

# 深度学习下随机基因开关模型的概率密度函数估计

石昊楠, 马晋忠\*

(山西大学 数学与统计学院, 山西 太原 030006)

**摘要:** 在求解随机基因开关模型的概率密度函数的过程中, 目前常用方法很难进行理论求解, 精确解只能在一些很严格的条件下才能得到或结果较粗糙, 精确度较低且耗时较长, 精度会受到样本路径长度和网格细分程度的限制。针对这些问题, 本文借助神经网络来估计该模型的概率密度函数。在研究过程中, 通过引入惩罚因子来克服局部最优化, 并给出了相应的设置准则; 同时将归一化条件作为监督条件, 以避免近似解为零。最终的结果表明, 用深度学习的方法估计随机基因开关模型的概率密度函数是非常可行和有效的, 这种方法不需要任何插值和坐标变换, 与蒙特卡罗方法相比耗时至少减少了 50 s, 且所得结果精度更高。同时研究了隐藏层数和节点数的影响, 表明通过适当构建神经网络可以提高机器学习的计算性能。

**关键词:** 概率密度函数; 蒙特卡罗模拟; 神经网络; 惩罚因子; 高斯白噪声; 随机基因开关模型

**中图分类号:** O212 **文献标志码:** A **文章编号:** 0253-2395(2024)06-1136-12

## Probability Density Function Estimation of Stochastic Gene Switching Model Under Deep Learning

SHI Haonan, MA Jinzhong\*

(School of Mathematics and Statistics, Shanxi University, Taiyuan 030006, China)

**Abstract:** In the process of solving the probability density function (PDF) of stochastic gene switching model, the commonly used methods are difficult to solve theoretically, and the accurate solutions can only be obtained under some very strict conditions. The results are relatively rough, with low accuracy and long time consumption, and the accuracy may be limited by sample path length and mesh subdivision degree. In response to these issues, this paper uses neural network to estimate the PDF of this model. In the research process, penalty factors were introduced to overcome local optimization, and corresponding setting criteria were provided. Normalization conditions were also used as supervisory conditions to avoid approximate solutions of zero. Our final results show that the deep learning method is very feasible and effective to estimate the PDF of stochastic gene switching model, and this method does not require any interpolation or coordinate transformation. Compared with the Monte Carlo method, the time consumption is reduced by at least 50 seconds, and the obtained results have higher accuracy. The effects of the number of hidden layers and nodes were also studied, indicating that the computational performance of machine learning can be improved by appropriately constructing neural network.

**Key words:** probability density function; Monte Carlo simulation; neural network; penalty factor; Gaussian white noise; stochastic gene switching model

收稿日期: 2023-05-30; 接受日期: 2023-10-12

基金项目: 国家自然科学基金(12102237); 山西省基础研究计划(20210302124387)

作者简介: 石昊楠(2000-), 女, 山西大同人, 硕士研究生, 主要研究方向为非线性和高频时间序列分析。E-mail: 632886127@qq.com

\* 通信作者: 马晋忠(MA Jinzhong), E-mail: majinzhong0414@163.com

引文格式: 石昊楠, 马晋忠. 深度学习下随机基因开关模型的概率密度函数估计[J]. 山西大学学报(自然科学版), 2024, 47(6):1136-1147. DOI:10.13451/j.sxu.ns.2023160

## 0 引言

基因开关模型也称为基因转录调控模型,基因转录表达的过程中常伴随着 mRNA 与蛋白质的合成与降解,它们数目的变化是典型的随机过程。随机基因开关模型的概率密度函数可以反映蛋白浓度的变化,通过研究它的概率密度函数可以掌握基因转录表达的机理,弄清各因素在基因转录调控过程中的作用及影响,从而更好地了解遗传信息的传递。随机基因开关模型的概率密度函数通常有两种估计方法:一种是直接理论求解 FP (Fokker-planck) 方程,但这种方法的解只是通过计算在网格上的点得出的,任何其他点的评估都需要插值或其他重构技术。对于一维系统,这些方法尚且有效,但在二维和更高维的系统中,它们将消耗大量计算资源,甚至难以克服维数灾难。同时,计算的精度会受到网格稀疏性的影响,而一个密集的网络又不可避免地会导致计算量的急剧增加。另一种方法是求解随机微分方程,然后通过蒙特卡罗方法统计计算其概率密度函数。与第一种方法相比,这种方法不需要插值,但它需要大量的样本路径,并且其解的准确性与生成的数据量有关。对于二维系统,可能因求解的联合概率密度函数的计算成本太大而无法获得。因此,寻找一种无须插值计算且计算复杂度较低的替代技术可以大大提高效率<sup>[1-2]</sup>。

人工神经网络 (Artificial Neural Network, ANN) 技术的应用为随机微分方程概率密度函数的求解提供了一种新的思路。人工神经网络是函数逼近的强大且有效的工具,由于人工神经网络拟合连续函数的能力强,大量学者开始使用人工神经网络来求解常微分方程和偏微分方程<sup>[3-5]</sup>。通过设置一个监督学习模型,将计算区间和边界区间离散化后作为网络的输入,并以微分方程的初始条件和边界条件作为网络训练的监督条件,从而使近似解通过最小化标准接近真实解。而深度学习是在人工神经网络及各种数据处理技术不断发展的基础上发展起来的。深度学习方法的提出,解决了多层神经网络模型的参数优化问题,突破了神经网络技术发展的瓶颈,之后深度学习的方法便开始被应用在各个领域,且正在不断完善和丰富<sup>[6]</sup>。使

用深度学习中的神经网络技术求解微分方程相对于另外两种传统估计概率密度函数方法具有以下优势:(1)神经网络可以学习近似解;(2)解的搜索不需要任何坐标变换;(3)随着样本点数量的增加,计算复杂度不会迅速增加;(4)求解速度更快。与文献[7]中基于逃逸概率和吸收区域来量化系统稳定性的研究方法不同,本文从高斯白噪声诱导的基因开关模型出发,分别用蒙特卡罗模拟方法和深度学习求解 FP 方程的方法进行概率密度函数的估计,并将两种方法进行对比,同时研究了隐藏层数与各层节点数对结果的影响,最后验证了用深度学习估计随机基因开关模型的概率密度函数的精确性和有效性。

## 1 模型分析

随机基因开关模型的表达式为<sup>[7-8]</sup>:

$$\frac{dx}{dt} = \frac{k(2x^2 + 50x^4)}{25 + 29x^2 + 52x^4 + 4x^6} - rx + 1 + \zeta(t),$$

这是一个生物学上的单基因双稳态模型。其中  $x$  表示抑制蛋白的浓度,  $k$  用来度量抑制蛋白结合所带来的增强转录,降解率用  $r$  表示,抑制蛋白的平衡态浓度由  $k$  和  $r$  决定。 $\zeta(t)$  为高斯白噪声过程,它满足的统计性质为:

$$\langle \zeta(t) \rangle = 0,$$

$$\langle \zeta(t)\zeta(s) \rangle = 2D\delta(t-s),$$

其中  $D$  为噪声强度<sup>[9-10]</sup>。

图 1 中我们给出了确定情形下 ( $D=0$ ) 系

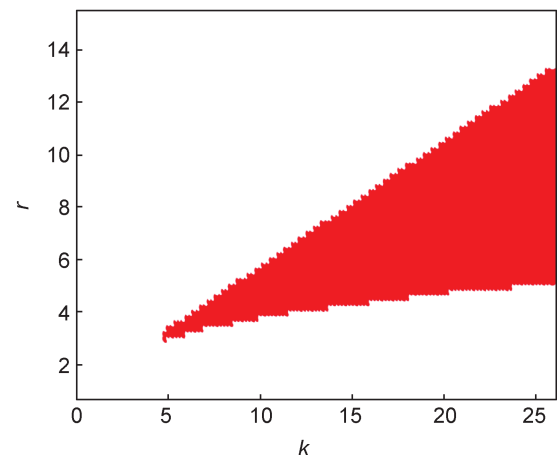


图 1 确定性基因开关模型对参数  $k$  和  $r$  的分岔图

Fig. 1 The bifurcation diagram of the deterministic gene switching model for the parameters  $k$  and  $r$

统对参数  $k$  和  $r$  的分岔图。结果显示当参数  $k$  和  $r$  的取值在图中的白色区域时,系统处于单稳态,此时的平衡点唯一,任意的初始蛋白浓度都会逐渐趋于这个平衡点。当  $k$  和  $r$  的取值在图中的红色区域时,此时系统有三个平衡点。图 2 给出了系统在这种情形的相图。此时,系统有三个平衡点  $x_1$ 、 $x_2$  和  $x_3$ ,其中  $x_1$  和  $x_3$  是稳定的。当系统处于  $x_1$  时,基因开关是关闭的,此时的蛋白质含量偏低,系统处于一个低转录的状态。当系统处于  $x_3$  时,基因开关打开,这时蛋白质含量偏高,系统开始进行高转录。 $x_2$  是不稳定点,小于  $x_2$  的蛋白质浓度会逐渐向  $x_1$  靠近,而大于  $x_2$  的蛋白质浓度会逐渐趋于  $x_3$ <sup>[11-12]</sup>。

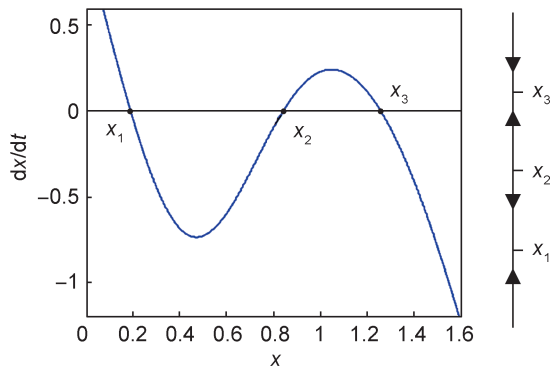


图2 确定性基因开关模型为双稳态时的相图

Fig. 2 The phase diagram of a bistable deterministic gene switching model

通过图 3 的势函数,可确定出一个较为理想的双稳态模型对应的参数值为  $r = 5.23, k = 10$ 。

要想进一步探索该基因模型的一些特点规

律,了解它的概率密度函数是至关重要的。接下来我们分别用蒙特卡罗模拟方法和本文主要研究的基于深度学习的方法估计基因开关模型的概率密度函数,通过对比来说明用深度学习的方法估计随机基因开关模型概率密度函数的可行性和有效性。

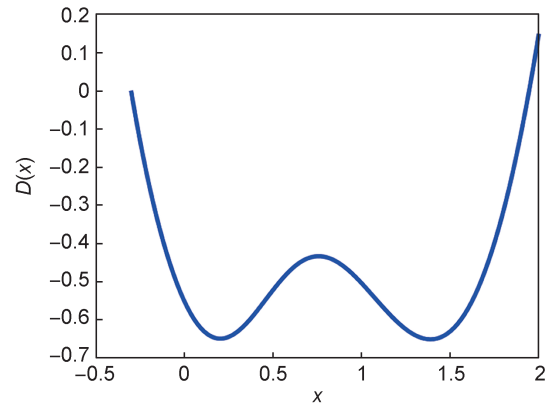


图3 确定性基因开关模型的势函数( $r = 5.23, k = 10$ )

Fig. 3 The potential function of the deterministic gene switching model ( $r = 5.23, k = 10$ )

## 2 稳态概率密度函数求解

### 2.1 蒙特卡罗求解

#### 2.1.1 随机二阶龙格-库塔模型

龙格-库塔(Runge-Kutta)方法<sup>[13]</sup>是一种常用的以数学为基础的典型微分方程数值解法,通过插值来对结果进行优化。本文研究的随机基因开关模型对应的经典二阶龙格-库塔公式如下(设置时间步长为  $h = 0.1$ )<sup>[14]</sup>:

$$k_1 = \frac{k(2x^2 + 50x^4)}{25 + 29x^2 + 52x^4 + 4x^6} - rx + 1,$$

$$k_2 = \frac{k[2(x + h \cdot k_1 + \sqrt{2Dh} \phi_n)^2 + 50(x + h \cdot k_1 + \sqrt{2Dh} \phi_n)^4]}{25 + 29(x + h \cdot k_1 + \sqrt{2Dh} \phi_n)^2 + 52(x + h \cdot k_1 + \sqrt{2Dh} \phi_n)^4 + 4(x + h \cdot k_1 + \sqrt{2Dh} \phi_n)^6} - r(x + h \cdot k_1 + \sqrt{2Dh} \phi_n) + 1,$$

$$x_{n+1} = x_n + \frac{h}{2} (k_1 + k_2) + \sqrt{2Dh} \phi_n,$$

其中  $\phi_n$  是一个标准正态分布随机数。

结合随机二阶龙格-库塔公式,利用 Matlab 得到高斯白噪声诱导的基因开关模型的时间历程图,如图 4(a) 所示。之后再借助蒙特卡罗模拟技术画出对应的概率密度函数图。为了使图形看起来更美观,我们将  $x$  的取值范围设置为  $[-0.4, 2]$ ,时长设置为 100 s。研究发现,该方

程在 20 s 以后的值是稳定的,因此我们取 20 s 之后的点。根据得到的  $x$ ,利用直方图法估计概率密度函数,将 20 s 之后的数据进行分组统计,计算每个区间内数据点的频率,将统计得到的频率除以总数和区间宽度,来得到概率密度函数。用蒙特卡罗进行模拟的实质是通过生成的随机数取平均后作为预期值的近似。因

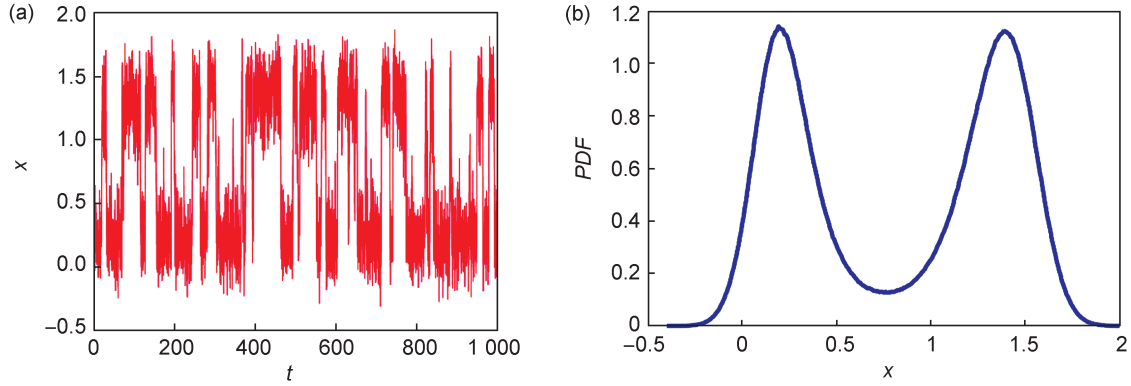


图4 稳定状态( $r=5.23, k=10, D=0.1$ )下的时间历程图(a)和PDF图(b)

Fig. 4 The time history diagrams (a) and the corresponding PDF diagrams (b) under steady state ( $r=5.23, k=10, D=0.1$ )

此,我们通过前面得到的点便能用蒙特卡罗模拟出一条PDF(probability density function)来,重复进行1 000次,则对应得到1 000条PDF曲线,将所得到的这1 000条函数图取平均,从而得到对应的概率密度函数图,如图4(b)所示。

一般在试验中,基因开关的状态往往是无法直接看出的,而是要依据当前细胞中该基因合成的蛋白质浓度的高低来判断。如果蛋白质的浓度很高,则说明基因开关处于开的状态。相反,如果蛋白质的浓度很低,我们则认为基因开关处于关闭状态。当系统存在两个稳定点时,受某些参数变化的影响,蛋白质浓度会由一个稳定点转移到另一个稳定点,引起蛋白质浓度的改变,进而触发基因开关<sup>[15]</sup>。而图4的状态是一个稳定状态,左右峰值高度基本一致,这个时刻抑制蛋白质的浓度可以作为判断基因开关的临界值。

### 2.1.2 不同参数对概率密度函数的影响

接下来我们通过改变参数值来研究不同参数值对模型概率密度函数的影响,即不同的降解率、增强转录和噪声强度对蛋白质浓度的影响。

首先只改变降解率 $r$ ,所得的概率密度函数对比如图5所示。

通过对比我们发现在其他条件一定的情况下,当降解率在5.2左右时双峰都比较明显,当降解率逐渐减小时,左侧的峰明显减小,右侧的峰会更突出,而当降解率逐渐增大时,右侧的峰会明显减小,左侧的峰会更突出。这说明当降解率减小时抑制蛋白浓度将主要处于高浓度状态,此时基因开关处于开的状态,而当降解率增大时抑制蛋白浓度将集中在低浓度态,

此时基因开关便会关闭,可见降解率的增大能够实现基因开关从开到关的转化,可以作为基因开关的控制参量。

同理改变增强转录 $k$ 所得的概率密度函数对比如图6所示。

通过对比发现在其他条件一定的情况下,当增强转录值在9.9左右时为双峰,增强转录值逐渐减小时,由双峰变为单峰,峰值在左侧,而当增强转录值逐渐增大时,仍由双峰变为单峰,峰值在右侧。这说明抑制蛋白结合所带来的增强转录值逐渐增大时,抑制蛋白浓度会逐渐由低浓度向高浓度状态移动,使得基因开关打开,因此该参数同样能当作基因开关的控制参量<sup>[16]</sup>。

同理改变噪声强度 $D$ 所得的概率密度函数对比如图7所示。

可以看出与前面所得结果不同,原来已是双峰,在其他条件一定的情况下,只改变噪声强度仍为双峰,只是明显程度不同,噪声强度越小越明显,说明此时噪声强度并不能作为基因开关的控制参数。通过上面的分析可知,三个参量中,降解率和用来度量抑制蛋白结合所带来的增强转录可以作为基因开关,通过调节这两个参量的用量可以起到调节抑制蛋白浓度的作用,从而也说明了研究概率密度函数的意义。

## 2.2 深度学习求解

### 2.2.1 确立平稳的FP方程

对于一个普通的一维随机微分方程:

$$\dot{x} = f(x) + \zeta(t), \quad (1)$$

其中 $x$ 是一个变量, $\zeta(t)$ 是一个噪声强度为 $D$ 的高斯白噪声过程, $f(x)$ 是关于 $x$ 的函数。对

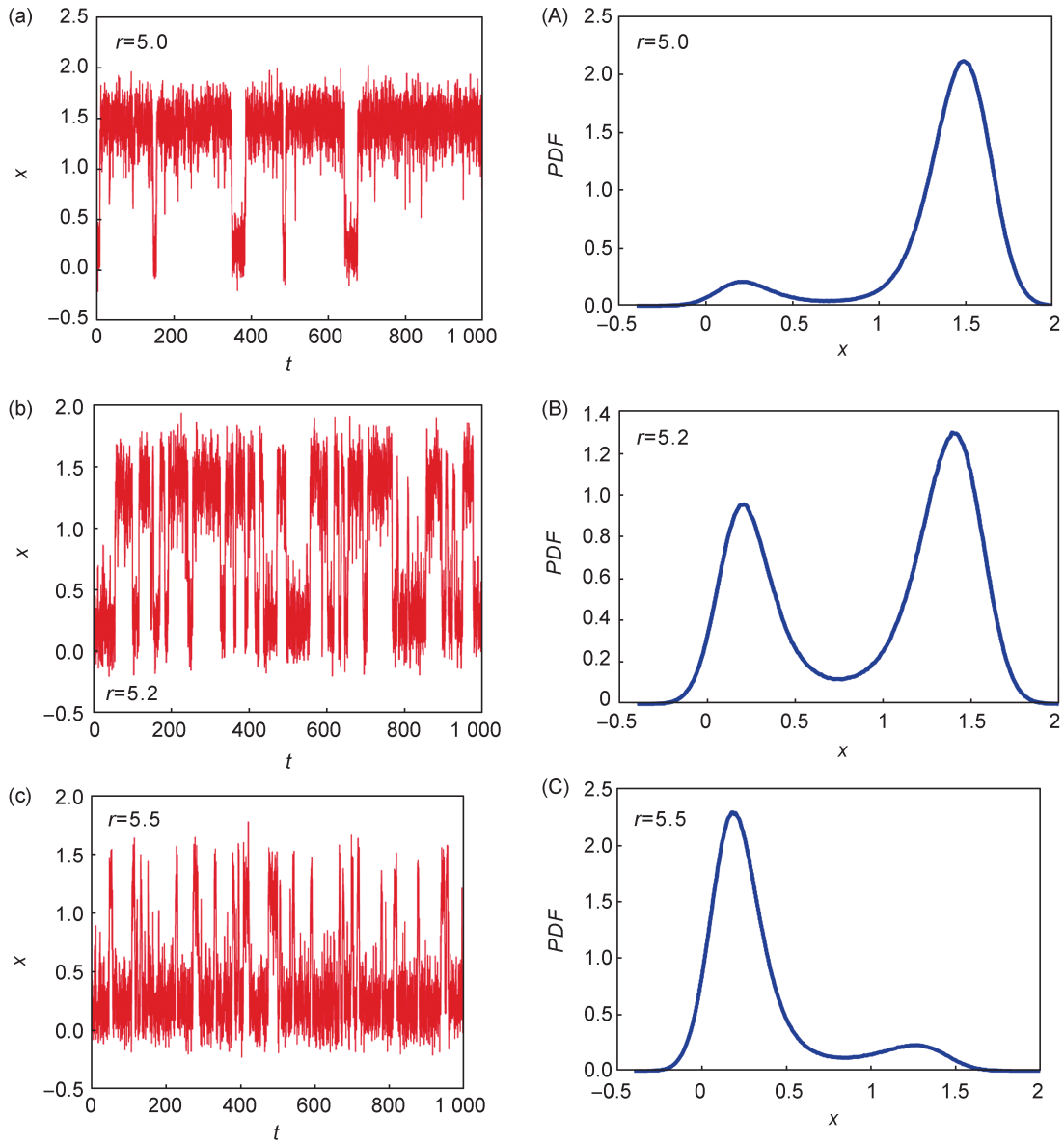


图5 不同r下模型的时间历程图和相应的PDF图

(a)  $r = 5.0$ 时的时间历程图;(b)  $r = 5.2$ 时的时间历程图;(c)  $r = 5.5$ 时的时间历程图;  
 (A)  $r = 5.0$ 时的PDF图;(B)  $r = 5.2$ 时的PDF图;(C)  $r = 5.5$ 时的PDF图。

Fig. 5 Time history plots and corresponding PDF plots of the model under different  $r$

(a) Time history plot when  $r = 5.0$ ; (b) time history plot when  $r = 5.2$ ; (c) time history plot when  $r = 5.5$ ;  
 (A) PDF plot when  $r = 5.0$ ; (B) PDF plot when  $r = 5.2$ ; (C) PDF plot when  $r = 5.5$ .

于式(1), 求解过程是马尔可夫过程, 且平稳响应的概率密度函数满足以下的FP方程<sup>[17-18]</sup>:

$$-\frac{\partial}{\partial x}[f(x)p(x)] + D\frac{\partial^2}{\partial x^2}p(x) = 0,$$

其中  $p(x)$  是方程(1)的概率密度函数, 它必须满足标准化条件并且当  $x$  趋于无穷大时,  $p(x)$  趋于零。

研究的基因开关模型对应的平稳FP方程为:

$$-\frac{\partial}{\partial x}\left[\left(\frac{k(2x^2 + 50x^4)}{25 + 29x^2 + 52x^4 + 4x^6} - rx + 1\right)p(x)\right] + D\frac{\partial^2}{\partial x^2}p(x) = 0,$$

整理后为:

$$-\left(\frac{k(100x + 5000x^3 + 2692x^5 - 32x^7 - 400x^9)}{(25 + 29x^2 + 52x^4 + 4x^6)^2} - r\right) \cdot p(x) - \left(\frac{k(2x^2 + 50x^4)}{25 + 29x^2 + 52x^4 + 4x^6} - rx + 1\right) \frac{\partial}{\partial x}p(x) +$$

$$D \frac{\partial^2}{\partial x^2} p(x) = 0.$$

2.2.2 深度学习求解方法介绍

接下来我们将介绍如何通过深度学习的方法求解FP方程<sup>[5, 19-20]</sup>。

图8是一个人工神经网络的典型结构,它也被称为具有L层的深度神经网络,其中x是输入层,y是输出层。 $H^l = [h_1^l, h_2^l, h_3^l, \dots, h_{n_l}^l]$  ( $2 \leq l \leq L-1$ )表示输出的第l-1个隐藏层, $n_l$ 是每层

的节点数。 $W = [\omega^1, \omega^2, \dots, \omega^l, \dots, \omega^{L-1}]$ 是网络的权重, $\omega^l$ 代表从第l层到第l+1层的权重。矩阵 $\omega^l$ 的大小为 $n_l \times n_{l+1}$ 。Bias =  $[b^1, b^2, \dots, b^l, \dots, b^{L-1}]$ 是层的偏差, $b^l$ 代表大小为 $n_l \times 1$ 的第l+1层的偏差。

图8所示的神经网络便是最简单的前馈神经网络,通常可以用来近似普通的连续型非线性函数。因此,我们可以将前馈过程描述为:

$$h_i^l = F(\omega_i^l x + b_i^l),$$

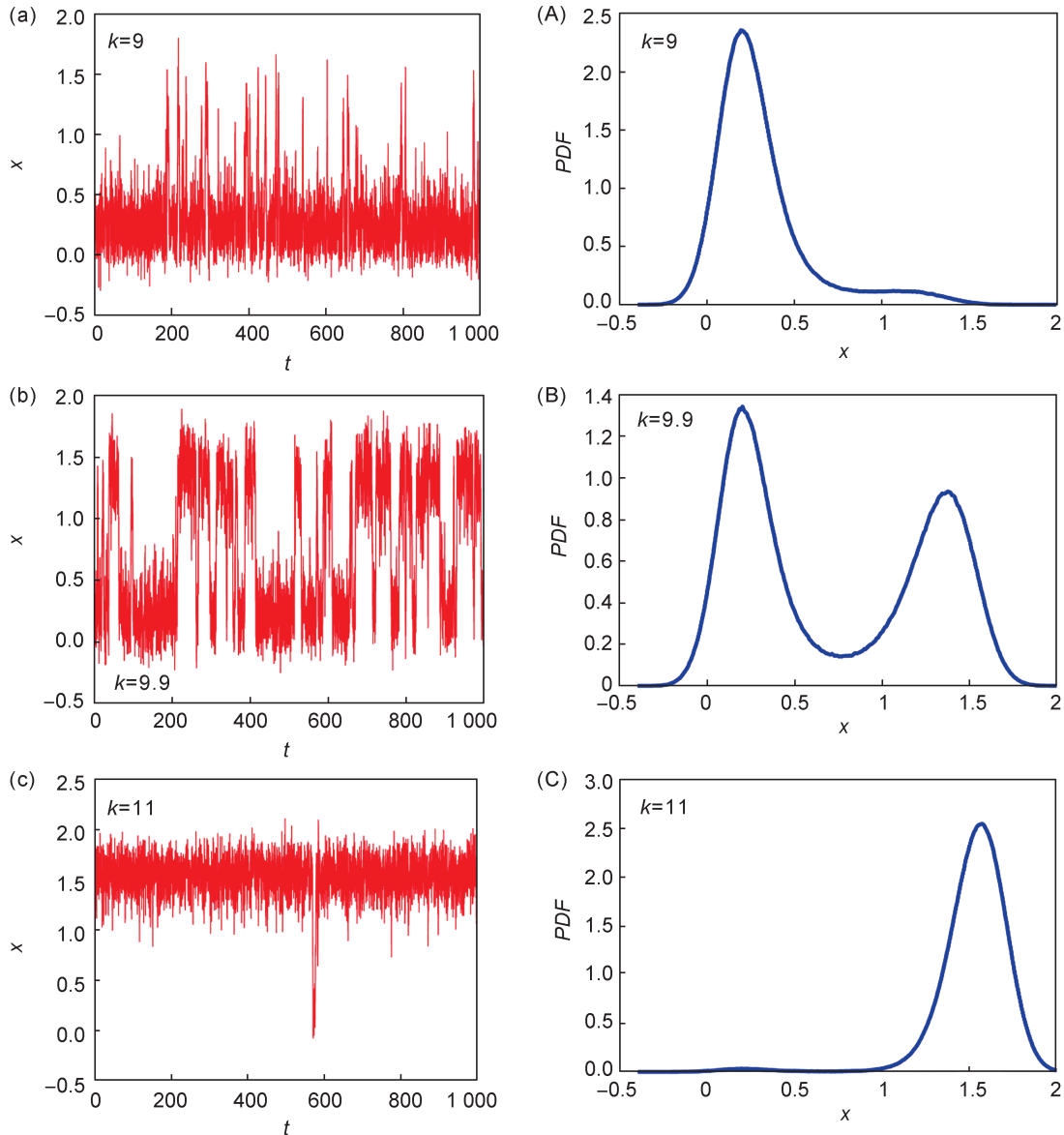


图6 不同k下模型的时间历程图和相应的PDF图

(a)  $k = 9$ 时的时间历程图;(b)  $k = 9.9$ 时的时间历程图;(c)  $k = 11$ 时的时间历程图;  
 (A)  $k = 9$ 时的PDF图;(B)  $k = 9.9$ 时的PDF图;(C)  $k = 11$ 时的PDF图。

Fig. 6 Time history plots and corresponding PDF plots of the model under different k

(a) Time history plot when  $k = 9$ ; (b) time history plot when  $k=9.9$ ; (c) time history plot when  $k=11$ ;  
 (A) PDF plot when  $k=9$ ; (B) PDF plot when  $k=9.9$ ; (C) PDF plot when  $k=11$ .

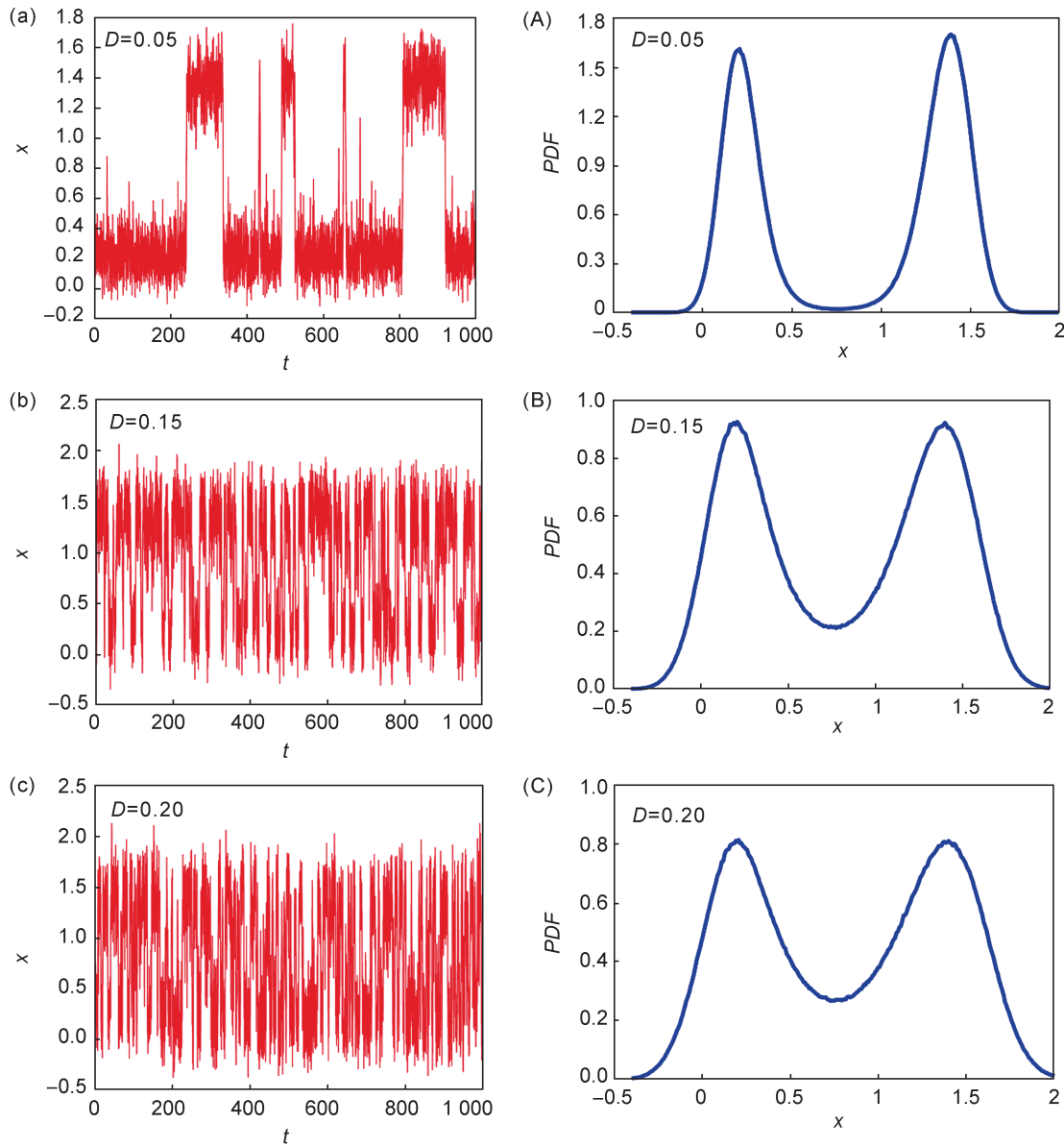


图7 不同D下模型的时间历程图和相应的PDF图

(a)  $D = 0.05$ 时的时间历程图; (b)  $D = 0.15$ 时的时间历程图; (c)  $D = 0.20$ 时的时间历程图;  
 (A)  $D = 0.05$ 时的PDF图; (B)  $D = 0.15$ 时的PDF图; (C)  $D = 0.20$ 时的PDF图。

Fig. 7 Time history plots and corresponding PDF plots of the model under different  $D$

(a) Time history plot when  $D = 0.05$ ; (b) time history plot when  $D = 0.15$ ; (c) time history plot when  $D = 0.20$ ;  
 (A) PDF plot when  $D = 0.05$ ; (B) PDF plot when  $D = 0.15$ ; (C) PDF plot when  $D = 0.20$ .

$$h_l^t = F\left(\sum_{j=1}^{n_{l-1}} \omega_{lj}^{t-1} h_j^{t-1} + b_l^{t-1}\right), 3 \leq l \leq L - 1,$$

$$y = F\left(\sum_{i=1}^{n_{L-1}} \omega_{Li}^{t-1} h_i^{t-1} + b^t\right),$$

其中  $F$  表示激活函数, 常用的激活函数有 Sigmoid、Tanh 和 ReLU 函数等, 输出可以描述为  $y = F(x; \theta)$ , 其中  $\theta = [W, \text{Bias}]$  表示为网络参数。通常还会在输出上添加绝对值函数, 以保

证 PDF 是非负的。

在求解微分方程时, 控制方程提供了一种灵活的方式, 通过调节外部控制变量或控制函数的值来实现对系统行为的调节和优化。微分方程中未知函数在求解之前所需满足的条件为初始条件; 需要在特定的区间端点满足一定的条件时求解微分方程的解, 这些条件称为边界条件, 边界条件是一个方程具有确定解的先决条件。通常, 当要求解的方程是平稳状态下的

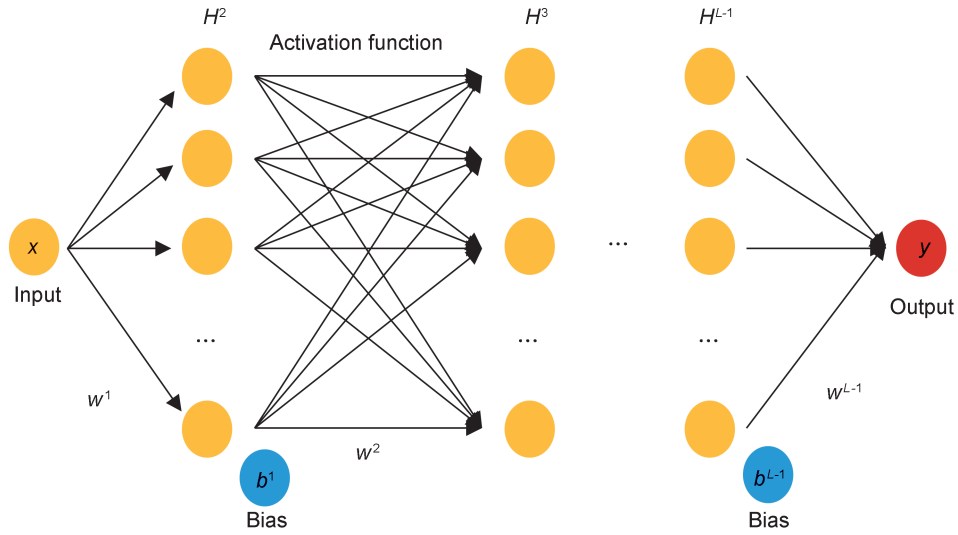


图8 深度神经网络的结构

Fig. 8 The structure of a deep neural network

方程时,只有边界条件而没有初始条件,对应使用人工神经网络求解微分方程时的约束条件为微分方程的控制方程和边界条件<sup>[1]</sup>:

$$\begin{cases} D(y(X_D))=f(X_D), X_D \in \Omega, \\ B(y(X_B))=g(X_B), X_B \in \partial\Omega, \end{cases}$$

其中  $D[\cdot]$  是包含关于  $X$  的导数的微分算子,  $B[\cdot]$  是边界算子。  $T_D = \{X_i^D \in \Omega\}_{i=1}^{N_D}$  的数据域和  $T_B = \{X_i^B \in \partial\Omega\}_{i=1}^{N_B}$  的数据边界是被离散化为人工神经网络的输入数据,相应的输出表示为  $\hat{y} = \hat{y}(X; \theta)$ , 其中  $N_D$  和  $N_B$  分别表示训练集和边界集中的数据量,  $\theta$  是人工神经网络的参数。损失函数表示为  $L = E_1 + E_2$ , 其中

$$E_1 = \frac{1}{N_D} \sum_{i=1}^{N_D} |D(\hat{y}(X_i^D; \theta)) - f(X_i^D)|^2,$$

$$E_2 = \frac{1}{N_B} \sum_{i=1}^{N_B} |B(\hat{y}(X_i^B; \theta)) - g(X_i^B)|^2.$$

然后,采用优化方法来训练网络,这种训练属于监督学习。损失部分  $E_1$  和  $E_2$  分别保证训练解满足给定方程的形式和边界条件。但是用这一方法处理许多FP方程往往是不可行的,仅仅只有方程形式和边界条件会使训练结果为零,需要添加归一化条件作为监督条件来避免这种困境。同时,由于增加了监督项,在高维案例中的训练结果将失效,有必要引入惩罚因素来解决这个问题,最终形成了用深度学习求解FP方程的方法(下面简称为DL-FP方法)<sup>[21]</sup>。

### 2.2.3 深度学习求解的具体操作过程

在本文中,我们借助深度学习来求解随机基因开关模型的概率密度函数,用以克服上文中提到的局限。除了以方程的形式和边界条件作为监督条件外,我们还将以归一化条件作为监督条件,可以避免近似解为零的情况。同时,为了很好地训练损失函数的三个分量,引入了惩罚因子。接下来,将描述用深度学习的方法求解的详细过程<sup>[1]</sup>。

首先要选择合适的训练集。我们知道人工神经网络的输入应当是有限集,因此要大致确定出FP方程自变量的取值范围。因为处于平衡状态的FP方程,当自变量远离平衡状态时概率为零,因此,必须选择一个足够大的范围。为了方便,我们可以根据前面的蒙特卡罗方法求解时的区间来确定训练区间。找到训练区间后,便可得到训练集,记步长为  $\Delta x$ 。在一维情况下,假设区间为  $[a, b]$ , 则  $N_p = (b - a) / \Delta x$ , 并且训练集中的数据量为  $N_D = N_p$ , 边界集中的数据量为  $N_B = 2$ 。将训练集和边界集作为同一个人工神经网络的输入,通过使用前馈神经网络得到它们的输出。

接着便是要确立损失函数。在使用深度学习的方法进行研究时,损失函数扮演了一个非常关键的角色。通过将损失函数最小化,从而达到减小预测值偏差的目的。可见合理地选取损失函数对模型有重要影响,因此我们要严格

定义损失函数。要想确保用深度学习的方法获得的FP方程的解是目标方程的解,应满足三个要求:方程的形式、概率密度函数的归一化条件和边界条件。因此,我们将损失函数定义为

$$L = \sum_{i=1}^3 a_i \cdot E_i, \text{ 其中}$$

$$E_1 = \frac{1}{N_D} \sum_{i=1}^{N_D} |D(\hat{y}(X_i^D; \theta)) - f(X_i^D)|^2, \quad (2)$$

$$E_2 = \left| \sum_{i=1}^{N_B} \Delta x \cdot \hat{y}(X_i^B; \theta) - 1 \right|^2, \quad (3)$$

$$E_3 = \frac{1}{N_B} \sum_{i=1}^{N_B} |B(\hat{y}(X_i^B; \theta)) - g(X_i^B)|^2, \quad (4)$$

这里  $a_i, i=1,2,3$  表示惩罚因子,同样用  $\hat{y}(X; \theta)$  表示近似解。方程(2)保证了近似解满足方程的形式,方程(3)保证了概率密度函数的归一化条件,方程(4)保证了边界条件,这三个监督条件缺一不可。

最后要使损失函数  $L$  最小化,本文使用Adam方法来最小化损失函数。同时,为了避免损失函数陷入局部最小值,我们设计了惩罚因子来影响监督训练的过程。通过我们不断地实验,发现各惩罚因子在一维情况下可以视为1,在高维情况下,各惩罚因子的取值要根据方程的具体情况进行设置。

将此方法对应到我们研究的基因开关模型,我们设置惩罚因子为  $a_1=1, a_2=1, a_3=1$  以及变量  $x$  的训练区间为  $[-1, 2]$ , 则  $p(-1)=0, p(2)=0$ 。我们可以得到步长为  $\Delta x=0.01$  的训

练集和边界集为  $\{-1, 2\}$ 。选择激活函数为双曲正切函数,设置网络的大小为4层,每个隐藏层25个节点,应用Adam优化技术,在  $3.5 \times 10^4$  次迭代后损失函数的最小值约为  $10^{-6}$ 。

### 2.3 方法对比

接下来将举几个不同参数的例子来对比说明两种方法。

将两种方法放在一起,通过图9—图11对比可以发现,用深度学习方法所得的基因开关模型的概率密度函数图与用蒙特卡罗方法所得图形一致,但放大一些局部细节后发现,新方法所得图像更光滑,精度更高,根据表1对比也可以发现新方法耗时更短。

### 3 隐藏层数与节点数对计算性能的影响

对于不同的模型来说,隐藏层数以及节点数的取值往往是不同的,需要考虑诸多影响因素。若隐藏层数少,可能会使误差较大,精度不高;若隐藏层数多,反而可能出现过拟合的情况。若节点数少,会使模型效果不显著;若节点数多,可能陷入局部最优,甚至可能出现过拟合的情况。可见,隐藏层数与节点数的正确选择对于一个神经网络的研究至关重要。以固定  $r=5, k=10, D=0.2$  为例考虑模型,假设网络有1、2、3、4、5和6个隐藏层,每个隐藏层有25个节点。在图12可以清楚地观察到当隐藏层为4时,损失函数将达到  $10^{-6}$  的数量级。然而,当隐藏层数超过4时,损失函数的趋势反

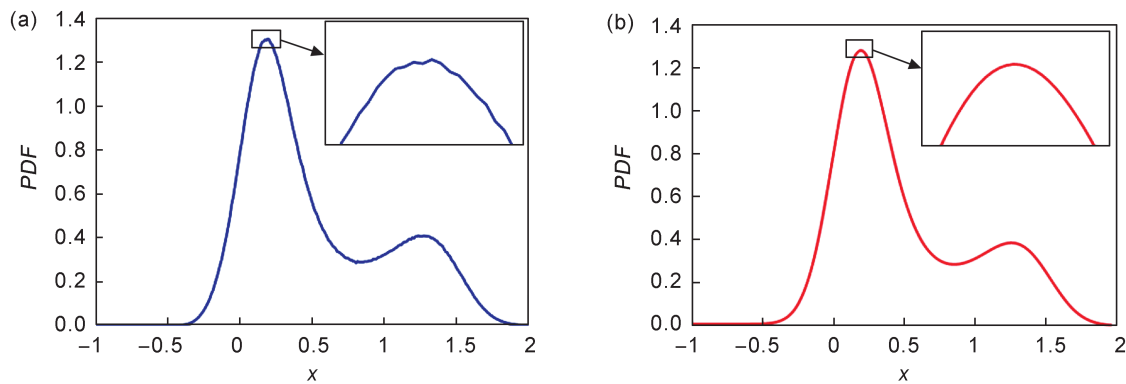


图9 用DL-FP算法与蒙特卡罗方法在固定  $r=5.5, k=10, D=0.2$  时得到的解的比较结果  
(a)为蒙特卡罗解,(b)为深度学习解,里面的小图为其局部放大图。

Fig. 9 A comparison of the solutions obtained by using the DL-FP algorithm with using the Monte Carlo method under fixed  $r=5.5, k=10, D=0.2$

(a) is the Monte Carlo solution; (b) is the deep learning solution, and the small images inside are their locally enlarged images.

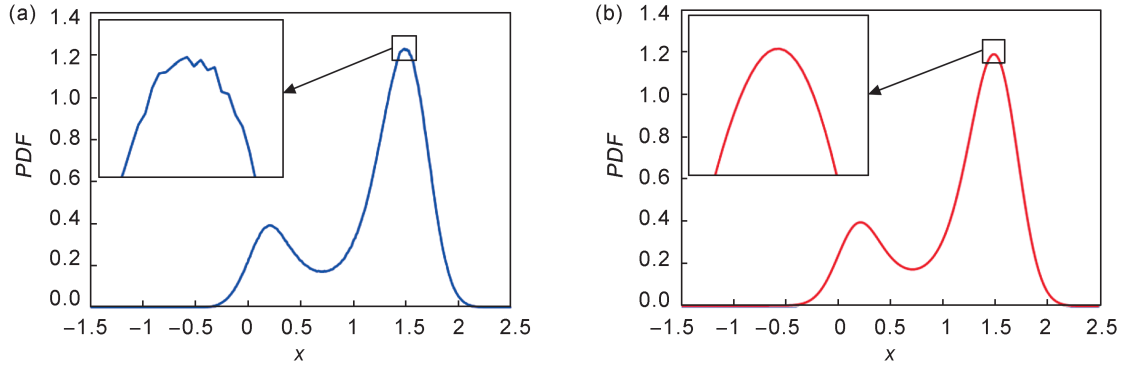


图10 用DL-FP算法与蒙特卡罗方法在固定 $r=5, k=10, D=0.2$ 时得到的解的比较结果

(a)为蒙特卡罗解; (b)为深度学习解,里面的小图为其局部放大图。

Fig. 10 A comparison of the solutions obtained by using the DL-FP algorithm with using the Monte Carlo method under fixed  $r=5, k=10, D=0.2$

(a) is the Monte Carlo solution, (b) is the deep learning solution, and the small image inside is its locally enlarged image.

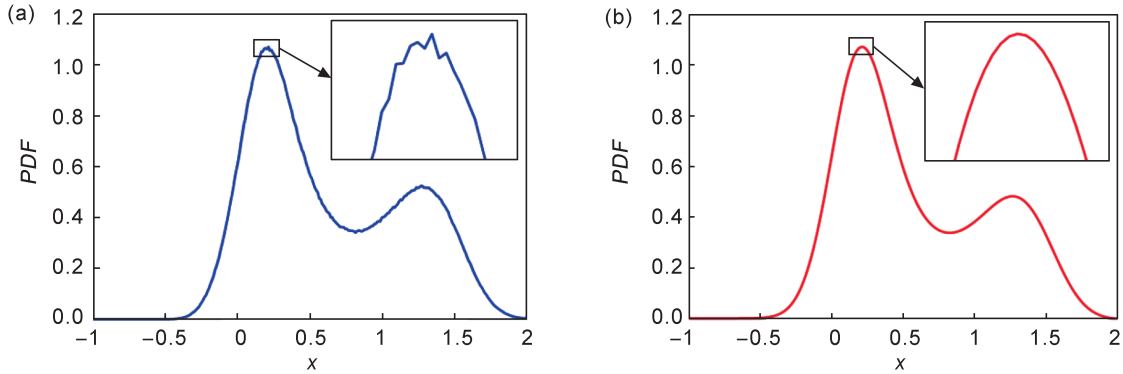


图11 用DL-FP算法与蒙特卡罗方法在固定 $r=5, k=9, D=0.2$ 时得到的解的比较结果

(a)为蒙特卡罗解; (b)为深度学习解,里面的小图为其局部放大图。

Fig. 11 A comparison of the solutions obtained by using the DL-FP algorithm with using the Monte Carlo method under fixed  $r=5, k=9, D=0.2$

(a) is the Monte Carlo solution, (b) is the deep learning solution, and the small image inside is its locally enlarged image.

表1 DL-FP算法与蒙特卡罗方法耗时对比

Table 1 Time consuming comparison of the DL-FP algorithm with the Monte Carlo method

参数	方法	耗时/s
$r=5.5, k=10, D=0.2$	蒙特卡罗模拟	181.62
	DL-FP算法	122.28
$r=5.0, k=10, D=0.2$	蒙特卡罗模拟	193.69
	DL-FP算法	118.42
$r=5.0, k=9, D=0.2$	蒙特卡罗模拟	157.88
	DL-FP算法	102.49

而变差。因此,隐藏层的最佳数量是4层,每层有25个节点。同时我们发现,更多的隐藏层确实提高了达到高精度的速度。

同样,假设网络有4个隐藏层,每个隐藏层有15、20、25、30和35个节点。在图13中可以看

到当每个隐藏层的节点数为25时,损失函数将达到 $10^{-6}$ 的数量级。然而,当节点数超过25时,损失函数的计算精度反而逐渐下降。因此,隐藏层的最佳数量是4层,每层有25个节点。

根据前面的分析可知,隐藏层数和节点数均会对计算精度产生影响,在具体分析方程时要具体考虑,并不是隐藏层数越多,节点数越多精度越高。

#### 4 结论

本文使用深度学习对随机基因开关模型的概率密度函数进行估计,来说明深度学习方法具有很高的准确性。与传统的求解随机基因开关模型概率密度函数的方法相比,深度学习方

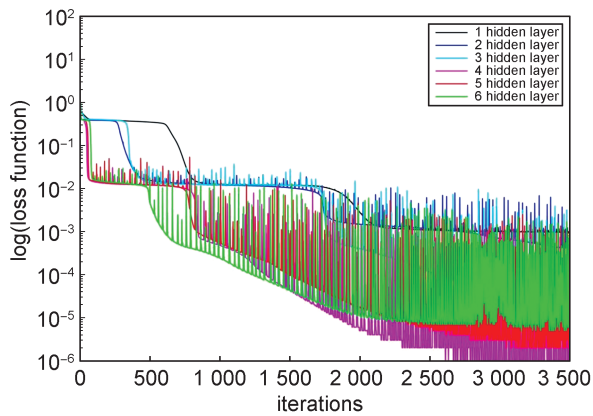


图12 不同隐藏层数的估计精度(每层25个节点)

Fig. 12 The estimation accuracy of different hidden layers (25 nodes per layer)

法不需要插值以及其他不同于有限元法和有限差分算法的重构技术,得到的解不需要任何坐标变换,且比蒙特卡罗模拟的结果更平滑,所耗时间更短。此外,还讨论了人工神经网络中隐藏层数和节点数的设置,发现在网络的设置中,并不是更深的隐藏层以及更多的节点数获得的精度越高,通过适当构建神经网络可以提高机器学习技术的性能。在训练所需的迭代次数方面,我们建议根据损失函数的精度来确定训练次数是更合理的。

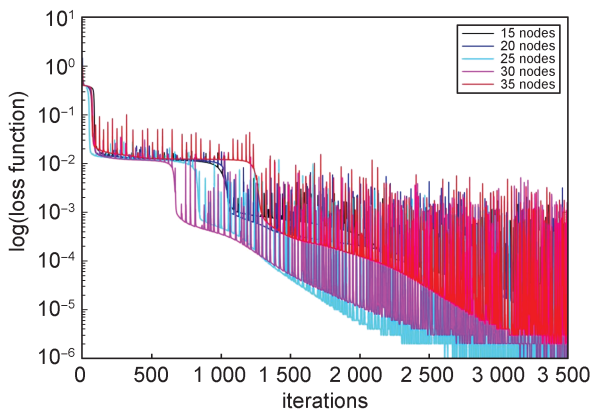


图13 不同节点数的估计精度(4个隐藏层)

Fig. 13 The estimation accuracy of different number of nodes (4 hidden layers)

因此,通过前面的对比可知,与传统的求解方法相比,用深度学习的方法估计随机基因开关模型的概率密度函数是可行的,并且深度学习的方法克服了传统方法的一些缺陷。它可以学习分析解、解的搜索也不需要任何坐标变换,并且随着采集样本点的个数增加,计算复杂度不会快速增加,可以快速计算出解的值,

不会像用蒙特卡罗模拟那样耗时严重。值得一提的是,本文仅以简单的一维基因开关模型为例应用了深度学习的方法,之后还可以研究更高维的情况,来对比出深度学习方法的优势。

#### 参考文献:

- [1] XU Y, ZHANG H, LI Y G, *et al.* Solving Fokker-Planck Equation Using Deep Learning[J]. *Chaos Interdiscip J Nonlinear Sci*, 2020, **30**(1): 013133. DOI: 10.1063/1.5132840.
- [2] SUN J Q. *Stochastic Dynamics and Control*[M]. Boston: Elsevier, 2006.
- [3] LAGARIS I E, LIKAS A, FOTIADIS D I. Artificial Neural Networks for Solving Ordinary and Partial Differential Equations[J]. *IEEE Trans Neural Netw*, 1998, **9**(5): 987-1000. DOI: 10.1109/72.712178.
- [4] SHEKARI BEIDOKHTI R, MALEK A. Solving Initial-boundary Value Problems for Systems of Partial Differential Equations Using Neural Networks and Optimization Techniques[J]. *J Frankl Inst*, 2009, **346**(9): 898-913. DOI: 10.1016/j.jfranklin.2009.05.003.
- [5] HAN J Q, JENTZEN A, WEINAN E W N. Solving High-dimensional Partial Differential Equations Using Deep Learning[J]. *Proc Natl Acad Sci USA*, 2018, **115**(34): 8505-8510. DOI: 10.1073/pnas.1718942115.
- [6] 周念成, 廖建权, 王强钢, 等. 深度学习在智能电网中的应用现状分析与展望[J]. *电力系统自动化*, 2019, **43**(4): 180-191. DOI: 10.7500/AEPS20180323002.  
ZHOU N C, LIAO J Q, WANG Q G, *et al.* Analysis and Prospect of Deep Learning Application in Smart Grid[J]. *Autom Electr Power Syst*, 2019, **43**(4): 180-191. DOI: 10.7500/AEPS20180323002.
- [7] SONG Y, XU W, JIAO Y Y. Bifurcation- and Noise-induced Tipping in Two-parametric Gene Transcriptional Regulatory System[J]. *Eur Phys J Plus*, 2021, **137**: 68. DOI: 10.1140/epjp/s13360-021-02300-3.
- [8] 冯晶. Lévy噪声激励下几类典型生物系统的分岔和首次穿越[D]. 西安: 西北工业大学, 2013.  
FENG J. Bifurcation and First Crossing of Several Typical Biological Systems under Lévy Noise Excitation[D]. Xi'an: Northwestern Polytechnical University, 2013.
- [9] MA J Z, XU Y, XU W, *et al.* Slowing Down Critical Transitions via Gaussian White Noise and Periodic Force[J]. *Sci China Technol Sci*, 2019, **62**(12): 2144-2152. DOI: 10.1007/s11431-019-9557-2.
- [10] MA J Z, XU Y, LI Y G, *et al.* Predicting Noise-induced Critical Transitions in Bistable Systems[J]. *Chaos Inter-*

- discip J Nonlinear Sci*, 2019, **29**(8): 081102. DOI: 10.1063/1.5115348.
- [11] 冯晶, 许勇, 李娟娟. 非高斯噪声激励下的基因转录调控系统[J]. *动力学与控制学报*, 2012, **10**(4): 325-328. DOI: 10.1103/PhysRevE.83.056215.
- FENG J, XU Y, LI J J. A Genetic Regulatory Model with a Kind of Non-Gaussian Noise[J]. *J Dyn Control*, 2012, **10**(4): 325-328. DOI: 10.1103/PhysRevE.83.056215.
- [12] XU Y, FENG J, LI J J, *et al.* Lévy Noise Induced Switch in the Gene Transcriptional Regulatory System[J]. *Chaos Interdiscip J Nonlinear Sci*, 2013, **23**(1): 013110. DOI: 10.1063/1.4775758.
- [13] TOCINO A, ARDANUY R. Runge-Kutta Methods for Numerical Solution of Stochastic Differential Equations [J]. *J Comput Appl Math*, 2002, **138**(2): 219-241. DOI: 10.1016/s0377-0427(01)00380-6.
- [14] WANG X L, FENG J, LIU Q, *et al.* Neural Network-based Parameter Estimation of Stochastic Differential Equations Driven by Lévy Noise[J]. *Phys A Stat Mech Appl*, 2022, **606**: 128146. DOI: 10.1016/j.physa.2022.128146.
- [15] 王参军. 基因转录调控系统中的色噪声诱导转化研究[J]. *物理学报*, 2012, **61**(1): 59-66. DOI: 10.3969/j.issn.1673-260X.2015.16.005.
- WANG C J. Colored Noise Induced Switch in the Gene Transcriptional Regulatory System[J]. *Acta Phys Sin*, 2012, **61**(1): 59-66. DOI: 10.3969/j.issn.1673-260X.2015.16.005.
- [16] 王九云. 基因转录调节系统中非线性时间延迟效应的研究[D]. 武汉: 华中师范大学, 2006.
- WANG J Y. Study on Nonlinear Time Delay Effect in Gene Transcription Regulation System[D]. Wuhan: Central China Normal University, 2006.
- [17] 胡岗. 随机力与非线性系统[M]. 上海: 上海科技教育出版社, 1994.
- HU G. Stochastic Forces and Nonlinear Systems[M]. Shanghai: Shanghai Scientific & Technological Education Publishing House, 1994.
- [18] HIGHAM D J. An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations[J]. *SIAM Rev*, 2001, **43**(3): 525-546. DOI: 10.1137/s0036144500378302.
- [19] CYBENKO G. Approximation by Superpositions of a Sigmoidal Function[J]. *Math Control Signal Syst*, 1989, **2**(4): 303-314. DOI: 10.1007/BF02551274.
- [20] WEINAN E, WANG Q C. Exponential Convergence of the Deep Neural Network Approximation for Analytic Functions[J]. *Sci China Math*, 2018, **61**(10): 1733-1740. DOI: 10.1007/s11425-018-9387-x.
- [21] BURY T M, SUJITH R I, PAVITHRAND I, *et al.* Deep Learning for Early Warning Signals of Tipping Points[J]. *PNAS*, 2021, **118**(39): e2106140118. DOI: 10.1073/pnas.2106140118.