机器学习和计算数学

渭南市[1]

普林斯顿大学和北京大数据研究所

纪念冯康教授（1920-1993年）

内容

摘要

基于神经网络的机器学习能够以前所未有的效率和精度在非常高的维度上逼近函数。这开辟了许多令人兴奋的新可能性，不仅在传统的人工智能领域，而且在科学计算和计算科学领域。 同时，机器学习也获得了作为一套“黑匣子”类型的技巧的声誉，没有基本的原则。 这是在机器学习方面取得进一步进展的真正障碍。

[1]威南@math.princeton.edu

在本文中，我们试图解决以下两个非常重要的问题：（1）机器学习已经如何影响并将进一步影响计算数学、科学计算和计算科学？ （2）计算数学，特别是数值分析如何影响机器学习？ 我们介绍在这些问题上取得的一些最重要进展。 我们的希望是把事物放在一个有助于将机器学习与计算数学结合起来的角度。

1 导言

基于神经网络的机器学习(ML)在传统人工智能中的各种任务上取得了令人印象深刻的成功。 这包括对图像进行分类，生成新的图像，如（假）人脸，并玩复杂的游戏，如围棋。 所有这些任务的一个共同特点是它们涉及到非常高维的对象。 事实上，当用数学术语表示时，图像分类问题是一个将在图像空间上定义的高维函数逼近每个图像类别对应的离散值集的问题。 输入空间的维数通常是图像中像素数的 3 倍，其中 3 是颜色空间的维数。 图像生成问题是一个从未知的高维分布中生成样本的问题，给定一组来自该分布的样本。 围棋博弈问题是在动态规划中求解一个类贝尔曼方程，因为最优策略满足这样一个方程。对于像 Go 这样复杂的游戏，这个类似于 Bellman 的方程是在一个巨大的空间上制定的。

所有这些都是通过使用现代机器学习技术精确地逼近高维函数的能力来实现的。 这为解决遭受“维度诅咒”(CoD)的问题开辟了新的可能性：随着维度的增长，计算成本成倍增长。 长期以来，这个 CoD 问题一直是科学界的一个基本障碍。

以数值求解偏微分方程(PDE)为例。 利用传统的数值方法，如有限差分、有限元和谱方法，我们现在可以常规地求解三个空间维数加时间维数的 PDE。目前在计算数学中研究的大多数 PDE 都属于这一类。 众所周知的例子包括泊松方程、麦克斯韦方程、欧拉方程、Navier-Stokes 方程和线性弹性的 PDE。 稀疏网格可以增加我们处理 PDE 的能力，例如 8 到 10 维。 这使我们能够尝试解决问题，如简单分子的 Boltzmann 方程。 但是当我们面对 PDE 时，我们完全迷失了，比如说在 100 维。 这使得基本上不可能求解复杂分子的 Fokker-Planck 或 Boltzmann 方程、多体薛定谔方程或实际控制问题的 Hamilton-Jacobi-Bellman 方程。

这正是机器学习可以帮助的地方。 事实上，从[20,10,21]的工作开始，基于机器学习的求解高维 PDE 和控制问题的数值算法一直是近年来科学计算中最令人兴奋的新发展之一，这为计算数学开辟了一系列新的可能性。 我们请[17]回顾这一令人兴奋的发展。

解决 PDE 只是冰山一角。 还有许多其他问题，其中 CoD 是主要障碍，包括：

- 经典的多体问题，例如。 蛋白质折叠
- 湍流。 尽管湍流可以用三维 Navier-Stokes 方程来建模，但它具有如此多的主动自由度，一个有效的湍流模型应该涉及许多变量。
- 固体力学。在固体力学中，我们甚至没有 NavierStokes 方程的模拟。为什么是这样？ 那么，真正的原因是固体的行为本质上是一个多尺度的问题，它涉及从原子化到宏观的尺度。
- 多尺度建模。 事实上，没有区分尺度的大多数多尺度问题都属于这一类。 一个直接的例子是聚合物流体或聚合物熔体的动力学。

机器学习能帮助解决这些问题吗？ 更广泛地说，我们能否将机器学习的成功扩展到传统 AI

之外？ 我们将设法使读者相信，许多问题确实如此。

基于神经网络的机器学习除了具有强大的功能外，还获得了一套技巧而不是一套系统的科学原理的声誉。它的性能敏感地取决于超参数的值，如网络宽度和深度、初始化、学习速率等。事实上，就在几年前，参数调整被认为是一门艺术。 即使现在，一些任务仍然是这样。 因此，一个自然的问题是：我们能否理解这些微妙之处，并提出更好的机器学习模型，其性能更健壮？

在本文中，我们回顾了在这两个问题上学到的东西。 我们讨论了机器学习已经或将对计算数学产生的影响，以及如何利用计算数学的思想，特别是数值分析，帮助理解和更好地制定机器学习模型。 在此基础上，我们将主要讨论基于 ML 算法的新问题。 尽管机器学习也提出了解决计算数学中一些传统问题的新方法，但我们在这方面不会说太多。

2 基于机器学习的计算科学问题算法

在本节和下一节中，我们将讨论如何使用神经网络模型来开发新的算法。 对于不熟悉神经网络的读者来说，只要把它们看作是多项式的一些替换。 之后我们将讨论神经网络。

2.1 非线性多网格法和蛋白质折叠

在传统的多网格方法[5]中，对于求解由线性椭圆方程的某种有限差分或有限元离散而产生的方程的线性系统，我们的目标是使二次函数最小化

$$Ih(u_h) \quad \text{呃} \; L_h u_h \; —\text{FHU}_h$$

这里 h 是离散化的网格大小。 多网格方法的基本思想是在求解这个问题和网格大小为 H 的粗网格上的一个约简问题之间迭代。为了做到这一点，我们需要以下几点

- 投影算子：P：UhT 呃，它将在精细网格上定义的函数映射到在粗网格上定义的函数。
- 标度 H：L 的有效算子 $_h$=p^{t}Lh P。 这定义了粗网格上的目标函数：

$$\text{啊}^{(u}\text{h)=2}^{\text{u}}{}_{\text{h}}{}^{l}{}^{u}\boldsymbol{h}\text{-}{}^{f}{}^{u}\boldsymbol{h}$$

- 延长算子 Q：Uht 呃，它将在粗网格上定义的函数映射到在细网格上定义的函数。 通常一个人可以把 Q 取为 P^{t}.

这里的关键思想是粗粒化，在细尺度和粗粒度问题之间迭代。 粗粒度中的主要成分是一组粗粒度变量和有效的粗粒度问题。 通过这种方式，这些显然是一般的想法，可以与各种各样的问题相关。 然而，在实践中，困难在于如何获得有效的粗粒度问题，对于线性问题来说，这是一个微不足道的步骤，而这是机器学习可以帮助的地方。

我们将以蛋白质折叠问题为例来说明非线性问题的一般思想。

设{xj}是蛋白质和周围溶剂中原子的位置，U=U({xj})是组合蛋白质-溶剂体系的势能。 势能由化学键、范德华相互作用、静电相互作用等引起的能量组成。 蛋白质折叠问题是寻找能量 U 的“基态：

“最小化”U。

在这里，我们增加了引号，因为我们真正想做的是抽样分发

所=：“即顶所=(如 t

为了定义粗粒度问题，我们假设我们得到了一组集合变量：s=(si，•...，s_n)，Sj=Sj(xi，...，Xn)，(n<N)。 一种可能是使用

二面角作为粗粒度变量。 原则上，人们也可以使用机器学习方法来学习粗粒度变量的“最佳”集合，但这里不会追求这个方向。

$$A(\boldsymbol{s}) \quad -\frac{1}{\beta}\ln p(\boldsymbol{s}), \quad p_\beta(\boldsymbol{s}) \quad \frac{1}{Z}\int e^{-\beta U(\boldsymbol{x})}\delta(\boldsymbol{s}(\boldsymbol{x})-\boldsymbol{s})\,d\boldsymbol{x},$$

定义了粗粒度变量后，有效的粗粒度问题只是与这组粗粒度变量相关的自由能：

与线性问题的情况不同，在目前的情况下，我们必须首先找到函数 A。

其思想是用神经网络近似 A。 这里的问题是如何获取培训数据。

与大多数预先收集训练数据的标准机器学习问题相反，在计算科学和科学计算的应用中，随着学习的进行，训练数据被“即时”收集。 这被称为“并发学习”协议[11]。 在这方面，预先收集训练数据的标准机器学习问题是“顺序学习”的例子”。 并行学习的关键问题是以最佳方式生成数据的有效算法。 培训数据集一方面应具有足够的代表性，另一方面应尽可能小。

[11]提出了生成这类数据集的一般程序。 它被称为 EELT（探索检查-标记-训练）算法，它包括以下步骤：

- 探索：探索 S 空间。 这可以通过采样*e 来完成^A(s)用 A 的电流近似。
- 检查：对于每个探索的状态，决定是否应该标记该状态。 这样做的一种方法是使用后验误差估计器。 这种后验误差估计的一个可能是机器学习模型集合预测的方差，见[41]。
- 标记：计算平均力（例如使用约束分子动力学）

  F(s)=-VS A(s)。

  其中自由能 A 可以用标准热力学积分计算。

训练：训练适当的神经网络模型。 要想建立一个好的神经网络模型，就必须考虑到问题中的对称性。 例如，如果我们通过消除氢原子的位置来粗化水分子集合的全原子表示，那么所得到的系统的自由能函数应该具有排列对称性，在设计神经网络模型时应该考虑到这一点（见下一小节）。

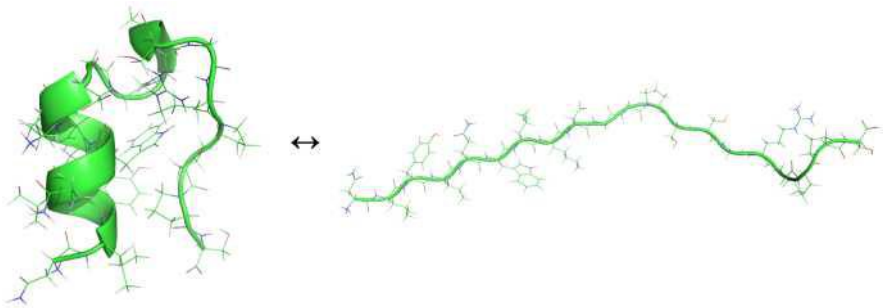


图 1：Trp 笼的折叠和扩展状态，经[37]许可复制

这也可以看作是一种非线性的多网格算法，因为它迭代粗粒度变量空间上的采样和全原子描述的（受约束的）吉布斯分布伽。

这是一个一般的过程，应该适用于一大类非线性“多网格”问题。

图 1 显示的是 Trp 笼的扩展和折叠结构。 这是一种含有 20 个氨基酸的小蛋白质。 我们选择了 38 个二面角作为集体变量。 [37]给出了全部结果。

分子动力学是通过跟踪系统中所有核的轨迹来研究分子和材料系统行为的一种方法。 假设原子核的动力学服从牛顿定律，一些势能函数(通常称为势能面或 PES)V 来模拟原子核之间的有效相互作用：

2.2 具有从头计算精度的分子动力学

$$m_i \frac{d^2 x_i}{dt^2} = -\frac{\partial V(x_1, x_2, \ldots, x_n)}{\partial x_i}$$

怎样才能得到函数 V？ 传统上，有两种相当不同的方法。 第一种是利用量子力学模型计算飞行中的原子间力(-W)，最流行的是密度泛函理论(DFT)。 这被称为 Car-Parrinello 分子动力学或从头算分子动力学[6,8]。这种方法是相当准确的，但也是非常昂贵的，限制了系统的大小，一个人可以处理大约 1000 个原子，即使高性能超级计算机。 另一种方法是提出经验潜力。 基本上，一个人猜测一个函数形式的 V 与一组小的拟合参数，然后由一组小的数据确定。 这种方法非常有效，但不可靠。 这种准确性和效率之间的困境长期以来一直是分子动力学的一个重要障碍。

对于机器学习，新的范式是使用 DFT 生成数据，然后使用机器学习生成对 V 的近似。这种方法有

系统			Al		镁		铝镁合金	
类型	格	#原子	奥菲	#数据	#C6ins	#数	奥菲	#数据
	联邦通信委	3	15.174.000	1.32	15.174.000	860	39.266.46	7.313
	HCP	1	15,174,000	908	15,174,000	760	18,999,90	2,461
	钻石	1	5,058,000	1,02	5,058,000	543	5,451,30	2,607
	sc	8	5,058,000	713	5,058,000	234	2,543,94	667

	FCC（100）	1	3,270,960	728	3,270,960	251	62,203,68	1,131
	FCC（110）	16[a],20[b]	3,270,960	838	3,270,960	353	10,744,272	2,435
地面	FCC（111）	1	3,270,960	544	3,270,960	230	62,203,68	1,160
	HCP（0001）	1	3,270,960	3	3,270,960	109	62,203,68	176
	HCP（1010）	1	3,270,960	7	3,270,960	167	62,203,68	203
	HCP（1120）	16[a],20[b]	3,270,960	293	3,270,960	182	107,442,72	501
和			60,089,760	6,489	60,089,760	3,689	529,961,76	18,654

[a]纯铝
[b]镁和铝镁合金

图 2：EELT 算法的结果：探索的配置数量与标记的数据点的数量。 只有很小比例的配置被标记。 经张林峰许可转载。 另见[40]。

产生与 V 的近似的潜力，它与 DFT 模型一样精确，并且与经验电位一样有效。

为了实现这一目标，我们必须解决两个问题。 一是数据的生成。 二是提出了合适的神经网络模型。 这两个问题是我们在这里讨论的所有问题的共同特征。

自适应数据生成的问题与以前非常相同。 仍然可以使用 EELT 程序。 每个步骤如何实现的细节有点不同。 详情请参阅[40]。

图 2 显示了使用 EELT 算法的效果。 正如人们所看到的，探索的配置中很小一部分实际上被标记。 对于 Al-Mg 示例，只选择~0.005%的配置进行标记。

对于第二个问题，设计合适的神经网络，最重要的考虑因素是：

1. 扩展，神经网络应该是广泛的，因为如果我们想扩展系统，我们只需要相应地扩展神经网络。 Behler 和 Parrinello[4]提出了实现这一目标的一种方法。

2. 保持对称性。 除了通常的平移对称性和旋转对称性外，还具有置换对称性：如果我们重新标记一个铜原子系统，它的势能不应该改变。 它在神经网络模型的准确性方面有很大的不同，是否考虑到这些对称性（见[23]和图 3）。

解决对称性问题的一种非常好的方法是将神经网络模型设计为两个网络的组成：嵌入网络和拟合网络。 嵌入网络的任务是表示足够多的保持对称性的函数，以便输入到拟合网络[39]中。

通过适当地解决这些问题，我们可以提出非常满意的基于神经网络的 V 表示（见图 4）。 这种表示被命名为深度势[23,39]，基于深度势的分子动力学被命名为 DeePMD[38]。 正如最近在[30,25]中所证明的那样，DeePMD 结合了状态

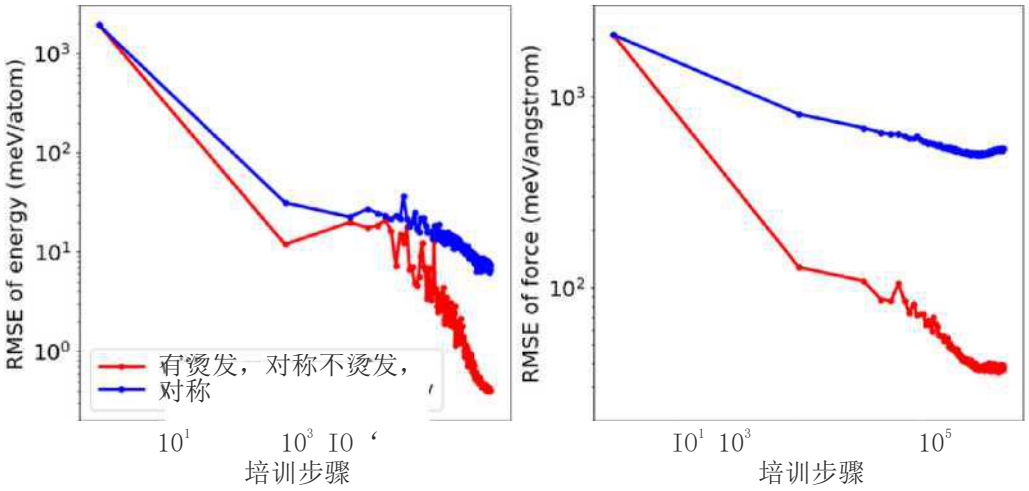


图 3：对称保存对测试精度的影响。 红色显示是穷人强加对称的结果（解释见正文）。 可以看出，测试精度大大提高。 经张林峰许可转载。

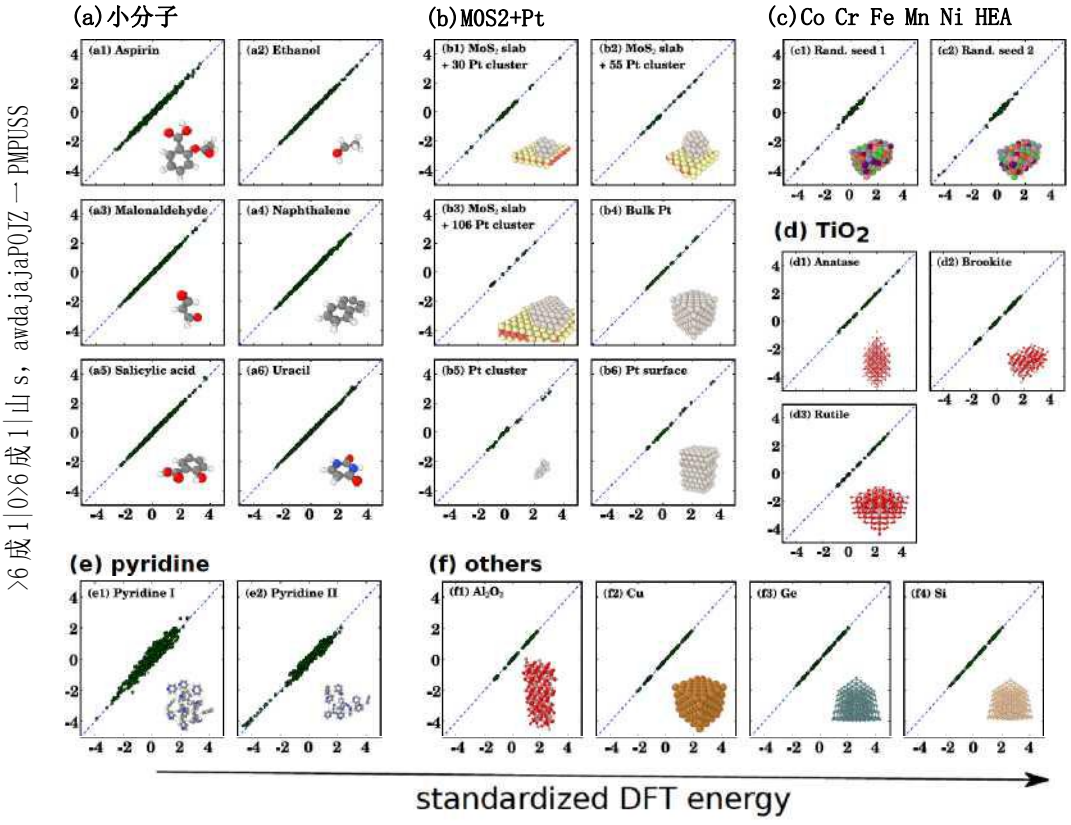


图 4：各种系统的深势测试精度。 经张林峰许可转载。 另见[39]。

艺术高性能超级计算机，可以帮助增加系统的大小，一个人可以建模的从头精度 5 个数量级。

3 基于机器学习的科学计算高维问题算法

3.1 随机控制

[20]介绍了机器学习在科学计算中解决高维问题的第一个应用。 随机控制由于其与机器学习的相似性而被选择为第一个例子。 考虑离散随机动力系统：

$$s \quad \text{不+我=}^{s}\text{不+bt0，at)+}$$

这里 st 和 at 分别是时间 t 的状态和控制，&是时间 t 的噪声。我们的目标是解决：

$$\text{}^{m}\text{在 I 阳}\}\{\text{Ect(ST t)+ct(s t)}\}\{\text{心=o 不=0} \tag{2}$$

在一组反馈控件中：

$$a \quad \text{不=在}^{(s}\text{t)-} \tag{3}$$

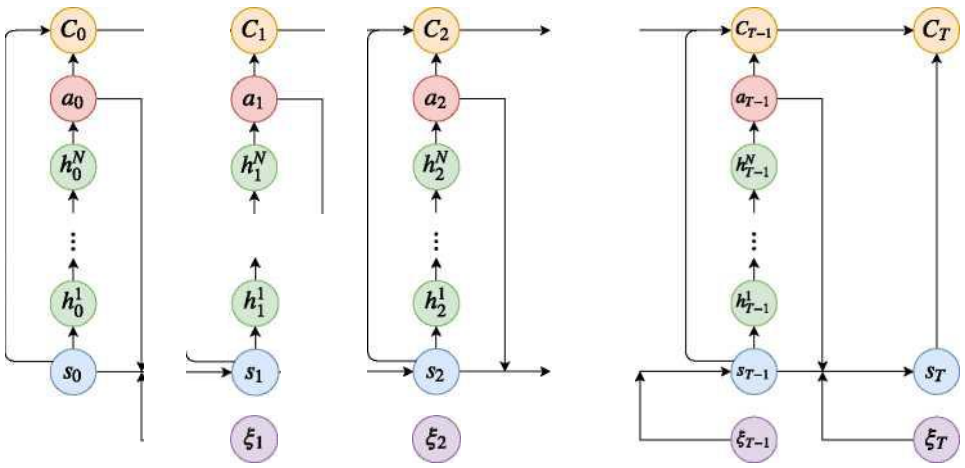
我们用神经网络模型逼近函数：

$$\text{在}^{(s)} \ \&\text{在}^{(s}\text{GT)，t=}^{0,}\text{，…' t-1} \tag{4}$$

优化问题（2）则变为：

$$t\text{-}1$$
$$\text{偌虬}^{e}\{\&\}\{\text{£CT\&AT^TGT)+CT(ST)}\}\text{。} \tag{5}$$

与标准监督学习的情况不同，这里我们有 T 组神经网络要同时训练。 网络架构如图 5 所示



f=0 *不=1* *不=2* ..... *t=T—1 t=T*

图 5：求解离散时间随机控制的网络体系结构。 整个网络总共有(N+1)T 层，涉及自由参数，需要同时优化。 每一列（除&）对应于一个子网络在 T.转载得到了杰群汉的许可。 另见[20]。

与机器学习的标准设置相比，可以看到一个明确的类比，其中（1）对残差网络起作用，噪声{&}起着数据的作用。 事实上，随机梯度下降(SGD)可以很容易地用于解决优化问题（5）。

该算法的应用示例如图 6 所示，用于多个设备的储能问题。 这里 n 是设备的数量。 更多细节，我们参考[20]。

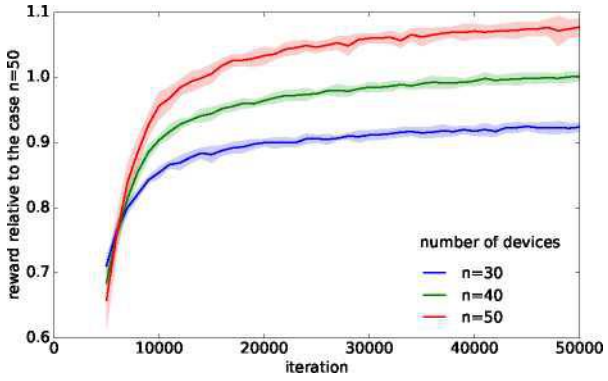


图 6：储能问题的相对奖励。 控制函数的空间为 $R^{n+2}t\ r^{3n}$ 对于 n=30，40，50，具有多重平等和不平等约束。 经韩洁群许可转载。 另见[20]。

3.2 非线性抛物线 PDE

考虑抛物型 PDE 的形式：

$$DDU+1AA^{t}*VXu+\ P-Vu+f(a^{t}v\hat{}=0,$$
$$u(T,\ x)=g(x)$$

我们研究的是一个终端价值问题，而不是初始值问题，因为我们考虑的主要应用之一是金融。 为了开发基于机器学习的算法，我们首先将其重新定义为一个随机优化问题。这可以使用反向随机微分方程(B SDE)[33]来完成。

$$\inf_{Y\text{不}，\ f\text{不}<T} \text{例如}|g(XT)-Yt|^{2}, \tag{6}$$

$$s.。\ X=e+\int_{0}^{T}\text{诚 s，Xs)ds}+\int_{0}^{T}E(s,\ Xs)d\ W_{s}, \tag{7}$$

$$y_{t}=\int_{0}^{T}\text{小时，}\ Z)DS+\&)^{t}d\ W_{s}. \tag{8}$$

可以看出，这个问题的唯一极小值是 PDE 的解有：

$$你=你(t,\ Xt)\quad \text{还有}\quad ZT=a^{t}(t,\ Xt)Vu(t,\ Xt)。 \tag{9}$$

利用这种公式，可以沿着以下路线开发一种基于机器学习的算法，采用前面[10,21]讨论的随机控制问题的思想：

时间离散后，近似未知函数

$$xo1u(0,\ xo)\qquad \text{还有}\quad x_{t^{n}\ 1a^{t}}(TJ,\ XTVu(TJ,\ X_{t}.)$$

通过前馈神经网络明和

- 使用 BSDE，构造一个逼近 u，它接受路径{Xt}o<n<N
  和{WT}o<n<N 作为输入数据，并给出最终输出，用 u({Xt)表示}o<n<N，{W 万亿}o<n<N 作为 u(t^，Xt 的近似值).

- 在 u 与给定终端条件之间匹配的误差定义了预期损失函数

$$W)=E[|g(XtN)-u(\{Xtn\}o<n,\ \{Wtn\}o<n,\ N)$$

这种算法被称为深度 BSDE 方法。

作为应用，让我们首先研究一个随机控制问题，但我们现在使用 Hamilton-Jacobi-Bellman(HJB)方程来解决这个问题。 考虑维数 d=100 的著名 LQ G（线性二次高斯）问题：

$$d\ Xt=2y/\backslash mtdt+a/2dwt, \tag{10}$$

成本函数：J({m,}o<,<,) = m,腭 DT+g(X,)]。 相应的给出了 HJB 方程

杜

$$DT+\text{亩}^{a}{}_{xr}\ \backslash\ |\ =\ 0 \qquad (11)$$

利用 Hopf-Cole 变换，可以用形式表示解：

$$u(t,\ x)=-y\ln \mathrm{EXP}(-Ag(x+V2Wt-t))。 \qquad (12)$$

这可以用来校准 准确性 的 深 B SDE 方法。

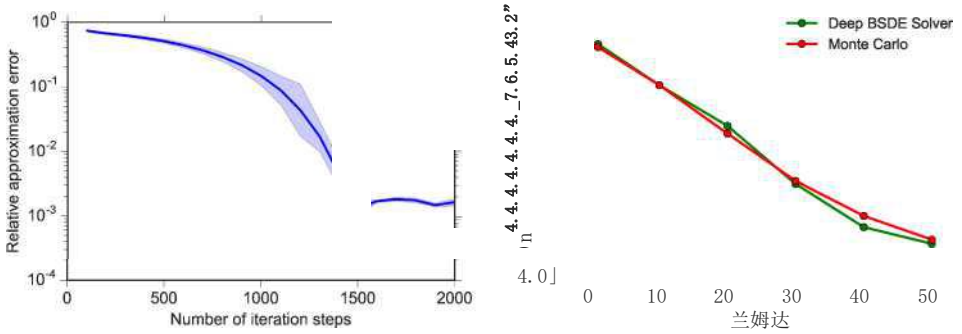


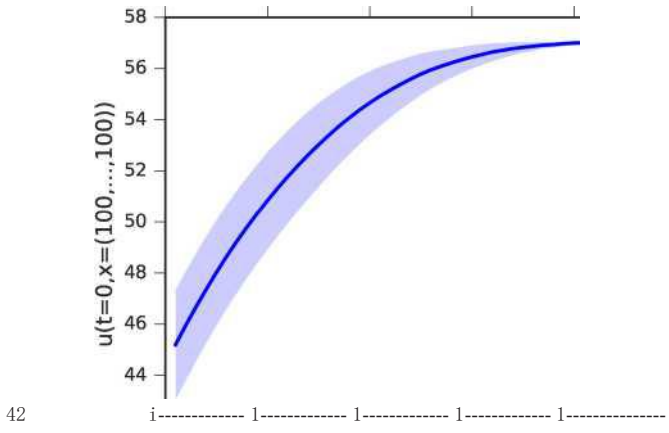
图 7：左：当人=1 时，u(t=0，x=（0，...，0）的深 B SDE 方法的相对误差，在 330 秒的运行时达到 0.17。右：针对不同人的最优成本 u(t=0，x=(0，...，0。经韩洁群许可转载。 另见[21]。

作为第二个例子，我们研究了默认风险的 Black-Scholes 方程：

杜

$$+Au-（1-6）Q(u(t,\ x)u(t,\ x)-ru(t,\ x)$$

其中 Q 是一些非线性函数。 这种建模默认风险的形式是针对低维情况提出和/或在文献中使用的(d=5，参考见[21])。 在 d=100[21]的情况下，对这一问题采用了深 B SDE 方法。



42

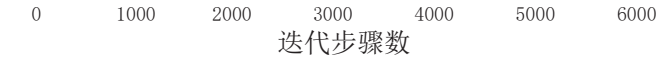


图 8:默认风险在 d=100 的 Black-Scholes 方程的解。深度 BSDE 方法在运行时 617 秒内实现了 0.46%大小的相对误差。 经韩洁群许可转载。 另见[21]。

深 B SDE 方法已应用于定价篮子选项、利息评级依赖选项、Libor 市场模型、百慕大交换、障碍选项（参考见[17]）。

3.3 气体动力学动力学方程的动量闭包

气体的动力学可以用著名的 Boltzmann 方程非常精确地建模：

$$DF+v \cdot Vxf=-Q(F),\ ve\ R^3,\ x\ e\ Q\ c\ R^3, \tag{13}$$

其中 f 是相位空间密度函数，g 是 Knudsen 数：

$$\frac{\text{意思是自由的道路}}{\text{宏观长度}}$$

q 是模拟气体粒子碰撞过程的碰撞算子。 当 g《1》时，这可以用欧拉方程近似：

$$d\ U+Vx \cdot F(U)=0, \tag{14}$$

在哪里

$$u = \int (p,\ p^{你,\ E})T,\ p=I\ f\ dV,$$

还有

$$f(u) = (p^{你,}\ p^{u}\ \otimes u+pI,\ (E+p)u)^{t}$$

通过将玻尔兹曼方程投影到所涉及的低阶矩上，并利用分布函数 f 接近局部 Maxwellian 的 ansatz，可以得到欧拉方程。

图 9：气体动力学的不同状态。 经韩洁群许可转载。 另见[22]。

当£不小的时候会发生什么？ 在这种情况下，一个自然的想法是寻求一些推广欧拉方程使用更多的矩。 这个程序是由哈罗德·格拉德发起的，他利用{1，v，(v-u)®(v-u)，\v-u|的时刻构造了著名的 13 矩系统 ²(v-u)}。 这一行工作遇到了几个困难。 首先，不能保证得到的方程是合适的。其次，总是有“闭包问题”：当将玻尔兹曼方程投影到一组矩上时，总是有一些术语涉及到所考虑的矩集之外的矩。 为了获得一个封闭的系统，人们需要以某种方式对这些术语进行建模。 对于欧拉方程，这是使用局部麦克斯韦近似来完成的。 当色不小时，这是准确的，但当色不小时，就不再是这样了。 目前还不清楚应该用什么来代替。

在[22]，Han 等人开发了一种基于机器学习的矩量法。 总体目标是构造一个均匀精确（广义）弯矩模型。 方法包括两个步骤：

1：通过自动编码器学习一组最优广义矩。 在这里，通过最优性，我们意味着广义矩集合保留了关于原始分布函数的最大信息量，并且可以用于以最小的精度损失恢复分布函数。 这可以做如下：找到编码器中和解码器小，从 U，W 恢复原始 f

$$W=\hat{}(f)=/wfdv，\ 收\ U，\ W)(v)=h(v；\ U，\ W)。$$

$$^{尽量减少}W，\ H_t e||^f 以中（/）||^2.$$

U 和 W 构成了一组广义水动力变量，我们将用来模拟气体流动。

2：学习 PDE 中投影 PDE 的通量和源项。 通过在这组（广义）矩上正式投影玻尔兹曼方程，可以得到 U 和 W 的有效 PDE。 这给了我们一组形式的 PDE：

$$\begin{aligned} &e\ DT\ U+Vx-F(U，\ W；\ £)=0，\\ &我知道\ ,W+Vx-G(U，\ W；\ £)=R(U，\ W；\ £)。 \end{aligned} \tag{15}$$

其中 F(U，W；£)=/vUfdv，G(U，W；£)=vWfdv，R(U，W；£)=£⁻¹ *f WQ(f)dv*。 我们现在的任务是从原始动力学方程中学习 F，G，R。

同样重要的问题是（1)获得最优数据集，(2）强制物理约束。 这里两个值得注意的物理约束是（1)守恒定律和(2）对称性。 在这种方法中，自然会尊重保护法律。 关于对称性，除了通常的静态对称性外，现在还有一种新的动态对称性：伽利略不变性。 这些问题都在[22]讨论。 我们还参考[22]的数值结果的模型，以获得这种方式。

4 机器学习的数学理论

虽然基于神经网络的机器学习已经展示了广泛的非常令人印象深刻的成功，但它也获得了“黑魔法”的声誉，而不是坚实的科学技术。 这是由于（1)我们对其成功背后的基本原因缺乏基本的理解；(2)这些模型和算法的性能对超参数的选择非常敏感，例如网络的体系结构和学习速率；(3）

一些技术，如批量归一化，似乎是一种黑魔法。

为了改变这种情况，我们需要（1)提高我们对基于神经网络的模型和算法成功背后的原因和脆弱性的理解；(2）寻找方法来制定更稳健的模型和设计更稳健的算法。 在本节中，我们讨论第一个问题。 下一节将专门讨论第二个问题。

下面列出我们需要解决的最基本问题：

- 为什么它在如此高的维度上工作？
- 为什么简单梯度下降用于训练神经网络模型？
- 过度标准化是好是坏？
- 为什么神经网络建模需要如此广泛的参数调优？

在这一点上，我们尚未对所有这些问题有明确的答案。 但一些连贯的画面开始浮现。 我们将重点讨论监督学习的问题，即使用有限数据集逼近目标函数。 为了简单起见，我们将把自己限制在物理感兴趣的领域是 X=[0,1]的情况下.

4.1 介绍基于神经网络的监督学习

监督学习的基本问题如下：给定输入输出数据对的自然数 neN 和序列{(Xj)=(xj，f*(Xj)，je{1，2，...，n}，我们希望尽可能准确地恢复目标函数 f。 我们将假设输入数据{xj，je{1，2，...，n}，是从 R 上的概率分布中采样的.

第一步。 选择一个假设空间。 这是一组试验函数，其中 m，n 是 H 的维数。 人们可以选择分段多项式或小波。

在现代机器学习中，最流行的选择是神经网络函数。 双层神经网络函数（一个输入层，一个通常不计算的输出层，一个隐藏层）采取这种形式

$$\text{我£a。（凹，}^{x}\text{））}\quad (16)$$

其中 a：R-R 是一个固定的标量非线性函数，其中 O={(AJ，Wjare 是要优化的参数（或训练的参数）。非线性函数 a：RTR 的一个流行例子是 ReLU（校正线性单元）激活函数：a(Z)=max{z，O}，对于所有 zeR。我们将把注意力限制在这个激活函数上。 粗略地说，如果一个人多次组合两层神经网络函数，则得到深度神经网络(DNN)函数。 其中一类重要的 DNN 模型是残差神经网络(ResNet)。 它

$$\text{齐+我}\quad \text{£aw((2)，凹，i)，j=i}\quad \textit{动物园=VX,}\quad (17)$$

们与离散常微分方程非常相似，并采取这种形式

对于 le{0，1，...，L-1}，其中 L，M，N。这里的参数是 O=(a，V，（%]），(wj，ij)。 ResNet 是真正深层神经网络模型的首选模型。

第二步。 选择一个损失函数。 损失函数的选择的首要考虑是拟合数据。 因此，最明显的选择是 L²损失：

$$\text{}^{r}\text{n}^{(}/)=\text{n£}|/^{(+}\text{j)}^{-\text{是的}}\text{ji}^{2}=\text{n£}|/^{(+}\text{j)}^{-}/\ *^{(+}\text{j}^{)|2}.\quad (18)$$

这也被称为“经验风险”。 有时我们还添加正则化项。

第三步。 选择一个优化算法。 机器学习中最流行的优化算法是不同版本的梯度下降(GD)算法，或其随机模拟，随机梯度下降(SGD)算法。 假设我们目标最小化的目标函数是形式的

$$F（0）=\text{或[l(0，O]。}\quad (19)$$

最简单的 SGD 迭代形式

$$\textit{0k+我=0k-”桐 l(0k，&k)，}\quad (20)$$

对于 keNo，其中{&，keNo={0，1，2，...}是一个 i.i.d 序列。 从分布 v 和 n 中采样的随机变量是在迭代过程中也可能发生变化的学习速率。 相比之下，GD 采取的形式

$$\text{+我=0k-西唤 3[l(0k，£)]。}\quad (21)$$

显然，这种形式的 SGD 可以适应形式（18）的损失函数，它可以看作是一个期望，v 是训练数据集上的经验分布。 这种 DNN-SGD 范式是现代机器学习的核心。

4.2 近似理论

逼近函数最简单的方法是使用多项式。 对于多项式逼近，有两种定理。 第一个是 Weierstrass 定理，它断言连续函数可以用紧域上的多项式统一逼近。 其次是泰勒定理，它告诉我们收敛速度取决于目标函数的光滑性。

利用神经网络理论中的术语，Weierstrass 定理是“普遍逼近定理”(UAT)。 这是一个有用的事实。 但泰勒定理更有用，因为它告诉我们一些关于收敛速度的东西。 近似理论中使用的泰勒定理的形式是直接和逆近似定理，它断言给定的函数可以用具有特定速率的多项式逼近，当且仅当该函数的某些规范是有限的。 这个特殊的范数，它测量函数的规律性，是表征这个近似方案的关键量。 对于分段多项式，这些规范是一些贝索夫空间规范[36]。 为了 L²一个典型的结果如下：

$$_{f}\text{蜡 11/-/m\^L2(X)<C,}\hbar^{a}\text{\^F||H“X)} \tag{22}$$

给你 ª 代表第一阶 Sobolev 范数[36]，m 是自由度的数目。 在规则网格上，网格大小由

$$\text{m\_Ud} \tag{23}$$

需要注意的一件重要的事情是，（22）中的收敛速度受到 CoD 的影响：如果我们想将误差减少一个 e 因数，我们需要将 m 增加一个因数 me_ᵈ 如果=1。 对于 d=100，根据机器学习的标准，它不是很高的维数，这意味着我们必须将 m 增加一个 e 的因子⁻¹⁰⁰。 这就是为什么多项式和分段多项式在高维中不有用的原因。

另一种理解这一点的方法如下。 度 p 在维数 d 中的单项式个数为 C：*，。对于 d 和 p 的大值来说，这增长非常快。

在高维中我们应该期望什么？ 我们可以借鉴的一个例子是用于集成的蒙特卡罗方法。 考虑近似问题

$$^{I}\ (g)=$$

使用

$$\text{我(g)}\ =\text{吗？}\ ^{g(xj)}$$

其中{x je[m]}是直接计算给出的概率分布川 A 一组 i.i.d 样本.

$$^{e(i}(g)-m(g))^{2}= \qquad ,\ ^{var}(g)=^{e}xg^{2(x)}\ —\ (^{e}xwg(^{X}))^{2}$$

这个确切的关系告诉我们两件事。 （1）蒙特卡罗积分的收敛速度与维数无关。 （2）误差常数由被积方差给出。 因此，为了减少误差，必须做方差减少。

如果我们使用基于网格的正交规则，精度也会受到 CoD 的影响。

通过更复杂的选择{xj，je[m]}的方法，例如使用基于数论的求积规则，可以提高蒙特卡罗率。但这些通常会导致收敛速度的 O(1/d)提高，并且随着 dTx 的减小而减小。

基于这些考虑，我们的目标是找到满足：的函数近似：

$$\underset{FCM}{} \quad ||f-f*=\text{釦疵 W} \quad \text{fCM JII£即)m}$$

自然的问题是：

我们如何做到这一点？ 即我们应该选择什么样的假设空间？

什么应该是“规范”||·||*(与 H 的选择有关 „)？ 在这里，我们把规范放在引号中，因为它不必是真正的规范。 我们只需要它控制近似误差。

关于第一个问题，让我们举一个说明性的例子。

$$f(x)=/a\ (3)\ e\ '(3,\ ^{x})\mathbf{d3},\qquad \frac{1}{m}\sum_{j} a(\omega_j)e^{i(\omega_j,x)}$$

考虑函数 f 及其近似 fm 的傅里叶表示(例如基于 FFT 的)：

这里{3j}是一个固定的网格，例如。均匀的网格。对于这个近似，我们有 $\mathbf{f-f}_{m}||L2(X)<C_{o}^{m\sim a/d\ \sim r}||ff^{a}\mathbf{(x)}$ 患有 CoD。

现在考虑替代代表

$$f(x)=/a\ (3)\ e^{-(3,\ x)}n(d3)=a(w)e^{-(\sim,\ x)} \tag{24}$$

其中 n 是概率分布。现在要近似 f，使用蒙特卡罗是很自然的。设{凹}为 i.i.d。样本 n，FM(X)=

$$^{a}(\hat{}j)e^{\text{我(WJ}},\ x)$$

那我们就有了

$$E|f(x)-f_{m}(\text{十})|^{2}=$$

这种近似不受 CoD 的影响。 请注意，FM(X)=和工旗 1AJA(^JX)只不过是一个具有激活函数 a(Z)=的两层神经网络(这里是 AJ= 3j))。

我们认为，这个简单的论点确实是为什么神经网络模型在高维中表现如此出色的核心。

现在让我们来举一个神经网络模型近似理论的具体例子。 我们将考虑两层神经网络。

$$=\ \{f_{m}^{(x)}=\ \underset{m}{\underline{\rangle :}}\ a_{jb}{}^{(w_j x)}\},\ \mathbf{0}\qquad \{(a_{j,}{}^{w}{}_{j}),\ je[m]\}$$

考虑函数 f：X=[0,1]ᵈ以下表格的 TR

$$f(x)= \quad AA(W^{t}x)p(da,\ dw)=E_{(a,w)\sim p}[aa(w^{t}x)],\ x$$

其中 Q=R¹x R^{r+1}，p 是 Q 上的概率分布。 定义：

好吧ᶠ二 B=厲(ᴱp 同刎 2^{1})$^{1/2}$

其中 Pf：={p：f(X)=Ep[aa(W¹x)]}。 这被称为巴伦规范[2,12,13]。 空间

$$B=\{f\ e\ C^{0}||f<x\}$$

被称为 Barron 空间[2,12,13]（另见[3,26,14]）。

类比经典逼近理论，还可以证明一些直接逼近定理和逆近似定理[13]。

. *定理 1（直接逼近定理）如果||f||b<x，那么对于任意整数 m>0，存在两层神经网络函数 fₘ就这样*

好吧ᶠ − f | m l^{2}(x)w

定理 2（逆逼近定理）设.

NC =ᶠ{m E a s x) 1 米 1 米 我瓜|侦 m^{c2}, +}

*设 f*为连续函数。 假设存在一个常数 C 和一个函数序列 fmeNC，使得*

f_{m} (x) 1 1 *(X)

对于所有的 xeX。然后在 Q 上存在概率分布 p，这样*

$$f^{*}(x)=/aa(w^{t}x)p^{*}(da,\ dw),$$

对于所有 x、e、x 和|//||b、m、C。*

4.3 估计误差

我们必须担心的另一个问题是训练数据集之外的机器学习模型的性能。 这一问题也出现在经典近似理论中。 图 10 说明了在均匀网格上多项式插值的经典 Runge 现象。 可以看出，远离网格点，插值的误差可能很大。 这是我们希望避免的情况。

我们在实践中所做的是尽量减少训练误差：

$$Rn\ (0)=n\ E(f(xj,\ 0)-f^{*}(xj))^{2}$$

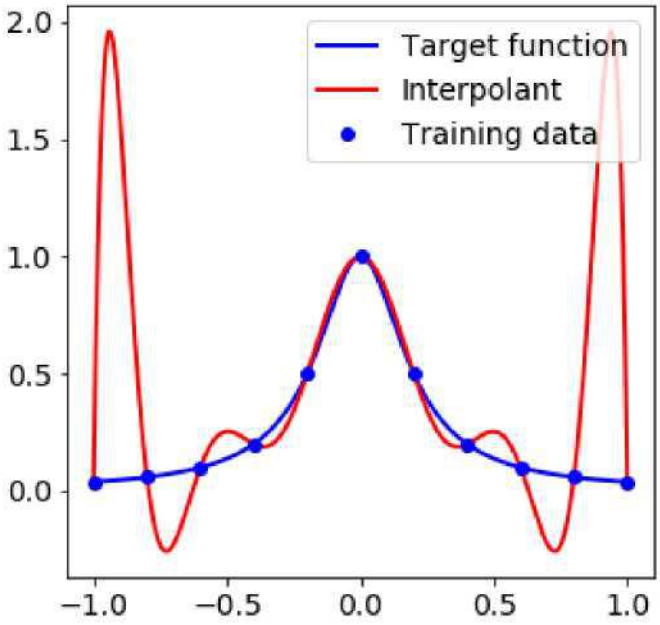


图 10：龙格现象：f*(X)=搭训 Reproduced 得到潮马的许可。

但我们对测试错误感兴趣，这是人口风险的抽样版本：

叫)=E 项 f f*(x)2

问题是我们如何控制这两个错误之间的差异。

这样做的一个方法是使用拉德马赫复杂性的概念。对我们来说，这里的重要事实是，拉德马赫的复杂性控制了训练和测试错误之间的差异（也称为“泛化差距”）。 事实上，让 H 是一组

吃 例[h(x)]- Rads(H)
母

函数，S=(xi，x_2，十$_n$)成为一个数据集。 然后，直到对数项，我们有

$$径(H)二一 Etn \frac{吃}{母}{}^{n}_{我=} \&h(x《) \quad (25)$$

其中，H 对 S 的拉德马赫复杂性被定义为

在那里{&}n=我是 i.i.d 的。 随机变量取值±1，概率相等。

然后问题变成了约束一个假设空间的拉德马赫复杂性。 对于巴伦空间，我们有[2]：

定理 3。 设 f q={f∈B，||f||b<q}。 我们就有了

雷达(FQ)<2Q^H^n(*其中 n=|S，数据集 S 的大小。*

4.4 正规化模型的先验估计

考虑正则化模型

l_n（0）=r_n（0）+X\l 吳 V 昭 Ip，R=argmin L 拉(0)

其中路径范数由：定义： (26)

1 米 i/²

(m 匚|AK|²|” k J

[12]证明了以下结果：

定理 4。 假设 f：XT[0,1]eB。 存在常数 Co，因此对于任何常数 6>0，如果 A>Co，那么概率至少为 1-6 超过训练数据集的选择，我们有*

如)<典+A|f*，J IOG(2d)+”1 0g(1”)+1 0g(n).

m n n

类似的近似理论和先验误差估计已经被证明适用于其他机器学习模型。 以下是对这些结果的简要总结。

随机特征模型：相应的函数空间是再现核 Hilbert 空间(RKHS)。

剩余网络(Res 网)：相应的函数空间是在[13]中引入的所谓流动诱导空间。

多层神经网络：相应的函数空间的候选是[15]中引入的多层空间。

真正重要的是控制近似误差和泛化间隙的“规范。 这些量是为相应空间中的函数定义的。 在逼近定理和 Rademach 复杂度估计到位后，人们可以很容易地证明正则化模型的以下类型的定理：直到对数项，正则化模型的最小化子满足：

RF<y+平

mn

其中 m 是自由参数的数目，n 是训练数据集的大小。 请注意，对于多层空间，[15]中证明的结果不是那么尖锐。

我们只讨论了假设空间的分析。 还有许多其他问题。 我们参考[19]来更多地讨论目前对基于神经网络的机器学习的理解。

从连续的角度 5 机器学习

现在我们转向机器学习的替代配方。 由于 PDE 的情况，我们希望首先在连续设置中制定机器学习，然后离散化，以获得具体的模型和算法。 这里的关键是，我们提出的连续问题应该是好的数学问题。 对于 PDE，这是通过要求它们“适口”来实现的”。 对于变分问题，我们要求问题在某种意义上是“凸”的，并且是较低的半连续的。 这些要求的要点是确保问题有唯一的解决

方案。 直观地说，对于机器学习问题，“好”意味着变分问题应该有一个简单的景观。 如何准确地表述这一点，是未来研究的一个重要问题。

正如[16]所指出的，连续配方的关键成分如下：

- 功能的表示（作为期望）
- 制定变分问题（作为期望）
- 优化，例如。 梯度流动

5.1 职能的表述

[16]中考虑了两种表示：基于积分变换的表示和基于流的表示。 最简单的基于积分变换的表示是（24)的推广）：

$$f(x;\ 0)= \quad a(w)a(w^{f}x)n(dw)$$

$$a(w)a(w^{t}十)$$

$$=耻“paa(w^{t}十)$$

$$p^{©(十,\ u)}$$

这里 0 表示模型中的参数：0 可以是 a（·）或 Prob 分布 n 或 p。

这种表示对应于两层神经网络。 [15]对多层神经网络进行了推广。

接下来我们转向基于流的表示：

$$与=时(w,\ T)a(w^{r}z) \tag{27}$$

博士

$$=E(a,\ w)阵\ AA(w^{r}z) \tag{28}$$

$$=阵©(z,\ u),\ z(0,\ x)=x \tag{29}$$

$$/^{(x,\ 0)} = {}^{1(1,\ x)}$$

在此表示中，参数 0 可以是$\{a_{t}\}$或$\{n_{t}\}$或$\{p_{t}\}$

5.2 随机优化问题

随机优化问题类型有：

随机算法是机器学习中的关键组成部分，可以很容易地解决这些问题。 例如，代替梯度下降算法：

$$DK+我= \quad - \quad vg(0, w)=e_k- \sum_{j=i}^{n} g(°, wj)$$

人们可以使用随机梯度下降：

$$+我=觅-g(们 Wk)$$

其中{wk}是从 v 采样的一组随机变量。

以下是现代机器学习中出现的随机优化问题的一些例子：

监督学习：在这种情况下，人口风险最小化

$$R（/）=Ex 睥(f(x)-f*(x))2$$

- 量子多体哈密顿量的特征值问题：

$$I(4)-箜业-e 认如(如-10(X)3x$$
$$^{I}(“ — (© ©) — ^{e}xq© \quad ©(x)2 \quad , \quad ”^{(d\ X)} — / |^{\wedge(x)}| \ d\ X$$

这里 H 是量子系统的哈密顿量。

$$升(\{a_t\}^{}\ 不=不\mathbb{E}\{\sum_{t=0}^{T-1}$$

- 随机控制问题：

将前面讨论的表示替换为随机优化问题的这些表达式，我们得到了我们需要解决的最终变分问题。

人们可以直接对这些变分问题进行离散化，然后使用一些优化算法求解离散化问题，也可以写下一些优化算法的连续形式，通常是梯度流动动力学，然后对这些连续流动进行离散化。 我们将讨论第二种方法。

5.3 优化：梯度流动

为了写出梯度流动的连续形式，我们从统计物理中得到了一些启示。 以监督学习为例。 我们认为人口风险是“自由能”，根据 Halperin 和 Hohenberg 的[24]，我们将参数分为保守和非保守两种。 例如，a 是一个非保守参数，n 是保守的，因为它的总积分必须是 1。

对于非保守参数，如[24]所建议的，可以使用“模型 A”动力学：

$$\frac{da}{DT} \quad \frac{b\ R}{巴}$$

这只是通常的 L^2 梯度流动。

对于 n 这样的保守参数，应该使用“模型 B”动力学，其工作原理如下：首先定义“化学势”

$$\frac{b\ R}{十亿}$$

由化学势得速度场 v 和电流 J：

$$J=NV,\ v=-W$$

连续性方程给出了梯度流动动力学：

这也是 Wasserstein 度量下的梯度流。

5.4 分散梯度流动

为了获得实际的模型，需要对这些连续问题进行离散化。 第一步，利用训练数据用经验风险代替人口风险。

更重要的问题是如何离散参数空间中的梯度流。 参数空间具有以下特点：（1）它具有简单的几何形状-与实际空间不同，它可能具有复杂的几何形状。 （2）它通常也是高维的。 由于这些原因，参数空间离散最自然的数值方法是粒子法，这是蒙特卡罗的动态版本。 雾化粒子法可能有助于提高性能。 在相对较低的维数中，由于参数空间的几何形状相对简单，人们也可以考虑谱方法，特别是谱方法的一些稀疏版本。

例如，基于积分变换的表示的保守流的离散化。 用表示法：f(x；0)=E(„„)¸AA(Wx)，梯度流动方程变为：

DTP=▽（ˆ▽五），　　五=bR(30)

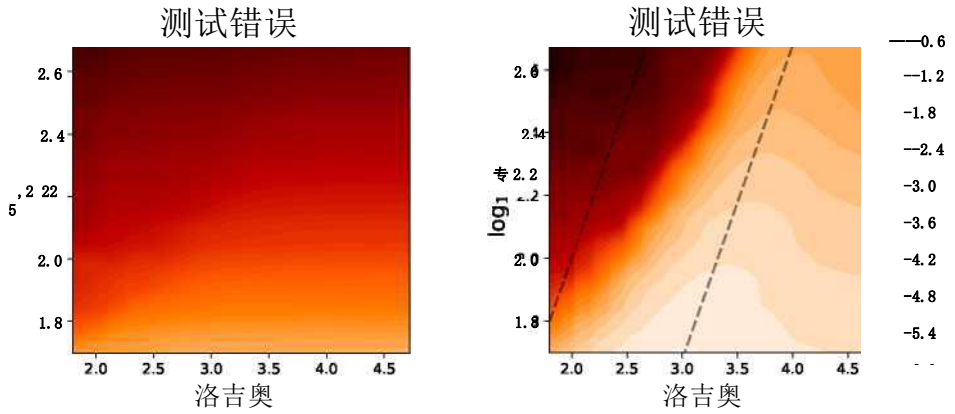


图 11：（左)连续视点；(右）常规 NN 模型。 目标函数是单个神经元。 经雷武许可转载。

粒子法离散化是基于：

$$\text{pam } E8(aj(t),\ w(t)=m\ E \text{ 知（*）}$$

j j

其中 Uj(T)=(Aj(T)，w(T))。 可以证明在这种情况下，（30）减少到

$$-^{1(u}\ 我，。。。。，^{u}m)$$

在哪里

$$^{I(u}1,\ \bullet\bullet\bullet\ ,\ ^{u}m) = \ m),\quad ^{u}j = \ j,\ ^{w}j),\quad m^{(x)} = \underline{\rangle :}\ ^{a}jb^{(w}j^{x)}$$

这正是“缩放”两层神经网络的梯度下降。 在这种情况下，连续公式也与两层神经网络[7,32,34,35]的“平均场”公式相吻合。

在“缩放”两层神经网络前面的缩放因子 1/m 实际上是相当重要的，对网络的测试性能有很大的影响。 图 11 显示了具有和不具有此缩放因子的两层神经网络模型测试误差的热图。 目标函数是简单的单神经元函数：f*(X)=a(Xi)。 重要的观察是，在没有这种缩放因子的情况下，测试误差在神经网络模型表现为相关随机特征模型的相位和显示比随机特征模型更好的性能的另一个相位之间显示出“相变。 这样的相变是选择正确的超参数集的原因之一，这里的网络宽度 m 是如此重要。 然而，如果使用缩放形式，则避免了这种相变现象，[31]性能更健壮。

5.5 基于流量的表示的最佳控制问题

基于流的表示自然会导致控制问题。 这种观点在机器学习中已经显式或隐式地使用了相当长的一段时间(例如，参见

[27])。 例如，反向传播算法是基于控制理论的算法的一个例子。 另一个最近的例子是最大基于原理的算法的发展，第一次介绍在[28]。尽管取得了这些成功，但我们认为仍然有很大的空间使用控制理论的观点来开发新的算法。

我们认为基于流的表示形式稍微更一般

新西兰

=博士，臥 z，u)，　　z(0，x)=x

其中 z 是状态，Pt 是时间 r 的控制。我们的目标是在{p}上最小化 R}

铂})=　　(/(x) — / *(x))2 =　(/(x) — *(+)Σ　　如

就像以前一样

$$/(x) = 1^{\text{兹}(1,\ x)} \tag{31}$$

这个控制问题的一个最重要的结果是 Pontryagin 的最大原理(PMP)。 为了说明这一结果，让我们将哈密顿量 H：xP2(Q)：TR 定义为 H(z，p，讪=Eu=[p臥 z，u)]。

Pontryagin 的最大原理断言控制问题的解必须满足：

$$Pt=\text{argmax}p^{e}x^{[h}(^{z}\text{不}\ ‘+,\ PT\ ‘+,\ p)^{]},\ Vr\ g[0,1], \tag{32}$$

每个 x，{(z*+}由前向/后向方程定义：dz，

你=VPH=Eus(t)[0(zT+，u)]dp，

具有边界条件的 DR：

$$-Vz\ H=Eu*(。[VT\ 心，u)pt+], \tag{33}$$

$$z_{0}^{t,\ x} = {}^{x}$$

$$pl^{+} = {}^{-2(}/(+;\ P(▪;\ t))^{-}/*(+)1 \tag{34}$$

(35)

对于平稳点，Pontryagin 的最大原则略强于 KKT 条件，因为（32）是最优性而不是临界性的陈述。 事实上，（32）也适用于参数是离散的，这在[29]中已经被用来为这种情况开发有效的数值算法。

在 PMP 的帮助下，也很容易写出优化问题的梯度下降流。 形式上，可以简单地写下每个 r：（32）的梯度下降流：

$$DT\ Pr(u,\ t)=\nabla\cdot(p_{r}\ (u,\ t)W(u;\ p),\ Vt\ g[0,1], \tag{36}$$

在哪里

$$五(u\ \ p\ ;\ ) = {}^{e}x[\tilde{}\hat{}p(^{z}\text{不}\ ‘^{x,\ p}t\ ,\ x,\ p\ \ \text{不}(▪;\ 功$$

和{(z*+}由前向/后向方程定义。

为了离散梯度流，我们可以简单地使用：

- 前向欧拉表示 t 变量中的流，步长为 1/L(L 是 t 变量中网格点的数目)；
- 梯度下降动力学的粒子法，每个 t 格点有 M 个样本。

这给了我们

$$^{ZT}\text{+我=}^{\text{中兴通讯，}x}\text{+巾伝件，}^{UJ(t)，1}_{j=i} = 0,\ldots,\ ^{1} - 1 \tag{37}$$

$$P\text{算=P；+1+LMEVz©(W+1，}^{UJ}_{j=1}\text{+一}^{(t)})\text{p；+i，}\ ^{1} = {}^{0}\text{，...，}\ ^{l-1} \tag{38}$$

$$\text{衅=-Ex[大众臥才，UJ(T)p 門。} \tag{39}$$

这将恢复（缩放）ResNet 的 GD 算法(反向传播：

$$^{z}\text{我+我=}^{z}\text{+巾(2)}^{uz}).$$

由于因子 1/(LM)的存在，我们称之为“缩放”ResNet)。

本着类似的精神，人们也可以使用 PMP[28]获得算法。采用控制理论中的术语，这种算法被称为“逐次逼近法”(MSA)。 基本 MSA 如下：

初始化：0° cU

关于 k=0,1,2，：

解决 d

解决

DP。 $d\ T$ -Vz H(球 pk、郎、pi=-2(/(x；俨)-/*(x))1

$$\sim\!\!=_{P}^{H(Z\quad)},\qquad {}^{z}\text{。}=^{X}$$

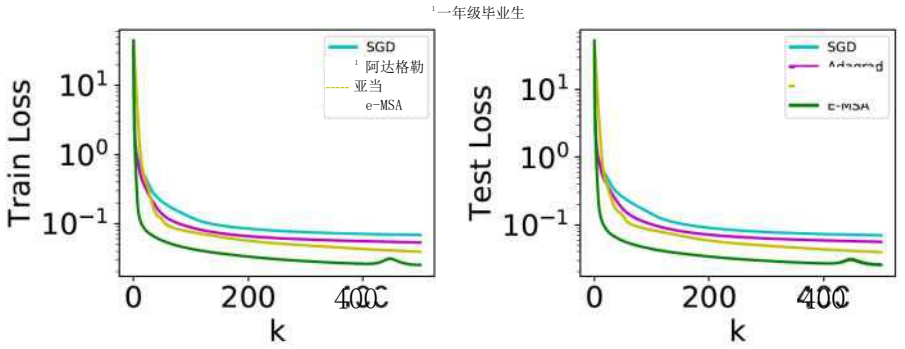
-Set

为每个人[0,1]

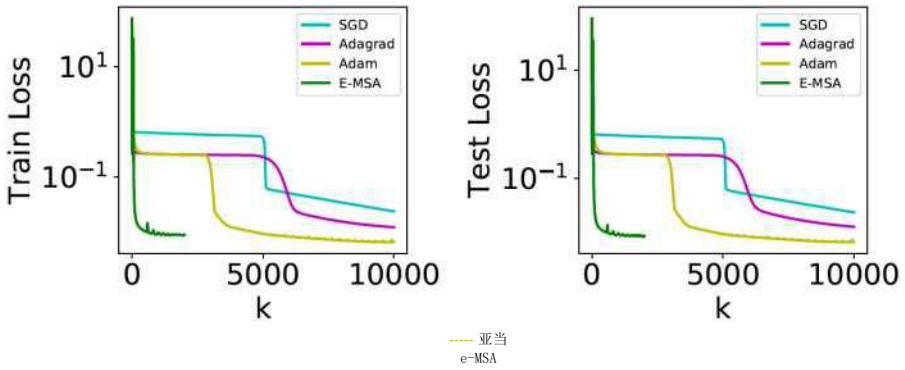
$$\circ\ r^{t1}\text{=argmax 心}^{H(z}\text{不，p。，幻}$$

在实践中，这个基本版本的性能不如“扩展 MSA”，它的工作方式与 MSA 相同，只是哈密顿被扩展的哈密顿[28]所取代：

$$^{H(z}\text{，}p^{0,\ v,}\ q)\text{：}=\ ^{H(z,\ \ ,0)}_{\mathbf{p}}\ -\ ^{(z,\ 0)}_{2\text{入皿-/}}\text{『—}^{1}\text{入恆+}^{v\ H(z,}_{z}\ \text{pW)。}$$



(a)



(b)

图 12：不同版本随机梯度下降算法的扩展 MSA 比较。 顶部的数字显示了小初始化的结果。 底部数字显示了更大初始化的结果。 经李千晓许可转载。 另见[28]。

图 12 显示了扩展 MSA 与不同版本的 SGD 在两种初始化中的结果。 可以看到，在迭代次数方面，扩展 MSA 的性能优于所有 SGD。 在挂钟时间方面，扩展 MSA 的优势明显减弱。 这可能是由于用于求解（32）的优化算法(这里是 BFGS)的实现效率低下)。 详情请参阅[28]。 无论如何，显然还有很大的改进空间。

6 结束语

最后，我们讨论了基于机器学习的算法已经和/或将产生显著差异的广泛问题。这些问题对于计算数学来说是比较新的。 我们坚信，基于机器学习的算法也会显著影响我们解决计算数学中更传统问题的方式。 然而，这方面的研究仍处于非常早期的阶段。

机器学习的另一个重要领域是多尺度建模。 上面讨论的闭矩问题就是这个方向的一个例子。有更多可能的应用程序，见[8]。机器学习似乎能够提供缺失的环节，使先进的多尺度建模技术真正实用。 例如，在异构多尺度方法(HMM)[1,9]中，一个重要的组成部分是从微观尺度模拟数据中提取相关的宏观尺度信息。 这一步骤一直是 HMM 的主要障碍。 显然，机器学习技术在这里有很大的帮助。

我们还讨论了数值分析的观点如何有助于改善机器学习的数学基础，并提出了新的和可能更稳健的公式。 特别是，我们已经尝到了高维近似理论应该是什么样的。 我们还证明，常用的机

器学习模型和训练算法可以从连续模型的某些特定离散化中恢复，以缩放的形式。 从这一讨论中可以看出，神经网络模型是相当自然和相当不可避免的。

我们从机器学习中真正学到了什么？ 嗯，似乎机器学习最重要的新见解是将函数表示为期望。 我们在这里复制它们是为了方便：

- 基于积分变换：

$$f(x)=E(\text{„}\cdot)\text{`}AA(W\text{十})$$

$$/^{(x)}=\text{电} \qquad a(\text{匾姑知}\ (a^{\circ}\ ,^{-x))}...)$$

- 基于流程：

$$\begin{array}{l} \textit{新西兰} \\ dT=E(a,\ w)\text{阵}\ aa(w^{t}z),\ z(0,\ x)=x \\ f^{(^{x}w)=1\&^{(1,\ x)}} \end{array} \qquad (40)\ (41)$$

从计算数学的角度来看，这表明中心问题将从特定的离散化方案转向更有效的函数表示。

这篇评论相当粗略。 有兴趣的读者可以参考三篇评论文章[17,18,19]了解更多细节。

致谢：我非常感谢我的合作者对本文所述工作的贡献。 特别是，我要衷心感谢罗伯托·卡尔、杰群·汉、阿努夫·詹岑、钱晓丽、潮马、韩王、斯蒂芬·沃伊托维奇和雷武对这里讨论的问题进行了多次讨论。 这项工作的一部分支持来自 iFlytek 给普林斯顿大学的礼物以及 ONR 赠款 N00014-13-1-0338。

参考资料

[1] Assyr Abdulle，渭南 E，Bjorn Engquist 和 Eric Vanden-Eijnden，异源多尺度方法，Acta Numerica，vol。 21，pp。 1-87,2012.

[2] 弗朗西斯·巴赫，“用凸神经网络打破维数的诅咒”，“机器学习研究杂志”，18（19）：1-53，2017 年。

[3] 安德鲁·R·巴伦，“乙状结肠函数叠加的通用近似界”，IEEE 信息论交易，39(3)：930-945，1993 年。

[4] 乔格·贝勒和米歇尔·帕里内洛，“高维势能表面的广义神经网络表示”，物理评论信件，98（14）：146401，2007 年。

[5] Achi Brandt，“多尺度科学计算：回顾 2001 年”。在 Barth，T.J.，Chan，T.F.和 Haimes，R。(编辑。)：多尺度多分辨率方法：理论与应用，Springer Verlag，Heidelberg，2001，pp。 196.

[6] 罗伯托·卡和米歇尔·帕里内洛，“分子动力学和密度功能理论的统一方法”，“物理评论信函”，55（22）：2471，1985 年。

[7] 莱纳奇·奇扎特和弗朗西斯·巴赫，“关于使用最优传输的过度参数化模型的梯度下降的全局收敛性”，“神经信息处理系统的进展”，第 3036-3046 页，2018 年。

[8] 渭南 E，多尺度建模原理，剑桥大学出版社，2011。

[9] 渭南 E 和 BjornEngquist，异构多尺度方法，Comm。 数学。 SCI，第二卷。 1，不。 1，pp。 87-132,2003.

[10] 渭南 E，韩杰群和阿努夫·金岑，“基于深度学习的高维抛物偏微分方程和后向随机微分方程的数值方法”，数学与统计通信 5，4（2017），349-380。

[11] 渭南 E，韩洁群，张林峰，“将机器学习与基于物理的建模相结合”，https://arxiv.org/pdf/2006.02619.pdf，2020 年。

[12] 渭南 E，马超和雷武，“两层神经网络种群风险的先验估计”，“数学科学中的通信”，17（5）：1407-1425，2019 年；ArXiv：1810.06397，2018 年。

[13] 渭南 E，马超和吴磊，“Barron 空间和神经网络模型的流诱导函数空间”，arXiv：1906.08039，2019。

[14] 渭南 E 和 StephanWojtowytsch，“Barron 函数的表示公式和点性质”，arXiv：2006.05982，2020。

[15] 渭南 E 和 StephanWojtowytsch，“关于与多层 ReLU 网络相关的 Banach 空间：函数表示、近似理论和梯度下降动力学”，https://arxiv.org/pdf/2007.15623.pdf，2020 年。

[16] 渭南 E，马超，雷武，“机器学习从一个连续的观点”，ar 希夫：1912.12777，2019 年。

[17] 渭南 E,韩洁群和 ArnulfJentzen，“求解高维 PDE 的算法:从非线性蒙特卡罗到机器学习”，https://arxiv.org/pdf/2008.13333.pdf，2020 年。

[18] 渭南 E，韩洁群和张林峰，“将机器学习与基于物理的建模相结合”，https://arxiv.org/pdf/2006.02619.pdf，2020 年。

[19] 渭南 E，马超，斯蒂芬·沃伊托维奇和雷武，“对机器学习的数学理解：什么是已知的，什么是不知道的”，https://arxiv.org/pdf/2009.10713.pdf.

[20] 韩洁群和渭南 E，“随机控制问题的深度学习近似”，深度强化学习讲习班，NIPS（2016），https://arxiv.org/pdf/1611.07422.pdf.

[21] 韩杰群，阿努夫·金岑和渭南 E，“用深度学习求解高维偏微分方程”，“国家科学院学报”，115，34（2018），8505-8510。

[22] 韩杰群，马超，郑马，渭南 E，“均匀精确的基于机器学习的动力学方程流体模型”，国家科学院学报，116（44）21983-21991；DOI：10.1073/PNAS。 1909854116,2019.

[23] 韩洁群，张林峰，罗伯托·卡尔和渭南 E，“深势：多体势能面的一般表示”，计算物理中

的通信，23（3）：629-639，2018。

[24] 皮埃尔·霍恩伯格和伯特兰·I·哈尔佩林，“动态临界现象理论”，“现代物理学评论”，49（3）：435，1977。

[25] 贾伟乐，王汉，陈墨汉，陆登辉，刘吉端，林林，罗伯托·卡尔，渭南 E，张林峰，“用机器学习将分子动力学的从头精度极限推到一亿个原子”，arXiv：2005.00223，2020。

[26] Jason M Klusowski 和 Andrew R Barron，“包括神经网络在内的高维脊函数组合的风险界限”，ar Xiv：1607.01434，2016。

[27] 延勒村，“反向传播的理论框架”，载于：图雷茨基，D.，Hinton，G.，Sejnouski，T。（编辑。） 1988 年连接主义模型暑期学校论文集，卡内基-梅隆大学，摩根考夫曼，1989 年。

[28] 李千晓，龙晨，程泰和渭南 E，“基于最大原理的深度学习算法”，“机器学习研究杂志”，第二卷。18,不。165 页。2018 年 1 月 1 日至 29 日,https://arxiv.org/pdf/1710.09513.pdf.

[29] 李千晓和郝树基，“一种最优的深度学习控制方法和应用于离散视觉神经网络”，第 35 届国际机器学习会议记录，2018 年。

[30] 鲁登辉，王汉，陈墨汉，刘吉端，林林，罗伯托卡，渭南 E，贾伟乐，张林峰，“86Pflops 深势分子动力学模拟 1 亿原子的从头计算精度”，arXiv：2004.11658，2020。

[31] 马超，雷武，渭南 E，“两层神经网络模型梯度下降动力学的猝灭激活行为”，arXiv：2006.14450，2020。

[32] 宋梅，AndreaMontanari 和 Phan-MinhNguyen，“两层神经网络景观的平均视野”，“国家科学院学报”，115（33）：E7665-E7671，2018。

[33] Etienne Pardoux 和 Shige Peng，“反向随机微分方程和拟线性抛物偏微分方程”，在随机偏微分方程及其应用中(夏洛特,NC,1991),第二卷。176 在控制和通知的讲座笔记。Sci.，Springer，Berlin，1992，pp。 200-217.

[34] 格兰特·罗茨科夫和埃里克·范登-埃因登，“作为相互作用粒子的参数：神经网络的长时间收敛和渐近误差缩放”，“神经信息处理系统的进展”，第 7146-7155 页，2018 年。

[35] 贾斯汀·西里尼亚诺和康斯坦蒂诺斯·斯皮里奥普洛斯，“神经网络的平均场分析：中心极限定理”，arXiv：1808.09372，2018 年。

[36] 汉斯·崔贝尔，函数空间理论，Birkhauser，1983 年。

[37] 韩旺，张林峰和渭南 E，未发表。

[38] 张林峰，韩洁群，王汉，罗伯托·卡，渭南 E，“深势分子动力学：具有量子力学准确性的可扩展模型”，“物理评论信函”，120：143001，2018 年 4 月。

[39] 张林峰，韩杰群，王汉旺，维萨姆·A·赛迪，罗伯托·卡，渭南 E，“有限系统和扩展系

统原子间势能模型的端到端对称性保持”，“神经信息处理系统的进展”，2018 年。

[40] 张林峰，林德业，王汉，罗伯托，渭南 E，“主动学习均匀精确的原子间势用于材料模拟”，“物理评论材料”，3（2）：023804，2019。

[41] 张林峰，王汉和渭南 E，“增强动力学在大原子和分子系统中的增强采样。i. 基本方法”，J.化学。 菲斯，第二卷。 148，pp。 124113,2018.