



# 长短期记忆网络预测混沌时间序列

熊有成, 赵鸿\*

厦门大学物理科学与技术学院, 厦门 361005

\*联系人, E-mail: zhaoh@xmu.edu.cn

收稿日期: 2019-04-09; 接受日期: 2019-04-23; 网络出版日期: 2019-05-31

国家自然科学基金(编号: 11335006)资助项目

**摘要** 近年来,深度学习技术的快速发展推动了机器学习的广泛应用. 其中,储备池计算(Reservoir Computing)方法由于在混沌时间序列预测方面的卓越效果而受到了越来越多的重视,形成了一个新的研究热点. 本文以传统的长短期记忆网络(LSTM)和全连接层作为模型的基础,构建了基于循环神经网络的LSTM学习机. 在仿真实验中,我们以平均有效预测时间作为度量指标,使用模型对洛伦兹系统的状态变量进行预测,并针对混沌系统特殊的动力学性质,为模型搭配了四项策略来辅助预测. 结果表明,LSTM学习机搭配合适的组合策略,其预测能力也能达到与储备池计算相比拟的效果,且模型复杂度更低. 因此,储备池计算并没有超越于传统方法的绝对优势,这启发我们通过进一步研究学习机预测时间序列的机制和方法,寻找更加有效的混沌时间序列预测学习机.

**关键词** 机器学习, 混沌时间序列, 预测, 长短期记忆网络

**PACS:** 05.45.Tp, 07.05.Mh, 87.85.dq

## 1 引言

非线性动力学系统在自然界中大量存在,如天气系统、生态系统、金融系统等. 这类系统的典型特征是其状态随时间的变化表现为混沌运动,对初始条件具有敏感依赖性,由于受噪声和计算机精度的影响,混沌系统不可能被长时间预测. 因此,利用实际系统某一状态的输出时间序列,在未知系统运动方程的情况下,如何尽可能长地预测后续时间序列的变化成为具有重要应用价值的科学问题,长期以来受到了不同领域研究者的重视. 近年来,随着

深度学习的巨大突破,学习机(Learning Machine)技术在自然科学的各个领域获得了广泛的应用,不仅在传统领域如图像分类<sup>[1]</sup>和下围棋<sup>[2]</sup>等方面取得了令人瞩目的成功,而且也推动了时间序列预测等课题的迅速发展. 2018年,科学家将储备池计算(Reservoir Computing)<sup>[3]</sup>方法应用于一类时空动力学系统的高维混沌时间序列的预测<sup>[4]</sup>,引起了研究者对储备池计算的高度重视,使得机器学习预测时间序列成为新的前沿热点. 另一方面,传统上已经发展出了一些可以用于时间序列预测的人工神经网络方法,如循环神经网络学习机<sup>[5]</sup>,它作为深度

**引用格式:** 熊有成, 赵鸿. 长短期记忆网络预测混沌时间序列. 中国科学: 物理学 力学 天文学, 2019, 49: 120501

Xiong Y C, Zhao H. Chaotic time series prediction based on long short-term memory neural networks (in Chinese). Sci Sin-Phys Mech Astron, 2019, 49: 120501, doi: 10.1360/SSPMA-2019-0115

学习的重要分支,不仅在自然语言处理<sup>[6]</sup>和语音识别<sup>[7]</sup>方面展现出了强大的能力,而且也被用于时间序列预测.

传统的混沌时间序列预测方法有基于实际观测数据的相空间重构法<sup>[8]</sup>和Volterra自适应滤波器<sup>[9]</sup>等.由于神经网络强大的非线性拟合能力,在混沌时间序列的重构相空间中,利用神经网络逼近相点演化规律的方法,在混沌时间序列上也已经有了大量应用,如多层感知机<sup>[10]</sup>、径向基函数<sup>[11]</sup>和循环神经网络<sup>[12]</sup>.循环神经网络(RNN)是一类专门用于处理序列数据的神经网络.它能将时间序列信息编码成隐藏层状态,使网络学习到不同时刻序列之间的关联.在实际的应用中,RNN容易受梯度消失或梯度爆炸<sup>[13]</sup>的影响,使得训练起来异常困难.储备池计算也属于循环神经网络的一种,最近关于储备池计算的工作展现出它在混沌系统中强大的预测能力<sup>[14,15]</sup>.值得注意的是,储备池计算由于不需要训练隐藏状态到隐藏状态的神经元参数,能够避免梯度消失和爆炸的问题.受此启发,在预测混沌系统的任务中,我们选择长短期记忆网络(LSTM)<sup>[16]</sup>作为基础模型来一定程度上缓解该问题.据调研,基于LSTM预测的方法已经被运用在低维混沌系统中<sup>[17,18]</sup>,但其预测精度远远不如储备池计算<sup>[15]</sup>.

学习机的效果依赖于很多因素,不同的初始化、不同的优化算法、不同的训练策略等都会导致巨大的差异.因此一个学习机是否具有好的效果,需要优化了这些因素之后才能给出可靠的判断.本研究基于循环神经网络的长短期记忆网络学习机,结果表明通过选择合适的组合策略,能够有效预测混沌时间序列,达到和储备池计算相比拟的结果.

## 2 模型

简要介绍循环神经网络和长短期记忆网络的概念.RNN就像一个标准的多层感知机,关键的不同是在延迟的时间步上连接起隐藏神经元.给定一段输入 $(x_1, x_2, \dots, x_t)$ ,RNN依靠下面的方程,迭代产生一段输出 $(y_1, y_2, \dots, y_t)$ :

$$\begin{aligned} h_t &= \sigma(W_{hh}h_{t-1} + W_{hi}x_t + b_h), \\ y_t &= W_{oh}h_t + b_o, \end{aligned} \quad (1)$$

其中,  $x_t \in \mathbb{R}^{d_i}$ ,  $y_t \in \mathbb{R}^{d_o}$ ,  $h_t \in \mathbb{R}^{d_h}$ 是RNN在时间步数 $t$ 上的输入、输出和隐藏状态,  $W_{hi} \in \mathbb{R}^{d_h \times d_i}$ ,  $W_{oh} \in \mathbb{R}^{d_o \times d_h}$ ,  $W_{hh} \in \mathbb{R}^{d_h \times d_h}$ 分别是输入到隐藏状态、隐藏状态到输出和隐藏状态到隐藏状态的权值矩阵,  $b_h \in \mathbb{R}^{d_h}$ 和 $b_o \in \mathbb{R}^{d_o}$ 是对应的偏差值,  $\sigma$ 是网络的激活函数.

RNN被证明能有效地处理序列问题,然而实际应用中,由于RNN编码序列长,在反向传播算法的优化中容易受到梯度消失和梯度爆炸的影响,难以捕获到序列之间的长期依赖.而LSTM作为缓解该问题最成功的模型被广泛使用.因此在预测混沌系统的任务中,我们选择LSTM作为基础单元,其内部的计算如下:

$$\begin{aligned} g_t^f &= \sigma(W_f[h_{t-1}, x_{t-1}] + b_f), \\ g_t^i &= \sigma(W_i[h_{t-1}, x_{t-1}] + b_i), \\ \tilde{C}_t &= \tan h(W_C[h_{t-1}, x_{t-1}] + b_C), \\ C_t &= g_t^f C_{t-1} + g_t^i \tilde{C}_t, \\ g_t^o &= \sigma(W_o[h_{t-1}, x_{t-1}] + b_o), \\ h_t &= g_t^o \tanh(C_t). \end{aligned} \quad (2)$$

我们将内部计算如上的计算单元,称作LSTM单元.其中,  $g_t^f, g_t^i, g_t^o \in \mathbb{R}^{d_h}$ 是遗忘门、输入门和输出门的信号.  $x_t \in \mathbb{R}^{d_i}$ 是输入信号,  $h_t \in \mathbb{R}^{d_h}$ 是隐藏状态,  $C_t \in \mathbb{R}^{d_h}$ 是细胞状态,  $W_f, W_i, W_o, W_C \in \mathbb{R}^{d_h \times (d_h + d_i)}$ 是控制每个门输出的权值矩阵,  $b_f, b_i, b_C, b_o \in \mathbb{R}^{d_h}$ 是每个门的偏差值,激活函数 $\sigma$ 为sigmoid函数.LSTM在循环过程中,将输入的历史时间序列编码成隐藏状态和细胞状态,和一般的循环神经网络不同的是,通过门的设计,LSTM可以自适应地控制积累信息的强弱.例如,遗忘门 $g_t^f$ 点乘 $C_{t-1}$ ,可以通过 $g_t^f$ 的大小决定有多少信息记入下一时刻的 $C_t$ .同时,LSTM还构造了一条基于时间的通道 $C_t$ ,梯度在这条通道上回传时不会连乘上隐藏状态到隐藏状态的矩阵,使得训练时能够缓解梯度消失和爆炸的影响.

我们使用LSTM单元作为基础元素构造了两种模型用于训练和预测,分别称作训练模型和预测模型.值得注意的是,两种模型的参数是共享的,将训练模型训练好后,再把其中的参数迁移到预测模型用于预测.

## 2.1 训练模型

在训练模型中, 我们往模型中输入一段长度为 $W$ 的时间序列 $(x_0, x_1, \dots, x_{W-1})$ , 并使用模型做直接单步预测,  $W$ 代表窗口长度. 在初始的时间步 $t = 0$ 时, 我们将两个全为零的向量 $\mathbf{h}_0$ 和 $\mathbf{C}_0$ 作为初始隐藏状态和初始细胞状态输入到模型. 在时间步 $t = i(0 < i \leq W)$ 时, 模型通过输入 $x_i, h_{i-1}$ 和 $C_{i-1}$ 计算下一时刻的 $h_i$ 和 $C_i$ , 它们包含了编码自时间序列 $(x_0, x_1, \dots, x_i)$ 中的信息, 接着通过将 $h_i$ 输入全连接层得到该时刻的输出 $\tilde{y}_i$ . 下一步再把 $h_i, C_i$ 和 $x_{i+1}$ 代入LSTM计算单元输出 $h_{i+1}$ 和 $C_{i+1}$ . 通过这样的循环, 就产生了一组预测输出 $(\tilde{y}_0, \tilde{y}_1, \dots, \tilde{y}_{W-1})$ . 训练模型如图1所示.

在训练阶段, 我们的目标是使模型输出的 $(\tilde{y}_0, \tilde{y}_1, \dots, \tilde{y}_{W-1})$ 尽可能地逼近真实的观测值 $(x_1, x_2, \dots, x_W)$ . 我们将真实的观测值用 $(y_0, y_1, \dots, y_{W-1})$ 代替, 并使用预测时间序列与真实观测值的均方误差(MSE)作为训练阶段的损失函数:

$$L = \frac{1}{W} \sum_{t=0}^{W-1} \|y_t - \tilde{y}_t\|^2. \quad (3)$$

为了更新LSTM单元和全连接层的参数, 我们通过基于时间的反向传播算法(BPTT)<sup>[19]</sup>来最小化损失函数, 设置学习率为 $l$ , 并选择在深度学习任务中表现优秀的Adam算法<sup>[20]</sup>作为优化算法.

## 2.2 预测模型

当参数在训练模型中优化好后, 我们将所学得的LSTM单元和全连接层的参数迁移至预测模型. 预测模型分为预热阶段和预测阶段. 在预热阶段, 我们首先接受一段长度为 $T$ 的时间序列 $(x_0, x_1, \dots, x_{T-1})$ , 同样以全为零的向量作为初始状态来启动LSTM, 并进行和训练架构一样的迭代, 经过 $T$ 步的迭代得到 $h_T$ 和 $C_T$ . 在预热阶段由于受全为零的初始状态影响, 模型输出精度十分低, 所以我们将其舍弃, 只保留最后一步的 $h_T$ 和 $C_T$ . 由于最后一步的 $h_T$ 和 $C_T$ 已经记忆好预热阶段时间序列的信息, 模型可从中产生精准的预测, 故将其作为预测阶段的初始状态.

在预测阶段, 和训练模型不同的是使用预测模型做间接多步预测, 不再把真实的观测值在每个时间步输入, 而是迭代地将模型预测的输出作为下一步模型的输入, 由此可产生指定长度为 $P$ 的一段时间序列 $(\tilde{y}_T, \tilde{y}_{T+1}, \dots, \tilde{y}_{T+P})$ 作为模型的预测序列. 预测模型可表示为图2.

## 2.3 训练策略

为了能更有效率地训练网络, 同时更好地学到混沌时间序列内在的动力学特性. 经过大量尝试和比较, 我们决定在训练时加入了如下4项适合混沌时

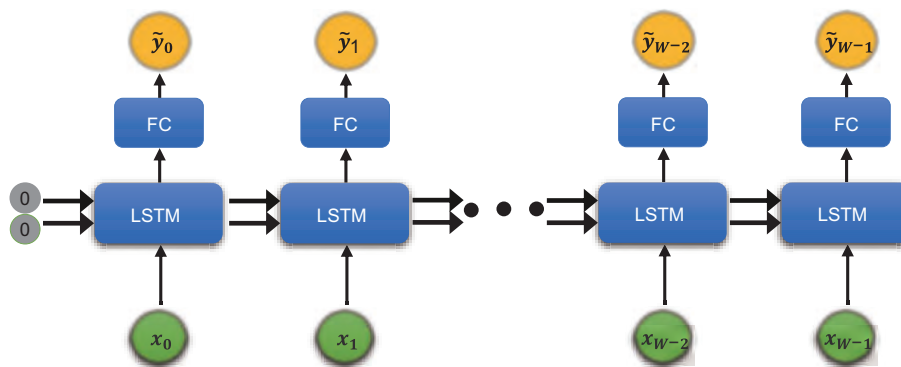
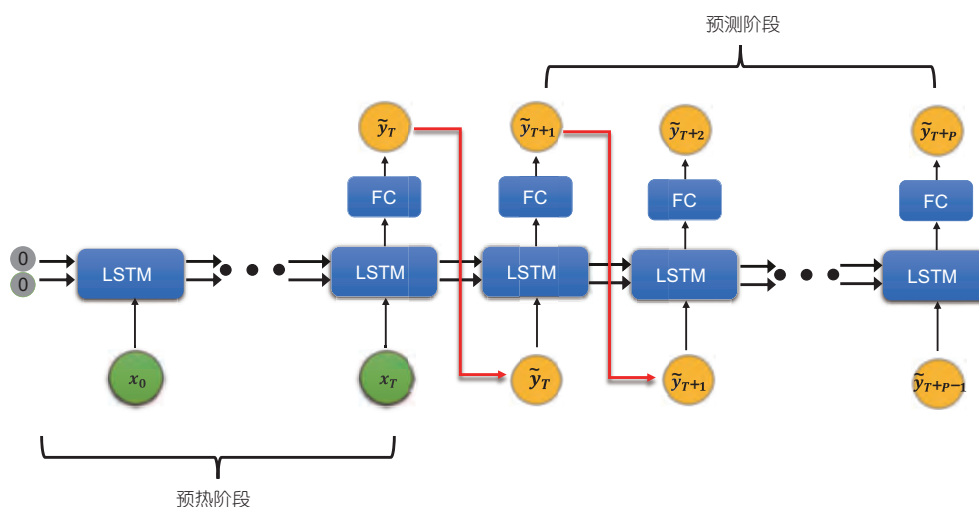


图1 (网络版彩图)训练模型. 左边两个标记为零的灰色节点为初始隐藏状态, 下面的绿色节点为每个时间步输入的真实观测值, 上面的黄色节点为每个时间步预测的下一步的输出, 中间标记为LSTM和FC的蓝色方块代表LSTM单元和全连接层

Figure 1 (Color online) The training model. The two gray nodes marked zero on the left are initially hidden states, the green nodes below are true observations for each time step, the yellow nodes above are the prediction for the next time step, the blue squares labeled LSTM and FC in the middle represent LSTM units and fully connected layers.



**图 2** (网络版彩图)预测模型. 预测模型分为预热阶段和预测阶段. 预热阶段的模型和训练模型一样, 但只需要最后一步的输出  $\tilde{y}_T$ ,  $h_T$  和  $C_T$  去启动预测阶段. 在预测阶段, 通过迭代产生  $P$  步的输出, 需要注意的是, 红色线代表模型将预测值重新输入模型进行间接多步预测, 而不是像训练模型一样输入真实的观测值

**Figure 2** (Color online) The prediction model. The prediction model is divided into a warm-up phase and a predictive phase. The warm-up phase is the same as the training model, but only need the last output  $\tilde{y}_T$ ,  $h_T$  and  $C_T$  to start up the prediction phase. In the prediction phase, we generate the output of  $P$ -step by iterating. It should be noted that the red line represents re-entering the prediction into the model for multi-step predictions, rather than input the true observations as the training model does.

间序列的关键策略.

标准化和还原策略. 标准化预处理是机器学习领域常见的预处理方式. 多维混沌时间序列在各个维度上均值方差往往差异较大, 且我们使用的损失函数是基于欧式距离的均方差损失函数, 为了使各个维度对损失函数的影响权重一致, 对时间序列数据使用标准化的预处理:

$$X = \frac{X - \text{mean}(X)}{\text{var}(X)}, \quad (4)$$

其中,  $\text{mean}(X)$ ,  $\text{var}(X) \in \mathbb{R}^d$  代表数据集  $X$  的均值和方差. 接着, 对模型输出结果  $Y$ , 做还原处理:

$$Y = Y \times \text{var}(X) + \text{mean}(X). \quad (5)$$

标准化和还原策略使得数据在各个维度的尺度一致, 导致反向传播算法时每个维度对模型参数影响一致, 模型可以取得更快的收敛效果和更好的泛化误差.

梯度截断策略. 循环神经网络的参数空间通常存在像悬崖一样斜率较大的区域, 这是因为随着时间步数, 较大参数的连乘导致参数空间尖锐而非线性. 在这样的悬崖区域会产生非常大的梯度, 梯度

下降更新使得参数弹射到很远的地方, 可能使得之前已完成的大量优化成为无用功. 因此机器学习领域在使用循环神经网络的任务中, 会在参数更新之前截断梯度的范数, 这一策略叫做梯度截断<sup>[13]</sup>. 在训练过程中, 梯度截断会为梯度  $g$  的范数定义一个阈值 (Threshold), 当梯度范数大于这个阈值时, 做如下截断操作:

$$g = \frac{\text{threshold}}{\|g\|} g. \quad (6)$$

考虑到混沌系统具有极端的初值敏感性, 极小的预测误差会随着演化呈指数增长. 因此在循环神经网络模拟混沌系统的动力学行为时, 前期的误差也会随着时间积累呈爆炸式增长, 容易带来极大的梯度, 使参数无法稳步地收敛到最优值. 综合循环神经网络本身的优化问题和混沌系统所具有的特性, 我们决定在训练过程中使用梯度截断的策略, 并调试阈值以达到最好的效果.

基于储备池计算的初始化策略. 大量实验表明, 神经网络参数的初始化对网络训练速度和网络最终的测试效果有着很大的影响<sup>[21]</sup>. 经过不断的尝试和启发性的思考, 我们找到一种适合混沌系统的

神经网络初始化方法: 基于储备池计算的初始化方法. 储备池计算有一种非常特殊的训练方法: 控制隐藏状态到输出的参数由数据学得, 但是隐藏状态到隐藏状态的参数( $W_{hh}$ )从一个特殊的分布中采样, 且在训练和预测的过程中固定住不做更新. 因为不做更新, 所以 $W_{hh}$ 的初始化至关重要. 初始化 $W_{hh}$ 可归纳为三步: 第一步, 从均匀分布 $[-a, a]$ 中采样得到 $W_{hh}$ ; 第二步, 随机选取比例为 $S$ 的一部分参数赋值为零, 来控制神经元之间的稀疏连接. 这个条件将使 $W_{hh}$ 分解为许多相互耦合的子系统, 更加丰富储备池的表达能力; 第三步, 控制 $W_{hh}$ 的谱半径 $R$ (矩阵的特征值的模的最大值). 谱半径对于储备池计算是否具有回声状态性质<sup>[3]</sup>至关重要, 当 $R < 1$ 时, 模型会趋向于忘掉前面时间序列积累的状态信息. 在实验中, 我们将LSTM单元中隐藏状态到隐藏状态的矩阵( $W_f, W_i, W_o, W_c$ )全部使用基于储备池计算的初始化方法, 初始化成稀疏度为 $S$ , 谱半径为 $R$ 的权值矩阵. 这个策略使矩阵在开始训练时能将混沌时间序列映射到一个具有丰富表达的高维的储备池, 使得初始的优化步骤能够往适合混沌时间序列的最优参数上发展.

保存最优模型策略. 在训练过程中, 我们发现训练步数也是非常重要的超参数. 训练误差不再下降后, 随着训练步数的增加, 模型在测试集上的预测效果会在一定范围内振荡, 不会趋于稳定. 对于这种现象, 我们采取的方法是当训练误差不再下降后, 每隔一定的训练步数, 将学得的模型参数保存下来作为模型训练到该步数下的一个版本, 共保存 $n$ 个模型版本. 训练结束后, 从训练集后到测试集前的时间序列中采取10段作为验证集, 将所有保存下来的模型版本在验证集上验证, 选出在验证集上效果最好的版本作为最终版本, 并将其用于测试集测试, 测试结果作为最终的结果. 值得注意的是, 在训练时间及硬件存储容量允许的条件下, 用更小的保存步数, 取得更多的版本, 将能找到效果更好的模型.

### 3 仿真实验

我们选择Lorenz系统<sup>[22]</sup>作为测试模型能力的

混沌系统. Lorenz系统方程为

$$\begin{aligned}\dot{x} &= -ax + ay, \\ \dot{y} &= bx - y - xz, \\ \dot{z} &= -cz + xy.\end{aligned}\quad (7)$$

我们用 $a=10, b=28, c=3/8$ 作为参数, 用四阶Runge-Kutta法产生以 $\Delta t$ 为时间间隔的混沌时间序列, 舍弃掉瞬态. 接着将前20000个点划为训练集, 接着2000个点划为验证集, 最后2000个点划为测试集.

以有效时间来衡量模型的预测能力, 有效时间定义为从开始预测到归一化误差 $E(t)$ 的值首次到达阈值 $f$ 所跨越的时间间隔, 用 $t_e$ 来表示.  $E(t)$ 定义为

$$E(t) = \frac{\|y(t) - \tilde{y}(t)\|}{\|y(t)\|^2)^{1/2}}. \quad (8)$$

为了充分验证模型与四种策略的效果, 避免随机性, 每次实验分别初始化10个不同的模型在训练集上训练, 并在测试集上随机选取30段互不相交的时间段用于测试, 将所有训练好的模型在所有测试时间段上求得有效时间, 最后对求得的300个有效时间做平均, 表示为平均有效时间( $\bar{t}_e$ ), 将其作为衡量模型预测能力的最终结果.

当同时使用四种策略时, 实验具体实现步骤如下:

- (1) 对数据做标准化预处理.
- (2) 使用基于储备池计算的初始化策略初始化训练模型.
- (3) 在训练集中随机选取窗口大小为 $W$ , 批量为 $n_{batchsize}$ 的一批时间序列, 用形状为 $(n_{batchsize}, W, 3)$ 的矩阵来表示.
- (4) 将该批数据输入到训练模型中, 将模型输出值使用还原处理, 然后基于损失函数求得反向传播的梯度, 对梯度运用梯度截断后再更新模型参数.
- (5) 循环3-4步, 直至达到训练步数或者训练误差不再下降.
- (6) 继续训练, 每间隔100步保存该训练步数下的模型参数, 共保存 $n$ 个模型.
- (7) 在验证集上比较所有模型来选取最优模型.
- (8) 使用最优模型在测试集上预测求得平均有效时间作为最终结果.

对于实验所用超参数, 我们把LSTM模型超参数分为必备的基础参数, 还有和策略相关的策略参数. 为了与储备池计算在Lorenz系统预测的平均有效时间相比较<sup>[15]</sup>, 我们选择同样的阈值( $f=0.4$ ). 经过大范围调试, 模型的最优参数如表1所示.

我们以结合所有四种策略的模型对测试集中一段轨迹的预测结果作为例子来展示模型的预测效果, 如图3. 可以看出在 $t=5$ 之前, LSTM都以极低的误差预测着洛伦兹系统的动力学演化, 但随着误差越来越大, 两条轨道迅速分离.

图4展示了预测结果的归一化误差 $E(t)$ 随时间的变化, 在 $t<5.5$ 的时间内,  $E(t)$ 都小于阈值( $f=0.4$ ). 定性地来看, 直到 $t=8$ 的时间内预测能力都还

表1 模型超参数

Table 1 Hyperparameters for the model

基础参数	值	策略参数	值
$\Delta t$	0.05	$R$	1.1
$d_h$	20	$S$	0.5
$n_{batchsize}$	200	$W$	60
$l$	$10^{-4}$	threshold	0.01
$f$	0.4	$n$	10

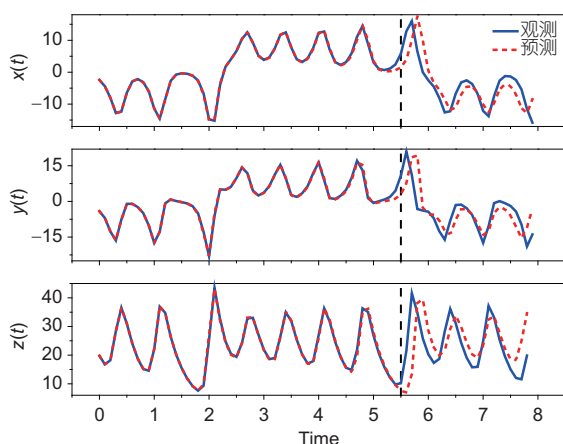


图3 (网络版彩图)长短期记忆模型结合四种策略预测Lorenz系统. 蓝色实线为真实的观测值, 红色虚线为模型预测值, 黑色竖直线代表归一化误差到达阈值 $f$

Figure 3 (Color online) Prediction of the Lorenz system with the LSTM learning machine and four strategies. The blue solid line shows the true observation of the Lorenz system, the red dashed line shows the prediction of the model, and the vertical black dashed line represents the normalized error reaching the threshold  $f$ .

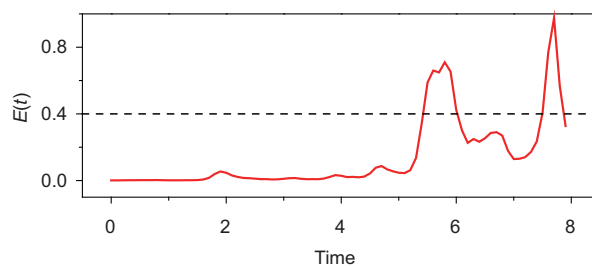


图4 (网络版彩图)图3中展示的预测轨迹的归一化误差 $E(t)$ 随时间的变化曲线. 当 $t=5.5$ 时, 归一化误差 $E(t)$ 第一次达到阈值 $f$

Figure 4 (Color online) Normalized error  $E(t)$  versus time of the Lorenz prediction trail shown in Figure 3. When  $t=5.5$ , the normalized error  $E(t)$  first reaches the value  $f$ .

存在. 注意储备池计算预测Lorenz系统混沌时间序列所能达到的平均有效时间也接近5.5<sup>[15]</sup>, 因此基于LSTM的模型的预测效果可以与它相比拟.

我们比较了使用搭配不同策略组合的模型来预测的结果. 对于不加任何策略的原始模型, 我们用O来表示. 对于加上策略的模型, 对应标准化和还原、梯度截断、基于储备池计算的参数初始化、保存最优模型四种策略, 我们分别用编号A, B, C, D来表示. 当一个不加策略的O模型加上策略, 就用对应策略编号组合来表示, 如原始模型加上策略A和B, 表示为AB. 对于不加基于储备池计算的初始化策略的模型, 我们使用深度学习中常用的Xavier初始化<sup>[23]</sup>作为初始化方法. 对于所有模型, 保证训练集、验证集和测试集的一致, 并且都使用adam作为优化算法, 其他基础参数均与表1相同.

图5展示了原始模型搭配上不同组合的策略所带给模型的提高. 每个策略带来的效果各有不同, 与原始模型比较, 标准化与还原策略能显著提升模型的预测效果. 同样的, 梯度截断策略和保存最优模型策略也能一定程度地提升模型的预测能力. 值得注意的是, 基于储备池计算的初始化策略在不结合标准化与还原策略时会降低模型预测的有效时间, 如C, BC, CD和BCD所示. 但是对比A和AC, AD和ACD, ABD和ABCD, 发现当数据使用了标准化与还原的预处理后, 各个维度对损失函数的影响保持平衡时, 这种初始化方法才能够发挥它的效果.

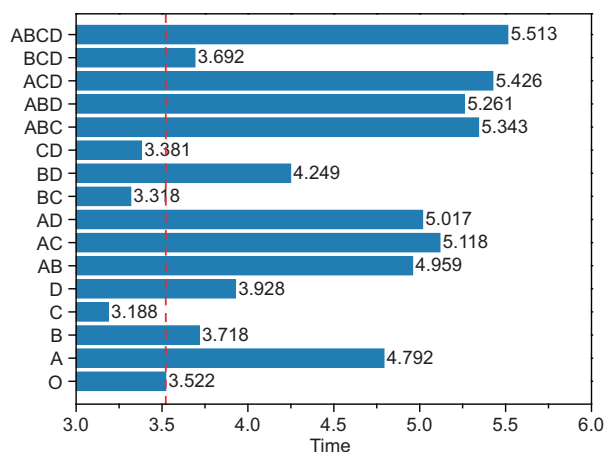


图5 (网络版彩图)模型搭配各种策略组合的平均有效时间对比

Figure 5 (Color online) The comparison of average valid time between models with the combination of various strategies.

## 4 讨论与结论

本文的研究表明通过改进长短期记忆网络模型并搭配以合适的训练策略,可以有效地实现混沌时间序列的长时间预测,效果完全可以与流行的储备池计算相比拟.实际上,为了达到较好的预测效

果,储备池模型使用了超过500维的隐藏状态向量来表示混沌时间序列的特征<sup>[15]</sup>,而本文仅仅使用了20维(见表1,  $d_h=20$ )的隐藏状态向量就达到类似的效果,因此模型复杂度更低.本文的结果意味着储备池计算并没有明显超越传统的长短期记忆模型,学习机预测时间序列的机制和方法有待进一步深入研究.

## 参考文献

- Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. Lake Tahoe, 2012. 1097–1105
- Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search. Nature, 2016, 529: 484–489
- Jaeger H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. Science, 2004, 304: 78–80
- Pathak J, Hunt B, Girvan M, et al. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. Phys Rev Lett, 2018, 120: 024102
- Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors. Nature, 1986, 323: 533–536
- Wu Y, Schuster M, Chen Z, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv: 1609.08144
- Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks. In: Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). New York: IEEE, 2013. 6645–6649
- Takens F. Detecting strange attractors in turbulence. In: Rand D, Young L S, eds. Dynamical Systems and Turbulence, Warwick 1980. Heidelberg: Springer, 1981. 366–381
- Zhang Y M, Wu X J, Bai S L. A DFP-method-based second-order Volterra filter and its application to chaotic time series prediction (in Chinese). Sci Sin-Phys Mech Astron, 2013, 43: 530–537 [张玉梅, 吴晓军, 白树林. 基于DFP的二阶Volterra滤波器及其在混沌序列预测中的应用. 中国科学: 物理学 力学 天文学, 2013, 43: 530–537]
- Karunasinghe D S K, Liong S Y. Chaotic time series prediction with a global model: Artificial neural network. J Hydrol, 2006, 323: 92–105
- Kim K B, Park J B, Choi Y H, et al. Control of chaotic dynamical systems using radial basis function network approximators. Inf Sci, 2000, 130: 165–183

- 12 Han M, Shi Z W, Xi J H. Learning the trajectories of periodic attractors using recurrent neural networks (in Chinese). *Control Theory Appl*, 2006, 23: 497–502 [韩敏, 史志伟, 席剑辉. 应用递归神经网络学习周期运动吸引子轨迹. *控制理论与应用*, 2006, 23: 497–502]
- 13 Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. arXiv: 1211.5063
- 14 Tian Z D, Li S J, Wang Y H, et al. A hybrid prediction model of smoothed monthly mean sunspot number (in Chinese). *Sci Sin-Phys Mech Astron*, 2016, 46: 119601 [田中大, 李树江, 王艳红, 等. 太阳黑子数平滑月均值的混合预测模型. *中国科学: 物理学 力学 天文学*, 2016, 46: 119601]
- 15 Pathak J, Wikner A, Fussell R, et al. Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. *Chaos*, 2018, 28: 041101
- 16 Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*, 1997, 9: 1735–1780
- 17 Gers F A, Eck D, Schmidhuber J. Applying LSTM to time series predictable through time-window approaches. In: *Neural Nets WIRN Vietri-01*. London: Springer, 2002. 193–200
- 18 Vlachas P R, Byeon W, Wan Z Y, et al. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proc R Soc A*, 2018, 474: 20170844
- 19 Werbos P J. Backpropagation through time: What it does and how to do it. *Proc IEEE*, 1990, 78: 1550–1560
- 20 Kingma D P, Ba J. Adam: A method for stochastic optimization. arXiv: 1412.6980
- 21 Sutskever I, Martens J, Dahl G E, et al. On the importance of initialization and momentum in deep learning. In: *Proceedings of 30th International Conference on Machine Learning*, Georgia, 2013. 5
- 22 Lorenz E N. Deterministic nonperiodic flow. *J Atmos Sci*, 1963, 20: 130–141
- 23 Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Sardinia, 2010. 249–256

## Chaotic time series prediction based on long short-term memory neural networks

XIONG YouCheng & ZHAO Hong\*

*College of Physical Science and Technology, Xiamen University, Xiamen 361005, China*

Recently, the development of deep learning technology has promoted the wide application of machine learning. In particular, the technique known as reservoir computing has attracted more and more attention due to its excellent behavior on chaotic time series prediction, and has formed a new research hotspot. In this paper, we use traditional long short-term memory (LSTM) neural networks and fully connected layers as the fundamental elements to build a LSTM learning machine based on the recurrent neural network architecture. The motivation for using LSTM is that it can effectively prevent the vanishing and exploding of gradients. In the simulation experiment, we quantify the duration of accurate prediction with the average valid time and use the model to predict the state of Lorenz system. Aiming at the special dynamic properties of chaotic system, we propose four strategies to assist prediction, which are normalization and restoration, scaling down the gradients, reservoir-based initialization and preserving the optimal model. The results show that the prediction ability of the LSTM learning machine with suitable strategies is comparable to that of reservoir computing, furthermore, the complexity of the LSTM is lower. Therefore, our results indicate that there is no obvious evidence that reservoir computing can surpass the traditional method, which inspires us to further study the mechanism and methods of learning machine for predicting time series, and find more effective learning machines.

**machine learning, chaotic time series, prediction, long short-term memory neural networks**

**PACS:** 05.45.Tp, 07.05.Mh, 87.85.dq

**doi:** 10.1360/SSPMA-2019-0115