

基于脉冲序列核的脉冲神经元监督学习算法

蔺想红,王向文,党小超

(西北师范大学计算机科学与工程学院,甘肃兰州 730070)

摘要: 脉冲神经元应用脉冲时间编码神经信息,监督学习的目标是对于给定的突触输入产生任意的期望脉冲序列.但由于神经元脉冲发放过程的不连续性,构建高效的脉冲神经元监督学习算法非常困难,同时也是该研究领域的重要问题.基于脉冲序列的核函数定义,提出了一种新的脉冲神经元监督学习算法,特点是应用脉冲序列核构造多脉冲误差函数和对应的突触学习规则,并通过神经元的实际脉冲发放频率自适应地调整学习率.将该算法用于脉冲序列的学习任务,期望脉冲序列采用 Poisson 过程或线性方法编码,并分析了不同的核函数对算法学习性能的影响.实验结果表明该算法具有较高的学习精度和良好的适应能力,在处理复杂的时空脉冲模式学习问题时十分有效.

关键词: 脉冲神经元; 监督学习; 脉冲序列核; 内积; 脉冲序列学习

中图分类号: TP183 **文献标识码:** A **文章编号:** 0372-2112 (2016)12-2877-10

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2016.12.010

A New Supervised Learning Algorithm for Spiking Neurons Based on Spike Train Kernels

LIN Xiang-hong, WANG Xiang-wen, DANG Xiao-chao

(School of Computer Science and Engineering, Northwest Normal University, Lanzhou, Gansu 730070, China)

Abstract: The purpose of supervised learning with temporal encoding for spiking neurons is to make the neurons emit arbitrary spike trains in response to given synaptic inputs. However, due to the discontinuity in the spike process, the formulation of efficient supervised learning algorithms for spiking neurons is difficult and remains an important problem in the research area. Based on the definition of kernel functions for spike trains, this paper proposes a new supervised learning algorithm for spiking neurons with temporal encoding. The learning rule for synapses is developed by constructing the multiple spikes error function using spike train kernels, and its learning rate is adaptively adjusted according to the actual firing rate of spiking neurons during learning. The proposed algorithm is successfully applied to various spike trains learning tasks, in which the desired spike trains are encoded by Poisson process or linear method. Furthermore, the effect of different kernels on the performance of the learning algorithm is also analyzed. The experiment results show that our proposed method has higher learning accuracy and flexibility than the existing learning methods, so it is effective for solving complex spatio-temporal spike pattern learning problems.

Key words: spiking neuron; supervised learning; spike train kernel; inner product; spike train learning

1 引言

神经科学的研究成果表明,神经信息被编码为精确确定时的脉冲序列,不仅仅是简单的脉冲发放频率^[1].脉冲神经网络由更具生物真实性的脉冲神经元模型为基本单元构成^[2],应用精确确定时的脉冲序列表示与处理信息,是新一代神经网络计算模型.脉冲神经网络与

基于脉冲频率编码信息的传统人工神经网络相比,拥有更强大的计算能力,是进行复杂时空信息处理的有效工具^[3].实际上,要将脉冲神经元构成的网络应用于实际问题,特别是复杂时空模式识别问题的求解,关键在于构建高效的监督学习算法.脉冲神经元监督学习的目标是对于给定的突触输入脉冲序列,通过学习规则对突触权值的调整,产生任意的期望脉冲序列^[4].对

收稿日期:2015-04-07;修回日期:2015-07-01;责任编辑:梅志强

基金项目:国家自然科学基金(No. 61165002, No. 61363059);甘肃省自然科学基金(No. 1506RJZA127);甘肃省高等学校科研项目(No. 2015A-013)

于脉冲神经网络来说,神经元内部状态变量及误差函数不再满足连续可微的性质,传统人工神经网络的学习算法已不能直接使用,需要研究者进一步构建具有广泛适用性的脉冲神经网络监督学习算法.

脉冲神经元及其网络的监督学习是新兴技术,对于监督学习算法的研究越来越受到研究者的关注,近年来提出了较多的监督学习算法^[5,6]. 借鉴传统人工神经网络的误差反向传播算法,Bohte 等人^[7]提出了适用于前馈脉冲神经网络的 SpikeProp 算法,该算法限制网络中所有层的神经元只能发放一个脉冲. McKennoch 等人^[8]在该算法的基础上,提出了收敛速度更快的 RProp 和 QuickProp 算法. 对于 SpikeProp 算法更加重要的扩展工作是 Multi-SpikeProp 算法,这类算法对脉冲神经网络输入层和隐含层神经元的脉冲发放没有限制,但仍然限制输出层神经元只能发放一个脉冲^[9,10]. 最近, Xu 等人^[11]进一步提出了基于梯度下降的多脉冲监督学习算法,对网络中所有层神经元的脉冲发放没有限制,实现了输出层多脉冲的时空模式学习. SpikeProp 及其扩展算法是一种数学分析方法,在学习规则的推导过程中,要求神经元模型的状态变量必须有解析的表达式. 文献[12,13]将脉冲神经元的监督学习转换为分类问题,分别应用感知器学习规则和支持向量机实现了脉冲序列时空模式的在线学习算法. 考虑神经元的脉冲时间依赖可塑性(Spike Timing-Dependent Plasticity, STDP)机制,Legenstein 等人^[14]给出了脉冲神经元的监督 Hebbian 学习算法,通过注入外部输入电流使学习神经元发放特定的期望脉冲序列. Ponulak 和 Kasiński^[15]将突触权值的调整表示为 STDP 和 anti-STDP 两个过程的结合,提出了一种可对脉冲序列的复杂时空模式进行学习的远程监督方法(Remote Supervised Method, ReSuMe),该算法具有学习规则的局部特性和最优解的稳定性,并且可适用于各种神经元模型. 虽然 ReSuMe 仅适用于单神经元或单层神经网络的学习,但基于其良好的适用性,已被广泛地应用在各类时空模式分类和识别问题. 文献[16]结合 SpikeProp 算法,进一步将 ReSuMe 扩展到适用于线性神经元模型的多层前馈网络. Wade 等人^[17]应用 BCM(Bienenstock-Cooper-Munro)规则与 STDP 机制,提出了多层前馈脉冲神经网络的 SWAT(Synaptic Weight Association Training)算法. 此外, Moheemmed 和 Schliebs^[18]基于 Widrow-Hoff 规则给出了 SPAN(Spike Pattern Association Neuron)学习规则,将脉冲序列应用卷积计算转换为模拟信号,通过转化后的输入脉冲序列,神经元期望输出和实际输出脉冲序列的误差调整突触权值. Yu 等人^[19]受 SPAN 的启发,提出了 PSD(Precise-Spike-Driven)监督学习算法,突触权值的调整根据期望与实际输出脉冲的误差来判断,正的

误差将导致长时程增强,负的误差将导致长时程抑制.

本文通过脉冲序列核的表示形式,将离散的脉冲序列转换为连续函数的分析过程,并可解释为特定的神经生理信号,比如神经元的突触后电位或脉冲发放的密度函数. 将脉冲序列集合映射到与核函数对应的再生核 Hilbert 空间^[20,21],这样便实现了脉冲序列的统一表示,以及脉冲序列相似性度量的形式化定义. 我们通过定义脉冲序列随时间变化的误差函数以及神经元输入脉冲序列和输出脉冲序列之间的关系,给出了基于脉冲序列核的脉冲神经元的监督学习算法,突触权值的学习规则表示为脉冲序列的内积形式,实现了脉冲序列复杂时空模式的学习. 为了描述方便,我们将所提学习规则命名为 STKLR(Spike Train Kernel Learning Rule). 该学习算法的特点在于:(1)突触学习规则的推导过程仅依赖于精确定时的脉冲序列,与神经元状态变量的表达方式无关,因此可适用于不同的神经元模型.(2)在学习算法的迭代过程中,学习率根据神经元实际输出脉冲序列的频率自适应地调整.(3)对神经元输入输出的脉冲序列编码方式没有限制,可适用于 Poisson 过程或线性方法等不同的时间编码策略.

2 脉冲序列核的表示

核方法是解决非线性模式分类问题的一种有效途径,其核心思想是通过某种非线性映射将原始数据嵌入到合适的高维特征空间,找出并学习一组数据中的相互关系^[22]. 设 x_1, x_2 为初始空间中的点(在这里可以是标量,也可以是向量),非线性函数 ϕ 实现输入空间 X 到特征空间 F 的映射,核函数可表示为如下的内积形式:

$$\kappa(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle \quad (1)$$

其中, $\kappa(x_1, x_2)$ 为核函数. 下面我们将核函数的定义应用于神经元发放的脉冲序列,并推导突触权值的学习规则.

脉冲序列 $s = \{t^f \in \Gamma; f = 1, \dots, N\}$ 表示脉冲神经元在时间区间 $\Gamma = [0, T]$ 所发放脉冲时间的有序数列,脉冲序列可形式化的表示为:

$$s(t) = \sum_{f=1}^N \delta(t - t^f) \quad (2)$$

其中, t^f 表示第 f 个脉冲发放时间, $\delta(x)$ 表示 Dirac delta 函数,当 $x = 0$ 时, $\delta(x) = 1$, 否则 $\delta(x) = 0$.

在定义脉冲序列的核函数之前,先给出脉冲时间的核函数表示,因为脉冲时间的核将组合构成脉冲序列核的表示. 对于两个脉冲对应的发放时间 t^m 和 t^n ,可定义脉冲时间的核函数形式:

$$\kappa(t^m, t^n) = \langle \delta(t - t^m), \delta(t - t^n) \rangle, \forall t^m, t^n \in \Gamma \quad (3)$$

对于核函数 κ 要求具有对称、平移不变和正定特性,一般取 Laplacian 或 Gaussian 等核函数. 比如 Gaussian 核

表示为 $\kappa(t^m, t^n) = \exp(-|t^m - t^n|^2 / 2\sigma^2)$.

由于脉冲序列是离散时间的集合,为了方便分析和计算,可以选择特定的光滑函数 h ,应用卷积将脉冲序列唯一地转换为一个连续函数:

$$f_s(t) = s(t) * h(t) = \sum_{j=1}^N h(t - t_j^s) \quad (4)$$

由于脉冲序列所对应时间区间的有限性和连续函数 $f_s(t)$ 的有界性,可以得到:

$$\int_{\Gamma} f_s^2(t) dt < \infty \quad (5)$$

也就是说,函数 $f_s(t)$ 是 $L_2(\Gamma)$ 空间的一个元素. 脉冲序列空间中的元素 $s_i \in S(\Gamma)$ 对应的函数 $f_{s_i}(t)$ 将构成一个 Hilbert 空间,表示为 $L_2(f_{s_i}(t), t \in \Gamma)$, 并且是 $L_2(\Gamma)$ 的一个子空间^[20].

对于任意给定的两个脉冲序列 $s_i, s_j \in S(\Gamma)$, 可在 $L_2(f_{s_i}(t), t \in \Gamma)$ 空间上定义其对应函数 $f_{s_i}(t)$ 和 $f_{s_j}(t)$ 的内积如下表示:

$$F(s_i, s_j) = \langle f_{s_i}(t), f_{s_j}(t) \rangle_{L_2(\Gamma)} = \int_{\Gamma} f_{s_i}(t) f_{s_j}(t) dt \quad (6)$$

应用脉冲时间核表示,进一步将式(6)重写为脉冲时间对的累加形式:

$$\begin{aligned} F(s_i, s_j) &= \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} \int_{\Gamma} h(t - t_i^m) h(t - t_j^n) dt \\ &= \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} \kappa(t_i^m, t_j^n) \end{aligned} \quad (7)$$

其中, $\kappa(t^m, t^n) = \int_{\Gamma} h(t - t^m) h(t - t^n) dt$. 因此, $F(s_i, s_j)$ 计算的时间复杂度为 $O(N_i N_j)$, N_i 和 N_j 分别表示脉冲序列 s_i 和 s_j 对应的脉冲数.

3 脉冲神经元的监督学习算法

3.1 脉冲序列的转换关系

脉冲神经元输入输出表示为脉冲序列的形式,即脉冲序列编码神经信息或外部刺激信号. 假设突触前神经元输入的脉冲序列为 $s_i(t) \in S(\Gamma)$, $i = 1, \dots, N$, 突触后神经元输出的脉冲序列为 $s_o(t) \in S(\Gamma)$, 为简化计算,将应用式(4)转换后的多个输入脉冲序列和输出脉冲序列在时间 t 的关系表达为线性组合关系:

$$f_{s_o}(t) = \sum_{i=1}^N w_i f_{s_i}(t) \quad (8)$$

其中,权值 w_i 表示突触前神经元 i 和突触后神经元之间的连接强度, N 表示突触前输入神经元的数目. 做这种简化的原因在于:(1)将脉冲序列表达为线性组合关系,可以构造相应的学习规则进行神经元突触权值的学习, Carnell 和 Richardson 给出了初步的结果^[23]; (2)如果将 $f_s(t)$ 解释为脉冲序列随时间变化的密度函数,对于广义的线性神经元模型,突触后神经元脉冲序列

的密度函数可以表达为突触前神经元脉冲序列密度函数的线性组合形式.

3.2 突触权值的学习规则

构建脉冲神经元监督学习算法的关键是定义脉冲序列的误差函数,以及突触权值的学习规则. 脉冲神经元在时间 t 的误差可定义为实际输出脉冲序列 $s_o \in S(\Gamma)$ 和期望脉冲序列 $s_d \in S(\Gamma)$ 对应函数 $f_s(t)$ 的差值平方:

$$E(t) = \frac{1}{2} [f_{s_o}(t) - f_{s_d}(t)]^2 \quad (9)$$

因此,脉冲神经元在时间区间 Γ 总的误差为

$$E = \int_{\Gamma} E(t) dt.$$

应用脉冲序列误差函数对于突触权值的梯度计算值,使用 delta 更新规则对所有的突触权值进行调整. 从突触前神经元 i 到突触后神经元的突触权值 w_i 计算如下:

$$\Delta w_i = -\eta \nabla E_i \quad (10)$$

其中, η 表示学习率, ∇E_i 表示脉冲序列误差函数 E 对于突触权值 w_i 的梯度计算值,可表示为误差函数 $E(t)$ 对权值 w_i 的导数在时间区间的积分:

$$\nabla E_i = \int_{\Gamma} \frac{\partial E(t)}{\partial w_i} dt \quad (11)$$

对于突触权值的学习规则,应用误差函数和链式规则,根据式(8)和式(9),突触权值 w_i 在时间 t 的变化量可推导得到:

$$\frac{\partial E(t)}{\partial w_i} = \frac{\partial E(t)}{\partial f_{s_o}(t)} \frac{\partial f_{s_o}(t)}{\partial w_i} = [f_{s_o}(t) - f_{s_d}(t)] f_{s_i}(t) \quad (12)$$

其中, $s_i(t) \in S(\Gamma)$ 表示输入神经元 i 所发放的脉冲序列. 根据式(11),突触权值 w_i 的调整值计算如下:

$$\nabla E_i = \int_{\Gamma} [f_{s_o}(t) - f_{s_d}(t)] f_{s_i}(t) dt = F(s_o, s_i) - F(s_d, s_i) \quad (13)$$

根据以上所讨论的推导过程,我们给出了一个基于脉冲序列核的脉冲神经元监督学习新算法 STKLR, 突触学习规则表示如下:

$$\begin{aligned} \Delta w_i &= -\eta [F(s_o, s_i) - F(s_d, s_i)] \\ &= \eta \left[\sum_{g=1}^{N_d} \sum_{f=1}^{N_i} \kappa(t_d^g, t_i^f) - \sum_{h=1}^{N_o} \sum_{f=1}^{N_i} \kappa(t_o^h, t_i^f) \right] \end{aligned} \quad (14)$$

其中, t_i^f 、 t_o^h 和 t_d^g 分别表示输入脉冲序列 s_i 、实际输出脉冲序列 s_o 和期望脉冲序列 s_d 中对应脉冲的发放时间.

3.3 学习率的自适应方法

学习率的取值大小对学习过程的收敛速度有较大的影响,直接影响训练时间和训练精度. 如果学习率取值太小,每次迭代中权值的有效更新值太小,将导致网络突触权值的收敛速度变慢;反之,容易使学习过程出现振荡,影响网络的收敛速度,甚至网络训练失败. 根据

神经元实际发放脉冲的频率自适应地调整学习率,可提高学习算法对突触权值训练的适应能力.

首先定义一个缩放因子 c , 使该算法的学习率 η 能够按照不同的脉冲发放频率自适应调整. 假设神经元脉冲序列的发放频率为 v , 基准频率区间为 $[v_{\min}, v_{\max}]$, 当 $v \in [v_{\min}, v_{\max}]$ 时, 缩放因子 $c = 1$, 否则, c 的表达式为:

$$c = \begin{cases} \frac{v_{\min} - v}{v_{\max} - v_{\min}}, & v < v_{\min} \\ \frac{v - v_{\max}}{v_{\max} - v_{\min}}, & v > v_{\max} \end{cases} \quad (15)$$

学习率在基准频率区间的取值称为基准学习率 η^* , 其值为给定频率区间的最佳学习率, 具体分析过程在 4.2 节中给出. 本文中基准频率区间的最小值 $v_{\min} = 40\text{Hz}$, 最大值 $v_{\max} = 60\text{Hz}$. 根据缩放因子 c 和频率区间内的基准学习率 η^* , 学习率 η 的自适应调整方式为:

$$\eta = \begin{cases} (1+c)\eta^*, & v < v_{\min} \\ \eta^*, & v_{\min} \leq v \leq v_{\max} \\ \eta^*/(1+c), & v > v_{\max} \end{cases} \quad (16)$$

3.4 脉冲序列的相似性度量

在神经元的脉冲序列学习过程中, 学习性能的评价就是判断在学习结束后实际发放的脉冲序列与期望输出脉冲序列接近的程度, 这实际上就是度量两个脉冲序列之间的相似性. 根据脉冲序列的内积所满足 Cauchy-Schwarz 不等式特性^[20]:

$$F^2(s_i, s_j) \leq F(s_i, s_i) F(s_j, s_j), \forall s_i, s_j \in S(I) \quad (17)$$

应用式(17)定义两个脉冲序列的相似性度量 C , 其表达式为:

$$C = \frac{\langle f_s(t), f_s(t) \rangle}{\|f_s(t)\| \|f_s(t)\|} = \frac{F(s_i, s_j)}{\sqrt{F(s_i, s_i)} \sqrt{F(s_j, s_j)}} \quad (18)$$

其中, $\|f_s(t)\| = \sqrt{\langle f_s(t), f_s(t) \rangle}$ 称为脉冲序列的 Euclidean 范数. 当两个脉冲序列完全相同时, C 的值为 1, 随着它们的相似性越来越低逐渐趋于 0.

4 实验结果

在实验中, 神经元模型采用脉冲响应模型^[24], 假设神经元有 N 个输入突触, 第 i 个突触有 G_i 个脉冲输入, t_i^g 表示神经元 i 的第 g 个脉冲的发放时间, 神经元在时间 t 的膜电位 $u(t)$ 可以表示为:

$$u(t) = \eta(t - t^f) + \sum_{i=1}^N \sum_{g=1}^{G_i} w_i \varepsilon(t - t_i^g) \quad (19)$$

其中, $\varepsilon(s) = \frac{s}{\tau} \exp(1 - \frac{s}{\tau}) H(s)$ 表示脉冲响应函数, 时间常量 $\tau = 2\text{ms}$; $\eta(s) = -\theta \exp(-s/\tau_r) H(s)$ 表示不应期函数, 时间常量 $\tau_r = 50\text{ms}$; $H(s)$ 表示 Heaviside 阶跃函数, 当 $s \leq 0$ 时, $H(s) = 0$, 否则 $H(s) = 1$. 神经元的激发阈值 θ

$= 1$, 脉冲激发后的绝对不应期 $t_r = 1\text{ms}$. 网络中初始突触权值随机生成, 并在区间 $[0, 0.2]$ 上均匀分布. 神经元进行脉冲序列学习时, 基准学习率 $\eta^* = 0.005$, 迭代过程中学习率 η 根据式(16)自适应地调整. 每一次实验的学习周期上限数设定为 1000 步, 且每个实验结果都是运行 100 次实验的平均结果. 此外, 为了对脉冲序列学习过程中的相关参数进行敏感性分析, 基准参数设置如下: 神经元的输入突触数目为 500, 输入与期望输出脉冲序列的时间长度为 200ms, 并分别由频率为 20Hz 和 50Hz 的 Poisson 过程随机生成. 如果没有特别说明, STKLR 中的核函数取 Gaussian 核, 参数 $\sigma = 2$.

由于 ReSuMe 具有良好的学习能力和适用性, 被广泛地应用在各类时空模式学习问题. 因此, 本文将所提 STKLR 与 ReSuMe 进行学习性能的比较. 将 ReSuMe 突触权值随时间变化的调整量积分, 可得突触权值的离线学习规则如下^[19]:

$$\Delta w_i^{\text{ReSuMe}} = \eta \left\{ \sum_{g=1}^{N_d} \left[a_s + \sum_{t_i^g < t_d^g} \exp\left(-\frac{t_d^g - t_i^g}{\tau_s}\right) \right] - \sum_{h=1}^{N_o} \left[a_s + \sum_{t_i^h < t_o^h} \exp\left(-\frac{t_o^h - t_i^h}{\tau_s}\right) \right] \right\} \quad (20)$$

其中, 参数 $a_s = 0.05$ 表示 non-Hebbian 项, 用于加速训练过程的收敛; 指数形式的核函数定义了 STDP 机制所决定的 Hebbian 项, 时间常量 $\tau_s = 5\text{ms}$; N_o, N_d 分别表示神经元实际输出脉冲序列 s_o 和期望脉冲序列 s_d 的脉冲总数.

4.1 脉冲序列学习任务

4.1.1 脉冲序列的学习过程分析

首先分析 STKLR 对于脉冲序列的学习过程和学习前后的突触权值变化情况, 神经元学习的结果如图 1 所示. 图 1(a) 展示了脉冲神经元的脉冲序列学习过程, 图中 ∇ 表示期望输出脉冲序列, Δ 表示脉冲神经元学习之前的输出脉冲序列, \cdot 表示学习过程中一些学习周期的实际输出脉冲序列. 从学习过程可以看到, 神经元经过大概 45 步就从初始的输出脉冲序列学习得到期望的输出脉冲序列. 图 1(b) 表示学习过程中学习精度曲线的变化, 可以看出在 45 个学习周期以后, 实际输出和期望脉冲序列一样, 即两者的相似度量值 C 达到 1. 图 1(c) 和图 1(d) 分别表示学习前和学习后的 500 个突触权值的变化情况. 在学习过程中, 神经元突触权值的变化范围为 $[0, 0.2]$. 通过脉冲序列学习过程的分析可以看出, STKLR 可以对复杂的脉冲序列时空模式进行学习, 并具有较好的学习能力.

4.1.2 参数变化时的学习性能分析

为了全面评价 STKLR 的学习性能, 变化神经元的输入突触数目和脉冲序列的长度, 并考察这些因素对学习性能的影响. 第 1 组实验分析 STKLR 在输入突触

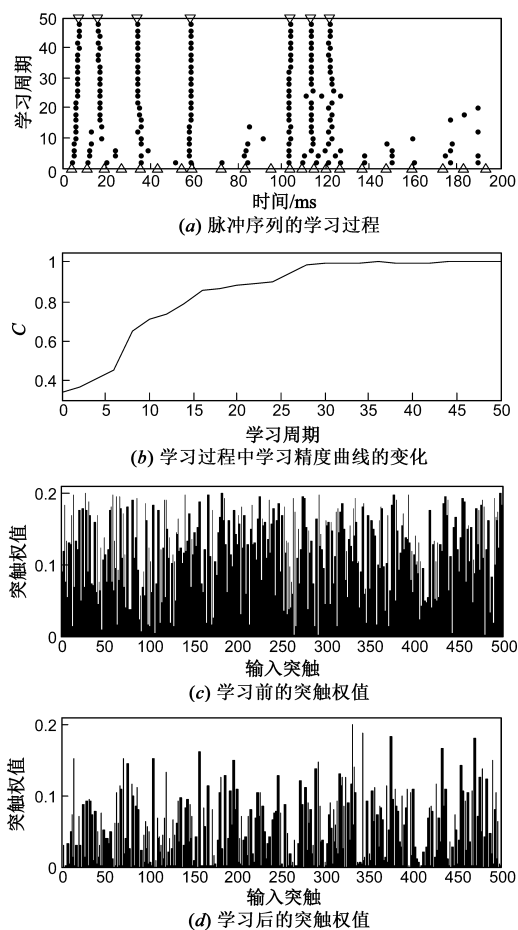


图1 脉冲序列的学习过程和学习前后的突触权值

数目逐渐增加时的学习性能. 实验中神经元的输入突触数目从 100 到 1000 以间隔 100 逐渐增加, 其它设定保持不变. 图 2(a) 表示 STKLR 和 ReSuMe 在输入突触数目逐渐增加时的学习精度, 从图中可以看出, 随着神经元输入突触数目的逐渐增加, 两个算法的学习精度都在增加, 最终其度量 C 值可以非常接近 1, 且 STKLR 的学习精度高于 ReSuMe 的学习结果. 例如, 当输入突触数目为 300 时, STKLR 的学习精度为 0.9577, 而 ReSuMe 的学习精度为 0.9302; 当输入突触数目为 800 时, STKLR 的学习精度为 0.9923, 而 ReSuMe 的学习精度为 0.9670. 同时, 学习精度在上升时期较大的标准方差显示此时 STKLR 学习性能的稳定性比较低, 但是当学习精度达到很高的数值时, STKLR 学习精度的标准方差则较小. 图 2(b) 表示当学习精度达到最高时所需的学习周期, 从图中可以看出, 除过输入突触数目为 200 的情况外, STKLR 所需的学习周期比 ReSuMe 的要少. 例如, 当输入突触数目为 500 时, STKLR 所需的学习周期为 522.81, 而 ReSuMe 所需的学习周期为 613.97; 当输入突触数目为 1000 时, STKLR 所需的学习周期为 449.16, 而 ReSuMe 的学习周期为 702.74.

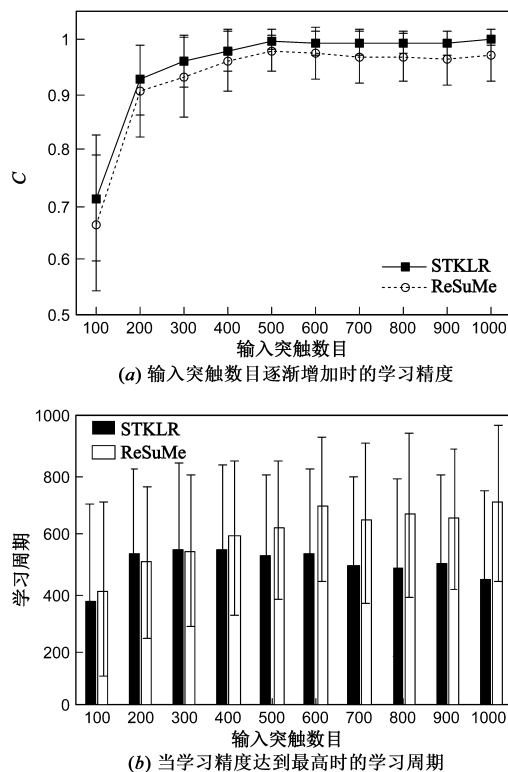


图2 输入突触数目逐渐增加时的脉冲序列学习结果

第 2 组实验分析 STKLR 在神经元输入与期望输出脉冲序列长度逐渐增加时的学习性能. 实验中神经元输入与期望输出脉冲序列的长度从 100ms 到 1000ms 以间隔 100ms 逐渐增加, 其它参数取基准值. 图 3 所示为神经元输入与期望输出脉冲序列长度变化时的脉冲序列学习结果. 图 3(a) 表示 STKLR 和 ReSuMe 在神经元输入与期望输出脉冲序列长度逐渐增加时的学习精度, 从图中可以看出, 随着脉冲序列长度的增加, 两个算法的学习精度都在减小. 此外, STKLR 的学习精度高于 ReSuMe 学习精度, 例如, 当脉冲序列长度为 200ms 时, STKLR 的学习精度为 0.9933, 而 ReSuMe 的学习精度为 0.9785; 当脉冲序列长度为 1000ms 时, STKLR 的学习精度为 0.7522, 而 ReSuMe 的学习精度为 0.6773. 图 3(b) 表示随着神经元脉冲序列长度的增加, 当学习精度达到最高时所需的最小学习周期. 从图中可以看出, 当脉冲序列长度为 100ms、200ms、300ms、700ms、800ms、900ms 和 1000ms 时, STKLR 所需的学习周期少于 ReSuMe. 例如, 当脉冲序列长度为 300ms 时, STKLR 所需的学习周期为 534.18, 而 ReSuMe 所需的学习周期为 587.94; 当脉冲序列长度为 1000ms 时, STKLR 所需的学习周期为 432.61, 而 ReSuMe 所需的学习周期为 464.70.

4.1.3 期望脉冲序列不同编码的学习

为了进一步分析 STKLR 对脉冲序列的学习性能,

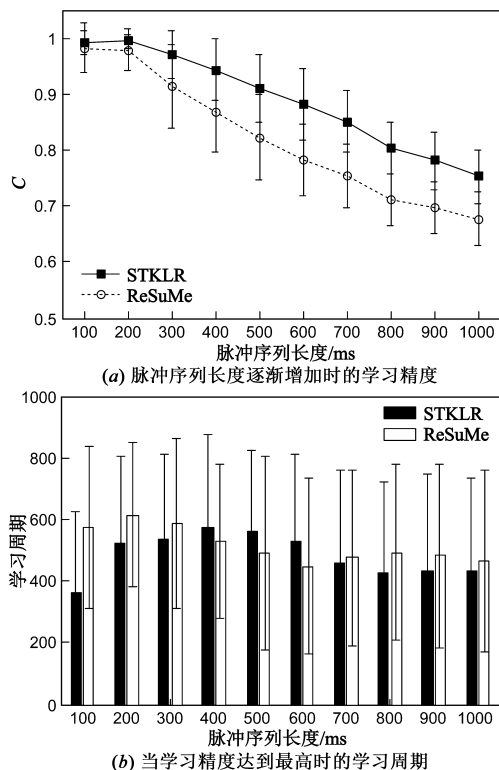


图3 脉冲序列长度逐渐增加时的脉冲序列学习结果

输入脉冲序列采用 Poisson 过程生成,而期望脉冲序列分别采用 Poisson 过程和线性方法^[17]两种不同的编码方式生成.实验中输入与期望输出脉冲序列的发放频率从 20Hz 到 200Hz 以间隔 20Hz 逐渐增加,并且输入脉冲序列与期望输出脉冲序列的发放频率相等,其它设定保持不变.图 4 所示为期望输出脉冲序列采用 Poisson 过程编码的学习结果.图 4(a)表示 STKLR 和 ReSuMe 在输入与期望输出脉冲序列由不同发放频率的 Poisson 过程生成时的学习精度,随着脉冲序列发放频率的增加,两个算法的学习精度先减小后增大,且 STKLR 的学习精度高于 ReSuMe.例如,当脉冲序列的发放频率为 100Hz 时,STKLR 的学习精度为 0.8860,而 ReSuMe 的学习精度为 0.7906;当脉冲序列发放频率为 200Hz 时,STKLR 的学习精度为 0.8362,而 ReSuMe 的学习精度为 0.7942.图 4(b)表示当学习精度达到最高时算法所需的学习周期,从图中可以看出,STKLR 算法所需的学习周期要少于 ReSuMe,例如,当脉冲序列发放频率为 200Hz 时,STKLR 所需的学习周期为 449.15,而 ReSuMe 所需的学习周期为 514.24.

图 5 给出了期望输出脉冲序列采用线性方法编码的学习结果,这时脉冲序列中的相邻脉冲之间具有相同的时间间隔.从图 5(a)中可以看出,随着脉冲序列发放频率的增加,STKLR 的学习精度基本保持不变且接近于 1,而 ReSuMe 的学习精度在逐渐减小之后又逐渐

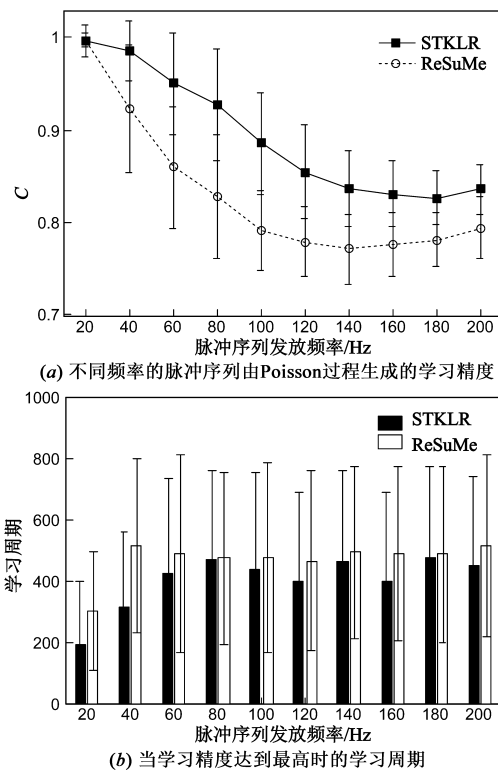


图4 期望脉冲序列采用Poisson过程编码的学习结果

增大,并且 ReSuMe 的学习精度具有较大的标准方差,其学习稳定性比较差.图 5(b)表示当学习精度达到最高时所需的学习周期,从图中可以看出,当脉冲序列的

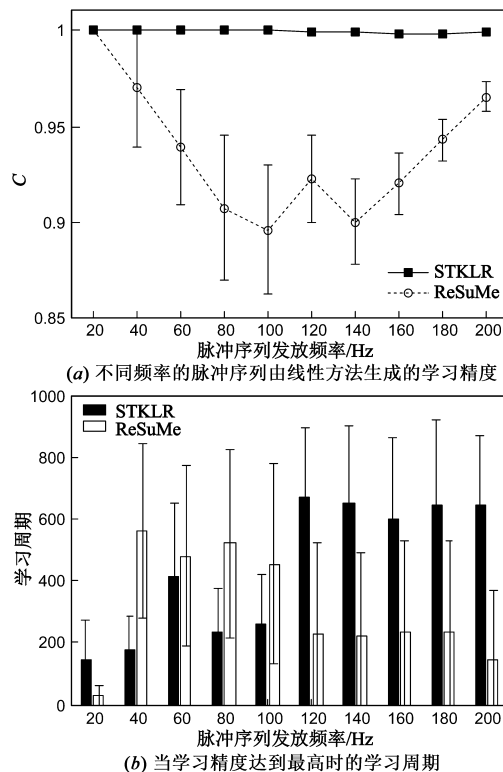


图5 期望脉冲序列采用线性方法编码的学习结果

发放频率较小时, STKLR 所需的学习周期少于 ReSuMe, 而当脉冲序列的发放频率大于 120Hz 时, STKLR 所需的学习周期多于 ReSuMe.

4.2 学习过程中学习率分析

4.2.1 基准学习率的确定

本组实验通过不同学习率的测试来确定在基准参数取值下适合 STKLR 和 ReSuMe 的最佳学习率. 实验中期望脉冲序列的发放频率为 50Hz, 学习率不采用式 (16) 调整, 图 6 所示为学习率的取值分别为 0.00005、0.0001、0.0005、0.001、0.005 和 0.01 时的脉冲序列学习结果. 图 6(a) 表示两个算法在学习率不同时的学习精度, 从图中可以看出, 随着学习率的增大, 两个算法的学习精度都在增加. 学习率为 0.005 时, STKLR 的学习精度为 0.9855, 而 ReSuMe 的学习精度为 0.9676. 当学习率取 0.01 时两个算法的学习精度与学习率取 0.005 时的学习精度相比变化不大, 但是学习精度的标准方差增大. 当学习率进一步增大时将导致算法学习失败的次数增加. 图 6(b) 表示两个算法在达到最高学习精度时所需的学习周期, 从图中可以看出随着学习率的增大, 两个算法所需的学习周期都在逐渐减少, 且在学习率大于 0.001 时 STKLR 小于 ReSuMe 所需的学习周期. 例如, 当学习率为 0.005 时, STKLR 所需学习周期的平均值为 452.15, 而 ReSuMe 的学习周期为 631.21. 因此, 在我们的实验测试与分析中, 学习率的取值 0.005 为两种算法的基准学习率.

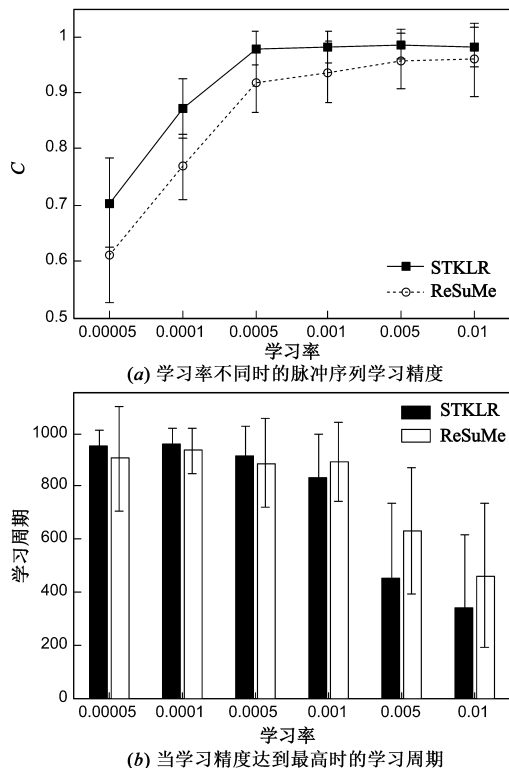


图6 学习率不同时的脉冲序列学习结果

4.2.2 学习率的变化过程

本组实验分析当期望输出脉冲序列由不同发放频率的 Poisson 过程或线性编码生成时, STKLR 对脉冲序列学习中学习率的变化过程. 实验中, 输入脉冲序列由发放频率为 20Hz 的 Poisson 过程生成, 期望输出脉冲序列的发放频率取 20Hz、50Hz、100Hz 和 150Hz 四个值, 其它设定保持不变. 图 7 表示期望输出脉冲序列的发放频率不同时学习率的变化过程, 当学习周期大于 200 时学习率趋于稳定, 因此图中只画出了前 200 个学习周期中的学习率. 图 7(a) 所示为期望输出脉冲序列由不同发放频率的 Poisson 过程生成时学习率的变化过程, 开始时神经元实际输出的脉冲较多, 学习率从 0.005 迅速减小, 之后随着学习周期的增加, 学习率逐渐增大, 最后趋于稳定. 当期望输出脉冲序列的发放频率逐渐增大时, 最终学习率逐渐减小. 期望输出脉冲序列的发放频率分别取 20Hz、50Hz、100Hz 和 150Hz 时, 对应的最终学习率分别为 0.0101、0.0049、0.0022 和 0.0011. 图 7(b) 所示为期望输出脉冲序列由线性方法编码时学习率的变化过程. 从图中可以看出, 学习率在学习过程中的变化情况与图 7(a) 相似. 期望输出脉冲序列的发放频率分别取 20Hz、50Hz、100Hz 和 150Hz 时, 对应的最终学习率分别为 0.01、0.005、0.0017 和 0.0009. 该实验说明本文提出的学习率自适应方法在学习过程中是有效的, 能够在学习过程中根据神经元实际输出脉冲序列的发放频率对学习率进行自适应地调整.

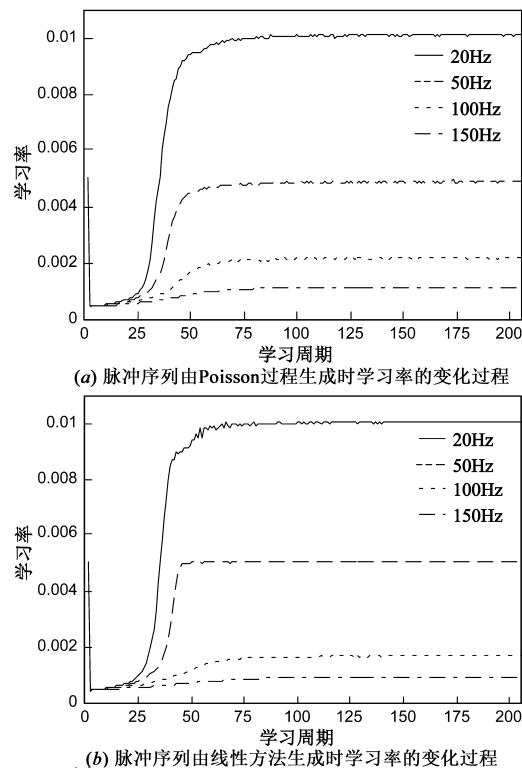


图7 期望输出脉冲序列的发放频率不同时学习率的变化过程

4.3 核函数的比较与分析

4.3.1 不同核函数的表示

对于 STKLR 来说,突触权值的学习规则最终表示为脉冲序列核的累加形式,为了考察不同的核函数对 STKLR 学习性能的影响,我们选择 Gaussian 核函数、Laplacian 核函数、 α -核函数以及 Inverse Multiquadratic 核函数,使用这些核函数对算法进行测试.本文中采用的核函数如表 1 所示.

表 1 脉冲序列的核函数

核函数	表达式
Gaussian	$\kappa(x, y) = \exp\left(-\frac{ x-y ^2}{2\sigma^2}\right)$
Laplacian	$\kappa(x, y) = \exp\left(-\frac{ x-y }{\sigma}\right)$
Inverse Multiquadratic	$\kappa(x, y) = \frac{1}{\sqrt{ x-y ^2 + c^2}}$
α -function	$\kappa(x, y) = \frac{ x-y }{\sigma} \exp\left(-\frac{ x-y }{\sigma}\right)$

4.3.2 核函数的学习性能

由于本文前面的实验中 STKLR 均采用 Gaussian 核函数,首先在基准参数下对 Gaussian 核的参数 σ 进行分析.图 8 给出了 Gaussian 核函数在参数不同时的脉冲序列学习结果, Gaussian 核的参数 σ 取值分别为 0.1、0.5、1、2、5 和 10 共六个值.图 8(a) 所示为核函数参数不同时的学习精度.从图中可以看出,随着核函数参数 σ 的增大, STKLR 的学习精度在 $\sigma = 0.1$ 时为 0.6025;

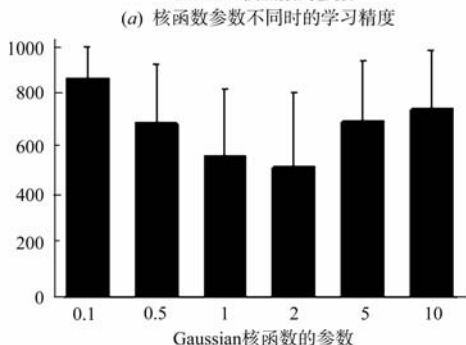
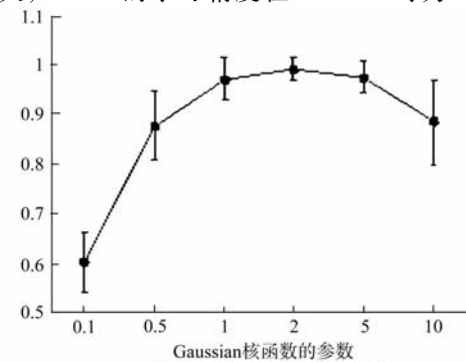


图8 Gaussian核函数在参数不同时的脉冲序列学习结果

随着 σ 的增大,学习精度逐渐增大,当 $\sigma = 2$ 时 C 达到最大值 0.9933,但当 σ 进一步增大时,学习精度逐渐减小.图 8(b) 所示为学习精度达到最高时所需的学习周期,其变化过程表明所需学习周期先减少,然后逐渐增多.当 $\sigma = 0.5$ 时,所需学习周期为 693.95;当 $\sigma = 2$ 时,所需学习周期为 522.81.根据以上分析,核函数的参数对学习精度有较大的影响,我们在实验中, Gaussian 核函数的最佳参数 σ 的值取为 2.

图 9 表示核函数分别为 Gaussian 核(G), Laplacian 核(L), α -核(A), Inverse Multiquadratic 核(IM) 时的脉冲序列学习结果,其中 Gaussian 核的参数为 0.5, Laplacian 核的参数为 5, α -核的参数为 0.5, Inverse Multiquadratic 核的参数为 1.图 9(a) 所示为核函数不同时的学习精度,图 9(b) 所示为当学习精度达到最高时所需的学习周期.此外,通过实验发现多项式核, Sigmoid 核, Multiquadratic 核等不适用于 STKLR,使用这些核函数时将导致算法学习失败或学习效果太差,而 Gaussian 核, Laplacian 核, α -核, Inverse Multiquadratic 核等可以得到理想的学习效果,实现对复杂脉冲序列模式的学习.

从以上实验结果可以看出,应用监督学习算法 STKLR 对脉冲序列时空模式进行学习时,核函数类型和参数的选择非常重要.核函数的类型选择和参数优化方法主要有^[25]: (1) 对于选定的核函数,可采用遍历方式对一个区间内的核参数逐个进行测试和比较.首先为核函数的参数赋初始值,然后开始实验测试,根据

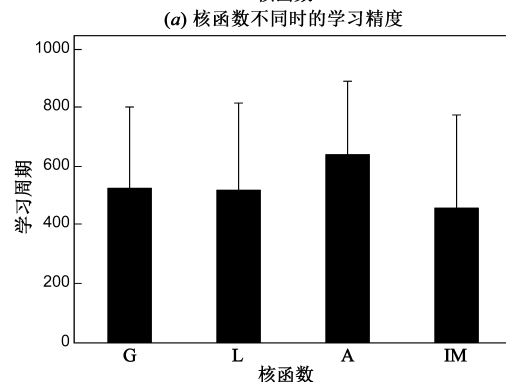
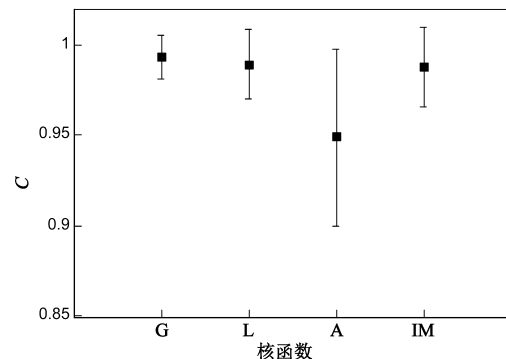


图9 核函数不同时的脉冲序列学习结果

测试精度重复调整核参数值,直至得到满意的测试精度为止.该方法简单可行,特别是对于本文中核函数参数只有一个的情况,容易得到合适的参数值.但缺点是凭经验调整,缺乏足够的理论依据,在参数调整过程中带有一定的盲目性;(2)使用目标函数自适应地选择最佳的核参数,目前较多应用遗传算法优化核函数类型以及参数.但是该类方法在每一次进化过程中,对于群体中的每一个参数,都要进行一次脉冲神经网络的监督学习迭代过程,增加了算法的时间和空间复杂性.

5 结论

脉冲神经元应用精确定时的脉冲序列编码信息,本文基于脉冲序列核函数的定义,提出了一种基于脉冲序列核的脉冲神经元监督学习算法 STKLR,并在学习过程中通过脉冲序列的实际发放频率自适应地调整学习率,实现了对脉冲序列复杂时空模式的有效学习.通过脉冲序列学习过程的分析,以及不同输入突触数目、不同脉冲序列长度、不同输入与期望输出脉冲序列发放频率、不同核函数和核函数参数等因素对算法学习性能影响的分析,表明本文所提出的 STKLR 具有较好的脉冲序列学习能力,并且比 ReSuMe 具有较高的学习精度和较小的学习周期.此外,从 STKLR 的推导过程可以看出,其学习规则表现为以核函数表示的脉冲序列内积形式,与具体脉冲神经元状态变量的表达方式无关,可适用于不同神经元模型.脉冲序列的核函数及其内积理论目前主要用于脉冲序列的相似性度量,较少直接应用于脉冲神经网络监督学习算法的构造,STKLR 可以看作是一种基于脉冲序列内积操作的脉冲神经网络监督学习算法的一般性框架.

进一步的工作考虑将 STKLR 应用到实际的时空模式分类和识别问题,如图像分类和语音识别等.应用脉冲神经网络的监督学习对具体问题的求解可分为三个步骤:首先,将样本数据(如数值、图像或语音数据)的每个分量应用特定的编码策略编码为精确定时的脉冲序列,并对样本所属类别设定期望的脉冲序列;然后,将多个脉冲序列输入网络中的脉冲神经元,得到实际输出的脉冲序列;最后,根据实际输出脉冲序列和期望脉冲序列的误差,应用 STKLR 算法的学习规则对神经网络的突触权值进行调整.通过算法的多次迭代过程,当误差小于设定的最小值或达到最大学习周期时,学习过程结束,并用得到的脉冲神经网络进行模式分类或识别.

参考文献

[1] Quiroga R Q, Panzeri S. Principles of Neural Coding[M]. Boca Raton, FL: CRC Press, 2013.

[2] 蔺想红, 张田文. 分段线性脉冲神经元模型的动力学特性分析[J]. 电子学报, 2009, 37(6): 1270 - 1276.
Lin Xiang-hong, Zhang Tian-wen. Dynamical properties of piecewise linear spiking neuron model[J]. Acta Electronica Sinica, 2009, 37(6): 1270 - 1276. (in Chinese)

[3] Ghosh-Dastidar S, Adeli H. Spiking neural networks[J]. International Journal of Neural Systems, 2009, 19(4): 295 - 308.

[4] Memmesheimer R M, Rubin R, Ölveczky B P, et al. Learning precisely timed spikes[J]. Neuron, 2014, 82(4): 925 - 938.

[5] Kasiński A, Ponulak F. Comparison of supervised learning methods for spike time coding in spiking neural networks[J]. International Journal of Applied Mathematics and Computer Science, 2006, 16(1): 101 - 113.

[6] 蔺想红, 王向文, 张宁, 等. 脉冲神经网络的监督学习算法研究综述[J]. 电子学报, 2015, 43(3): 577 - 586.
Lin Xiang-hong, Wang Xiang-wen, Zhang Ning, et al. Supervised learning algorithms for spiking neural networks: A review[J]. Acta Electronica Sinica, 2015, 43(3): 577 - 586. (in Chinese)

[7] Bohte S M, Kok J N, La Poutré J A. Error-backpropagation in temporally encoded networks of spiking neurons[J]. Neurocomputing, 2002, 48(1-4): 17 - 37.

[8] McKennoch S, Liu D, Bushnell L G. Fast modifications of the SpikeProp algorithm[A]. Proceedings of the International Joint Conference on Neural Networks[C]. Vancouver, Canada: IEEE, 2006. 3970 - 3977.

[9] Booi O, Nguyen T H. A gradient descent rule for spiking neurons emitting multiple spikes[J]. Information Processing Letters, 2005, 95(6): 552 - 558.

[10] Ghosh-Dastidar S, Adeli H. A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection[J]. Neural Networks, 2009, 22(10): 1419 - 1431.

[11] Xu Y, Zeng X, Han L, et al. A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks[J]. Neural Networks, 2013, 43: 99 - 113.

[12] Xu Y, Zeng X, Zhong S. A new supervised learning algorithm for spiking neurons[J]. Neural Computation, 2013, 25(6): 1472 - 1511.

[13] Le Mouel C, Harris K D, Yger P. Supervised learning with decision margins in pools of spiking neurons[J]. Journal of Computational Neuroscience, 2014, 37(2): 333 - 344.

[14] Legenstein R, Naeger C, Maass W. What can a neuron learn with spike-timing-dependent plasticity? [J]. Neural Computation, 2005, 17(11): 2337 - 2382.

[15] Ponulak F, Kasiński A. Supervised learning in spiking neu-

- ral networks with ReSuMe; Sequence learning, classification, and spike shifting [J]. *Neural Computation*, 2010, 22 (2): 467 – 510.
- [16] Sporea I, Grüning A. Supervised learning in multilayer spiking neural networks [J]. *Neural Computation*, 2013, 25 (2): 473 – 509.
- [17] Wade J J, McDauid L J, Santos J A, et al. SWAT: A spiking neural network training algorithm for classification problems [J]. *IEEE Transactions on Neural Networks*, 2010, 21 (11): 1817 – 1830.
- [18] Mohammed A, Schliebs S. SPAN: Spike pattern association neuron for learning spatio-temporal spike patterns [J]. *International Journal of Neural Systems*, 2012, 22 (4): 786 – 803.
- [19] Yu Q, Tang H, Tan K C, et al. Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns [J]. *PLoS One*, 2013, 8 (11): e78318.
- [20] Paiva A R C, Park I, Príncipe J C. A reproducing kernel Hilbert space framework for spike train signal processing [J]. *Neural Computation*, 2009, 21 (2): 424 – 449.
- [21] Park I M, Seth S, Paiva A R C, et al. Kernel methods on spike train space for neuroscience: A tutorial [J]. *IEEE Signal Processing Magazine*, 2013, 30 (4): 149 – 160.
- [22] Xu J, He M, Han J, et al. A comprehensive estimation method for kernel function of radar signal classifier [J]. *Chinese Journal of Electronics*, 2015, 24 (1): 218 – 222.
- [23] Carnell A, Richardson D. Linear algebra for times series of spikes [A]. *Proceedings of the 13th European Symposium on Artificial Neural Networks [C]*. Evere, Belgium: d-side, 2005. 363 – 368.
- [24] Gerstner W, Kistler W M. *Spiking Neuron Models: Single Neurons, Populations, Plasticity [M]*. Cambridge: Cambridge University Press, 2002.
- [25] Zhao M, Ren J, Ji L, et al. Parameter selection of support vector machines and genetic algorithm based on change area search [J]. *Neural Computing and Applications*, 2012, 21 (1): 1 – 8.

作者简介



蔺想红 男, 1976 年 1 月生于甘肃天水. 2009 年获哈尔滨工业大学计算机应用技术专业博士学位, 现任西北师范大学计算机科学与工程学院教授, 硕士生导师. 研究方向为神经网络、进化计算、人工生命、图像处理.
E-mail: linxh@nwnu.edu.cn



王向文 男, 1991 年 3 月生于甘肃天水. 2015 年获西北师范大学软件工程专业硕士学位. 研究方向为神经网络、机器学习.
E-mail: wangxiangwen2@163.com



党小超 (通信作者) 男, 1963 年 9 月生于甘肃兰州. 1995 年获西北工业大学计算机应用技术专业硕士学位, 现任西北师范大学计算机科学与工程学院院长、教授, 硕士生导师. 研究方向为计算机网络、无线传感器网络、智能信息处理.
E-mail: dangxc@nwnu.edu.cn