员, 主要研究领域为并行算法设计与分析, 并行编程和优化, 计算机体系结构, 生物信息学, 大数据.