

# 自动微分数值计算理论-地震波模拟为例<sup>\*</sup>

刘金国<sup>1)</sup> 许开来<sup>2)</sup>

1) (哈佛大学物理系, 坎布里奇 02138)

2) (斯坦福大学, 斯坦福 94305)

自动微分是机器学习中核心的数值技术, 它被用于高效, 自动的对程序求导。如今越来越多的物理学者意识到高效, 自动的求导可以对很多物理问题的求解提供新的思路。本文介绍自动微分技术相关的主要理论, 并以地震波的模拟为例介绍它的应用。

关键词: 自动微分, 科学计算, 编程语言

**PACS:** 02.60.Pn, 02.30.Jr

## 1 引言

自动微分是指自动获取一个目标计算机程序导数的技术。很多人了解它是因为它在机器学习中的成功应用, 人们可以用它优化带有千亿参数的神经网络 [Rosset(2019)]。与很多人印象不同的是, 自动微分其实是个很古老的技术。Nolan 曾在他 1953 年的博士论文中就提出过 [Nolan(1953)] 自动化求导计算机程序的构想, 而最近十几年, 自动微分在科学计算中的应用越来越广泛。过去人们求解导数很多靠手动推导, 很多时候限制了一个算法可处理问题的范围, 一个典型的例子是变分蒙特卡洛算法。过去, 人们会将变分基态假设为 Gutzwiller 波函数 [Gutzwiller(1963)] 这样具有良好解析性质的函数。直到 2017 年, Carleo 等人 [Carleo and Troyer(2017)] 把机器学习的一些变分模型限制波尔兹曼机 (RBM) 带入了大家的视野, 由于 RBM 的解析性质很好, 当时求导过程仍然可以通过解析推导。但受此启发, 大家

---

\*

† 通信作者. E-mail: jinguoliu@g.harvard.edu

意识到不光是 RBM 这样解析性质好的函数，任何机器学习中的神经网络也可以被用做波函数的猜测形式。而且如果可以用流行的机器学习框架，比如脸书公司的 PyTorch 和谷歌公司开发的 TensorFlow 来写波函数，人们可以避免首推导数的麻烦而可以把波函数的表达变得更加自由。而现在，人们对自动微分的认知更加深刻了，波函数的猜测形式不光是神经网络，而几乎可以是任意包含物理过程的计算机函数，因为计算机完全有能力微分一段看似随意的通用代码。因为不管程序有多么复杂，总可以被分解成一些加减乘除这样的基础解析函数。知晓基础函数的自动微分规则，计算机总可以通过链式法则用计算机推导出整个程序的导数。除了上面这个例子，科学家们利用方便的，自动化的导数解决了很多包括量子模拟 [Luo *et al.*(2019)Luo, Liu, Zhang, and Wang], 分子动力学, 组合优化, 海洋学 [Heimbach *et al.*(2005)Heimbach, Hill, 和地震学 [Symes(2007), Zhu *et al.*(2020)Zhu, Xu, Darve, and Beroza] 等领域中的变分问题，控制问题和逆向工程问题。甚至是一些非解析的函数，比如蒙特卡洛抽样过程，人们也设计出了一些办法对其自动化的求导 [Zhang *et al.*(2019)Zhang, Wan, and Yao]。

那物理学家们是否只需要奉行“拿来主义”，仅阅读机器学习库的文档，就可以很好的驾驭自动微分呢？并不完全是的，虽然利用现有的机器学习库固然可以解决很多科学问题，尤其是与张量应用有关的问题，但是很多更加先进的自动微分技术需要研究者一定的理解才能更好的驾驭。机器学习库中定义了一些神经网络中常用的函数族（矩阵乘法，卷积函数，relu 函数等）手动定义导数传递规则，并通过上述链式法则求得用该库中提供的函数写出的程序的导数，正如流行的机器学习框架 PyTorch 和 TensorFlow 中那样。手动定义常用函数导数的优点是可以对函数有针对性的优化，尤其是可以保证张量运算的性能。这种方案的缺点是，它往往无法满足科学计算的研究中，对函数求导的多元化需求。有些时候人们可以通过添加一些求导规则来实现对程序的求导，比如在张量网络等模拟中需要用到的复数奇异值分解函数 [Wan and Zhang(2019), Liao *et al.*(2019)Liao, Liu, Wang, and Xiang], 模拟变分量子算法中么正量子线路的求导 [Luo *et al.*(2019)Luo, Liu, Zhang, and Wang]。但是也有一些时候，添加求导规则也无济于事，比如，严格对教化求解基态过程中用到的极大极小本正值求解器 [Hao Xie and Wang()] 里面对稀疏矩阵构造过程的求导就不太容易指定。甚至很多时候，这些规则在书写之初就没有很好的考虑复变函数的求导规则可能不同而导致给出错误的梯度。比如奇异值分解函数的求导规则直到 2019 年，才有人意识到复数的求导规则中有一个规范不变性带来的一项被忽略了 [Wan and Zhang(2019)]。我们不妨称这种需要手动添加求导规则的自动微分为“狭义”自动微分。而接下来我们主要围绕“广义”的可以真正微分包含控制流的通用代码的自动微分，它直接微分最基础的有限的计算机指令，并给出正确性有保证的梯度。自动微分

的强大威力同时也带来了一些门槛，为了让研究者们更好的利用广义自动微分，本文的向物理学家们介绍该领域中的几种核心技术。章节 ??介绍了前向和后向自动微分，以及它们适合的应用场景。章节 3介绍了后向自动微分中的时间和空间交换的理论，了解它可以使得研究者可以更加好的权衡程序的运行时间和空间。章节 4介绍了地震波模拟这个应用，它很好的展示了。

## 2 链式法则的两种传递方式

在上世纪 60 年代,自动微分发展出了两种不同的实践,一种叫做前向传播(Forward propagation) [Wengert(1964)],一种叫做后向传播(Backward propagation) [Boltianski *et al.*(1960) Boltianski, Gamkrelidze, Mishchenko, and Pontryagin]. 为了直观的理解它们的不同,这里考虑一个计算过程  $f: \mathbb{R}^m \rightarrow \mathbb{R}^n$

$$\begin{aligned} \mathbf{x}^1 &= f_1(\mathbf{x}^0) \\ \mathbf{x}^2 &= f_2(\mathbf{x}^1) \\ &\dots \\ \mathbf{x}^L &= f_L(\mathbf{x}^{L-1}) \end{aligned}$$

其中  $x^0 \in \mathbb{R}^m$ ,  $x^L \in \mathbb{R}^n$ ,  $L$  是计算的步骤数。这段程序的雅可比 (Jacobian) 矩阵  $J_{ij} \equiv \frac{\partial x_i^L}{\partial x_j^0}$  是一个  $n \times m$  的矩阵, 其中  $x_j^0$  和  $x_i^L$  代表了输入和输出中的一个元素。前向自动微分的链式法则表述为  $\frac{\partial \mathbf{x}^k}{\partial x_j^0} = \frac{\partial \mathbf{x}^k}{\partial \mathbf{x}^{k-1}} \frac{\partial \mathbf{x}^{k-1}}{\partial x_j^0}$ , 它计算了雅可比矩阵的第  $j$  列。后向自动微分的链式法则为  $\frac{\partial \mathbf{x}_i^L}{\partial \mathbf{x}^{k-1}} = \frac{\partial \mathbf{x}_i^L}{\partial \mathbf{x}^k} \frac{\partial \mathbf{x}^k}{\partial \mathbf{x}^{k-1}}$ , 它计算了雅可比矩阵的第  $i$  行。在大多数应用中, 计算一行雅可比矩阵比计算一列有用的多, 因为后向自动微分的实现难度远高于前向自动微分, 因为它的梯度计算方向和计算本身的方向相反, 如何回溯中间状态成了一个核心的问题。

NOTE: add an illustration of forward AD and reverse AD. Change column/row to Jv and vJ.

## 3 时间与空间的权衡

引言中, 我们提到后向自动微分有一个无法避免的共同问题, 那就是如何回溯中间状态。回溯中间状态最简单的做法是, 记录程序的初始状态, 当需要获取第  $k$  个计算状态  $s_k$ , 程序从初始状态  $s_0$  开始进行  $k$  步运算。这么做缺点明显, 那么就是需要  $O(T^2)$  的运行时间。当磁盘空间充裕, 程序可以在每一步都进行状态拷贝, 虽然有  $O(T)$  的额外空间开销, 但是没有额外的用于运算时间。但很多实践中, 不管是  $O(T^2)$  的时间开销或者是  $O(T)$  的空间开销都无法被接受, 人们更需要的是一个均衡的计算时间和计算空间的方案。在程序运行的何时, 何处进行状态拷贝, 成为了检查点方案的问题核心。最优的关于时间和空间交换

的检查点方案在 1992 年被 Griewank [?] 提出，它是很多广义自动微分框架的基础。算法的梗概如图 1 (a) 所示，算法把计算过程分割成  $d$  个区块，并记录每个区块的开始状态，每个区块的大小由函数  $\eta(\tau, \delta)$  决定，其中  $\delta = 1 \dots d$  为从末尾开始数的区块的指标， $\tau = 1 \dots t$  是一个超参数并被初始化为  $t$ 。最后的区块最小，因此可以用上述  $O(T^2)$  的算法来获取中间状态，回溯完最后一个区块后，程序释放最后一个区块开始处被缓存的状态。于是在接下来计算倒数第二个区块的过程中，我们可以在这个计算过程中多缓存一个状态而不增加最大内存开销。类似的，倒数第  $\delta$  个分块可以使用的额外状态缓存数为  $\delta - 1$ 。对每个分开递归的调用这个算法  $t$  次，每次将  $\tau \leftarrow \tau - 1$ ，最后得到的分块的大小为 1。整个算法的时间和空间开销的关系是

$$T_c = tT, S_c = dS, \quad (1)$$

其中， $T = \eta(t, d)$  是初始计算时间。选择适当的  $t$  或者  $d$ ，在时间和空间维度上的额外复杂度可以都是  $\log(T)$ 。

回溯中间状态还有另一种做法，那就是可逆计算。可逆计算的代码天然可以被回溯，但是也会因此带来额外的时间或空间的开销。由于直接赋值在可逆计算中不被允许，最直接的将不可逆代码变成可逆代码的方法是先分配内存，用累加的方式，

$$\mathbf{x}^1 \dots \mathbf{x}^L \leftarrow 0 \quad (2)$$

$$\mathbf{x}^1 += f_1(\mathbf{x}^0) \quad (3)$$

$$\mathbf{x}^2 += f_1(\mathbf{x}^1) \quad (4)$$

$$\dots \quad (5)$$

$$\mathbf{x}^L += f_1(\mathbf{x}^{L-1}) \quad (6)$$

$$T_r = T \left( \frac{T}{S} \right)^{\frac{\ln(2 - (1/k))}{\ln k}}, S_r = \frac{k-1}{\ln k} S \log \frac{T}{S}. \quad (7)$$

## 4 自动微分在物理中的应用

### 4.1 地震波的模拟

后来在地震学的模拟中，它被用来微分地震波传播的过程 [Symes(2007)]。

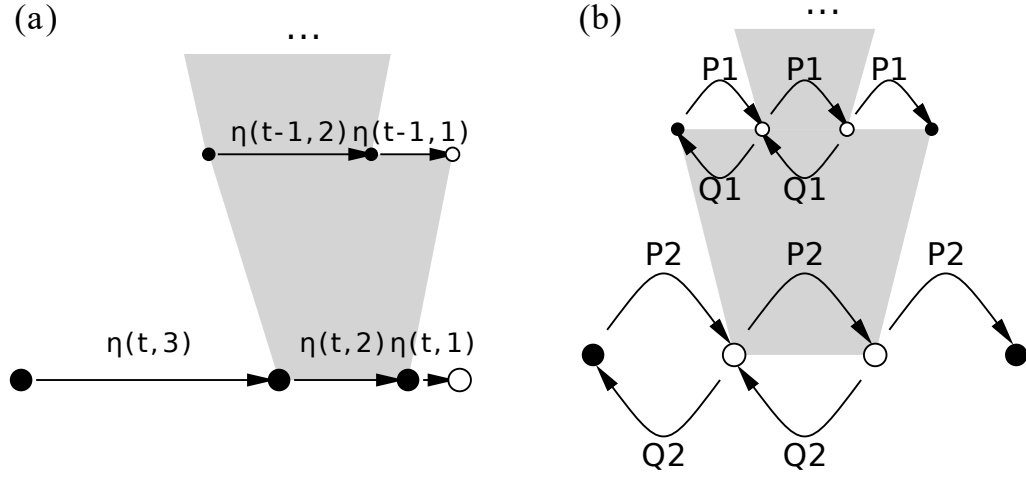


图 1: (a) 广义自动微分中常见的 Treeverse 算法 [?], 其中  $\eta(\tau, \delta) \equiv \binom{\tau + \delta}{\delta} = \frac{(\tau + \delta)!}{\tau! \delta!}$ 。 (b) 可逆计算时空交换的 Bennett 算法。 [Bennett(1973), Levine and Sherman(1990)] 其中,  $P$  和  $Q$  分别代表了计算和反计算。它的伪代码是代码块??。

#### 4.2 热带 (Tropical) 张量网络求解自旋玻璃最优构型

热带张量网络是张量网络的一个特别的应用, 它重新定义了张量中基础元素的代数

$$x \oplus y = \max(x, y), \quad x \odot y = x + y. \quad (8)$$

如果我们将张量中的元素定义为自旋玻璃的耦合强度, 我们就可以通过收缩这个张量结构得到自旋玻璃的最优构型。这个最优构型的能量表达式为

$$E(\{\sigma\}) = \max_{\sigma} - \sum_{i < j} J_{ij} \sigma_i \sigma_j - \sum_i h_i \sigma_i, \quad (9)$$

因此对它的微分就是。但是作了这样的替换, 张量收缩的微分规则就发生了变换。

## 5 结 论

### 致谢

感谢北京大学力学系某某教授和某某博士以及某某的讨论。

### 附录 A1

标题排列和编号方式为 A1, A2, A3.

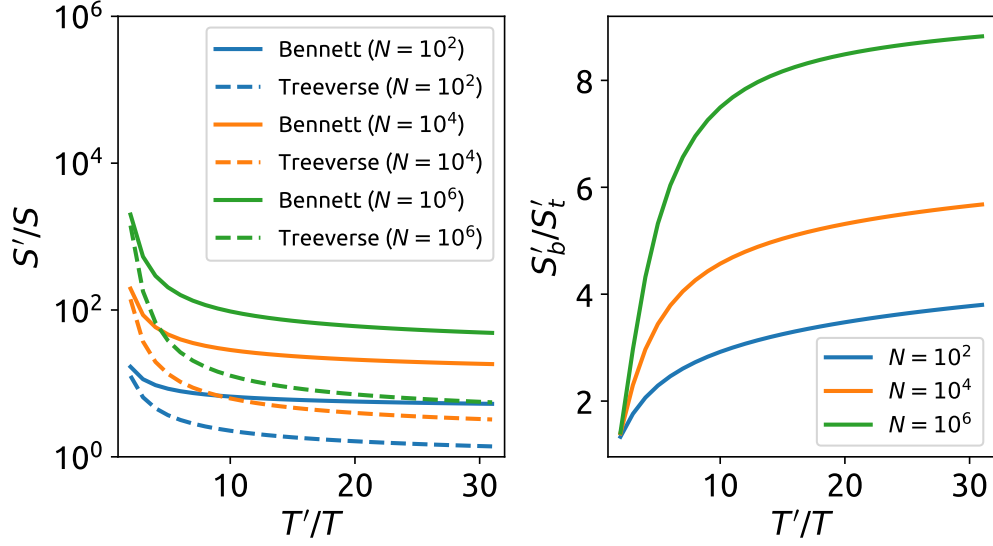


图 2: 为了回溯中间状态, 时间和空间在两种最优时间-空间交换策略下的关系。(a) 固定横轴为状态回溯的计算时间与原函数计算时间的比值, 对比再允许固定时间开销下, 内存的额外开销。其中 Bennett 算法代表了可逆计算下的最优策略, 而 Treeverse 则是传统计算允许的最优策略。(b) 对比 Bennett 算法与 Treeverse 算法空间开销的比值。

附录 A2

[Rosset(2019)] C. Rosset, Microsoft Blog (2019).

[Nolan(1953)] J. F. Nolan, *Analytical differentiation on a digital computer*, Ph.D. thesis, Massachusetts Institute of Technology (1953).

[Gutzwiller(1963)] M. C. Gutzwiller, \bibfield journal \bibinfo journal Phys. Rev. Lett.\ \textbf{\bibinfo volume 10}, \bibinfo pages 159 (\bibinfo year 1963).

[Carleo and Troyer(2017)] G. Carleo and M. Troyer, \bibfield journal \bibinfo journal Science\ \textbf{\bibinfo volume 355}, \bibinfo pages 602 — 606 (\bibinfo year 2017).

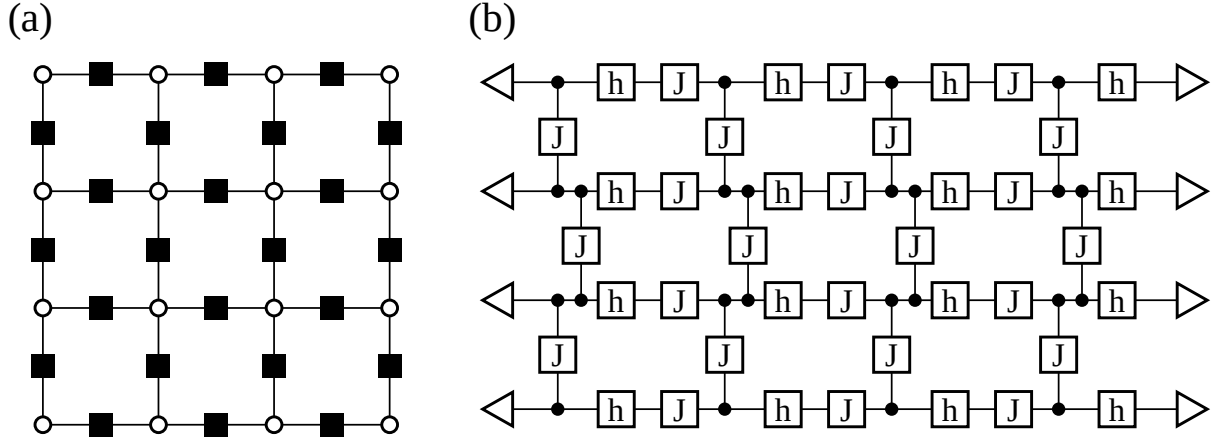


图 3: 正方晶格上的自旋玻璃问题, (a) 对应的热带张量网络表示。(b) 对应的“量子线路”表示。  
[Luo *et al.*(2019)Luo, Liu, Zhang, and Wang] X.-Z. Luo, J.-G. Liu, P. Zhang, and L. Wang, “Yao.jl: Extensible, efficient framework for quantum algorithm design,” (2019), arXiv:1912.10877 [quant-ph]

[Heimbach *et al.*(2005)Heimbach, Hill, and Giering] P. Heimbach, C. Hill, and R. Giering, Future Generation Computer Systems **21**, 1356 (2005).

[Symes(2007)] W. W. Symes, \bibfield journal \bibinfo journal Geophysics\ \textbf{\bibinfo volume 72}, \bibinfo pages SM213 (\bibinfo year 2007).

[Zhu *et al.*(2020)Zhu, Xu, Darve, and Beroza] W. Zhu, K. Xu, E. Darve, and G. C. Beroza, “A general approach to seismic inversion with automatic differentiation,” (2020), arXiv:2003.06027 [physics.comp-ph] .

[Zhang *et al.*(2019)Zhang, Wan, and Yao] S.-X. Zhang, Z.-Q. Wan, and H. Yao, “Automatic differentiable monte carlo: Theory and application,” (2019), arXiv:1911.09117 [physics.comp-ph] .

[Wan and Zhang(2019)] Z.-Q. Wan and S.-X. Zhang, “Automatic differentiation for complex valued svd,” (2019), arXiv:1909.02659 [math.NA] .

- [Liao *et al.*(2019)Liao, Liu, Wang, and Xiang] H.-J. Liao, J.-G. Liu, L. Wang, and T. Xiang, *Physical Review X* **9** (2019), 10.1103/physrevx.9.031041.
- [Hao Xie and Wang()] J.-G. L. Hao Xie and L. Wang, *arXiv:2001.04121*.
- [Wengert(1964)] R. E. Wengert, *Communications of the ACM* **7**, 463 (1964).
- [Boltyanski *et al.*(1960)Boltyanski, Gamkrelidze, Mishchenko, and Pontryagin] V. Boltyanski, R. Gamkrelidze, E. Mishchenko, and L. Pontryagin, *IFAC Proceedings Volumes* **1**, 464 (1960).
- [Bennett(1973)] C. H. Bennett (1973).
- [Levine and Sherman(1990)] R. Y. Levine and A. T. Sherman, *SIAM Journal on Computing* **19**, 673 (1990).



# English Title \*

Guan Yun-Chang<sup>1)2)</sup>   Liu Bei<sup>1)†</sup>   Zhuge Liang<sup>2)</sup>

1) ( *State Key Laboratory of Quantum Optics and Quantum Optics Devices, Institute of Opto-Electronics, Shanxi University, Taiyuan 030006, China* )

2) ( *Department of Physics, Tsinghua University, Beijing 100084, China* )

## Abstract

To determine the probe made of amino acids arranged in a linear chain and joined together by peptide bonds between the carboxyl and amino groups of adjacent amino acid residues. The sequence of amino acids in a protein is defined by a gene and encoded in the genetic code. This can happen either before the protein is used in the cell, or as part of control mechanisms.

**Key-**      Keyword1, Keyword2, Keyword3, Keyword4

**PAACS:** 02.10.Yn, 33.15.Vb, 98.52.Cf, 78.47.dc

---

\* Project supported by the State Key Development Program for Basic Research of China (Grant No. 2011CB00000), the National Natural Science Foundation of China (Grant Nos. 123456, 567890), and the National High Technology Research and Development Program of China (Grant No. 2011AA06Z000).

† Corresponding author. E-mail: aaa@bbb.ccc