

脉冲深度学习梯度替代算法研究综述

方维¹⁾ 朱耀宇²⁾ 黄梓涵³⁾ 姚满⁴⁾ 余肇飞⁵⁾ 田永鸿^{1),3),6)}

¹⁾(北京大学深圳研究生院信息工程学院, 深圳, 518055)

²⁾(中国科学院计算技术研究所, 北京, 100190)

³⁾(北京大学计算机学院, 北京, 100871)

⁴⁾(中国科学院自动化研究所, 北京, 100190)

⁵⁾(北京大学人工智能研究院, 北京, 100871)

⁶⁾(鹏城实验室, 深圳, 518000)

摘 要 被誉为第三代神经网络模型的脉冲神经网络(Spiking Neural Network, SNN)具有二值通信、稀疏激活、事件驱动、超低功耗的特性,但也因其复杂的时域动态和离散不可导的脉冲发放过程而难以训练.近年来以梯度替代法和人工神经网络(Artificial Neural Network, ANN)转换 SNN 方法为代表的深度学习方法被提出,大幅度改善 SNN 性能,形成了脉冲深度学习这一全新领域.本文围绕梯度替代法的研究进展,对其中的基础学习算法、编码方式、神经元和突触改进、网络结构改进、正则化方法、ANN 辅助训练算法、事件驱动学习算法、在线学习算法以及训练加速方法进行系统地回顾和综述,并选择其中的代表性方法进行实验对比分析,讨论了目前的研究挑战和可能的解决方案,最后展望了未来可能取得突破的研究方向.

关键词 脉冲神经网络; 梯度替代法; 类脑计算; 神经形态计算; 脉冲深度学习

中图法分类号 TP18 DOI 号 *投稿时不提供 DOI 号*

Review of Surrogate Gradient Methods in Spiking Deep Learning

Wei Fang¹⁾ Yaoyu Zhu²⁾ Zihan Huang³⁾ Man Yao⁴⁾ Zhao Fei Yu⁵⁾ Yonghong Tian^{1),3),6)}

¹⁾(School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen, 518055)

²⁾(Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190)

³⁾(School of Computer Science, Peking University, Beijing, 100871)

⁴⁾(Institute of Automation, Chinese Academy of Sciences, Beijing, 100190)

⁵⁾(Institute for Artificial Intelligence, Peking University, Beijing, 100871)

⁶⁾(Peng Cheng Laboratory, Shenzhen, 518000)

Abstract Spiking Neural Networks (SNNs) are regarded as the third generation of neural network models with binary communication, sparse activation, event-driven computations, and power-efficient characteristics. However, the training of SNNs is challenging because of their complex temporal dynamics and non-differentiable firing mechanisms. Recently, deep learning methods, including the surrogate gradient methods and the ANN to SNN conversion methods, have been proposed and have greatly promoted the performance of SNNs, forming the emerging spiking deep learning research area. This article focuses on the surrogate gradient methods and provides a systemic review including the following topics: (i) the basic learning methods; (ii) encoding methods; (iii) neuron and synapse model modifications; (iv) network structure designs; (v)

收稿日期: 年-月-日; 最终修改稿收到日期: 年-月-日 *投稿时不填写此项*. 本课题得到国家自然科学基金(No. 62425101, No.62332002, No.62027804, No.62088102)资助. 方维, 男, 博士, 助理研究员, 主要研究领域为脉冲神经网络、神经形态计算.E-mail: fwei@pku.edu.cn 朱耀宇, 男, 博士, 特别研究助理, 主要研究领域为类脑计算和代码自动生成.E-mail: zhuyaoyu@ict.ac.cn. 黄梓涵, 男, 博士研究生, 主要研究领域为脉冲神经网络.E-mail: hzh@stu.pku.edu.cn. 姚满, 男, 博士, 助理研究员, 主要研究领域为神经形态计算.E-mail: man.yao@ia.ac.cn. 余肇飞, 男, 博士, 助理教授, 主要研究领域为计算机视觉、神经形态计算和计算神经科学.E-mail: yuzf12@pku.edu.cn. 田永鸿, 博士, 教授, 中国计算机学会(CCF)高级会员, 主要研究领域为视频大数据分析处理、机器学习、类脑计算.E-mail: yhtian@pku.edu.cn.

第1作者手机号码(投稿时必须提供, 以便紧急联系, 发表时会删除): 13041160166, E-mail: fwei@pku.edu.cn

normalization methods; (vi) ANN-auxiliary training methods; (vii) event-driven learning methods; (viii) online learning methods; (ix) training acceleration methods. Additional experiments are conducted to compare representative methods from different categories. Then, the current challenging issues and potential solutions are discussed. Finally, the possible research direction to make breakthroughs is prospected.

Key words Spiking Neural Networks; Surrogate Gradient Methods; Brain-inspired Computing; Neuromorphic Computing; Spiking Deep Learning

1 引言

人工智能在近十几年取得了快速发展^[1],在图像分类^[2-5]、目标检测和跟踪^[6, 7]、语音识别^[8, 9]、机器翻译^[10-12]、游戏对战^[13, 14]、聊天机器人^[15-17]、图像生成^[18-20]等领域获得了巨大成功,引领了新一轮的经济发展和产业变革.在人工智能技术的演进过程中,神经科学提供的视野和灵感起到了重要作用^[21, 22],最典型的例子莫过于神经网络,其起源于神经科学,并在人工智能领域作为主要的计算模型.

第一代神经网络又称为感知机(Perceptron)^[23],接收多个输入并输出布尔值.感知机通过训练可以解决线性分类问题,引发了第一次神经网络热潮.感知机不能处理非线性的异或(Exclusive OR, XOR)问题,且训练算法只能用于单层网络,这些缺点使得对神经网络的关注逐渐衰退.第二代神经网络是人工神经网络(Artificial Neural Network, ANN),不再输出布尔值,而是改用 Sigmoid 等非线性激活输出,结合反向传播算法^[24]实现多层网络的构建和训练.ANN 解决了异或分类问题,引发了第二次神经网络热潮.但受限于芯片行业的发展,90 年代的算力无法支撑大规模神经网络的训练,而小规模神经网络在计算代价、任务性能、可解释性等方面相较于支持向量机^[25]等当时人工智能领域的主流方法并不占优,因而对神经网络的研究又逐渐陷入第二次低谷.

脉冲神经网络(Spiking Neural Network, SNN)被誉为第三代神经网络模型^[26],与生物神经元的机制更为相似,拥有积分发放、阈值触发、稀疏激活、脉冲通信的特性.SNN 凭借极高的生物可解释性,已经被计算神经科学领域广泛使用^[27-29],用于解释和探究生物神经系统的运行原理.由于复杂的时域动态、离散不可导的脉冲发放过程,训练 SNN 比 ANN 更为困难,因而 SNN 在任务性能为主要导向的人工智能领域关注度较少.

神经形态计算(Neuromorphic Computing)^[30, 31]

的蓬勃发展为 SNN 提供了新的机遇.神经形态计算是一种全新的计算范式,旨在借鉴和模仿大脑的运行机理,实现超越传统冯诺依曼架构(Von Neumann Architecture)的全新软件算法和硬件设备,代表性成果包括动态视觉传感器(Dynamic Vision Sensor, DVS)^[32]、视达(Vidar)^[33]等神经形态视觉传感器和 IBM True North^[34]、Intel Loihi^[35]、达尔文(Darwin)^[36]、天机芯(Tianjic)^[37]等神经形态计算芯片.SNN 被视作神经形态计算领域的主要计算模型,其目标是结合神经形态视觉传感器和计算芯片,充分利用脉冲计算的二值量化、稀疏激活特性,实现感算一体、事件驱动的超低功耗边缘智能(Edge AI)系统^[31].然而,这一设想受限于 SNN 高性能学习算法的缓慢发展,一度难以实现.

2006 年 Hinton 等^[38]使用神经网络在 MNIST 数据集^[39]上击败了基于径向基函数内核(Radial Basis Function Kernel)的支持向量机,以深度学习(Deep Learning)之名拉开了神经网络复兴的序幕^[40].2012 年 Alex 等^[41]构建了大规模深度卷积神经网络 AlexNet 并借助图形处理单元(Graphics Processing Unit, GPU)的强大并行计算能力训练,在 ImageNet 大规模图像识别挑战赛^[42]上取得第一,相较于第二名有着 10%正确率的断崖式性能领先,引发了第三次神经网络热潮.深度学习方法以革命般摧枯拉朽的力量将人工智能的各个领域重塑;在这一过程中,以梯度替代法(Surrogate Gradient Method)^[43]和 ANN 转换 SNN 方法(ANN to SNN Conversion, ANN2SNN)^[44]为代表的两大类深度学习方法被提出,并应用于 SNN 的训练,大幅提升 SNN 的任务性能至早期 ANN 的水平^[45],形成了脉冲深度学习(Spiking Deep Learning)这一研究领域.梯度替代法直接训练深度 SNN,训练开销大,但获得的 SNN 时间步数少、延迟低,不局限于频率编码且能够用于神经形态数据分类等时域任务;ANN2SNN 方法则是将训练好的 ANN 转换为 SNN,避开直接训练 SNN,转换速度快、任务精度高,但通常基于频率编码,时间步数多、延迟高且不能用于时域任务.

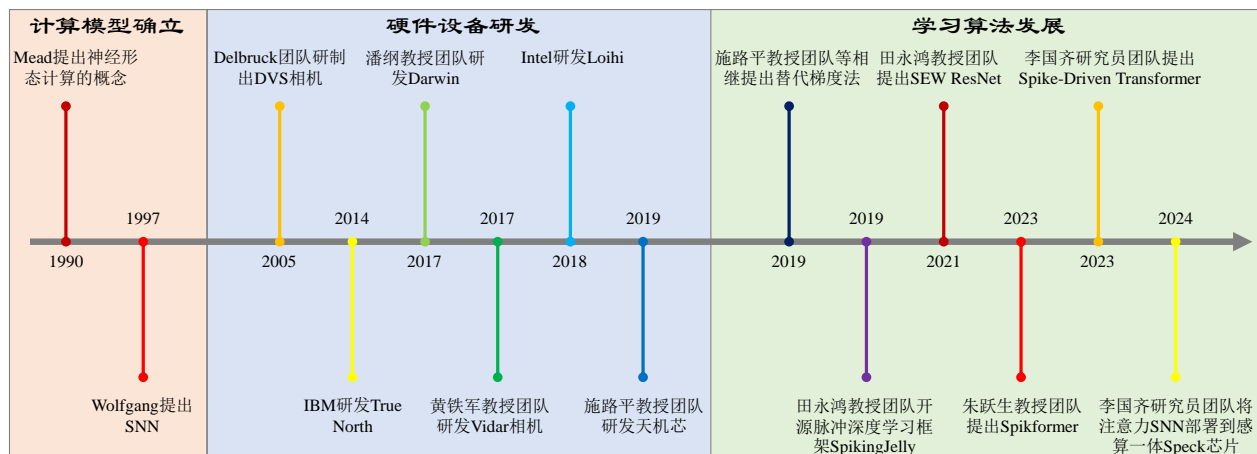


图1 梯度替代学习算法发展历程

本文聚焦于直接训练方法，对基于梯度替代法的深度SNN学习算法进行系统性介绍和总结。

图1总结了梯度替代法的发展历程。1990年Mead提出的神经形态计算概念^[30]，其后Wolfgang于1997年提出并确立了SNN类脑计算模型^[26]。2005年Delbruck团队研制出DVS相机^[32]；它是目前最常用的神经形态视觉传感器之一，基于该传感器的神经形态数据集现已在脉冲深度学习中大量使用。2014年IBM研发出基于异步电路实现的事件驱动神经形态计算芯片True North^[34]，使用非冯诺依曼架构，芯片能耗密度仅为20 mW/cm²，相较于能耗密度典型值为50W/cm²的CPU展示出SNN巨大的能耗优势。2017浙江大学潘纲教授团队研发出国内首个神经形态计算芯片达尔文(Darwin)^[36]，并用于手写数字识别和脑机信号识别任务。同年北京大学黄铁军教授团队模仿视网膜中央凹采样模型，研发出积分型脉冲相机视达(Vidar)^[33]，其工作原理与脉冲神经元积分发放的特性一致，能够实现高速摄像并重构任意时刻的图像数据。2018年Intel研发出Loihi芯片^[35]，并提供了完善的软硬件工具链，被大量研究者用于部署SNN。2019年清华大学施路平教授团队研发出全球首款神经形态异构芯片天机芯(Tianjic)^[37]，支持ANN和SNN混合运行，可以充分结合两者的性能和能耗优势。至此，神经形态视觉传感器和计算芯片已较为完善，脉冲深度学习的数据集和硬件载体已基本构建完成。2019年，施路平教授团队^[46]、Zenke等^[47]、Shrestha等^[48]分别独立提出了通过重定义脉冲发放过程梯度的方式来训练深度SNN的学习算法；该类算法被统称为梯度替代法，与ANN转换SNN算法共同开启了脉冲深度学习时代，其后各类学习算法大量涌现。同年

末，北京大学田永鸿教授团队开源了国际上首批脉冲深度学习框架之一的SpikingJelly框架^[49]，填补了深度SNN软件框架的空白。2021年田永鸿教授团队提出Spike-Element-Wise (SEW) ResNet^[50]，首次训练出超过100层、最高152层的深度SNN，实现了SNN的残差学习。2023年北京朱跃生教授团队提出了首个符合神经形态硬件计算特性的脉冲Transformer架构Spikformer^[51]，同年中国科学院自动化研究所李国齐研究员团队提出了Spike-Driven Transformer^[52]，标志着Transformer架构开始逐渐在SNN领域推广。2024年，李国齐研究员团队将注意力SNN部署到识科技(SynSense)的异步神经形态感算一体Speck芯片^[53]，利用注意力机制将脉冲的稀疏性再次提升，达到了比之前最先进的Intel Loihi芯片的还低的功耗水平。需要声明的是，在脉冲深度学习蓬勃发展的过程中，各类基于梯度替代法的学习算法层出不穷，因图片尺寸有限，难以一一列举，本文仅选取了其中部分影响力较大的代表性工作展示在图1中。

本文将在第二章介绍SNN的基本组分和评测基准作为背景知识，随后在第三章对现有的梯度替代法相关学习算法进行系统分类和讲解。在第四章，本文将设置统一的实验环境，对各类学习算法中的代表性方法进行横向对比，公平比较和分析各类方法的性能。在第五章，本文展望了目前的研究挑战与未来研究方向。在第六章，本文对现有方法进行了总结，讨论了这些方法目前的缺陷和对应的改进方法，并展望梯度替代学习算法未来可能的突破点，即结合神经科学的视角见解与深度学习的强大优化能力，设计脑启发的学习算法，实现如大脑般通用的人工智能。

2 深度 SNN 的基本组分和评测基准

深度 SNN 通常由多个突触层和脉冲神经元层堆叠而成.SNN 的突触层与 ANN 中的基本一致,主要包括卷积层、池化层、全连接层等.批量标准化(Batch Normalization, BN)^[54]和层标准化(Layer Normalization, LN)^[55]等正则化层也经常使用.

SNN 的脉冲神经元是其区别于 ANN 的显著标志,与生物神经系统中的神经元行为更为相似,具有较为复杂的神经动态.来自其他神经元的输入电信号通过树突(Dendrite)传递到神经元的胞体,累计为膜电位(Membrane Potential),当膜电位超过阈值(Threshold)电位时,神经元会将累计的电荷在极短的时间内(约为 1—2 毫秒)一次性释放,形成脉冲(Spike)并通过轴突(Axon)传递到其他神经元.神经元释放脉冲后,膜电位会瞬间降低,这一过程称之为放电后的重置(Reset).

计算神经科学中构建的脉冲神经元模型对生物神经元进行了精细建模,通常使用一个或多个微分方程去描述其神经动态.例如, SNN 中广泛使用的泄露积分发放(Leaky Integrate-and-Fire, LIF)神经元的阈下神经动态为:

$$\tau_m \frac{dV(t)}{dt} = -(V(t) - V_{rest}) + X(t), \quad (1)$$

其中 τ_m 是膜时间常数, $V(t)$ 是膜电位, V_{rest} 是静息电位, $X(t)$ 是输入电流.如果膜电位 $V(t)$ 超过了阈值,则释放脉冲,使用 Heaviside 阶跃函数 $\Theta(x)$ 描述这一过程:

$$S(t) = \Theta(V(t) - V_{th}), \quad (2)$$

其中 $x \geq 0$ 时 $\Theta(x) = 1$, $x < 0$ 时 $\Theta(x) = 0$.当神经元释放脉冲后,膜电位瞬间重置到 V_{reset} ,这一重置过程可以描述为:

$$\lim_{\Delta t \rightarrow 0^+} V(t + \Delta t) = V_{reset}. \quad (3)$$

诸如 Izhikevich 模型^[56]等更为精细的脉冲神经元模型通常需要更多数量的微分方程去描述,计算代价较高,因而在深度 SNN 中较少使用.

对脉冲神经元进行仿真时,一般做法是将连续时间微分方程转换为离散时间差分方程.Fang 等^[49]使用充电、放电、重置三个方程来构建通用离散时间脉冲神经元模型:

$$H[t] = f(V[t-1], X[t]), \quad (4)$$

$$S[t] = \Theta(H[t] - V_{th}), \quad (5)$$

$$V[t] = \begin{cases} H[t] \cdot (1 - S[t]) + V_{reset} \cdot S[t], & \text{硬重置} \\ H[t] - V_{th} \cdot S[t], & \text{软重置} \end{cases}, \quad (6)$$

其中 $H[t]$ 表示充电后、重置前的膜电位, $X[t]$ 表示输入电流, V_{th} 表示阈值, $S[t]$ 表示释放的脉冲, $V[t]$ 表示重置后的膜电位, V_{reset} 表示重置电压.公式(4)表示神经元的充电方程, f 因神经元而异,例如对于 LIF 神经元,参考其微分方程(1)式,可以得到充电的差分方程为:

$$H[t] = V[t-1] + \frac{1}{\tau_m} (X[t] - (V[t-1] - V_{reset})). \quad (7)$$

公式(5)为放电方程,使用 Heaviside 阶跃函数来比较膜电位和阈值,并生成二值脉冲.公式(6)为重置方程,目前在脉冲深度学习领域主要存在两种重置方法,分别为硬重置(Hard Reset)和软重置(Soft Reset).硬重置在释放脉冲后,将膜电位直接设置为 V_{reset} ,研究者们发现其用于梯度替代法训练的深度 SNN 性能较好^[57].软重置则是在神经元释放脉冲后,将膜电位减少 V_{th} ,使用这种重置方式的积分发放(Integrate-and-Fire, IF)神经元在理论上拟合 ReLU 函数的误差更小^[58],因而在 ANN2SNN 中普遍使用.

尽管多数深度 SNN 使用与 ANN 相同的无状态的突触,但神经元是有状态的,且状态是通过逐步迭代的方式生成,因此 SNN 相较于 ANN 引入了时间维度,其处理的输入数据是一个序列,通常用 T 表示输入序列长度,同时也表示运行 SNN 所需的时间步数(Time-step), T 有时也称为仿真步数.

脉冲深度学习蓬勃发展,大量实验结果不断涌现,其中静态图像数据集和神经形态数据集分类任务是最频繁使用的性能评测基准.静态图像数据集的“静态”是相较于动态的神经形态数据而言,因图像通常不包括时域信息,每个样本仅为单张图片.常用的静态图像数据集包括 MNIST^[39]、Fashion-MNIST^[59]、CIFAR^[60]和 ImageNet^[42]数据集,数据集规模和分类难度依次递增.神经形态数据集是从神经形态视觉传感器直接收集,或软件仿真算法将静态图片转换而得到的事件集合,其中每个事件通常以异步的地址事件协议(Address Event Representation, AER)来表示为 (x_i, y_i, t_i, p_i) ,其中 i 是事件索引, (x_i, y_i) 是事件的横纵坐标, t_i 是事件的时间戳, $p_i \in \{-1, 1\}$ 是事件的极性.神经形态数据

集中的事件稀疏但数量众多,一个样本通常包含百万个事件,难以被神经网络直接处理,因而需要通过切片积分等下采样方式转换成帧数据后才能使用^[61].常用的神经形态数据集包括 N-MNIST^[62]、CIFAR10-DVS^[63]、DVS Gesture^[64]、ASL-DVS^[65]、N-Caltech101^[62]、ES-ImageNet^[66]、Spiking Heidelberg Digits (SHD)^[67]等.神经形态数据集常用于评估 SNN 的时域信息处理能力.但 Laxmi 等^[68]等指出 N-MNIST^[62]等拍摄静态图片得到神经形态数据集的时域信息较少,因而对于网络的长期依赖学习能力评估,也有一些研究采用序列(Sequential)图像分类^[69, 70].在序列图像分类任务中,图像会被从左到右逐列输入,网络在同一个时刻只能看到一列图像,最终的性能能够体现网络的记忆能力.

SNN 的能源效率是其突出优势,因而网络的能耗也是评估各类学习算法的重要指标.由于进行实际硬件部署成本较高,目前多数研究都采用理论估算的方式汇报能耗结果.具体而言,该方法假设 SNN 在推理时,对于乘积累加运算(Multiply-Accumulate, MAC)运算,若参与计算的一方是脉冲,则脉冲为 0 的位置不需要计算,MAC 操作转换为选择脉冲为 1 的位置进行累加(Accumulate, AC)实现;对于每种操作的能耗,按照每个 MAC 操作消耗 $E_{MAC} = 4.6 \text{ pJ}$,每个 AC 操作则消耗 $E_{AC} = 0.9 \text{ pJ}$ ^[71]来估算;不考虑在内存中读写数据带来的功耗;统计网络中所有乘加数量 n_{MAC} 和加法操作数量 n_{AC} ,即可得到网络整体能耗为 $E_{total} = E_{MAC}n_{MAC} + E_{AC}n_{AC}$.

3 深度 SNN 的梯度替代训练算法

由于高性能学习算法的缺失,SNN 一度只能解决 MNIST 分类这种简单任务,不具备处理复杂现实世界问题的能力.近年来随着脉冲深度学习方法的相继提出,SNN 的性能大幅度提升至实用水平,研究者们甚至成功构建出基于脉冲计算的超低功耗边缘智能系统^[37, 53, 72].本章将对脉冲深度学习中的梯度替代法这一大类算法进行详细介绍,全面梳理现有研究成果和最新进展.

3.1 基础学习算法

反向传播和梯度下降是深度学习中参数优化的核心方法,其依赖于神经网络前向传播使用连续可导的操作.但 SNN 的脉冲发放过程,即(5)式使用

的 Heaviside 阶跃函数 $\Theta(x)$ 是离散、跃变的,其梯度为冲击函数 $\delta(x)$:

$$\Theta'(x) = \delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0, \end{cases} \quad (8)$$

在反向传播中使用 $\delta(x)$ 会破坏正常的梯度传播,使得网络无法训练.

量化神经网络(Quantized Neural Network)领域的研究者,对权重或激活值进行量化时也遇到了类似的问题.量化函数是典型的输入连续、输出离散的函数,其梯度也几乎处处为 0. Bengio 等^[73]提出了直通估计器(Straight-Through Estimator)以解决这一问题,其核心思路是在前向传播时使用量化函数,而在反向传播时重新定义前向传播的梯度.例如四舍五入量化的 Round 函数及其通过直通估计器定义的梯度可以表示为如下形式:

$$y = \text{Round}(x), \quad (9)$$

$$\frac{dy}{dx} = 1. \quad (10)$$

在上述定义下, Round 函数的梯度被视作恒等函数 $y = x$ 的梯度.

在 SNN 研究领域,施路平教授团队^[46]、Zenke 等^[47]、Shrestha 等^[48]在 2018 年分别独立地提出了梯度替代算法,其思路与直通估计器类似,成为目前直接训练深度 SNN 算法的基石.梯度替代法在前向传播时使用 Heaviside 阶跃函数 $\Theta(x)$ 生成二值脉冲,而在反向传播时重定义 $\Theta'(x)$ 为替代函数 $\sigma(x)$ 的导数 $\sigma'(x)$.具体而言,(5)式仍然用于前向传播,而其反向传播则按照重定义的梯度:

$$\frac{\partial S[t]}{\partial (H[t] - V_{th})} = \sigma'(H[t] - V_{th}). \quad (11)$$

替代函数 $\sigma(x)$ 通常是连续、光滑的函数,拥有数值正常的导.常用的替代函数包括 Rectangular^[46]、SuperSpike^[47]、ArcTan^[74]、Sigmoid 等,这些函数大多是单调递增的偶函数,且值域为(0, 1),可以视作 $\Theta(x)$ 的光滑近似.

除了光滑近似,对于替代函数梯度的另一种解释基于概率性发放脉冲的期望的梯度^[48, 75],这一思想也被量化神经网络用于解释直通估计器^[76].具体而言,将前向传播视作概率性的脉冲发放:

$$p(s=1) = \sigma(x), \quad (12)$$

$$p(s=0) = 1 - \sigma(x), \quad (13)$$

其中 $\sigma(x)$ 是 Sigmoid 等值域为(0,1)的函数,将输入转换为概率.相应的反向传播定义为:

$$\frac{ds}{dx} \approx \frac{d(E(s))}{dx} = \frac{d(\sigma(x))}{dx} = \sigma'(x), \quad (14)$$

从而得到了与将 $\sigma(x)$ 视作 $\Theta(x)$ 的光滑近似时完全相同的梯度表达式。

替代函数是深度 SNN 训练算法的基础组件,对网络性能有着重要影响.Zenke 等^[43]首先以数值模拟的方式形象展示了替代函数为何能够训练网络,他们在一个两层小网络上对比数值梯度和替代梯度,发现两者相似性较高,表明替代梯度所指示的梯度下降方向与真实梯度接近;数值梯度时而在 0 和较大值之间跳动,而替代梯度则连续且数值有限,展示出替代梯度数值稳定的性质.Zenke 等^[77]进一步研究了替代函数的形状对训练的影响.以 SuperSpike 替代函数为例,其定义为:

$$\sigma'(x) = (\beta|x|+1)^{-2}, \quad (15)$$

其中 β 为形状参数, β 越大则 $\sigma(x)$ 越接近 $\Theta(x)$, 但相应的梯度也越集中于 0 附近且数值越大, $\sigma'(x)$ 越接近 $\delta(x)$. 他们在含有一个隐藏层的 SNN 上进行学习率和替代函数形状参数的网格搜索实验,发现只要替代梯度不是常数,则训练能达到的最高正确率与形状参数无关,表明梯度替代法对替代函数的形状具有鲁棒性.但他们的这一实验也发现,不同的形状参数对应的能达到最高性能的学习率的范围也不同,如果使用不恰当的形状参数,则需要非常精细地调整学习率才能达到较好性能. Lian 等^[78]的研究表明, β 直接影响替代梯度函数的宽度,太大的 β 会导致替代梯度和真实梯度之间误差较大,造成梯度不匹配;而太小的 β 则使得替代函数的宽度狭窄,造成梯度消失;两者都会导致网络难以训练. Li 等^[79]则发现不同形状参数下,替代梯度和数值梯度的余弦相似度存在差异. 另一方面, Zenke 等^[72]也测试了替代梯度的尺度对学习的影响,他们将式(14)乘上 β 得到 $\beta\sigma'(x)$ 以改变梯度的尺度,发现不同的 β 对性能影响较大,这一问题可能是较大的梯度在 SNN 使用通过时间反向传播(Back Propagation Through Time, BPTT)累乘梯度时引发梯度爆炸所致.

如何选择合适的替代函数,已有少量研究进展. Li 等^[79]基于数值梯度和替代梯度的余弦相似度来设置替代梯度函数的超参数,但由于数值梯度计算代价高昂,该方法只被用于网络首层脉冲神经元. Lian 等^[78]根据膜电位的分布动态调整替代梯度的宽度,避免梯度不匹配或梯度消失问题. Che 等^[80]

对不同替代函数参数在训练时使用 Softmax 混合,而推理时使用 Argmax 选择,实现可微分参数搜索.

3.2 编码方式

神经编码(Neural Coding)泛指神经元如何将信息表示为电生理活动,是神经科学中的一个重要研究问题.在脉冲深度学习中,该问题细化为如何使用脉冲序列来表示信息,研究话题涵盖如何将非脉冲的输入编码成脉冲,以及 SNN 内部如何使用脉冲传递信息.从类型来看,编码方式可分为两类:频率编码(Rate Coding)和时间编码(Temporal Coding),前者使用脉冲的发放频率表示数值的大小,而后者则通过脉冲的发放时刻传递信息.

常见的图像、视频、语音等数据都以整数或浮点值形式存储,与 SNN 期望的脉冲形式不符,研究者们提出了多种输入编码方式解决这一问题.泊松编码(Poisson Coding)是频率编码的代表性方法.对于输入 $x \in (0,1)$,标准的泊松编码生成数量符合强度为 x 的泊松分布的脉冲,而简化的实现则用每个时间步中脉冲发放概率均为 x 的二项分布近似.泊松编码在早期的深度 SNN 研究中^[81, 82]较多使用.

时间编码方式则通过脉冲的发放时刻来表示信息,首达脉冲编码(Time-To-First-Spike Coding)^[83, 84]是其中的典型代表.首达脉冲编码遵循“刺激越强,发放越早”的规则,将输入转换成对应的发放时刻,即:

$$S[t] = \begin{cases} 1, & t = t_f, \\ 0, & t \neq t_f, \end{cases} \quad (16)$$

$$t_f = \text{Round}((T-1) \cdot (1-x)), \quad (17)$$

其中 Round 为四舍五入的量化函数, T 为时间步数.

泊松编码对于数值较小的输入很难触发脉冲,随机发放的特性需要较长时间步才能获取稳定结果,因而时间步数多、延迟高、性能低;而首达脉冲编码只能释放单个脉冲,且式(17)中的 Round 函数损失了一定信息,实际性能也欠佳.目前多数高性能深度 SNN^[50-52]都采用直接输入编码^[85]方式.如果输入是静态的图片,而 SNN 需要运行 T 次,则该方法将输入简单重复 T 次得到输入序列.在该输入方式下,连续浮点输入转换为离散二值脉冲的编码实际由首个突触层和脉冲神经元层完成,它们可以视作可学习的编码器^[74].Rathi 等^[85]的实验结果表明,尽管直接输入编码在首层引入了浮点计算,但相较于泊松编码,该方法的时间步数大幅度降

低,因而最终网络的能耗也低于使用泊松编码。

绝大多数 ANN2SNN 方法使用频率编码, SNN 内部通过脉冲的发放频率表示转换前的 ANN 中 ReLU 的激活值;此外,首达脉冲编码^[86-88]、相位编码(Phase Coding)^[89]和进制编码(Radix Coding)^[90]等效率更高的时间编码方式.而在基于直接训练的深度 SNN 中,梯度替代法提供了强大的端到端训练能力,因而无需手动设计网络内部的编码方式.从实际表现看,梯度替代法训练出的 SNN 所需的时间步数也远少于由转换法得到的 SNN,效果较好.但理解直接训练的 SNN 内部的编码方式,依然是一个值得探索的方向,现有研究较少.Li 等^[91]对基于梯度替代法训练的 SNN 和同结构 ANN 进行了相似性分析,发现 SNN 的特征与 ANN 具有高度相似性,时间维度并未提供太多额外信息;Hu 等^[92]则进而发现不同时间步的梯度相似性也很高;以上研究表明现有的深度 SNN 内部的编码方式可能较为接近频率编码.需要指出的是,上述研究使用直接输入编码、纯前馈 SNN,而其结论未必适用时间编码输入或者带有反馈连接的 SNN.

3.3 神经元和突触改进

深度脉冲神经网络的主要组分是神经元和突触,两者均对网络性能有着重要影响,有大量研究对其进行改进,提出了多种新型神经元和突触模型.

PLIF 神经元 (Parametric Leaky Integrate-and-Fire Neuron)模型^[74]是最早的神经动态可学习的神经元模型之一,其基于经典的 LIF 神经元模型,将膜时间常数 τ_m 参数化并设置为可学习.PLIF 神经元的阈下神经动态为:

$$H[t] = V[t-1] + k(a) \cdot (-(V[t-1] - V_{reset}) + X[t]), \quad (18)$$

其中膜时间常数的倒数,即 τ_m^{-1} 被重参数化为 $\tau_m^{-1} = k(a)$,而 a 是真正的可学习参数. $k(a) \in (0,1)$ 是限幅函数,确保 $\tau_m > 1$ 以防止神经元出现自充电的情况,在实践中通常取 $k(a) = \text{Sigmoid}(a)$.PLIF 神经元通常设置每一层只有一个可学习参数 a ,即该层神经元的膜时间常数是共享的,既大幅度减少了参数量,又与生理实验证据中相邻脑区神经元性质类似这一特性符合;而不同神经元层的参数 a 在训练后不尽相同,保持了神经元的异质性.以往的研究为了减少调参成本,倾向于在整个网络中使用相同的膜时间常数 τ_m ,丧失了神经元的异质性,并且只

训练网络权重,使得网络的表达能力有所下降;PLIF 神经元的提出解决了这一问题,并实现了突触权重和神经动态的联合学习.但 PLIF 神经元在训练完成后与 LIF 神经元无异,因而其可以视作一种参数化和训练技巧,而非一种新型神经元.

GLIF 神经元(Gated Leaky Integrate-and-Fire Neuron)^[93]进一步扩展了神经动态的学习范围,其将神经元对上一时刻的状态衰减、对输入的累计、释放脉冲引发的重置均进行参数化,分别表示为可学习的门控 $\mathbb{G}_\alpha, \mathbb{G}_\beta, \mathbb{G}_\gamma$,具体形式为:

$$\mathbb{G}_\alpha = (1 - \alpha(1 - \tau_{exp})) \cdot H[t-1] - (1 - \alpha)\tau_{lin}, \quad (19)$$

$$\mathbb{G}_\beta = (1 - \beta(1 - g[t])) \cdot X[t], \quad (20)$$

$$\mathbb{G}_\gamma = -\gamma \cdot \mathbb{G}_\alpha - (1 - \gamma) \cdot V_{reset}, \quad (21)$$

其中 α, β, γ 分别是可学习的门控系数; τ_{exp} 和 τ_{lin} 分别表示指数和线性衰减系数; $g[t]$ 表示随时间变化的突触权重.GLIF 神经元也使用了参数共享的技巧,其可学习参数支持设置为逐层或逐通道,因此也几乎不增加网络的参数量.GLIF 神经元通过可学习的门控,实现了指数衰减和线性衰减、无状态突触和有状态突触、硬重置和软重置的混叠,因此具有很强的表达能力,但也带来了较大的计算量,较于传统神经元,其训练速度有着较大下降.

MLF 方法(Multi-Level Firing Method)^[94]使用多个脉冲神经元构成一个神经元组,组内的神经元使用不同的阈值,并将输出的脉冲累计,相较于传统方法使用的单个神经元,具有更好的拟合能力,但神经元层输出的不再是纯二值脉冲,可能会难以在一些仅支持二值计算的神经形态计算芯片上实现.

CLIF 神经元 (Complementary Leaky Integrate-and-Fire Neuron)^[95]旨在解决 LIF 神经元中漏电行为导致的长期梯度衰减问题,通过增加补充电位(Complementary Potential)实现跨多个时间步的稳定梯度传播:

$$M[t] = M[t-1] \cdot \sigma\left(\frac{1}{\tau_m} H[t]\right) + S[t], \quad (22)$$

$$V[t] = H[t] - S[t] \cdot (V_{th} + \sigma(M[t])), \quad (23)$$

其中 $M[t]$ 表示补充电位, $\sigma(\dots)$ 是 Sigmoid 激活函数.公式(22)表示 $M[t]$ 的更新过程,其自身衰减与膜

电位的衰减程度相反,并在神经元释放脉冲,即膜电位瞬间下降时自增,实现了与膜电位的互补.公式(23)基于软重置的(6)式进行修改,引入了 $M[t]$ 使得膜电位能自适应调整,避免过高或过低的发放率.尽管 PLIF 神经元和 GLIF 神经元神经动态中都使用了 Sigmoid 函数,但该函数只用于包装可学习参数,其输出在训练完成后是常数,神经元推理时并不需要计算;而 CLIF 神经元的式(22)和式(23)中 Sigmoid 函数的输入是依赖于数据的,不能在推理时去除,而 Sigmoid 函数复杂的指数计算,可能带来较高的硬件实现代价.

传统神经元皆为串行计算,不能充分利用 GPU 的大规模并行计算能力加速,是深度 SNN 训练速度缓慢的一个重要原因.PSN(Parallel Spiking Neurons)^[70]是首个并行脉冲神经元模型,其灵感来自于传统串行脉冲神经元在不发放脉冲的一段时刻内,膜电位的逐时间步迭代求解可以写成非迭代形式的解析解.受此现象启发, Fang 等^[70]去除了传统脉冲神经元的重置过程,并发现对于大多数神经元而言, $H[t]$ 可以表达为输入 $X[i]$ 的线性组合,以此提出了 PSN 模型,其神经动态为:

$$H = WX, W \in \mathbb{R}^{T \times T}, X \in \mathbb{R}^{T \times N}, \quad (24)$$

$$S = \Theta(H - B), B \in \mathbb{R}^T, S \in \{0,1\}^{T \times N}, \quad (25)$$

其中 X 是输入序列, W 是可学习权重, H 是膜电位, B 是可学习阈值, S 是输出脉冲, N 是神经元数量, T 是时间步数.PSN 膜电位的生成需要用到所有时刻的信息,而在一些实际任务中,未来信息不可在当下获取,为解决这一问题, Fang 等^[70]提出 Masked PSN, 其对式(24)中使用的权重增加掩模,只使用包括 t 时刻在内的最新 k 个输入来生成 $H[t]$, 具体形式为:

$$H = (W \cdot M_k)X, W \in \mathbb{R}^{T \times T}, M_k \in \mathbb{R}^{T \times T}, X \in \mathbb{R}^{T \times N}, \quad (26)$$

其中 M_k 定义为:

$$M_k[i][j] = \begin{cases} 1, & j \leq i \leq j+k-1 \\ 0, & \text{其他情况} \end{cases}. \quad (27)$$

PSN 和 Masked PSN 的权重均是逐时刻的,难以处理变长序列.Fang 等^[70]进而将 Masked PSN 的权重设置成时域共享,得到 Sliding PSN, 其神经动态为:

$$H[t] = \sum_{i=0}^{k-1} W_i \cdot X[t-k+1+i], \quad (28)$$

$$S[t] = \Theta(H[t] - V_{th}), \quad (29)$$

其中 $W = [W_0, W_1, \dots, W_{k-1}] \in \mathbb{R}^k$ 是可学习权重, 约定 $j < 0$ 时 $X[j] = 0$, V_{th} 是可学习的阈值.PSN、Masked PSN、Sliding PSN 统称为 PSN 家族,相较于传统串行神经元, PSN 家族无需逐步迭代,可以使用并行度更高的矩阵乘法来计算膜电位,仿真速度大幅度提升;使用直接的权重连接替换传统神经元的基于马尔科夫链的依赖关系,长期依赖的学习能力也得到增强.PSN 家族最大的缺陷在于皆为高阶神经元,需要存储多个历史输入,推理的内存消耗会剧增.

与 PSN 并行化的思路类似的研究还包括随机并行脉冲神经元^[96],其也通过忽略重置来避免膜电位的迭代求解,但脉冲的生成不是直接使用 Heaviside 阶跃函数,而是采用概率性发放的形式,发放概率由膜电位决定,而梯度则使用替代函数来重新定义.

AMOS (At Most One Spike)神经元只能释放不超过一个脉冲,相较于不做任何限制的普通神经元,更少的脉冲发放次数带来更低的理论功耗.AMOS 神经元通常与首达脉冲编码结合用于 ANN2SNN 方法^[86],以单个脉冲精确的发放时刻来表示信息.而在 SNN 的直接训练算法中, AMOS 神经元的足迹最早可以追溯到早期的经典 SNN 有监督学习算法 SpikeProp^[86], Mostafa 等^[97]则是首次将 AMOS 神经元用于深度 SNN,其在训练算法上沿用了之前 Mostafa^[98]的方法,层之间传递的是脉冲发放时刻,借助于输入和输出脉冲发放时刻的因果(先后)关系来传递梯度,但确定时刻的先后关系需要排序和遍历,复杂度较高.Kheradpisheh 等^[83]提出的 S4NN (Single-Spike Supervised Spiking Neural Network)也使用 AMOS 神经元,但层之间传递的是脉冲的值,脉冲发放时刻则被隐式地用于通过链式法则定义梯度,相较于 Mostafa 等^[97]的方法复杂度大幅度降低,易于实现,且任务性能更好.总体而言, AMOS 神经元脉冲数量少的优势非常直观,但研究还处于早期阶段,与传统神经元的性能有着较大差距.

图 2 对本小节介绍的部分神经元进行了梳理,清晰地展示了神经元改进工作之间的脉络关系.表 1 总结了部分脉冲神经元改进研究在多个数据集上的时间步数和分类正确率,以“步数|正确率”的形式展示.整体来看,随着神经动态复杂度的提升,神

表 1 脉冲神经元分类任务仿真步数和正确率

神经元/数据集	CIFAR10	CIFAR100	ImageNet	DVS Gesture	CIFAR10-DVS
PLIF	8 93.50			20 97.57	20 74.80
GLIF	2 94.44	2 75.48	4 67.52		
	4 94.85	4 77.05	6 69.09		16 78.10
	6 95.03	6 77.35			
MLF	4 94.25			40 97.29	10 70.36
CLIF	4 96.01	4 79.69			
	6 96.45	6 80.58			
	8 96.69	8 80.89			
PSN 家族					4 82.30
	4 95.32		4 70.54		8 85.30
					10 85.90

神经元的表达能力得到提高,因而网络的任务性能也进一步提升,但这通常也会导致计算代价的提升和训练速度的降低,而神经元的并行化则可能是这一问题的解决途径.需要注意的是,AMOS 神经元类方法目前任务性能还较低,并且主要使用 MNIST 之类的简单数据集评测性能,因而没有列入到表 1 中进行对比.

深度 SNN 中所使用的突触模型通常与深度 ANN 中相同,但也有一些研究者对突触进行了更精细的建模,引入额外的时域动态或突触延迟.Fang 等^[99]将常用的无状态的突触更改为由差分方程描述的有状态突触,使得突触也具有了一定的记忆,增强了整个网络在记忆任务上的学习能力. Ilyass 等^[100]通过时间步维度上的扩张卷积来移动脉冲发

放的位置,从而对突触延迟进行建模,同时使得突触延迟也参与到网络的训练,在时域任务上以更少的参数超越了传统方法的性能.但这些方法都使得突触的复杂度大幅度提升,网络的训练速度下降、内存消耗激增,因而尚未应用于大规模深度 SNN.

3.4 网络结构改进

网络结构改进一直是深度学习领域的热门研究方向. ANN 领域已有诸多成熟的网络结构,但它们在设计时并未考虑神经形态计算的特性,直接用于 SNN 会引发性能退化问题,因而脉冲深度学习领域的相关研究主要集中于对已有网络结构的脉冲化改进.

梯度替代法的出现使得 SNN 研究者能够训练中等规模的深度脉冲卷积网络.然而,研究者们发现若继续采用简单堆叠卷积层的方式来增加网络规模,则性能难以继续提升.研究者们开始考虑构建基于残差连接的深度 SNN 解决上述问题.残差连接起源于 ResNet^[4],如图 3(a)所示,是现代深度神经网络结构中不可缺少的一部分. Spiking ResNet 是 ResNet 的 SNN 版本,最早用于 ANN 转换 SNN^[101]并取得了较好的效果,其结构如图 3(b)所示.但是,如果直接将 ResNet 的残差结构沿用至 SNN 中(即 Spiking ResNet),在训练十几层的网络时即出现性能退化^[102],表现为更深的模型相较于浅层模型,具有更高的训练集误差. Fang 等^[50]从恒等变换和梯度传播角度进行分析,发现 Spiking ResNet 难以实现恒等变换、易于引发梯度消失或梯度爆炸,因此无法有效加深 SNN.为解决这一问题, Spike-Element-Wise (SEW) ResNet^[50]被提出,残差块结构如图 3(c)所示,其将脉冲神经元的位置调换

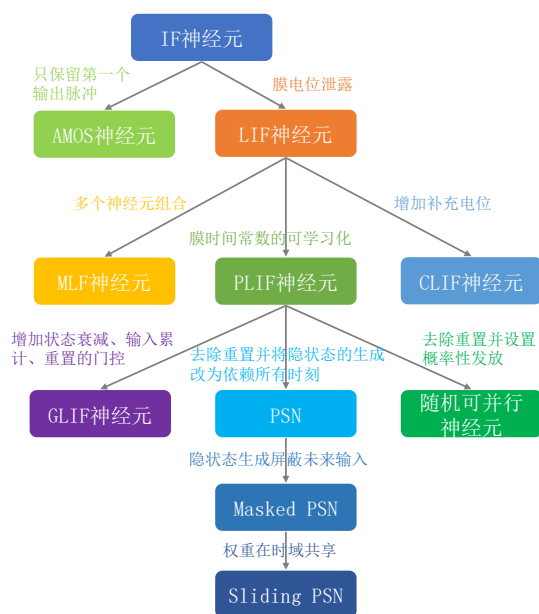


图 2 部分神经元改进工作之间的联系

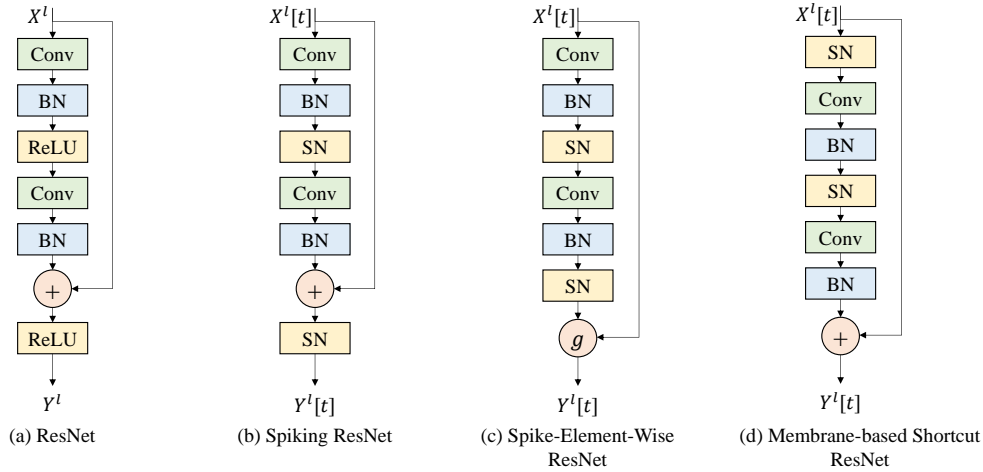


图3 常见的残差块结构

到残差连接之前, 然后使用一个逐元素操作函数 g 来实施残差连接, 其中 g 可以是加法、乘法、取反后再乘法等. SEW ResNet 在 ImageNet 数据集上进行了验证, 实验结果证实了模型性能随深度稳定增加, 首次实现了 SNN 中的残差学习, 并将 SNN 规模扩大至数百层. Membrane-based Shortcut (MS) ResNet^[103] 是另一种能够实现恒等变换的脉冲残差连接方式, 其将每个残差块中第一个脉冲神经元的输入和最后一个 BN 层的输出进行连接, 结构如图 3(d) 所示, 实现了神经元膜电位层次的残差学习, 同样能够将 SNN 规模扩大至数百层.

SEW ResNet 和 MS ResNet 都解决了深度 SNN 的退化问题, 但同时也引入了新的问题. SEW ResNet 主要使用实验性能最好的加法来连接残差块的输入和最后一个 SN 的输出脉冲, 导致残差块输出的实际上是脉冲之和, 是非负整数而非二值脉冲, 这可能丧失了 SNN 的二值特性以及对应的硬件实现时免乘法器的优势; MS ResNet 则是使用残差连接在网络层之间传递稠密的浮点值, 破坏了 SNN 事件驱动通信的特性, 难以在异步芯片实现.

在 ResNet 中添加额外的注意力(Attention)模块能够提升神经网络的全局建模能力, 从而有效提升任务性能^[11, 104, 105]. 这一做法在 Spiking ResNet 中同样有效. Yao 等^[106]提出了时域注意力(Temporal-wise Attention)机制, 将输入在宽、高和通道维度上进行平均后, 送入由 2 层多层感知机(Multilayer Perceptron, MLP)组成的小网络处理, 并输出注意力分数, 然后与不同时刻输入再进行点乘. 这个额外插入的 2 层 MLP 网络就是注意力模块, 起到辅助提取全局信息的作用. 通过设计更高效的注意力模块^[107-109], 或者将注意力机制应用于时间、空间、通

道等多个维度^[110, 111], SNN 在各种任务中的性能得到显著提升. 值得一提的是, 与注意力 ANN 相比, 受益于事件驱动计算特性, 在 SNN 中增加额外的注意力模块通常会使得整个网络的能耗进一步降低. Yao 等的一系列工作^[53, 107, 108, 110]以 Spiking ResNet 为例, 对这一特性进行了深入研究和应用. Spiking ResNet 包含了循环和卷积两种基本操作, 这可以提升参数在时间和空间上的利用效率, 但也使得 SNN 具有“时空不变性^[112]”, 从而导致较差的全局建模能力^[113]. 与此同时, Spiking ResNet 的时空不变性还会引入大量的噪声冗余特征^[107]. 注意力模块能够有效抑制 SNN 中的噪声脉冲, 且优化正常特征, 因此能够在带来性能提升的同时显著降低能耗. 注意力 SNN 的功能在边缘计算芯片上也得到了验证^[53, 114, 115]. 特别是, 将注意力 SNN 部署到识科技(SynSense)的异步神经形态感算一体 Speck 芯片^[53]后, 实测数据显示, 在 DVS128 Gesture 数据集上, 注意力机制能带来 9% 的性能提升, 同时平均功耗由 9.5mW 降低至 3.8mW.

以自注意力(Self Attention)机制为基础的 Transformer^[105]是继 ResNet 之后的影响力最大的网络结构, 自提出以来便在多个领域刷新了性能指标, 成为目前人工智能领域最常用的网络架构之一. 如何有效结合 Transformer 架构的高性能和 SNN 的低功耗引起了脉冲深度学习领域内学者们的广泛兴趣. 较早的 Spiking Transformer^[116-118]的主要设计思路是, 将 Transformer 中的部分人工神经元改为脉冲神经元, 并保留诸如自注意力机制、归一化等关键操作来保证任务精度. 这些 Spiking Transformer 架构事实上是 ANN 与 SNN 融合的异构设计, 难以

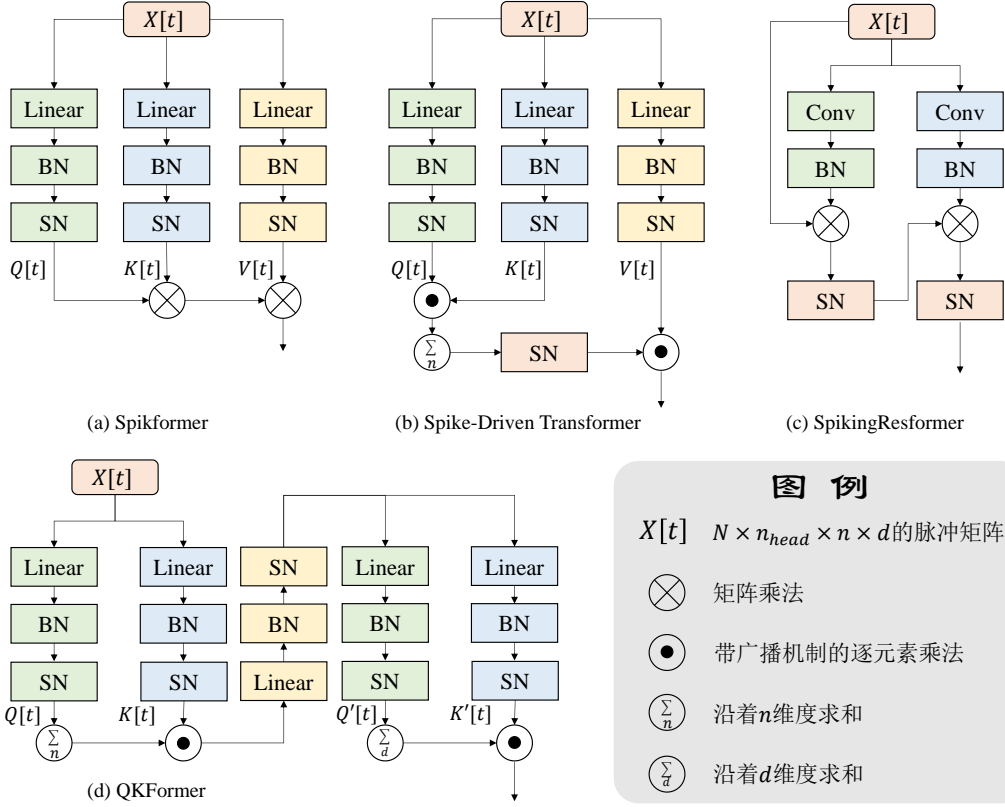


图 4 深度 SNN 中的自注意力机制

真正发挥出 SNN 低功耗的优势.脉冲深度学习领域的研究者们意识到发挥 Spiking Transformer 潜力的关键是如何设计脉冲自注意力算子,并围绕这一问题进行了大量改进.图 4 展示了目前 Spiking Transformer 中主流的自注意力机制.

Zhou 等^[51]指出,自注意力使用的浮点矩阵乘法以及 Softmax 激活涉及的指数运算难以在神经形态芯片上实现.为此,Zhou 等^[51]提出了 Spikformer,使用脉冲自注意力(Spiking Self Attention, SSA)机制,如图 4(a)所示.对于脉冲神经元的输出 $Q[t], K[t], V[t] \in \{0, 1\}^{N \times n_{head} \times n \times d}$,其中 N 表示批量大小, n_{head} 表示注意力头数, n 表示分块(Patch)的数量, d 表示嵌入(Embedding)的维度,则 SSA 按照如下形式计算注意力分数 $score$:

$$score[t] = SN(Q[t]K[t]^T V[t] \cdot s), \quad (30)$$

其中 s 是缩放因子, SN 表示脉冲神经元层. SSA 的两个矩阵乘法参与方都至少包含一个脉冲矩阵,而缩放因子则可以被吸收进脉冲神经元层的阈值;相较于原始的注意力,在 SSA 中 Softmax 激活被去掉了,可以根据 n 和 d 来选择先计算 $Q[t]K[t]^T$ 或 $K[t]^T V[t]$ 以降低复杂度至 $\min(\mathcal{O}(n^2 d), \mathcal{O}(nd^2))$.

Yao 等^[52]进一步提出了脉冲驱动 Transformer 架构,其核心是脉冲驱动自注意力(Spike-Driven Self Attention, SDSA)机制,如图 4(b)所示,其延续了 SSA 不使用 Softmax 激活的设计,但使用逐元素乘法替代矩阵乘法,同时去除了自注意力机制中的归一化操作:

$$score[t] = SN\left(\sum_n (Q[t]K[t])\right) \cdot V[t], \quad (31)$$

其中 \sum_n 表示沿着分块(Patch)的维度求和.需要注意的是, $SN(\dots) \in \{0, 1\}^{N \times n_{head} \times d}$ 而 $V[t] \in \{0, 1\}^{N \times n_{head} \times n \times d}$,

两者的逐元素乘法使用了广播(Broadcast)机制. SDSA 算子的计算复杂度降低至 $\mathcal{O}(nd)$,同时完全消除了乘法,从而使得整个脉冲驱动 Transformer 中仅有稀疏加法.

SpikingResformer^[119]使用了双脉冲自注意力机制(Dual Spike Self Attention, DSSA),如图 4(c)所示.这种注意力机制使用了双脉冲变换(Dual Spike Transformation, DST)算子来替换 Transformer 中的浮点矩阵乘法:

$$DST(X, Y; f(\cdot)) = Xf(Y) = XYW, \quad (32)$$

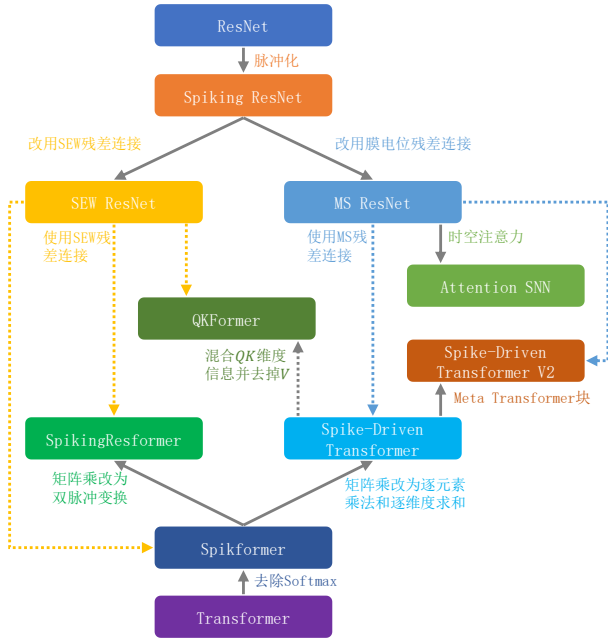


图5 部分网络结构改进研究的联系

$$\text{DST}_T(X, Y; f(\cdot)) = Xf(Y)^T = XW^TY^T, \quad (33)$$

其中 $f(\dots)$ 是 Y 上的广义线性变换, 可以是无偏置的线性层、卷积层和 BN 层等. Shi 等^[119]证明了这种算子是脉冲驱动的, 并且可以用这种算子替换 Transformer 中的浮点矩阵乘法. 利用 DST 算子, DSSA 按照如下形式计算注意力分数:

$$\text{AttnMap}(X[t]) = \text{SN}(\text{DST}_T(X[t], X[t]; f(\cdot)) \cdot c_1), \quad (34)$$

$$\text{score}[t] = \text{SN}(\text{DST}(\text{AttnMap}(X[t]), X[t]; f(\cdot)) \cdot c_2), \quad (35)$$

$$f(X[t]) = \text{BN}(\text{Conv}_p(X[t])), \quad (36)$$

其中 c_1, c_2 是缩放因子, BN 是批归一化层, Conv_p 是卷积核大小和步长为 p 的卷积. SpikingResformer 最大的成功之处在于将自注意力机制巧妙融合进传统卷积架构, 为 SNN 结构设计打开了新思路.

QKFormer^[120] 如其名字所暗示, 只使用 $Q[t], K[t]$, 并通过融合不同维度来提取信息, 其结构展示在图 4(d). QKFormer 首先使用 token 维度元素之和作为通道维度的掩码来提取特征:

$$\text{score}[t] = K[t] \cdot \text{SN}\left(\sum_n Q[t]\right), \quad (37)$$

其中 $K[t] \cdot \text{SN}(\dots)$ 用到了广播机制. QKFormer 进而用通道维度元素之和作为 token 维度的掩码:

$$\text{score}'[t] = K'[t] \cdot \text{SN}\left(\sum_d Q'[t]\right), \quad (38)$$

其中 $Q'[t], K'[t]$ 是不同于原来的 $Q[t], K[t]$, 是经过全连接层处理后的, 如图 4(d)所示. QKFormer 只涉及逐维度求和与逐元素乘法, 不使用矩阵乘法, 注意力机制的复杂度和脉冲驱动自注意力类似, 也低至 $\mathcal{O}(nd)$.

当脉冲化的自注意力机制被成功实现后, 研究者们不再使用原有的 ResNet 等卷积架构作为网络骨架, 而是使用 Transformer 类网络架构, 但 ResNet 中分多个阶段 (Stage) 的设计得到了保留. Spikformer^[51]和 Spike-Driven Transformer^[52]使用 Compact Convolutional Transformer^[121]的网络架构. SpikingResformer^[119]使用了 3 阶段的层级结构以提取不同尺度的特征, 并在 2 层 MLP 之间插入分组卷积层以提取局部特征. Spike-driven Transformer V2^[122]专门设计了 Meta Transformer 块, 由带残差连接的 Token 维度的脉冲驱动的自注意力^[52]和通道维度的 MLP 组成; 在网络架构层次, 前 2 个阶段使用带残差连接的大感受野的 7×7 可分离卷积和小感受野的 3×3 的普通卷积组成的卷积块, 而在后 2 个阶段使用 Meta Transformer 块. QKFormer^[120]则是使用类似 Swin Transformer^[123]的网络结构.

图 5 展示了本小节中介绍的部分网络结构改进研究之间的递进关系. 整体来看, 目前新的研究集中于 Transformer 架构改进, 但其中也使用了大量来自 ResNet 相关研究基础. 图 6 对比了常见深度 SNN 架构在 ImageNet 数据集的分类正确率、功耗和参数量. 除 MS-ResNet 外, 其他网络均使用时间步数 $T=4$; 默认使用 224×224 的图片分辨率进行推理, 但也有部分研究者额外汇报了使用 288×288 图片分辨率推理的结果, 在图中以方形点进行了标注. 图 6 的结果表明, 随着残差结构、自注意力机制的引入, 深度 SNN 的性能得到进一步提升, 在 ImageNet 数据集上已经达到 85% 的正确率, 同时能耗和参数量也不断优化, 新的网络架构向着正确率更高且功耗和参数量更低的方向迅猛发展.

除手动设计网络结构外, 也有研究者将网络结构搜索技术引入 SNN, 实现自动化的模型设计. Na 等^[124]首次将 NAS 引入 SNN, 该方法使用超网训练和遗传算法来优化 SNN 框架, 为了解决搜索方法带来的计算成本, 使用了遗传算法中常见的 one-shot weighting-sharing 方法^[125, 126], 将权重在不同候选框架中共享; 同时还提出了 Spike-Aware 优

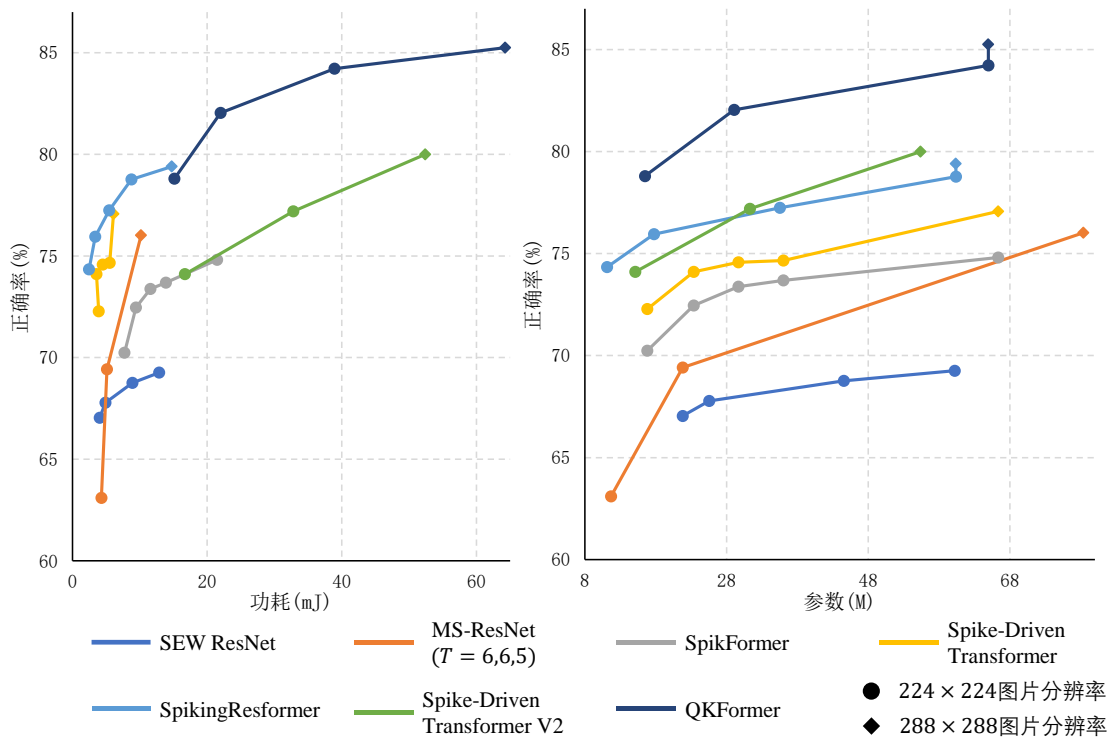


图6 常见深度SNN架构在ImageNet数据集的分类正确率、功耗和参数量

化方程,通过为更多脉冲的候选框架更低的评分作为惩罚,从而限制脉冲数量.Kim等^[124]把搜索空间延拓到前向和反馈连接,同时提出了 Sparsity-Aware Hamming Distance (SAHD)作为指标去评估 SNN 框架,通过使用该指标避免了 NAS 方法中最耗时的超网训练过程,提升了搜索速度.Che等^[80]把搜索空间延拓到层与细胞维度,使其能够应用于深度估计等稠密预测领域,同时把替代函数也纳入搜索空间,优化替代函数的梯度.该方法首次把可微分网络结构搜索方式引入 SNN,只训练代理参数来搜索网络结构,提升了训练速度和精度.Shen等^[127]更关注于搜索的生物原理性,提出了脑启发的神经电路演化策略,此策略包括反馈连接,兴奋性和抑制性神经元^[128],以及基于 STDP 的局部学习方法.同时该方法进一步将任务拓展到了强化学习,取得了与 ANN 相当的性能.为了更好地在准确性与计算成本之间权衡,Yan等^[129]采用了单路径 NAS 方法,即将所有候选框架编码在一个无分支的脉冲超网中,该框架不再训练不同大小的独立卷积核,而是训练一个无分支的超网卷积核,显著降低了计算成本和搜索时间.整体来看,网络结构搜索类方法本身训练开销较大,与需要多个时间步运行的 SNN 结合后更为明显,已有的研究也都在尝试解决这一问题.

近年来还有一些研究将 SNN 处理的数据类型从图片、神经形态数据集等扩展到图(Graph)、点云(Point Clouds)等,图神经网络(Graph Convolutional Network, GCN)、点云网络(PointNet)等架构的脉冲版本被相继提出.Gu等^[127]针对触觉数据构建图结构,并使用图卷积预处理,特征由 LIF 神经元转换为脉冲,继而输出到使用 LIF 神经元的 MLP,是较早的成功将 SNN 结合 GCN 的研究.Zhu等^[128]提出的脉冲卷积图神经网络,则是使用频率编码来将图卷积得到的特征转换为脉冲,且 MLP 中使用三值激活的 LIF 神经元,在 4 个引文网络公共数据集上取得了比部分经典 GCN 模型更好的性能.Li等^[130]使用了自适应阈值的 LIF 神经元,并将其用于聚合邻域信息,由于 LIF 神经元动态的放电机制,使得不同时刻的图结构也呈现动态变化,能够捕捉动态图的时序信息.Guo等^[131]将 PointNet 中的激活函数换成 LIF 神经元以构建脉冲 PointNet,并通过单步训练和多步推理以降低训练开销,同时通过训练时随机初始化膜电位的方式来降低和多步推理时的性能差异.总体而言,将 SNN 推广至其他数据类型的网络结构,通常能带来理论上的激活值存储开销和推理能耗的降低,且脉冲神经元的特性使得网络具备一定时序信息提取能力,而主要挑战则包括梯

表 2 深度 SNN 中的批量标准化方法及变体

方法	$t=0$	$t=1$...	$t=T-1$	统计量更新
BN	$\mu[0], \sigma^2[0]$	$\mu[1], \sigma^2[1]$		$\mu[T-1], \sigma^2[T-1]$	$\mu_{k+T} = (1-\rho)^T \mu_k + \sum_{t=0}^{T-1} (1-\rho)^{T-1-t} \rho \mu[t]$ $\sigma^2_{k+T} = (1-\rho)^T \sigma^2_k + \sum_{t=0}^{T-1} (1-\rho)^{T-1-t} \rho \sigma^2[t]$
	γ, β				
TDBN	μ, σ^2				$\mu_{k+1} = (1-\rho)\mu_k + \rho\mu$ $\sigma^2_{k+1} = (1-\rho)\sigma^2_k + \rho\sigma^2$
	γ, β				
BNTT	$\mu[0], \sigma^2[0]$	$\mu[1], \sigma^2[1]$		$\mu[T-1], \sigma^2[T-1]$	$\mu_{k+1}[t] = (1-\rho)\mu_k[t] + \rho\mu[t], t=0,1,\dots,T-1$ $\sigma^2_{k+1}[t] = (1-\rho)\sigma^2_k[t] + \rho\sigma^2[t], t=0,1,\dots,T-1$
	$\beta[0], \gamma[0]$	$\beta[1], \gamma[1]$		$\beta[T-1], \gamma[T-1]$	
TEBN	μ, σ^2				$\mu_{k+1} = (1-\rho)\mu_k + \rho\mu$ $\sigma^2_{k+1} = (1-\rho)\sigma^2_k + \rho\sigma^2$
	$\gamma p[0], \beta p[0]$	$\gamma p[1], \beta p[1]$		$\gamma p[T-1], \beta p[T-1]$	

度跨越多个时间步的衰减、BPTT 带来的较大的训练开销, 以及诸如图卷积、点云采样等操作难以在现有神经形态计算芯片上实现等问题。

3.5 正则化方法

正则化方法已经在神经网络优化过程中大量使用, 其中批量标准化(Batch Normalization, BN)^[54]是 SNN 中最为广泛使用的方式. 相较于层标准化(Layer Normalization, LN)^[55]等其他的正则化方法, BN 层常用于卷积层之后, 并且可以在推理阶段与卷积层融合, 无需额外的资源进行实现, 因此在 SNN 中备受青睐. 除 ANN 中已有的正则化方法外, 一些专用于 SNN 的正则化方法也被提出, 其中多数方法基于 BN 进行改进, 少数则是针对脉冲神经元的特性设计, 它们进一步提升了网络的训练效果。

NeuNorm^[132]专用于脉冲卷积层, 作用于脉冲神经元的输出, 对于每层神经元, 额外记录每个位置 (i, j) 在所有通道的脉冲发放次数之和, 并随时间步进行移动平均来持续更新:

$$O_{norm}[t][i][j] = k_{decay} \cdot O_{norm}[t-1][i][j] + \frac{1-k_{decay}}{C^2} \cdot \sum_{c=0}^{C-1} O[t][c][i][j], \quad (39)$$

其中 k_{decay} 是衰减因子, C 是通道数, $O[t][c][i][j]$ 是 c 通道位置为 (i, j) 处的神经元在 t 时刻的输出, 而 $O_{norm}[t][i][j]$ 则是 NeuNorm 正则化项, 该层传递给下一层的输出会减去该正则化项. 该方法具备一定

的生物可解释性, 与视网膜细胞中特定位置的响应会被临近细胞进行正则化的行为类似^[133, 134]. NeuNorm 在效果可能来源于两方面, 一是对脉冲进行了时间上指数移动平均, 使得当前时刻的输出更为平滑, 避免了单个时刻过低或过高的发放率造成的扰动; 二是其使得脉冲神经元层的输出是浮点值形式发放率, 相较纯二值脉冲携带了更多信息. 但 NeuNorm 无法融合进突触层, 使用后会引入浮点操作, 可能是这一原因导致其逐渐被 BN 类方法取代。

在具有时间维度的 SNN 中直接使用静态 ANN 中普通的 BN 层可能会引发问题, 研究者们对此进行了探究并提出了多种改进的 BN 变体. 表 2 对目前深度 SNN 中的 BN 类方法进行了总结和对比。

普通的 BN 在 SNN 中使用, 其训练时会在每个时间步都计算当前 t 时刻输入的均值 $\mu[t]$ 和方差 $\sigma^2[t]$ 并进行标准化; 而在推理时则是利用训练时的统计量来对推理输入标准化. 需要注意的是, BN 在训练时每次前向传播后都会按照动量的方式来更新均值和方差统计量. 记在本次训练前均值和方差统计量分别为 μ_k, σ^2_k , 其中下标 k 表示统计量更新次数, 则经过本次训练后, BN 实际上进行了 T 次统计量的动量更新并得到 $\mu_{k+T}, \sigma^2_{k+T}$, 展示在表 2

中, BN 层通常还设置可学习的仿射变换, 其权重和偏置项分别是 β, γ , 由梯度下降更新。

原始的 BN 这种随着时间步来动量更新统计量的方式可能并不准确, 阈值依赖的 BN(Threshold Dependent Batch Normalization, TDBN)^[102]解决了这一问题, 其将输入在时间维度上进行融合, 直接计算整个序列的均值和方差, 因而处理完一个序列后, BN 的统计量只会动量更新一次, 而不是按照原始 BN 的方式更新 T 次。TDBN 还根据后续神经元的阈值对标准化后的输出做相应的线性缩放, 以此抵消 SNN 中特有的阈值给权重的尺度带来的影响。考虑到 SNN 中不同时刻的数据分布可能并不相同, 因而通过时间批量标准化 (Batch Normalization Through Time, BNTT)^[135]在每个时间步都使用一个独立的 BN 层, 即均值、方差、统计量、仿射变换都是每个时间步一套单独的参数。时域有效批量标准化 (Temporal Effective Batch Normalization, TEBN)^[136]的思想则是介于 TDBN 和 BNTT 之间, 其统计整个输入序列的均值和方差, 但对每个时刻又设置单独的可学习仿射变换。为减少参数量, TEBN 中不同时间步的仿射变换是使用类似于广播机制的方式生成的, 其权重和偏置项 γ, β 只有一套, 而每个时间步在使用时则是由可学习参数 $p[t]$ 与 γ, β 相乘来生成 t 时刻的仿射变换参数。需要指出的是, BNTT 和 TEBN 均含有逐时刻的参数, 暗含输入序列长度固定不可变的要求, 这与 SNN 中参数时域复用的特性不符, 将使得网络不能直接处理变长序列。

SNN 中的正则化层通常被用于卷积层后、神经元前, 对脉冲神经元的输入电流进行正则化, 但也有例外, 例如 Guo 等^[137]对神经元每一步的膜电位也进行批量标准化并取得了性能提升。

正则化方法除使用正则化层外, 还包括使用正则化损失和数据增强等。Guo 等^[138]将神经元释放脉冲的过程视作信息的量化, 将神经元膜电位与输出脉冲的均方误差作为网络损失的一部分, 以此减少量化误差; Deng 等^[139]使用每个时间步的输出与目标做交叉熵, 然后在不同时间步上进行平均, 以此替换传统的先平均每个时间步的输出再做交叉熵的损失, 对神经形态数据分类等时域任务有较大的性能提升。数据增强方法通常在训练集样本上施加诸如亮度、尺寸等变换, 以提升网络的泛化能力。ANN 领域用于静态图片上的数据增强方法已经比较成熟, 而 Li 等^[140]则对神经形态数据增强进行

了探索, 表明剪切、旋转、平移等几何变换适用于神经形态数据; 而亮度、色彩、锐度、均衡化等色彩变换会改变事件极性、破坏事件流的稀疏性, 不适合使用; 其研究结论可以作为神经形态数据增强的经验性准则。

3.6 ANN辅助训练算法

一些研究者另辟蹊径, 通过 ANN 辅助的方式来训练高性能的 SNN, 主要分为两类方法: 基于共享权重训练的方法和基于蒸馏的 SNN 训练方法。

基于共享权重的训练中, Wu 等^[141]和 Kheradpisheh 等^[142]设计了共享相同权重的 SNN 网络和 ANN 网络, 以 SNN 的输出在时间上的累计近似 ANN 的激活值, 通过 ANN 的反向传播来获取参数梯度并更新共享权重。

具体而言, Wu 等^[141]提出一种串联学习框架, 该框架包括一个 SNN 和一个通过权重共享耦合的 ANN。图 7(a)展示了该串联学习框架, 在前向传播时 SNN 结构利用前一层输出的脉冲序列 S^{l-1} 计算当前层的输出脉冲序列 S^l 和脉冲数 $c^l = \sum_{t=0}^{T-1} S^l[t]$, 而

ANN 则利用前一层的脉冲数 c^{l-1} 计算当前层的激活值 a^l 来近似脉冲数。在反向传播时, 使用 ANN 的激活值 a^l 的梯度代替脉冲数 c^l 的梯度, 通过 ANN 的反向传播计算前一层激活值和权重的梯度:

$$\frac{\partial \mathcal{L}}{\partial a^{l-1}} \approx \frac{\partial \mathcal{L}}{\partial c^{l-1}} = \frac{\partial \mathcal{L}}{\partial a^l} \cdot \frac{\partial a^l}{\partial c^{l-1}}, \quad (40)$$

$$\frac{\partial \mathcal{L}}{\partial W^{l-1}} = \frac{\partial \mathcal{L}}{\partial a^{l-1}} \cdot \frac{\partial a^{l-1}}{\partial W^{l-1}}, \quad (41)$$

其中 \mathcal{L} 是模型的损失, W^{l-1} 是第 $l-1$ 层的权重。该方法在 ANN 中计算 SNN 输出的误差, 使用 ANN 的梯度代替 SNN 的梯度更新权重, 避开了脉冲释放过程不可导的问题, 且不需要在每个时间步都进行复杂的梯度计算。

Kheradpisheh 等^[142]设计了一对由 IF 神经元组成的 SNN 网络和由 ReLU 激活函数组成的 ANN 网络, 两个网络共享权重。该网络利用 IF 神经元输出的频率来近似 ReLU 神经元的输出, 用 SNN 的输出近似 ANN 的输出。不同于 Wu 等^[141]在前向传播时将 SNN 脉冲数作为 ANN 层的输入, Kheradpisheh 等^[142]在前向传播时分别运行 SNN 和 ANN, 在反向传播时, 该方法不是直接计算 ANN 的真实梯度, 而是将 ANN 输出替换为 SNN 输出, 从而在 ANN 中计算 SNN 的近似梯度:

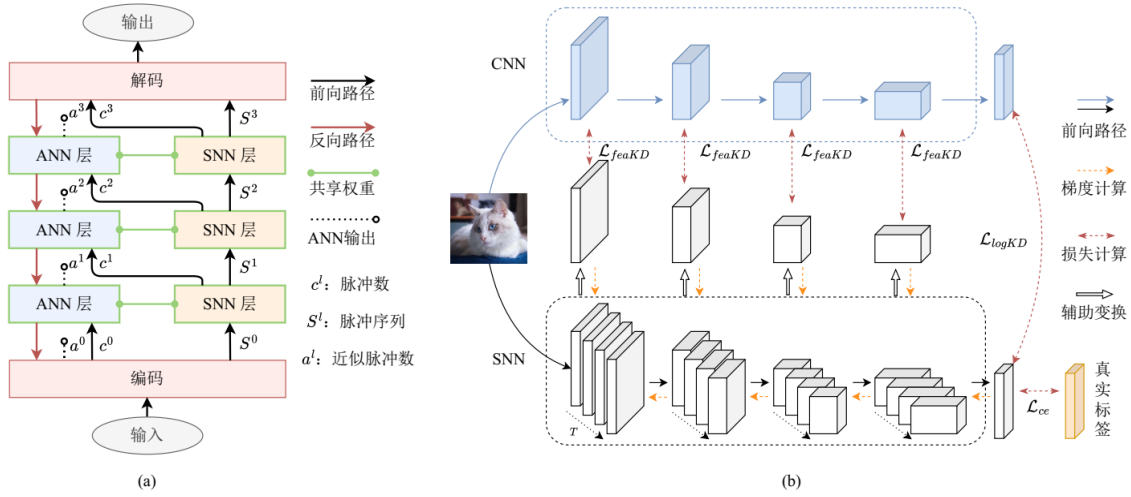


图7 两类 ANN 辅助训练方法 (a)共享权重法 (b)蒸馏法

$$\mathcal{L} = -\sum_k Y_k \ln(O_k^A) \approx -\sum_k Y_k \ln(O_k^S), \quad (42)$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial W_{ji}^l} &= \sum_k \frac{\partial \mathcal{L}}{\partial O_k^A} \sum_d \frac{\partial O_k^A}{\partial y_d^L} \frac{\partial y_d^L}{\partial W_{ji}^l} \\ &\approx \sum_k \frac{\partial \mathcal{L}}{\partial O_k^S} \sum_d \frac{\partial O_k^S}{\partial y_d^L} \frac{\partial y_d^L}{\partial W_{ji}^l}, \end{aligned} \quad (43)$$

其中, \mathcal{L} 是网络的损失函数, Y_k 是第 k 类的目标值, 如果样本为第 k 类, 则 Y_k 为 1, 否则为 0; y_d^L 是代理 ANN 网络最后一层第 d 个神经元的输出, O_k^A 、 O_k^S 是代理 ANN 网络和 SNN 网络的在第 k 个类别的输出, W_{ij}^l 是第 l 层的权重. 该方法用 SNN 的输出 O_k^S 替换 ANN 的输出 O_k^A , 从而在 ANN 模型中反向传播计算 SNN 模型的误差.

基于蒸馏的 SNN 训练中, Xu 等^[143]和 Qiu 等^[144]利用知识蒸馏方法, SNN 模型作为学生从教师 ANN 模型中学习, 该方法可以在很短的时间步长上有效地构建深层 SNN 网络.

Xu 等^[143]提出了基于响应的知识蒸馏和基于特征提取的知识蒸馏两种方法. 基于响应的知识蒸馏只从教师 ANN 模型的最后一层的输出中提取知识, 其损失函数包含 SNN 输出 Q_s 与真实标签 y_{true} 以及蒸馏标签 Q_T 的交叉熵损失:

$$\begin{aligned} \mathcal{L}_{KD} &= \alpha \tau^2 * \text{CrossEntropy}(Q_s^r, Q_T^r) \\ &+ (1 - \alpha) * \text{CrossEntropy}(Q_s, y_{true}), \end{aligned} \quad (44)$$

其中 τ 是用于平滑概率分布的温度参数, Q_s^r 和 Q_T^r 是利用模型最后一层第 i 个神经元的输出 Z_i 来计算的, 其中第 i 个元素 q_i 的计算公式为

$q_i = \text{Softmax}(Z_i / T)$, α 用于权衡两种损失的重要程度. 基于特征提取的知识蒸馏从教师 ANN 模型的中间层提取隐藏知识, 其损失函数包含学生 SNN 的输出与真实标签的损失 \mathcal{L}_{task} 以及中间层特征的 L2 距离损失 $\mathcal{L}_{distill}$:

$$\mathcal{L}_{KD} = \mathcal{L}_{task} + \alpha \cdot \mathcal{L}_{distill}, \quad (45)$$

$$\mathcal{L}_{distill} = \sum_i (T_i - S_i)^2, \quad (46)$$

其中 T_i 是经过边缘 ReLU 处理以抑制负信息影响后的教师 ANN 模型的中间层特征, S_i 是经过 1×1 卷积层匹配通道大小后的学生 SNN 的中间层特征.

Qiu 等^[144]通过神经网络结构搜索实验表明, 与更大规模、更高性能的教师模型相比, 具有相同架构的教师 ANN 模型在训练学生 SNN 模型时效果更好. 基于这一发现, 其提出了一个自架构知识蒸馏框架, 如图 7(b)所示, 该框架将教师 ANN 模型的知识转移到具有相同体系结构的学生 SNN 网络中. 该网络的总损失函数 \mathcal{L}_{all} 包含以下三部分: 传统的交叉熵损失 \mathcal{L}_{ce} 、让学生模型模仿教师模型特征图的特征蒸馏损失 \mathcal{L}_{feaKD} 以及约束学生模型的输出分布接近教师模型的输出分布的 logits 蒸馏损失 \mathcal{L}_{logKD} :

$$\mathcal{L}_{all} = \alpha * \mathcal{L}_{ce} + \beta * \mathcal{L}_{feaKD} + \gamma * \mathcal{L}_{logKD}, \quad (47)$$

$$\mathcal{F}_s = T_s(\mathcal{F}_s) = \text{BN}(\text{Conv}(\frac{1}{T} \sum \mathcal{F}_s)), \quad (48)$$

$$\mathcal{F}_t = T_t(\mathcal{F}_t) = \mathcal{F}_t, \quad (49)$$

$$\mathcal{L}_{feaKD} = \|\mathcal{F}_s - \mathcal{F}_t\|^2, \quad (50)$$

$$\mathcal{L}_{logKD} = \tau^2 \sum p_\tau^t \log\left(\frac{p_\tau^t}{p_\tau^s}\right), \quad (51)$$

$$p_\tau^s(i) = \frac{\exp(p^s(i)/\tau)}{\sum \exp(p^s/\tau)}, \quad (52)$$

其中 α 、 β 和 γ 是控制不同损失权重的超参数, \mathcal{F}_s 和 \mathcal{F}_t 分别表示学生 SNN 模型和教师 ANN 模型的中间层特征, BN 和 Conv 分别表示批量正则化层和卷积层, T_s 和 T_t 表示 SNN 和 ANN 模型的特征变换, T 表示时间步数, p_τ^t 和 p_τ^s 分别表示 ANN 和 SNN 的预测分布, τ 是平滑参数. 卷积层将 SNN 的特征映射到连续空间, 以解决特征维度不匹配的问题, 从而允许学生 SNN 模型模仿教师 ANN 模型的中间层输出特征.

共享权重类方法直接避开了 SNN 计算代价高、训练耗时长、内存消耗大的反向传播流程, 但由于 ANN 和 SNN 本身的差异, 共享权重和不精确的梯度会导致训练出的 SNN 性能较其耦合的 ANN 有较大程度下降, 因而这一类方法并未被广泛使用. 基于蒸馏类的 SNN 训练方法通常需要额外引入 ANN 的输出以计算损失, 训练代价比普通的梯度替代法更高, 但由于 ANN 的指导作用, 训练出的网络性能也强于只使用数据集中目标值计算损失的普通 SNN. 两类方法均需要 ANN 的辅助, 而 ANN 不具有时间维度, 无法处理时域任务, 这一缺陷使得 ANN 辅助类算法的应用范围非常受限.

3.7 事件驱动学习算法

事件驱动学习方法使用网络发放的脉冲传递梯度信息, 其反向传播也是稀疏的, 而普通方法则使用稠密的反向传播. 事件驱动方法中梯度表示脉冲发放时刻的改变量, 而普通方法的梯度则表示脉冲的取值应该增加还是减少.

在事件驱动学习方法中, 梯度在相邻层之间的传播一般从神经元的输出脉冲传递到释放脉冲时的膜电位, 再从该膜电位分别向输入脉冲和对应的突触连接权重传递. Zhang 等^[145]在事件驱动学习的基础上, 进一步考虑了脉冲响应模型 (Spike Response Model, SRM) 神经元中重置核导致的多个脉冲之间的相互作用, 从而推导出更为细致的反向传播公式. Zhu 等^[146]基于 SRM 神经元, 推导出了事

件驱动学习方法在含有神经元的网络层反向传播中具有梯度之和不变性:

$$\sum_j \sum_{t_m(s_j^{(l-1)})} \frac{\partial \mathcal{L}}{\partial t_m(s_j^{(l-1)})} = \sum_i \sum_{t_k(s_i^{(l)})} \frac{\partial \mathcal{L}}{\partial t_k(s_i^{(l)})}, \quad (53)$$

其中等式左边是第 $l-1$ 层所有脉冲携带的梯度之和, 其中 j 和 $t_m(s_j^{(l-1)})$ 分别对应第 $l-1$ 层的单个神

经元和单个脉冲, 等式右边是第 l 层所有脉冲携带的梯度之和. 该工作进一步分析了不含神经元的池化层, 改进了平均池化层使其满足梯度之和不变性. 在此基础上, Zhu 等^[147]进一步探究了损失函数对时序的事件驱动学习方法的影响. 该研究发现, 基于频率的损失函数同样适用于时序的事件驱动学习方案, 并针对先前损失函数在目标类别输出神经元上梯度之和与脉冲发放数量差异不成正比的问题, 提出了改善型计数损失. 此外, 该工作还将权重归一化中所使用的比例因子的训练转移至阈值, 提升了网络的性能.

目前事件驱动的学习算法研究还处于起步阶段, 性能远低于传统算法, 但其稀疏的反向传播在理论上能够部署于事件驱动的神形态计算芯片, 使得 SNN 的片上训练成为可能, 前景广阔.

3.8 在线学习算法

在线学习方法为 SNN 这种需要多个时间步进行学习和推理的模型提供了一种实时更新权重的策略. 这种学习方式避免了通过时间反向传播 (Back Propagation Through Time, BPTT) 需要存储大量中间状态的需求. 在线学习方法的内存消耗量通常为 $\mathcal{O}(1)$, 而 BPTT 则是 $\mathcal{O}(T)$. 因此, 在线学习适用于资源受限或时间步数较多的场景.

Deep Continuous Local Learning (DECOLLE)^[148]是最早的深度 SNN 在线学习方法之一, 其针对双指数脉冲响应神经元, 通过在每层的输出脉冲后引入一个读取层获取局部损失, 实现了学习规则在时间和空间上的局部化. 其后的在线学习算法则是对 BPTT 的梯度进行单步近似, 相较于仅使用局部信息的 DECOLLE 性能更好. 不失一般性, 对于使用 LIF 神经元的多层 SNN, 基于 BPTT 的第 l 层的权重 W^l 上的梯度为:

$$\frac{dL}{dW^l} = \sum_{t=0}^{T-1} \frac{dL}{dV^{l+1}[t]} \frac{\partial V^{l+1}[t]}{\partial S^l[t]} \frac{\partial S^l[t]}{\partial V^l[t]} \frac{dV^l[t]}{dW^l}$$

$$= \sum_{t=0}^{T-1} \frac{dL}{d\mathbf{V}^{l+1}[t]} \frac{\partial \mathbf{V}^{l+1}[t]}{\partial \mathbf{S}^l[t]} \frac{\partial \mathbf{S}^l[t]}{\partial \mathbf{V}^l[t]} \cdot \sum_{j=0}^t \left[\prod_{i \in [j+1, t], i \in \mathbb{N}} \left(\frac{\partial \mathbf{V}^l[i]}{\partial \mathbf{V}^l[i-1]} + \frac{\partial \mathbf{V}^l[i]}{\partial \mathbf{S}^l[i-1]} \frac{\partial \mathbf{S}^l[i-1]}{\partial \mathbf{V}^l[i-1]} \right) \right] \frac{\partial \mathbf{V}^l[j]}{\partial \mathbf{W}^l}. \quad (54)$$

在线学习方法通常希望在网络运行到当前时间步 t 时,就能近似计算出式(54)的最左侧求和符号内的第 t 项. Online Training Through Time (OTTT)^[149] 的主要改动为: 忽略重置过程的梯度, 使 $\frac{\partial \mathbf{V}^l[i]}{\partial \mathbf{S}^l[i-1]} = 0$; 使用软重置, 使得 $\frac{\partial \mathbf{V}^l[i]}{\partial \mathbf{V}^l[i-1]} = \lambda$ 为常数; 令累乘项中的 $\frac{\partial \mathbf{V}^l[i]}{\partial \mathbf{S}^l[i-1]} = 0$. 在上述改动下, 展开来自后续层的梯度, 式(54)简化为:

$$\frac{dL}{d\mathbf{W}^l} = \sum_{t=0}^{T-1} \left[\left(\frac{\partial L}{\partial \mathbf{S}^{l_{\max}}[t]} \frac{\partial \mathbf{S}^{l_{\max}}[t]}{\partial \mathbf{V}^{l_{\max}}[t]} \prod_{j=l+1}^{l_{\max}-1} \frac{\partial \mathbf{V}^{j+1}[t]}{\partial \mathbf{S}^j[t]} \frac{\partial \mathbf{S}^j[t]}{\partial \mathbf{V}^j[t]} \right) \right]^T \cdot \left(\sum_{\tau \leq t, \tau \in \mathbb{N}} \lambda^{t-\tau} \mathbf{S}^{l-1}[\tau] \right)^T. \quad (55)$$

定义资格迹变量:

$$\mathbf{E}^{l-1}[t] = \lambda \mathbf{E}^{l-1}[t-1] + \mathbf{S}^{l-1}[t], \quad (56)$$

其中第 0 层初始化为 $\mathbf{E}^{l-1}[0] = \mathbf{S}^{l-1}[0]$. 式(55)中的

$\sum_{\tau \leq t, \tau \in \mathbb{N}} \lambda^{t-\tau} \mathbf{S}^{l-1}[\tau]$ 即可转换为 $\mathbf{E}^{l-1}[t]^T$, 此时该式的求和项 t 只需要 t 时刻的信息. 该工作还从理论上论证了其梯度与基于脉冲表征的 Differentiation on Spike Representation 方法^[150]之间的正相关性.

Spatial Learning Through Time (SLTT)^[151] 延续了 OTTT 忽略重置梯度和使用软重置的设置, 并完全忽略膜电位在时刻之间的梯度, 即令 $\frac{\partial \mathbf{V}^l[i]}{\partial \mathbf{V}^l[i-1]} = 0$. 在上述设置下, 式(54)简化为:

$$\frac{dL}{d\mathbf{W}^l} = \sum_{t=0}^{T-1} \left[\left(\frac{\partial L}{\partial \mathbf{S}^{l_{\max}}[t]} \frac{\partial \mathbf{S}^{l_{\max}}[t]}{\partial \mathbf{V}^{l_{\max}}[t]} \prod_{j=l+1}^{l_{\max}-1} \frac{\partial \mathbf{V}^{j+1}[t]}{\partial \mathbf{S}^j[t]} \frac{\partial \mathbf{S}^j[t]}{\partial \mathbf{V}^j[t]} \right) \right]^T \cdot \left[\frac{\partial \mathbf{V}^{l+1}[t]}{\partial \mathbf{S}^l[t]} \frac{\partial \mathbf{S}^l[t]}{\partial \mathbf{V}^l[t]} \right] \mathbf{S}^{l-1}[t]^T. \quad (57)$$

因而 SLTT 只需要保存当前时间步的脉冲 $\mathbf{S}^{l-1}[t]$ 即可实现在线学习.

在 OTTT 的基础上, Neuronal Dynamics-based Online Training (NDOT)^[152] 对层内的时间依赖性进行了更细致的建模, 不像式(55)中简单的使用膜电位的衰减 λ , 而是将重置过程也纳入:

$$\mathbf{E}^{l-1}[t] = \mathbf{E}^{l-1}[t-1] \cdot \frac{U^l[t] - V_{th} \mathbf{S}^l[t]}{U^l[t-1] - V_{th} \mathbf{S}^l[t-1]} + \mathbf{S}^{l-1}[t]. \quad (58)$$

Zhu 等^[153] 则考虑在 SNN 在线学习中加入归一化机制. 由于在线学习过程中无法使用未来信息, 而直接在每一步进行 BN 存在协方差漂移问题, 该工作提出了包含 BN 和线性变换的 Online Spiking Renormalization (OSR) 模块以保证训练和推理时归一化变换参数的一致性, 还引入了在线阈值稳定器以稳定时间步之间的神经元发放率. OSR 模块训练时的过程如下:

$$\hat{I}[t] = \frac{I[t] - \mu[t]}{\sqrt{\sigma^2[t] + \epsilon}}, \quad (59)$$

$$\tilde{I}[t] = \gamma \cdot \left(\hat{I}[t] \cdot \text{NoGrad} \left(\frac{\sqrt{\sigma^2[t] + \epsilon}}{\sqrt{\sigma^2 + \epsilon}} \right) + \text{NoGrad} \left(\frac{\mu[t] - \hat{\mu}}{\sqrt{\sigma^2 + \epsilon}} \right) \right) + \beta, \quad (60)$$

在第 t 个时间步中, $I[t]$ 是未经变换的输入电流, $\mu[t]$ 和 $\sigma^2[t]$ 分别是 $I[t]$ 的均值和方差, $\hat{\mu}$ 、 σ^2 分别是 BN 层内记录的均值和方差统计量, $\hat{I}[t]$ 是 BN 变换后的值, $\tilde{I}[t]$ 是二次线性变换之后的值, NoGrad(...) 内的运算不参与反向传播. 在推理时, OSR 的行为则和 BN 完全一致.

以往的在线学习方法偏向理论研究, 而 Hu 等^[92] 则是从工程实践角度提出解决方法, 其通过实验发现常规 BPTT 训练中只有最后一层的时序信息对训练所得权重影响大, 于是在前向传播中断开了除最后一层外的时序传播过程. 这一操作简单易行, 且实验效果良好. 对于保留了前向时序传播的最后一层, 该工作使用可逆模块解决了使用 $\mathcal{O}(1)$ 存储空间记录 T 步信息的问题, 其关键是推导出前一时间步膜电位用后一时间步信息来表示的方法. 此外该工作在网络中使用了 ConvNeXt 块^[154], 并将前一时间步的高层信息融合到了当前时间步的低层信息中以提升网络表现.

未来信息不可在当下获取, 故已有的在线学习

方法都假设 $\frac{d\mathcal{L}}{dS'[t]} \approx \frac{\partial \mathcal{L}}{\partial S'[t]}$, 这一近似相当于认为当前时刻的输出脉冲, 对未来时刻的损失不会产生影响, 其实并不符合实际情况. 这一特性也使得在线学习方法难以处理时序任务.

3.9 训练加速方法

相较于 ANN, SNN 额外增加了时间维度, 在不使用在线学习方法、默认使用 BPTT 方法训练的情况下, 网络的训练耗时和内存消耗通常和总时间步 T 近似成正比, 带来了显著高于 ANN 的训练开销, 如何对 SNN 训练加速成为研究者们日益关心的话题. GPU 拥有强大的并行计算能力, 是训练 SNN 的首选设备, 目前已有的 SNN 训练加速方法都基于 GPU 和 SNN 的特性进行设计.

稀疏脉冲梯度下降^[155]在反向传播时, 将满足 $|H[t] - V_{th}| \geq B_{th}$ 的神经元视作不活跃的神经元, 并

将其脉冲释放过程的梯度 $\frac{\partial S[t]}{\partial H[t]}$ 视作 0, 从而使得

本应稠密的反向传播的计算图变得稀疏, 然后使用 PyTorch 中自带的稀疏计算库进行加速. 稀疏脉冲梯度下降方法相较于普通的梯度下降方法, 在 GPU 上最高可达 150 倍的训练反向传播加速和 85% 的内存消耗减少, 但其只在简单的全连接 SNN 上进行了实现和验证. SpikingJelly 框架^[49]提供了更为通用的深度 SNN 加速方法. SpikingJelly 框架首先定义了 SNN 传播模式的概念, 并提出逐步传播和逐层传播这两种计算图的构建方式. 在逐层传播模式下, 网络中的每层可以同时接收到尺寸为 $(T \times N \times \dots)$ 的整个序列作为输入, 其中 T 是序列长度, N 是批量大小. 对于无状态的卷积、全连接等突触层, SpikingJelly 框架提供了包装器, 将输入的时间和批量维度融合, 即将输入尺寸变换到 $(TN \times \dots)$, 然后再送入无状态层计算, 计算得到的结果再重新拆成序列, 恢复到尺寸为 $(T \times N \times \dots)$ 的序列. 由于时间维度被当作了批量维度, 不同时间步的计算也是并行的, 速度远快于传统的通过循环实现的逐步计算. 对于有状态的神经元等层, SpikingJelly 框架使用自定义的 CUDA 后端, 将神经元遍历所有时间步的迭代计算封装到单个 CUDA 内核, 相较于 PyTorch 实现的神经元在计算时调用多个小 CUDA 内核, 单个大 CUDA 内核的调度开销更小、计算速度更快, 在 T 较大时能有数十倍加速效果. 综合使用无状态层和有状态层的加速方法, SpikingJelly 框架相较于其他

SNN 框架实现的 SNN 仿真方式, 最高可达 11 倍的训练加速效果.

Luke 等^[156]提出了一种加速脉冲神经元的时间分组仿真方式, 在仿真脉冲神经元时, 将时间步分组, 每组时间步内忽略神经元的重置过程, 从而使得膜电位的计算从迭代计算改为直接求解, 并将膜电位与阈值比较, 输出脉冲, 这一思路与 PSN^[70]类似; 由于前述过程忽略了重置, 会导致输出脉冲数量多于有重置的正常神经元仿真方式, PSN 没有采取任何处理措施, 而该方法则对输出脉冲进行修正, 仅保留每组时间步内的第一个脉冲, 一定程度上缓解了 PSN 去掉重置可能导致的发放率升高问题. 该方法相较于正常仿真过程, 性能有所降低, 但仿真速度大幅度提升.

4 综合对比实验

此前尚未有工作将不同类别的方法进行统一的比较, 因而本章选取了各类学习算法中的代表性方法, 在相同的设置下进行实验, 参与比较的方法包括 SpikingJelly 框架^[49]中基于 CuPy 后端加速的 IF 神经元和 LIF 神经元作为基准(Baseline)、神经元和突触改进算法中的 CLIF 神经元^[95]和 PSN 家族^[70]、正则化方法中的 TEBN^[136]、在线学习算法中的 OSR^[153]、训练加速算法中的时间分组仿真方式加速的 BlockALIF 神经元^[156]、ANN 辅助训练算法中的 Tandem 学习方法^[141]和响应与特征蒸馏^[143]. 这些方法都提供了开源代码, 可以直接使用. 实验任务包括静态 CIFAR10 和序列 CIFAR10 的分类、神经形态的 SHD 语音数据集^[67]分类、神经形态的 Gen1^[157]数据集的目标检测. 需要指出的是, Tandem 学习方法^[141]和响应与特征蒸馏^[143]依赖于 ANN, 因而不适用于序列 CIFAR10 分类和神经形态数据集相关任务, 仅在静态 CIFAR10 图片分类任务上进行实验. 需要注意的是, 本次实验中并没有纳入网络结构改进类方法, 因为这些方法已经在复杂的 ImageNet 数据集上进行了公平的性能比较, 结果如前文图 6 所示. 本章还选取了代表性的训练加速方法, 包括 SpikingJelly 框架中融合内核实现的 LIF 神经元^[49]、并行脉冲神经元 PSN^[70]和时间分组仿真方式加速的 BlockALIF 神经元^[156]并对比了他们的加速效果. 关于本章中各个实验的详细超参数, 请参见附录.

4.1 CIFAR 图片分类任务性能

本文使用 Fang 等^[70]的网络结构, 测试各类方法



图 8 CIFAR 图片分类性能对比

分类静态 CIFAR10 和序列 CIFAR10 的任务性能, 以此检验各类方法的静态数据集分类性能和长期依赖学习能力.除 CLIF 神经元和 PSN 家族的网络外, ANN 辅助训练类算法的网络中使用 IF 神经元, 其它网络均使用 LIF 神经元. BlockALIF 神经元均使用每组 2 个时间步, 因为如果每组 1 个时间步与普通神经元无异, 则没有任何加速效果; 如果每组更多时间步, 则实验发现其分类性能剧烈下降.对于 PSN 家族的网络, CIFAR10 分类任务使用 PSN, 而序列 CIFAR10 分类使用 $k=4$ 的 Sliding PSN. 图 8 全面对比了各类方法的性能.

图 8(a)展示了各类方法的分类正确率.对于静态的 CIFAR10 分类, 神经动态中不带衰减的 IF 神经元表现强于 LIF 神经元, 而序列 CIFAR10 分类则是神经动态更为复杂的 LIF 神经元性能更好.在静态 CIFAR10 分类任务上, PSN 性能略高于 CLIF 神经元, 均强于 IF 神经元; 而在序列 CIFAR10 分类任

务上, CLIF 神经元相较于 LIF 神经元提升明显, 而 Sliding PSN 性能又大幅度超越 CLIF 神经元, 表明 PSN 家族通过直接权重连接替换马尔科夫链, 极大增强了长期依赖学习能力, 而 CLIF 神经元增加补充电位的神经动态也有利于缓解梯度随时间的衰减. TEBN 在两种任务上都相较于普通网络提升显著, 表明使用全部时刻的统计量和逐时刻的仿射变换, 相较于普通的 BN 有效捕捉到了输入的分布并提升了拟合能力. OSR 作为在线学习方法, 在 CIFAR10 分类任务上性能反而强于普通网络, 但在序列 CIFAR10 分类任务上性能大幅度下降, 表明静态任务中未来的梯度或可忽略, 而时域任务中未来的梯度则至关重要. BlockALIF 神经元性能较差, 而且在 $T=32$ 的序列 CIFAR10 分类任务上性能下降更严重, 表明时间上的分组限制了脉冲发放次数, 对性能有着很大的负面影响. Tandem 学习方法由于使用不精确的梯度, 性能弱于基于替代函数训练的 IF 神经元. 蒸馏方法相较于原始的使用 IF 神经元的

表 3 对比各类代表性方法任务神经形态数据集处理任务性能

评估指标\方法	IF	LIF	CLIF	Sliding PSN	TEBN	OSR	BlockALIF
SHD 分类正确率(%)	78.33	66.98	66.1	69.23	74.54	40.93	82.36
SHD 分类突触操作数(M)	0.0251	0.0262	0.0263	0.0264	0.0262	0.0228	0.0239
NCAR 预训练分类正确率(%)	81.00	91.69	85.34	91.41	90.96	56.23	68.14
Gen1 目标检测 mAP@0.5	0.0013	0.4399	0.2388	0.4781	0.5213	0.0007	0.1347
Gen1 目标检测 mAP@0.5:0.95	0.0002	0.2106	0.090204	0.2391	0.2656	0.0003	0.0457
Gen1 目标检测突触操作数(M)	385.86	296.33	353.03	293.53	214.72	305.93	232.69

网络,性能均有一定提升,其中特征蒸馏提升稍高,且均高于 Tandem 学习方法,表明来自 ANN 的知识帮助较大。

图 8(b)展示了各类方法的训练速度,可以发现基于 CuPy 后端加速的 IF 和 LIF 神经元速度最快,体现了 SpikingJelly 框架中融合 CUDA 内核带来的巨大优势.Tandem 方法速度较慢,原因在于其需要 ANN 部分的计算,且神经元使用 PyTorch 实现.蒸馏方法中仍然使用 CuPy 后端加速的 IF 神经元,但由于 ANN 部分的计算和蒸馏损失拖累了速度,其中特征蒸馏需要更多的损失项,因而速度比响应蒸馏更慢.CLIF 神经元原作者也是使用 PyTorch 实现,且神经动态较为复杂,故速度较慢.令人意外的是,PSN 家族的速度慢于 CuPy 后端加速的 IF 和 LIF 神经元,这可能是由于该实验设置下,批量数为 128、网络通道数为 256,内存读写需求较大,PSN 家族的核心运算矩阵乘法遭遇了内存瓶颈(Memory-Bound).通过额外实验发现,设置批量大小 32、网络通道数 32 时,内存读写需求大大降低,此时 PSN 训练速度达到 2345samples/s,高于 CuPy 后端加速的 IF 神经元 1649 samples/s 的训练速度,佐证了前述猜想.TEBN 方法中使用的仍然是 CuPy 后端加速的神经元,但由于逐时刻仿射变换的额外计算而拖慢了速度.OSR 速度大幅度落后于其他方法,因其是在线学习方法,只能使用 SpikingJelly 框架中的逐步(Step-By-Step)传播,而其他方法则可以使用逐层(Layer-By-Layer)传播并通过融合时间批量维度来加速无状态层.BlockALIF 神经元的加速效果较差,主要原因在于神经动态极为复杂,且使用时间上的卷积实现并行计算,而 Fang 等^[70]在设计 Sliding PSN 的经验表明卷积并行度较低,速度远慢于矩阵乘法.关于 BlockALIF 神经元的加速效果,将在后文予以详细讨论.图 8(c)展示了各类方法的推理速度,整体和训练速度一致,不再赘述。

图 8(d)展示了各类方法在训练时的 GPU 显存消

耗.神经网络在训练时,需要保存权重和各种计算的中间变量以用于反向传播.权重的数量在网络结构确定后就固定了,因而中间变量的数量对内存消耗起决定性作用.使用 BPTT 训练的 SNN,需要保存所有时刻的中间变量,而在线学习方法只需要保存单个时间步的中间变量,因而 OSR 方法是各类方法中消耗内存最少的.Tandem 方法的反向传播基于 ANN,也消除了时间维度,故内存消耗量与在线学习方法接近.其他方法基于 BPTT 训练,其中 CLIF 神经元引入了额外的补充电位以及 Sigmoid 激活函数;BlockALIF 神经元具有复杂的神经动态;TEBN 逐时刻的仿射变换使 BN 层的计算变为原来的 T 倍;三者都引入了大量中间状态,因而带来了较大的内存消耗.蒸馏类方法一方面需要 ANN 计算,另一方面引入了额外损失,故内存消耗量比直接训练的方法更大,且特征蒸馏的损失项更多,故内存消耗量也显著高于响应蒸馏方法.相较于 IF 神经元和 LIF 神经元,PSN 家族并未引入额外计算或中间变量,因而内存消耗量与它们接近.需要注意的是,尽管 LIF 神经元相较于 IF 神经元神经动态更复杂、中间变量更多,但在 CuPy 后端实现时这些中间变量由 CUDA 内核内部处理,而不是 PyTorch 的自动微分机制,因而这些变量在 CUDA 执行完毕后就自动释放了,所以两者的内存消耗完全一致。

图 8(e)展示了各类方法的突触操作数,在网络结构确定后,该指标主要由神经元的发放率决定.PSN 家族的突触操作数显著高于作为基准的 IF 神经元和 LIF 神经元,表明其去除重置确实导致了发放率的显著升高.OSR 和 TEBN 的突触操作数也较高,需要注意的是前者尽管是在线学习方法,却在静态 CIFAR10 上取得了接近 BPTT 类方法的正确率,而后者则取得了静态 CIFAR10 的最高正确率,可能是它们在训练过程中通过提升发放率来获得了性能增益.在该任务中,作为基准的 IF 神经元和 LIF 神经元每层脉冲神经元的发放率都低于 50%,

而 Guo 等^[158]则猜想发放率越接近 50% 则信息熵越大, 网络性能越好, OSR 和 TEBN 的表现与其猜想一致. 其他方法的突触操作数则较为接近, 但也存在一定差异. 例如, LIF 神经元低于 IF 神经元, 表明膜电位的泄露行为降低了整体的发放率. Tandem 方法使用脉冲频率传递信息, 而特征蒸馏则是直接拟合 ANN 的 ReLU 的输出, 两者都暗含使用频率编码表示信息, 因而发放率和突触操作数稍高; 而响应蒸馏中, 由于蒸馏温度以及 ANN 输出的“暗知识(Dark Knowledge)”, 即输出不是理想的独热编码形式, 故其损失相较于普通交叉熵损失有所松弛, 可能是这一特性导致其突触操作数反而低于作为基准的 IF 神经元. BlockALIF 神经元在序列 CIFAR10 分类的突触操作数极低, 考虑到其正确率只有 34.62%, 表明确实是时间维度分组限制发放导致的性能差.

4.2 神经形态语音数据SHD分类性能

本文使用神经形态的 SHD 语音数据集^[67]进行实验, 并使用 Ilyass 等^[100]的开源代码和网络结构. 需要注意的是, 该实验使用固定时间间隔积分来处理输入事件, 因而输入的帧数为 88~126.

实验结果展示在表 3 中. 在正确率方面, 该数据集上各种方法的表现与 CIFAR 分类存在较大差异. 作为基准的 IF 神经元和 LIF 神经元, 前者性能高于后者, 而在常见的神经形态数据集分类任务中则通常是 LIF 神经元性能更好, 这可能暗示了语音数据和视觉数据存在较大差异. TEBN 与性能更好的 IF 神经元配合使用, 但正确率低于作为基准的 IF 神经元, 可能是其难以处理变长序列的输入所致. BlockALIF 神经元性能最好. 通过额外的实验发现, 当去除 BlockALIF 神经元的自适应的阈值后, 其性能会下降到和 IF 神经元持平, 表明自适应阈值使其较好地捕捉了音频数据中的时域信息. 其他方法都表现较差, 低于作为基准的 IF 神经元, 尤其是 OSR 方法, 可能是该任务的时间步较大, 因而在线学习方法相较于 BPTT 难以补偿时间梯度的缺失. 而在突触操作数方面, 由于网络本身规模很小且层数浅, 故各类方法的差别不大.

4.3 神经形态数据集Gen1目标检测性能

本文使用神经形态的 Gen1^[157]数据集进行实验, 该数据集由 ATIS 传感器采集的驾驶场景组成, 相较于静态图片分类任务难度更大, 极具挑战性. 本文使用 Fan 等^[159]的开源代码和网络结构, 使用 Spiking DenseNet121-16^[160]作为检测的骨干网络,

首先在神经形态的 NCAR^[161]数据集上预训练分类任务, 然后在 Gen1 数据集上进行目标检测任务训练, 实验结果展示在表 3 中. NCAR 预训练结果显示, LIF 神经元优于 IF 神经元, 也强于 CLIF 神经元、Sliding PSN、TEBN、OSR 和 BlockALIF 神经元性能都较差. 需要指出的是, 根据 Fan 等^[159]的设计, 预训练设置了早停(Early Stopping), 连续 5 轮训练的最小验证集损失不减少则退出训练. 该预训练仅为后续目标检测任务提供较优的初始参数, 因而本身的分类正确率无需过多关注. 从目标检测性能看, 性能排序为 TEBN > Sliding PSN > LIF > CLIF > BlockALIF > IF > OSR, 且从 BlockALIF 神经元开始, 后面的方法几乎无精度, 可以认为是不收敛. 对于能够正常收敛的方法, 突触操作数排序为 TEBN < BlockALIF < Sliding PSN < LIF. 整体来看, TEBN 在该检测任务中表现极好, 精度最高、突触操作数最低, 可能是其凭借逐时刻的仿射变换增强了网络对时序数据的拟合能力; Sliding PSN 和 LIF 神经元表现稳定; IF 神经元不收敛, 可能是其仅积分不作衰减的特性难以捕捉时序动态所致; CLIF 神经元和 BlockALIF 神经元性能都较差, 它们都使用了自适应的阈值更新机制, 可能表明该机制未必总对性能有益; OSR 方法不收敛, 可能是该任务的时序信息较多, 在线学习方法在缺失完整梯度的情况下难以训练参数.

4.4 加速性能测试

已有的 SNN 加速的研究集中在神经元层次, 故本文选取 PyTorch 实现的 LIF 神经元、SpikingJelly 框架中融合内核实现的 LIF 神经元^[49]、并行脉冲神经元 PSN^[70]和时间分组仿真方式加速的 BlockALIF 神经元^[156]进行实验, 对比加速性能. 实验环境为 Intel Core i9-10900X CPU, 64G 内存, Nvidia RTX 2080 Ti GPU; 神经元数量为 4096; 分别测试不同神经元在时间步数 $T = 2, 4, 8, 16, 32, 64$ 时进行训练 (前向传播、反向传播和梯度下降) 的耗时. 以 PyTorch 实现的 LIF 神经元作为速度基准, 其他神经元与 LIF 神经元的速度之比展示在了表 4 中. 实验结果显示, 随着时间步数的增大, SpikingJelly 优势明显, 最高可达接近 15 倍训练加速效果, 原因在于 T 较大时 PyTorch 实现的神经元会调用大量琐碎的 CUDA 内核, 而 SpikingJelly 融合内核后可以大幅度降低琐碎内核的调度开销; PSN 加速效果比 SpikingJelly 更胜一筹, 最高可达近 44 倍加速, 展现了并行加速相较于串行计算的巨大优势;

表 4 对比加速方法性能

T	相较于 LIF 神经元的加速比						LIF 耗时(ms)
	SpikingJelly	PSN	BlockALIF 分组大小				
			2	4	8	16	
2	1.03	2.20	0.38				1.44
4	1.48	4.07	0.44	0.41			3.02
8	2.72	6.81	0.39	0.41	0.42		4.79
16	6.19	12.60	0.40	0.38	0.40	0.38	9.48
32	16.61	17.76	0.49	0.50	0.47	0.48	17.14
64	14.83	43.75	0.56	0.59	0.59	0.60	30.60

BlockALIF 则加速效果较差, 速度反而慢于 LIF 神经元, 一方面原因可能是在实验中时间步数最大为 $T=64$, 如此之大的时间步数在深度 SNN 中很少使用, 但还是不足以大到能够弥补神经元内部使用卷积本身的调度开销, 如果时间步数达到 Luke 等^[156]测试的数千, 则 BlockALIF 神经元有可能快于 LIF 神经元, 另一方面原因在于 BlockALIF 神经元在处理没有同层反馈连接的神经元时会增加计算复杂度, 其更适用于对有同层反馈连接的神经元进行加速, 尤其是在分组较大时候, BlockALIF 神经元可以将分组内部多个仿真步的反馈连接并行计算从而提高计算效率, 从而达到 Luke 等^[156]得到的较高加速比. 总体而言, SpikingJelly 对串行神经元加速效果好, 但仍弱于并行的 PSN, 后者可能代表了未来的神经元加速方向.

4.5 性能对比测试小结

本章对不同方法进行了对比测试, 并通过多个指标进行评价. 综合多个数据集的结果来看, 尚无某种方法能够在所有任务中都取得最好性能, 但这也符合机器学习领域著名的没有免费午餐定理(no free lunch theorem)^[162]. 研究者需要根据自身需求进行取舍来决定使用哪些方法. 根据本章的实验可以得到一些初步结论. 对于静态图片相关任务, 推荐的方法包括结构简单、代码易于实现 IF 神经元和 PSN; 如果对性能有较高追求, 还可以考虑牺牲一定训练速度, 使用 CLIF 神经元、TEBN 或蒸馏类方法; 如果显存受限, 则使用 OSR 方法. 对于时序数据处理, 在线学习方法性能较差, 不适合使用. 对于神经形态数据集相关任务, 使用 LIF 神经元、Sliding PSN 或 TEBN 通常能达到较好效果, 而在前者性能不好的情况下也值得尝试 IF 神经元和 BlockALIF 神经元. 如果研究者需要加速训练过程, 则推荐使用 SpikingJelly 中基于 CuPy 实现的神经

元, 其与 PyTorch 实现的计算等价, 且加速效果提升明显. 而突触操作数, 则随着方法和具体任务变化, 尚无统一结论, 例如 Sliding PSN 在 CIFAR 分类任务上突触操作数明显高于 LIF 神经元, 但在 Gen1 目标检测任务则与之接近; TEBN 在 Gen1 目标检测任务上突触操作数显著地少, 但在其他任务上则与其他方法差距不大.

5 研究挑战与未来研究方向

梯度替代算法近年来取得了飞速发展, 成绩斐然, 但仍有部分困扰整个研究领域的难题尚待解决. 一定程度上, 目前深度 SNN 性能的提升主要来自于深度学习方法的贡献, 这一方面带来的性能的飞跃, 另一方面也则使得研究集中于 ANN 的脉冲化, 而对 SNN 独有的编码方式、神经动态、学习算法等关注不够. 针对这一现状, 本文总结了以下研究挑战和对应的研究方向, 值得领域内研究者关注:

(1) 生物启发的高效神经编码算法设计: 生物神经系统中最早被发现的编码方式是频率编码^[163], 其后更多证据表明, 生物神经系统中还存在更高效的编码方式, 例如人类触觉感知系统可以通过单个脉冲的精确发放时刻来编码触感^[164], 苍蝇可以凭借 30 毫秒内到达的一两个脉冲对环境做出快速反应^[165]. 目前爆发编码(Burst encode)^[166, 167]、相位编码^[189]等已经用于 ANN2SNN 方法, 而在梯度替代学习算法中的应用较少. 研究者们可以考虑设计生物启发的高效神经编码算法, 并应用于 SNN 内部的信息表征, 该研究一方面能够充分利用时域信息以降低时间步数和能耗, 另一方面也有望与脑机接口相关研究领域形成合力, 加速对大脑工作原理的理解.

(2) 神经元动态过于简化: 现有的神经元改进方案^[70, 74, 94, 95]中, 所使用的神经元也都较为简化, 并

不具备计算神经科学中常用的 Izhikevich^[56]神经元模型相当的复杂神经动态。需要注意的是, SNN 与 ANN 最大的区别即在于神经元; Wolfgang 证明 SNN 能够实现与 ANN 相同的拟合能力, 且使用更少的神经元^[26], 其关键在于脉冲神经元相较于 ANN 中激活函数所不具备的神经动态; He 等^[168]通过简单网络结构与复杂的神经元动态结合, 实现与复杂网络结构相同的性能, 且内存消耗更少、运行速度更快。以上理论和实验结果表明, 在 SNN 中使用复杂神经元具有诸多优势, 但受限于其高昂的计算成本、复杂的参数调试, IF 神经元、LIF 神经元等高度简化的脉冲神经元模型仍然是深度 SNN 的首选。未来的研究可以考虑通过并行加速算法降低计算代价, 以梯度下降法自动优化神经元参数, 从而构建具有复杂神经动态和脉冲模式的神经元模型并应用于深度 SNN。

(3) 网络结构层次的时域动态被忽略: 现有的 SNN 结构与 ANN 类似, 包含堆栈式卷积层和池化层、残差连接和自注意力机制。这一结构擅长提取空间特征而非时域特征, 后者则通常被认为是由脉冲神经元负责。一个典型的例子是, 即便是在 SpikFormer^[51]和 Spike-Driven Transformer^[52]这样最先进的脉冲 Transformer 架构中, 其自注意力计算也是局限于单个时间步内, 而不跨越时间步。这一设计理念导向了目前深度 SNN 的纯前馈网络结构, 忽略了网络结构层次的时域动态, 但大脑结构却并非如此。大脑中存在大量的同脑区和跨脑区稠密链接, 共同构成了一个巨大的循环神经网络; 传统观点认为视觉信息处理是一系列前馈过程, 以此也衍生出卷积神经网络架构。但最近越来越多证据表明这一过程中存在反馈途径, 即高阶的认知和视网膜的信息相互作用^[169]。Yin 等^[170]和 Rao 等^[171]在 SNN 中增加了反馈连接, 大幅度改善了网络的长期依赖学习能力, 但遗憾的是其反馈只局限于脉冲神经元层内, 并只在小网络、简单数据集上进行了验证。网络结构层次的时域动态尚未在深度 SNN 中得到重视, 这一问题值得研究者们关注。

(4) 突触可塑性学习算法研究进展缓慢: 反向传播算法根据网络输出计算最终损失, 并将误差逐层回传, 同时计算网络参数的更新量, 是一种全局的学习方式。在反向传播算法通过替代梯度法引入 SNN 后, 诸如赫布学习规则^[172]、脉冲时间依赖可塑性(Spike-Timing-Dependent Plasticity, STDP)^[173]等突触可塑性学习算法则因性能低下而较少使用。

然而, 这些方法亦有独特优点: 在理论研究方面, 它们对应着生物实验中发现的现象和数据, 对其研究有助于理解大脑学习的奥秘, 因而备受计算神经科学的青睐; 在实际应用方面, 它们是局部的学习规则, 在硬件上实现时只需要记录神经元和前后突触的活动信息, 资源消耗远少于需要记录整个网络中间层信息的反向传播算法, 适合片上学习, 例如 Nabil 等^[174]在 Intel Loihi 芯片^[35]上实现了基于 STDP 的片上实时学习并用于气味识别, 并且能够减缓灾难性遗忘; Wu 等^[175]将突触可塑性学习算法与梯度替代法共同使用, 发现这种混合学习机制在小样本学习、持续学习和容错学习方面均优于纯梯度替代法, 同时将该算法部署到 Tianjic 芯片^[37], 受益于突触可塑性的局部性, 各个计算核心之间的通信开销也大幅度降低。突触可塑性学习算法尽管已经展现出诸多潜力, 但将结合梯度替代法其用于改善大规模深度 SNN 的学习, 则尚未有成功的先例报道, 这一无人区值得研究者们探索。

(5) 软件仿真和硬件部署的沟壑: SNN 的目标运行设备是神经系统计算芯片, 但现有学习算法更多关注于软件仿真, 而对硬件部署面临的挑战关注不够。首先总结硬件相关的研究问题如下:

(a)模型量化(Model Quantization): GPU 通常配备 GB 级别的显存, 使用 float32 精度的突触, 但神经形态计算芯片上资源有限, 例如 Loihi^[35]至多支持 9 比特表示的突触权重。已经有部分研究者在中等规模 SNN 上进行了量化并达到较好效果^[176-179], 其中 Wei 等^[179]将权重量化到 1 比特、膜电位量化到 2 比特, 在 CIFAR100、Tiny ImageNet 等多个中等规模的数据集上达到了较好性能, 展现出低精度 SNN 部署的巨大潜力。

(b)网络剪枝(Network Pruning): 神经形态计算芯片上的内存容量有限, 无法容纳太多突触和神经元, 例如 Loihi^[35]至多支持存储空间不超过 16MB 的突触权重和 12800 个神经元, 而标准的 VGG11 则含有 132.86MB 的突触数, 如果是以 float32 精度实现则需要 531.44MB 存储空间, 远超 Loihi 的容纳能力。对突触和神经元进行剪枝可以大幅度降低模型大小。目前 SNN 中的简直技术可以分为两类, 一类是通用剪枝技术^[180-185], 可以同时用于 ANN 和 SNN; 另一类则是 SNN 独有的, 例如基于突触可塑性的剪枝方法^[186-189]、基于神经元发放率的剪枝方法^[190, 191]等。

(c)硬件非理想性(Hardware Non-idealities): 基于模

拟电路/数模混合电路或忆阻器实现的神经形态芯片, 例如 Neurogrid 等, 受限于制作工艺和自身特性, 其运行 SNN 时存在一定噪声, 输出结果与在 CPU/GPU 上的仿真存在一定差异。目前有少量研究者对这一问题进行了探索。Bhattacharjee 等^[192]发现 SNN 多步运行的特性在忆阻器上会累计误差, 并通过利用 BN 的统计量来记录噪声, 减少软件仿真和硬件部署的差异。Moro 等^[193]则是通过训练时注入噪声来模拟硬件运行的环境, 使硬件推理和软件模拟的结果更一致。Christensen 等^[194]指出, 设计“硬件感知”的软件学习算法, 即在软件中模拟硬件特性, 是解决硬件非理想性的关键途径。

(d)同步仿真和异步部署: 深度 SNN 通常中通常使用元素取值仅为 0 或 1 的张量(tensor)表示脉冲, 多个时刻的脉冲序列类似于帧; 而 DVS 相机和 Loihi、Speck 等基于异步电路实现的神经形态芯片则是使用 AER 协议表示脉冲事件, 两者的差异导致 CPU/GPU 上的训练和神经形态芯片上推理的精度不一致。尽管增大时间步数可以降低两者的差异, 但训练代价会显著增加。关于这一问题的研究还较少, Yao 等^[53]通过在单步释放多个脉冲来降低误差并在 Speck 芯片上进行了验证。设计异步仿真算法可能是解决这一问题的关键, 目前已经有初步工作^[195]进行了尝试。

现有的梯度替代算法, 通常使用任务性能和理论功耗作为评估指标, 而对硬件部署的可行性未作过多考虑。未来的研究者们可以更多地考虑硬件约束, 例如在有限内存和精度的条件下设计高性能 SNN 神经元和网络架构、软件模拟硬件特性等, 进行软件-硬件协同设计 (Software Hardware Codesign), 提升 SNN 的片上推理性能, 这一设计理念也是诸多 SNN 研究者所倡导的^[31, 194]。

6 总结与展望

本文介绍了基于梯度替代法直接训练的深度脉冲神经网络学习算法研究进展, 将已有算法进行分类, 并详细介绍和比较。本文海选取了各类梯度替代算法中的代表性方法, 并在多个数据集和任务上比较了它们的多个性能指标。根据前文的系统性梳理和对比实验结果, 现对各类方法现状和可能的改进方向总结如下:

(1) **基础学习算法**是梯度替代法训练 SNN 的基石, 但目前关于不同替代函数优劣、网络收敛条件等理

论分析较少, 多数已有的研究都基于实验性结论, 需要研究者们重视。

(2) **编码方式**在 ANN2SNN 中研究较多, 而梯度替代法能够直接端到端训练, 因而研究者很少进行网络内部编码方式的手工设计。但目前的 SNN 在时间步数极小时性能会剧烈下降, 引入网络内部的时间编码或许能够解决这一问题, 从而实现超低延迟的 SNN 推理。降低时间步数的另一优势是, BPTT 训练的开销也会大幅下降, 因而对编码方式的研究也有助于降低 SNN 的训练开销。

(3) **神经元和突触改进方法**通常会不可避免地增加模型的复杂度, 甚至引入一些难以在现有硬件上实现的操作, 例如 CLIF 神经元^[95]中的 Sigmoid 激活函数涉及硬件上昂贵的指数运算; Sliding PSN^[70]作为 k 阶神经元需要 k 个历史输入的存储消耗; 有状态的突触^[99]也需要额外的资源存储和更新突触上电流的状态, 且在训练时会显著增加内存消耗。因而, 未来的研究中应更多的考虑神经元在 GPU 上的并行加速算法和神经形态硬件兼容性, 以增强模型的训练速度和实用性。

(4) **网络结构改进方法**已经取得了较大成功, 但也有一些遗留问题尚未解决, 例如前文已经讨论过的 SEW ResNet^[50]和 MS ResNet^[103]的硬件部署问题。此外, 目前 SNN 网络结构设计整体思路仍然延续自 ANN, 而生物神经系统中的反馈连接、侧向抑制等特性尚未得到探索, 这些特殊的结构可能是实现人脑级别通用人工智能的关键, 有待进一步探索。

(5) **正则化方法**中较新的 BN^[135]、TEBN^[136]等方法使用逐时刻的参数, 因此要求输入序列的长度不可变, 在处理实际任务时可能不够灵活, 这一问题有待改进。此外, 目前 BN 类方法较多, 而其他方法较少。考虑到脉冲化的 Transformer 架构目前性能更高, 而 ANN 中的结论已经表明 Transformer 架构使用 LN 性能更好, 故未来的研究可更多聚焦于 LN 的变体在 SNN 中的应用。一个典型的问题是, 原始的 LN 无法与卷积层合并, 这一问题仍有待解决。此外, 也可考虑针对脉冲神经元的特性, 设计诸如 NeuNorm^[132]类型的 SNN 专用正则化方法。

(6) **ANN 辅助训练算法**中基于 ANN 耦合的算法梯度误差较大, 但其相较于普通梯度替代法能够避免 BPTT 的巨大内存消耗量, 值得进一步研究。其本质可以认为是使用脉冲在时间上的累计来计算梯度, 因而未来的研究方向可以聚焦于设计低误差的脉冲累计表示方法。基于 ANN 蒸馏的算法则主要存在

计算代价高、超参数数量多且调试困难的缺陷需要改进.两类方法均不能用于时域任务,根源在于 ANN 不具有时间维度,而通过循环神经网络或 Transformer 辅助训练或许能够解决这一问题.

(7)事件驱动学习算法适合硬件实现,但目前研究还处在初级阶段,实际性能较为一般,且对超参数敏感、稳定性差、任务正确率较低,存在很大改进空间.值得注意的是,事件驱动算法使用脉冲传递梯度的这一特性,更适合基于稀疏计算的实现方式,即在前向传播和反向传播时使用脉冲发放时刻表示脉冲.而目前的事件驱动仿真方式仍然使用基于二值张量的方式表示脉冲,无脉冲的位置表示成 0,存在很大的表示冗余.如何设计一套稀疏加速仿真方式,也是值得整个领域内研究者们重视的话题.

(8) 在线学习算法有望解决 SNN 使用 BPTT 训练内存消耗量过大的问题,且适合在神经系统硬件上进行片上学习.该方法目前在静态数据集上表现优秀,但对时域任务还不能很好的处理,未来的相关研究可对此问题进行重点关注.

(9) 训练加速方法中 SpikingJelly^[49]框架加速效果较好且通用性最强,但其加速思路更类似于加速 RNN,没有充分利用脉冲的二值量化、稀疏激活特性;稀疏脉冲梯度下降^[155]则一定程度上利用了 SNN 的稀疏特性,但其受限于工程难度,只在 MLP 上进行了实验,没有在更常用的卷积架构上实现.研究者们如果能够充分利用 SNN 的特性,通过稀疏计算降低计算量和内存消耗,通过二值脉冲和浮点权重的混合精度运算提升计算速度,则 SNN 相较于 ANN 的低功耗优势或许能从仅推理阶段延申到更具实用价值的训练阶段,从而彻底解决现有的人工智能训练成本高昂的难题,这将使得 SNN 的科学和应用价值进一步提升.

从宏观视角来看,作为神经科学和计算科学融合产物的脉冲深度学习领域的梯度替代类学习方法,目前灵感和方法论多来自于深度学习已有的研究范式,技术路线与量化神经网络、循环神经网络、微型机器学习等领域也存在一定重合.这种研究范式是一把双刃剑,既带来了 SNN 性能的快速提升,也不可避免地引入了深度学习的固有缺陷,例如依赖大量有标注数据并进行多轮训练才能达到较好效果,而人脑则可以仅使用少量样本高效学习;在新任务上学习会引发旧任务的性能骤降,即灾难性遗忘(Catastrophic Forgetting),而人脑则擅长利用已有经验迅速学习新任务,具有很强的迁移学习

(Transfer Learning)能力;对随机扰动敏感,容易被对抗攻击(Adversarial Attack)诱导,而人类对攻击样本则展现出惊人的正确率和鲁棒性^[196].

考虑到神经科学在人工智能发展中的历史地位,以及人脑仍是已知最智能的系统这一现实,模仿大脑的结构功能和运行原理来设计 SNN 学习算法,或许能够解决传统深度学习方法面临的挑战,并推动人工智能领域取得新一次重大进展.令人欣喜的是,已经有部分研究者在这一方向进行了探索,例如通过突触可塑性^[175, 197, 198]或脉冲神经元的阈值调节^[199],缓解灾难性遗忘,提升持续学习和小样本学习能力;利用循环连接^[170]、超极化电流(After-hyperpolarizing Current)^[171]、精细神经元模型^[168]、非局部的突触可塑性^[200]、兴奋型/抑制型神经元^[170]、突触延迟^[100]等生理结构和机制改善网络记忆能力、任务性能或参数效率;利用离散的脉冲方法过程和神经元的积分泄露机制^[201]来增强对抗攻击的鲁棒性.这类研究在神经科学的指引下,充分利用了 SNN 独有的神经动态和突触可塑性机制,设计脑启发的结构和算法,并在特定任务上超越了传统深度学习方法的性能,开辟了有别于传统深度学习方法之外的研究道路,值得后来的学者参考.

参 考 文 献

- [1] Lecun Yann, Bengio Yoshua, Hinton Geoffrey. Deep Learning [J]. Nature, 2015, 521(7553): 436-44.
- [2] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification; proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, F, 2015 [C].
- [3] Szegedy Christian, Liu Wei, Jia Yangqing, et al. Going Deeper with Convolutions; proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, F, 2015 [C].
- [4] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep Residual Learning for Image Recognition; proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, F, 2016 [C].
- [5] Dosovitskiy Alexey, Beyer Lucas, Kolesnikov

- Alexander, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale; proceedings of the International Conference on Learning Representations, F, 2021 [C].
- [6] Girshick Ross, Donahue Jeff, Darrell Trevor, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation; proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, F, 2014 [C].
- [7] Redmon Joseph, Divvala Santosh, Girshick Ross, et al. You Only Look Once: Unified, Real-Time Object Detection; proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, F, 2016 [C].
- [8] Graves Alex, Mohamed Abdel-Rahman, Hinton Geoffrey. Speech Recognition with Deep Recurrent Neural Networks; proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, F, 2013 [C].
- [9] Graves Alex, Jaitly Navdeep, Mohamed Abdel-Rahman. Hybrid Speech Recognition with Deep Bidirectional LSTM; proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding, F, 2013 [C].
- [10] Sutskever Ilya, Vinyals Oriol, Le Quoc V. Sequence to Sequence Learning with Neural Networks; proceedings of the Advances in Neural Information Processing Systems, F, 2014 [C].
- [11] Bahdanau Dzmitry, Cho Kyunghyun, Bengio Yoshua. Neural Machine Translation by Jointly Learning to Align and Translate; proceedings of the International Conference on Learning Representations, F, 2015 [C].
- [12] Sennrich Rico, Haddow Barry, Birch Alexandra. Neural Machine Translation of Rare Words with Subword Units; proceedings of the Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, F, 2016 [C].
- [13] Mnih Volodymyr, Kavukcuoglu Koray, Silver David, et al. Human-Level Control through Deep Reinforcement Learning [J]. Nature, 2015, 518(7540): 529-33.
- [14] Silver David, Schrittwieser Julian, Simonyan Karen, et al. Mastering the Game of Go without Human Knowledge [J]. Nature, 2017, 550(7676): 354-9.
- [15] Brown Tom, Mann Benjamin, Ryder Nick, et al. Language Models are Few-Shot Learners; proceedings of the Advances in Neural Information Processing Systems, F, 2020 [C]. Curran Associates, Inc.
- [16] Zeng Wei, Ren Xiaozhe, Su Teng, et al. PanGu-\$\$: Large-scale Autoregressive Pretrained Chinese Language Models with Auto-parallel Computation [Z]. 2021
- [17] Openai, Achiam Josh, Adler Steven, et al. GPT-4 Technical Report [Z]. 2024
- [18] Goodfellow Ian, Pouget-Abadie Jean, Mirza Mehdi, et al. Generative Adversarial Nets; proceedings of the Advances in Neural Information Processing Systems, F, 2014 [C]. Curran Associates, Inc.
- [19] Radford Alec, Metz Luke, Chintala Soumith. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks; proceedings of the International Conference on Learning Representations, F, 2016 [C].
- [20] Rombach Robin, Blattmann Andreas, Lorenz Dominik, et al. High-Resolution Image Synthesis With Latent Diffusion Models; proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), F June, 2022 [C].
- [21] Hassabis Demis, Kumaran Dharshan, Summerfield Christopher, et al. Neuroscience-inspired Artificial Intelligence [J]. Neuron, 2017, 95(2): 245-58.
- [22] Zador Anthony, Escola Sean, Richards Blake, et al. Catalyzing next-generation Artificial Intelligence through NeuroAI [J]. Nature Communications, 2023, 14(1): 1597.
- [23] Rosenblatt Frank. The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain [J]. Psychological Review, 1958, 65(6): 386.
- [24] Rumelhart David E., Hinton Geoffrey E., Williams Ronald J. Learning Representations by Back-Propagating Errors [J]. Nature, 1986, 323(6088): 533-6.

- [25] Cortes Corinna, Vapnik Vladimir. Support-Vector Networks [J]. Machine Learning, 1995, 20(3): 273-97.
- [26] Maass Wolfgang. Networks of Spiking Neurons: the Third Generation of Neural Network Models [J]. Neural Networks, 1997, 10(9): 1659-71.
- [27] Gewaltig Marc-Oliver, Diesmann Markus. Nest (Neural Simulation Tool) [J]. Scholarpedia, 2007, 2(4): 1430.
- [28] Eliasmith Chris, Stewart Terrence C., Choo Xuan, et al. A Large-Scale Model of the Functioning Brain [J]. Science, 2012, 338(6111): 1202-5.
- [29] Stimberg Marcel, Brette Romain, Goodman Dan F. M. Brian 2, an Intuitive and Efficient Neural Simulator [J]. eLife, 2019, 8: e47314.
- [30] Mead Carver. Neuromorphic Electronic Systems [J]. Proceedings of the IEEE, 1990, 78(10): 1629-36.
- [31] Roy Kaushik, Jaiswal Akhilesh, Panda Priyadarshini. Towards Spike-based Machine Intelligence with Neuromorphic Computing [J]. Nature, 2019, 575(7784): 607-17.
- [32] Lichtsteiner Patrick, Posch Christoph, Delbruck Tobi. A 128x128 120 dB 15us Latency Asynchronous Temporal Contrast Vision Sensor [J]. IEEE Journal of Solid-State Circuits, 2008, 43(2): 566-76.
- [33] Dong Siwei, Huang Tiejun, Tian Yonghong. Spike Camera and Its Coding Methods; proceedings of the Data Compression Conference, F, 2017 [C].
- [34] Merolla Paul A., Arthur John V., Alvarez-Icaza Rodrigo, et al. A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface [J]. Science, 2014, 345(6197): 668-73.
- [35] Davies Mike, Srinivasa Narayan, Lin Tsung-Han, et al. Loihi: a Neuromorphic Manycore Processor with On-Chip Learning [J]. IEEE Micro, 2018, 38(1): 82-99.
- [36] Ma De, Shen Juncheng, Gu Zonghua, et al. Darwin: a Neuromorphic Hardware Co-processor Based on Spiking Neural Networks [J]. Journal of Systems Architecture, 2017, 77: 43-51.
- [37] Pei Jing, Deng Lei, Song Sen, et al. Towards Artificial General Intelligence with Hybrid Tianjic Chip Architecture [J]. Nature, 2019, 572(7767): 106-11.
- [38] Hinton Geoffrey E., Osindero Simon, Teh Yee-Whye. A Fast Learning Algorithm for Deep Belief Nets [J]. Neural Computation, 2006, 18(7): 1527-54.
- [39] Lecun Yann, Bottou Lon, Bengio Yoshua, et al. Gradient-Based Learning Applied to Document Recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-324.
- [40] Goodfellow Ian, Bengio Yoshua, Courville Aaron. Deep Learning [M]. MIT Press, 2016.
- [41] Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks; proceedings of the Advances in Neural Information Processing Systems, F, 2012 [C]. Curran Associates, Inc.
- [42] Russakovsky Olga, Deng Jia, Su Hao, et al. ImageNet Large Scale Visual Recognition Challenge [J]. International Journal of Computer Vision, 2015, 115(3): 211-52.
- [43] Neftci Emre O., Mostafa Hesham, Zenke Friedemann. Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks [J]. IEEE Signal Processing Magazine, 2019, 36(6): 51-63.
- [44] Cao Yongqiang, Chen Yang, Khosla Deepak. Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition [J]. International Journal of Computer Vision, 2015, 113(1): 54-66.
- [45] Tavanaei Amirhossein, Ghodrati Masoud, Kheradpisheh Saeed Reza, et al. Deep Learning in Spiking Neural Networks [J]. Neural Networks, 2019, 111: 47-63.
- [46] Wu Yujie, Deng Lei, Li Guoqi, et al. Spatio-Temporal Backpropagation for Training High-Performance Spiking Neural Networks [J]. Frontiers in Neuroscience, 2018, 12: 331.
- [47] Zenke Friedemann, Ganguli Surya. SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks [J]. Neural Computation, 2018, 30(6): 1514-41.
- [48] Shrestha Sumit Bam, Orchard Garrick. SLAYER: Spike Layer Error Reassignment in Time; proceedings of the Advances in Neural Information

- Processing Systems, F, 2018 [C].
- [49] Fang Wei, Chen Yanqi, Ding Jianhao, et al. SpikingJelly: An open-source machine learning infrastructure platform for spike-based intelligence [J]. Science Advances, 2023, 9(40): eadi1480.
- [50] Fang Wei, Yu Zhaoifei, Chen Yanqi, et al. Deep Residual Learning in Spiking Neural Networks; proceedings of the Advances in Neural Information Processing Systems, F, 2021 [C].
- [51] Zhou Zhaokun, Zhu Yuesheng, He Chao, et al. Spikformer: When Spiking Neural Network Meets Transformer; proceedings of the International Conference on Learning Representations, F, 2023 [C].
- [52] Yao Man, Hu Jiakui, Zhou Zhaokun, et al. Spike-Driven Transformer; proceedings of the Advances in Neural Information Processing Systems, F, 2023 [C].
- [53] Yao Man, Richter Ole, Zhao Guangshe, et al. Spike-based Dynamic Computing with Asynchronous Sensing-Computing Neuromorphic Chip [J]. Nature Communications, 2024, 15(1): 4464.
- [54] Ioffe Sergey, Szegedy Christian. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift; proceedings of the International Conference on Machine Learning, F, 2015 [C].
- [55] Ba Jimmy Lei, Kiros Jamie Ryan, Hinton Geoffrey E. Layer Normalization [J]. arXiv preprint arXiv:160706450, 2016.
- [56] Izhikevich Eugene M. Simple Model of Spiking Neurons [J]. IEEE Transactions on Neural Networks, 2003, 14(6): 1569-72.
- [57] Ledinauskas Eimantas, Ruseckas Julius, Jurnas Alfonsas, et al. Training Deep Spiking Neural Networks [J]. arXiv preprint arXiv:200604436, 2020.
- [58] Rueckauer Bodo, Lungu Iulia-Alexandra, Hu Yuhuang, et al. Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification [J]. Frontiers in Neuroscience, 2017, 11: 682.
- [59] Xiao Han, Rasul Kashif, Vollgraf Roland. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms [J]. arXiv preprint arXiv:170807747, 2017.
- [60] Krizhevsky Alex, Hinton Geoffrey. Learning Multiple Layers of Features from Tiny Images [J]. 2009.
- [61] !!! INVALID CITATION !!! [46, 49, 61].
- [62] Orchard Garrick, Jayawant Ajinkya, Cohen Gregory K., et al. Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades [J]. Frontiers in Neuroscience, 2015, 9.
- [63] Li Hongmin, Liu Hanchao, Ji Xiangyang, et al. CIFAR10-DVS: an Event-Stream Dataset for Object Classification [J]. Frontiers in Neuroscience, 2017, 11.
- [64] Amir Arnon, Taba Brian, Berg David, et al. A Low Power, Fully Event-Based Gesture Recognition System; proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, F, 2017 [C].
- [65] Bi Yin, Chadha Aaron, Abbas Alhabib, et al. Graph-Based Object Classification for Neuromorphic Vision Sensing; proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, F, 2019 [C].
- [66] Lin Yihan, Ding Wei, Qiang Shaohua, et al. ES-ImageNet: a Million Event-Stream Classification Dataset for Spiking Neural Networks [J]. Frontiers in Neuroscience, 2021, 15.
- [67] Cramer Benjamin, Stradmann Yannik, Schemmel Johannes, et al. The Heidelberg Spiking Data Sets for the Systematic Evaluation of Spiking Neural Networks [J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(7): 2744-57.
- [68] Iyer Laxmi R., Chua Yansong, Li Haizhou. Is Neuromorphic MNIST Neuromorphic? Analyzing the Discriminative Power of Neuromorphic Datasets in the Time Domain [J]. Frontiers in Neuroscience, 2021, 15.
- [69] Yin Bojian, Corradi Federico, Boht Sander M. Accurate and Efficient Time-domain Classification with Adaptive Spiking Recurrent Neural Networks [J]. Nature Machine Intelligence, 2021, 3(10): 905-13.
- [70] Fang Wei, Yu Zhaoifei, Zhou Zhaokun, et al. Parallel Spiking Neurons with High Efficiency and Ability to

- Learn Long-term Dependencies; proceedings of the Advances in Neural Information Processing Systems, F, 2023 [C].
- [71] Horowitz Mark. 1.1 Computing's Energy Problem (And What We Can Do about It); proceedings of the IEEE International Solid-State Circuits Conference Digest of Technical Papers, F, 2014 [C].
- [72] Yang Zheyu, Wang Taoyi, Lin Yihan, et al. A Vision Chip with Complementary Pathways for Open-World Sensing [J]. *Nature*, 2024, 629(8014): 1027-33.
- [73] Bengio Yoshua, Lonard Nicholas, Courville Aaron. Estimating or Propagating Gradients through Stochastic Neurons for Conditional Computation [J]. arXiv preprint arXiv:13083432, 2013.
- [74] Fang Wei, Yu Zhaofoei, Chen Yanqi, et al. Incorporating Learnable Membrane Time Constant to Enhance Learning of Spiking Neural Networks; proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, F, 2021 [C].
- [75] Gygax Julia, Zenke Friedemann. Elucidating the theoretical underpinnings of surrogate gradient learning in spiking neural networks [J]. arXiv preprint arXiv:240414964, 2024.
- [76] Hubara Itay, Courbariaux Matthieu, Soudry Daniel, et al. Quantized neural networks: Training neural networks with low precision weights and activations [J]. *Journal of Machine Learning Research*, 2018, 18(187): 1-30.
- [77] Zenke Friedemann, Vogels Tim P. The Remarkable Robustness of Surrogate Gradient Learning for Instilling Complex Function in Spiking Neural Networks [J]. *BioRxiv*, 2020.
- [78] Lian Shuang, Shen Jiangrong, Liu Qianhui, et al. Learnable Surrogate Gradient for Direct Training Spiking Neural Networks; proceedings of the IJCAI, F, 2023 [C].
- [79] Li Yuhang, Guo Yufei, Zhang Shanghang, et al. Differentiable Spike: Rethinking Gradient-Descent for Training Spiking Neural Networks; proceedings of the Advances in Neural Information Processing Systems, F, 2021 [C].
- [80] Che Kaiwei, Leng Luziwei, Zhang Kaixuan, et al. Differentiable Hierarchical and Surrogate Gradient Search for Spiking Neural Networks [J]. *Advances in Neural Information Processing Systems*, 2022, 35: 24975-90.
- [81] Lee Chankyu, Sarwar Syed Shakib, Panda Priyadarshini, et al. Enabling Spike-based Backpropagation for Training Deep Neural Network Architectures [J]. *Frontiers in Neuroscience*, 2020, 14.
- [82] Cheng Xiang, Hao Yunzhe, Xu Jiaming, et al. LISNN: Improving Spiking Neural Networks with Lateral Interactions for Robust Object Recognition; proceedings of the International Joint Conference on Artificial Intelligence, F, 2020 [C].
- [83] Kheradpisheh Saeed Reza, Masquelier Timothe. Temporal Backpropagation for Spiking Neural Networks with One Spike per Neuron [J]. *International Journal of Neural Systems*, 2020, 30(06): 2050027.
- [84] Stanojevic Ana, Woźniak Stanisław, Bellec Guillaume, et al. High-performance deep spiking neural networks with 0.3 spikes per neuron [J]. *Nature Communications*, 2024, 15(1): 6793.
- [85] Rathi Nitin, Roy Kaushik. DIET-SNN: a Low-Latency Spiking Neural Network with Direct Input Encoding and Leakage and Threshold Optimization [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [86] Rueckauer Bodo, Liu Shih-Chii. Conversion of analog to spiking neural networks using sparse temporal coding; proceedings of the IEEE International Symposium on Circuits and Systems, F, 2018 [C].
- [87] Zhang Lei, Zhou Shengyuan, Zhi Tian, et al. TDSNN: From Deep Neural Networks to Deep Spike Neural Networks with Temporal-Coding [J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, 33(01): 1319-26.
- [88] Stanojevic Ana, Woźniak Stanisław, Bellec Guillaume, et al. An exact mapping from ReLU networks to spiking neural networks [J]. *Neural Networks*, 2023, 168: 74-88.
- [89] Kim Jaehyun, Kim Heesu, Huh Subin, et al. Deep Neural Networks with Weighted Spikes [J]. *Neurocomputing*, 2018, 311: 373-86.

- [90] Wang Z., Gu X., Goh R. S. M., et al. Efficient Spiking Neural Networks With Radix Encoding [J]. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(3): 3689-701.
- [91] Li Yuhang, Kim Youngeun, Park Hyoungeob, et al. Uncovering the Representation of Spiking Neural Networks Trained with Surrogate Gradient [J]. Transactions on Machine Learning Research, 2023.
- [92] Hu Jiakui, Yao Man, Qiu Xuerui, et al. High-Performance Temporal Reversible Spiking Neural Networks with $\mathcal{O}(L)$ Training Memory and $\mathcal{O}(1)$ Inference Cost; proceedings of the International Conference on Machine Learning, F, 2024 [C].
- [93] Yao Xingting, Li Fanrong, Mo Zitao, et al. GLIF: a Unified Gated Leaky Integrate-and-Fire Neuron for Spiking Neural Networks; proceedings of the Advances in Neural Information Processing Systems, F, 2022 [C].
- [94] Feng Lang, Liu Qianhui, Tang Huajin, et al. Multi-Level Firing with Spiking DS-ResNet: Enabling Better and Deeper Directly-Trained Spiking Neural Networks; proceedings of the International Joint Conference on Artificial Intelligence, F, 2022 [C].
- [95] Huang Yulong, Xiaopeng L. I. N., Ren Hongwei, et al. CLIF: Complementary Leaky Integrate-and-Fire Neuron for Spiking Neural Networks; proceedings of the Forty-first International Conference on Machine Learning, F, 2024 [C].
- [96] Yarga Sidi Yaya Arnaud, Wood Sean U. N. Accelerating SNN Training with Stochastic Parallelizable Spiking Neurons; proceedings of the International Joint Conference on Neural Networks, F, 2023 [C].
- [97] Mostafa Hesham, Pedroni Bruno U., Sheik Sadique, et al. Fast Classification Using Sparsely Active Spiking Networks; proceedings of the IEEE International Symposium on Circuits and Systems, F, 2017 [C].
- [98] Mostafa Hesham. Supervised Learning Based on Temporal Coding in Spiking Neural Networks [J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 29(7): 3227-35.
- [99] Fang Haowen, Shrestha Amar, Zhao Ziyi, et al. Exploiting Neuron and Synapse Filter Dynamics in Spatial Temporal Learning of Deep Spiking Neural Network; proceedings of the International Joint Conference on Artificial Intelligence, F, 2020 [C].
- [100] Hammouamri Ilyass, Khalfaoui-Hassani Ismail, Masquelier Timothe. Learning Delays in Spiking Neural Networks using Dilated Convolutions with Learnable Spacings; proceedings of the The Twelfth International Conference on Learning Representations, F, 2024 [C].
- [101] Hu Yangfan, Tang Huajin, Pan Gang. Spiking Deep Residual Networks [J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(8): 5200-5.
- [102] Zheng Hanle, Wu Yujie, Deng Lei, et al. Going Deeper With Directly-Trained Larger Spiking Neural Networks; proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, F, 2021 [C].
- [103] Hu Yifan, Deng Lei, Wu Yujie, et al. Advancing Spiking Neural Networks Toward Deep Residual Learning [J]. IEEE Transactions on Neural Networks and Learning Systems, 2024: 1-15.
- [104] Graves Alex. Generating Sequences With Recurrent Neural Networks [Z]. 2014
- [105] Vaswani Ashish, Shazeer Noam, Parmar Niki, et al. Attention Is All You Need; proceedings of the Advances in Neural Information Processing Systems, F, 2017 [C].
- [106] Yao Man, Gao Huanhuan, Zhao Guangshe, et al. Temporal-Wise Attention Spiking Neural Networks for Event Streams Classification; proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, F, 2021 [C].
- [107] Yao Man, Hu Jiakui, Zhao Guangshe, et al. Inherent Redundancy in Spiking Neural Networks; proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, F, October, 2023 [C].
- [108] Yao Man, Zhang Hengyu, Zhao Guangshe, et al. Sparser Spiking Activity Can Be Better: Feature Refine-and-Mask Spiking Neural Network for Event-Based Visual Recognition [J]. Neural

- Networks, 2023, 166: 410-23.
- [109] Xu Qi, Gao Yuyuan, Shen Jiangrong, et al. Enhancing Adaptive History Reserving by Spiking Convolutional Block Attention Module in Recurrent Neural Networks; proceedings of the Advances in Neural Information Processing Systems, F, 2023 [C]. Curran Associates, Inc.
- [110] Yao Man, Zhao Guangshe, Zhang Hengyu, et al. Attention Spiking Neural Networks [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(8): 9393-410.
- [111] Zhu Rui-Jie, Zhang Malu, Zhao Qihang, et al. TCJA-SNN: Temporal-Channel Joint Attention for Spiking Neural Networks [J]. IEEE Transactions on Neural Networks and Learning Systems, 2024: 1-14.
- [112] Huang Ziyuan, Zhang Shiwei, Pan Liang, et al. TAda! Temporally-Adaptive Convolutions for Video Understanding; proceedings of the International Conference on Learning Representations, F, 2022 [C].
- [113] Wang Xiaolong, Girshick Ross, Gupta Abhinav, et al. Non-local Neural Networks; proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, F, 2018 [C].
- [114] Kim Sangyeob, Kim Soyeon, Hong Seongyon, et al. C-DNN: a 24.5-85.8 TOPS/W Complementary-Deep-Neural-Network Processor with Heterogeneous CNN/SNN Core Architecture and Forward-Gradient-Based Sparsity Generation; proceedings of the IEEE International Solid-State Circuits Conference, F, 2023 [C].
- [115] Chang Muya, Lele Ashwin Sanjay, Spetalnick Samuel D., et al. A Heterogeneous RRAM In-Memory and SRAM Near-Memory SoC for Fused Frame and Event-Based Target Identification and Tracking; proceedings of the IEEE International Solid-State Circuits Conference, F, 2023 [C].
- [116] Zhang Jiqing, Dong Bo, Zhang Haiwei, et al. Spiking Transformers for Event-Based Single Object Tracking; proceedings of the Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, F, 2022 [C].
- [117] Zhang Jiyuan, Tang Lulu, Yu Zhaofei, et al. Spike Transformer: Monocular Depth Estimation for Spiking Camera; proceedings of the European Conference on Computer Vision, F, 2022 [C].
- [118] Han Minglun, Wang Qingyu, Zhang Tielin, et al. Complex Dynamic Neurons Improved Spiking Transformer Network for Efficient Automatic Speech Recognition [J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2023.
- [119] Shi Xinyu, Hao Zecheng, Yu Zhaofei. SpikingResformer: Bridging ResNet and Vision Transformer in Spiking Neural Networks; proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, F June, 2024 [C].
- [120] Zhou Chenlin, Zhang Han, Zhou Zhaokun, et al. QKFormer: Hierarchical Spiking Transformer using Q-K Attention [Z]. 2024
- [121] Hassani Ali, Walton Steven, Shah Nikhil, et al. Escaping the Big Data Paradigm with Compact Transformers [Z]. 2022
- [122] Yao Man, Hu Jiakui, Hu Tianxiang, et al. Spike-driven Transformer V2: Meta Spiking Neural Network Architecture Inspiring the Design of Next-generation Neuromorphic Chips; proceedings of the The Twelfth International Conference on Learning Representations, F, 2024 [C].
- [123] Liu Ze, Lin Yutong, Cao Yue, et al. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows; proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, F, 2021 [C].
- [124] Na Byunggook, Mok Jisoo, Park Seongsik, et al. AutoSNN: Towards Energy-Efficient Spiking Neural Networks; proceedings of the Proceedings of the 39th International Conference on Machine Learning, F, 2022 [C].
- [125] Pham Hieu, Guan Melody, Zoph Barret, et al. Efficient neural architecture search via parameters sharing; proceedings of the International conference on machine learning, F, 2018 [C]. PMLR.
- [126] Cai Han, Gan Chuang, Wang Tianzhe, et al. Once-for-all: Train one network and specialize it for efficient deployment [J]. arXiv preprint arXiv:190809791, 2019.
- [127] Shen Guobin, Zhao Dongcheng, Dong Yiting, et al.

- Brain-inspired neural circuit evolution for spiking neural networks [J]. *Proceedings of the National Academy of Sciences*, 2023, 120(39): e2218173120.
- [128] Suzuki Norimitsu, Bekkers John M. Microcircuits mediating feedforward and feedback synaptic inhibition in the piriform cortex [J]. *Journal of Neuroscience*, 2012, 32(3): 919-31.
- [129] Yan Jiaqi, Liu Qianhui, Zhang Malu, et al. Efficient spiking neural network design via neural architecture search [J]. *Neural Networks*, 2024, 173: 106172.
- [130] Li Jintang, Yu Zhouxin, Zhu Zulun, et al. Scaling up dynamic graph representation learning via spiking neural networks; proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, F, 2023 [C].
- [131] Ren Dayong, Ma Zhe, Chen Yuanpei, et al. Spiking pointnet: Spiking neural networks for point clouds [J]. *Advances in Neural Information Processing Systems*, 2024, 36.
- [132] Wu Yujie, Deng Lei, Li Guoqi, et al. Direct Training for Spiking Neural Networks: Faster, Larger, Better; proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, F, 2019 [C].
- [133] Carandini Matteo, Heeger David J. Normalization as a canonical neural computation [J]. *Nature reviews neuroscience*, 2012, 13(1): 51-62.
- [134] Mante Valerio, Bonin Vincent, Carandini Matteo. Functional mechanisms shaping lateral geniculate responses to artificial and natural stimuli [J]. *Neuron*, 2008, 58(4): 625-38.
- [135] Kim Youngeun, Panda Priyadarshini. Revisiting Batch Normalization for Training Low-Latency Deep Spiking Neural Networks From Scratch [J]. *Frontiers in Neuroscience*, 2021, 15.
- [136] Duan Chaoteng, Ding Jianhao, Chen Shiyang, et al. Temporal Effective Batch Normalization in Spiking Neural Networks; proceedings of the Advances in Neural Information Processing Systems, F, 2022 [C].
- [137] Guo Yufei, Zhang Yuhang, Chen Yuanpei, et al. Membrane Potential Batch Normalization for Spiking Neural Networks; proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, F October, 2023 [C].
- [138] Guo Yufei, Liu Xiaode, Chen Yuanpei, et al. RMP-Loss: Regularizing Membrane Potential Distribution for Spiking Neural Networks; proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, F October, 2023 [C].
- [139] Deng Shikuang, Li Yuhang, Zhang Shanghang, et al. Temporal Efficient Training of Spiking Neural Network via Gradient Re-weighting; proceedings of the International Conference on Learning Representations, F, 2022 [C].
- [140] Li Yuhang, Kim Youngeun, Park Hyoungseob, et al. Neuromorphic Data Augmentation for Training Spiking Neural Networks; proceedings of the European Conference on Computer Vision, Cham, F, 2022 [C].
- [141] Wu Jibin, Chua Yansong, Zhang Malu, et al. A Tandem Learning Rule for Effective Training and Rapid Inference of Deep Spiking Neural Networks [J]. *IEEE transactions on neural networks and learning systems*, 2023, 34(1): 446—60.
- [142] Kheradpisheh Saeed Reza, Mirsadeghi Maryam, Masquelier Timothee. Spiking Neural Networks Trained via Proxy [J]. *IEEE Access*, 2022.
- [143] Xu Qi, Li Yaxin, Shen Jiangrong, et al. Constructing Deep Spiking Neural Networks From Artificial Neural Networks With Knowledge Distillation; proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, F June, 2023 [C].
- [144] Qiu Haonan, Ning Munan, Song Zeyin, et al. Self-Architectural Knowledge Distillation for Spiking Neural Networks [J]. *Neural Networks*, 2024, 178: 106475.
- [145] Zhang Wenrui, Li Peng. Temporal Spike Sequence Learning via Backpropagation for Deep Spiking Neural Networks; proceedings of the Advances in Neural Information Processing Systems, F, 2020 [C].
- [146] Zhu Yaoyu, Yu Zhaoqi, Fang Wei, et al. Training Spiking Neural Networks with Event-driven Backpropagation; proceedings of the Advances in Neural Information Processing Systems, F, 2022 [C].
- [147] Zhu Yaoyu, Fang Wei, Xie Xiaodong, et al. Exploring Loss Functions for Time-based Training Strategy in Spiking Neural Networks [J]. *Advances in Neural*

- Information Processing Systems, 2024, 36.
- [148] Kaiser Jacques, Mostafa Hesham, Neftci Emre. Synaptic Plasticity Dynamics for Deep Continuous Local Learning (DECOLLE) [J]. *Frontiers in Neuroscience*, 2020, 14: 424.
- [149] Xiao Mingqing, Meng Qingyan, Zhang Zongpeng, et al. Online Training Through Time for Spiking Neural Networks; proceedings of the Advances in Neural Information Processing Systems, F, 2022 [C].
- [150] Meng Qingyan, Xiao Mingqing, Yan Shen, et al. Training High-Performance Low-Latency Spiking Neural Networks by Differentiation on Spike Representation; proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, F, 2022 [C].
- [151] Meng Qingyan, Xiao Mingqing, Yan Shen, et al. Towards Memory- and Time-Efficient Backpropagation for Training Spiking Neural Networks; proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, F October, 2023 [C].
- [152] Jiang Haiyan, De Masi Giulia, Xiong Huan, et al. NDOT: Neuronal Dynamics-based Online Training for Spiking Neural Networks; proceedings of the International Conference on Machine Learning, F, 2024 [C].
- [153] Zhu Yaoyu, Ding Jianhao, Huang Tiejun, et al. Online Stabilization of Spiking Neural Networks; proceedings of the International Conference on Learning Representations, F, 2024 [C].
- [154] Liu Zhuang, Mao Hanzhi, Wu Chao-Yuan, et al. A ConvNet for the 2020s; proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, F, 2022 [C].
- [155] Perez-Nieves Nicolas, Goodman Dan F. M. Sparse Spiking Gradient Descent; proceedings of the Advances in Neural Information Processing Systems, F, 2021 [C].
- [156] Taylor Luke, King Andrew, Harper Nicol S. Addressing the Speed-Accuracy Simulation Trade-off for Adaptive Spiking Neurons; proceedings of the Advances in Neural Information Processing Systems, F, 2023 [C]. Curran Associates, Inc.
- [157] De Tournemire Pierre, Nitti Davide, Perot Etienne, et al. A large scale event-based detection dataset for automotive [J]. *arXiv preprint arXiv:200108499*, 2020.
- [158] Guo Yufei, Chen Yuanpei, Zhang Liwen, et al. IM-Loss: Information Maximization Loss for Spiking Neural Networks; proceedings of the Advances in Neural Information Processing Systems, F, 2022 [C]. Curran Associates, Inc.
- [159] Fan Yimeng, Zhang Wei, Liu Changsong, et al. SFOD: Spiking Fusion Object Detector; proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), F June, 2024 [C].
- [160] Huang Gao, Liu Zhuang, Van Der Maaten Laurens, et al. Densely Connected Convolutional Networks; proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, F, 2017 [C].
- [161] Sironi Amos, Brambilla Manuele, Bourdis Nicolas, et al. Hats: Histograms of Averaged Time Surfaces for Robust Event-Based Object Classification; proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision, F, 2018 [C].
- [162] Wolpert David H. The lack of a priori distinctions between learning algorithms [J]. *Neural computation*, 1996, 8(7): 1341-90.
- [163] Adrian Edgar D., Zotterman Yngve. The Impulses Produced by Sensory Nerve Endings: Part 3. Impulses Set up by Touch and Pressure [J]. *The Journal of Physiology*, 1926, 61(4): 465.
- [164] Johansson Roland S., Birznieks Ingvars. First Spikes in Ensembles of Human Tactile Afferents Code Complex Spatial Fingertip Events [J]. *Nature neuroscience*, 2004, 7(2): 170-7.
- [165] Rieke Fred, Warland David, Van Steveninck Rob De Ruyter, et al. *Spikes: exploring the neural code* [M]. MIT press, 1999.
- [166] Park Seongsik, Kim Seijoon, Choe Hyeokjun, et al. Fast and Efficient Information Transmission with Burst Spikes in Deep Spiking Neural Networks [Z]. *Proceedings of the 56th Annual Design Automation Conference 2019*. Las Vegas, NV, USA; Association

- for Computing Machinery. 2019: Article 53.10.1145/3316781.3317822
- [167] Li Yang, Zeng Yi. Efficient and Accurate Conversion of Spiking Neural Network with Burst Spikes; proceedings of the Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, {IJCAI-22}, F 7, 2022 [C]. International Joint Conferences on Artificial Intelligence Organization.
- [168] He Linxuan, Xu Yunhui, He Weihua, et al. Network Model with Internal Complexity Bridges Artificial Intelligence and Neuroscience [J]. Nature Computational Science, 2024: 1-16.
- [169] Gilbert Charles D., Li Wu. Top-down Influences on Visual Processing [J]. Nature reviews neuroscience, 2013, 14(5): 350-63.
- [170] Yin Bojian, Corradi Federico, Bohté Sander M. Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks [J]. Nature Machine Intelligence, 2021, 3(10): 905-13.
- [171] Rao Arjun, Plank Philipp, Wild Andreas, et al. A Long Short-Term Memory for AI Applications in Spike-based Neuromorphic Hardware [J]. Nature Machine Intelligence, 2022, 4(5): 467-79.
- [172] Hebb Donald Olding. The Organization of Behavior: a Neuropsychological Theory [M]. 1949.
- [173] Bi Guo-Qiang, Poo Mu-Ming. Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type [J]. Journal of Neuroscience, 1998, 18(24): 10464-72.
- [174] Imam Nabil, Cleland Thomas A. Rapid online learning and robust recall in a neuromorphic olfactory circuit [J]. Nature Machine Intelligence, 2020, 2(3): 181-91.
- [175] Wu Yujie, Zhao Rong, Zhu Jun, et al. Brain-Inspired Global-Local Learning Incorporated with Neuromorphic Computing [J]. Nature Communications, 2022, 13(1): 1-14.
- [176] Putra Rachmad Vidya Wicaksana, Shafique Muhammad. Q-spinn: A framework for quantizing spiking neural networks; proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), F, 2021 [C]. IEEE.
- [177] Kheradpisheh Saeed Reza, Mirsadeghi Maryam, Masquelier Timothe. BS4NN: Binarized Spiking Neural Networks with Temporal Coding and Learning [J]. Neural Processing Letters, 2022, 54(2): 1255-73.
- [178] Shymyrbay Ayan, Fouda Mohammed E., Eltawil Ahmed. Low Precision Quantization-aware Training in Spiking Neural Networks with Differentiable Quantization Function; proceedings of the International Joint Conference on Neural Networks, F, 2023 [C].
- [179] Wei Wenjie, Liang Yu, Belatreche Ammar, et al. Q-SNNs: Quantized Spiking Neural Networks; proceedings of the Proceedings of the 32nd ACM International Conference on Multimedia, F, 2024 [C].
- [180] Bellec Guillaume, Salaj Darjan, Subramoney Anand, et al. Long Short-Term Memory and Learning-To-Learn in Networks of Spiking Neurons; proceedings of the Advances in Neural Information Processing Systems, F, 2018 [C].
- [181] Deng Lei, Wu Yujie, Hu Yifan, et al. Comprehensive snn compression using admm optimization and activity regularization [J]. IEEE transactions on neural networks and learning systems, 2021, 34(6): 2791-805.
- [182] Chen Yanqi, Yu Zhaofei, Fang Wei, et al. Pruning of Deep Spiking Neural Networks through Gradient Rewiring; proceedings of the International Joint Conference on Artificial Intelligence, F, 2021 [C].
- [183] Chen Yanqi, Yu Zhaofei, Fang Wei, et al. State Transition of Dendritic Spines Improves Learning of Sparse Spiking Neural Networks; proceedings of the International Conference on Machine Learning, F, 2022 [C].
- [184] Kim Youngeun, Li Yuhang, Park Hyoungeob, et al. Exploring Lottery Ticket Hypothesis in Spiking Neural Networks; proceedings of the European Conference on Computer Vision, Cham, F, 2022 [C].
- [185] Chen Yanqi, Ma Zhengyu, Fang Wei, et al. A Unified Framework for Soft Threshold Pruning; proceedings of the International Conference on Learning Representations, F, 2023 [C].
- [186] Neftci Emre O, Pedroni Bruno U, Joshi Siddharth, et al. Stochastic synapses enable efficient brain-inspired

- learning machines [J]. *Frontiers in neuroscience*, 2016, 10: 241.
- [187] Rathin Nitin, Panda Priyadarshini, Roy Kaushik. STDP-based pruning of connections and weight quantization in spiking neural networks for energy-efficient recognition [J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018, 38(4): 668-77.
- [188] Shi Yuhao, Nguyen Leon, Oh Sangheon, et al. A soft-pruning method applied during training of spiking neural networks for in-memory computing applications [J]. *Frontiers in neuroscience*, 2019, 13: 405.
- [189] Qi Yu, Shen Jiangrong, Wang Yueming, et al. Jointly Learning Network Connections and Link Weights in Spiking Neural Networks; proceedings of the Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, {IJCAI-18}, F 7, 2018 [C]. International Joint Conferences on Artificial Intelligence Organization.
- [190] Li Yaxin, Shen Jiangrong, Xu Hongming, et al. Towards efficient deep spiking neural networks construction with spiking activity based pruning [Z]. 2024
- [191] Shi Xinyu, Ding Jianhao, Hao Zecheng, et al. Towards Energy Efficient Spiking Neural Networks: An Unstructured Pruning Framework; proceedings of the The Twelfth International Conference on Learning Representations, F, 2024 [C].
- [192] Bhattacharjee Abhiroop, Kim Youngeun, Moitra Abhishek, et al. Examining the Robustness of Spiking Neural Networks on Non-ideal Memristive Crossbars [Z]. Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design. Boston, MA, USA; Association for Computing Machinery. 2022: Article 1.10.1145/3531437.3539729
- [193] Moro Filippo, Esmanhotto E., Hirtzlin T., et al. Hardware calibrated learning to compensate heterogeneity in analog RRAM-based Spiking Neural Networks; proceedings of the 2022 IEEE International Symposium on Circuits and Systems (ISCAS), F, 2022 [C].
- [194] Christensen Dennis V., Dittmann Regina, Linares-Barranco Bernabe, et al. 2022 roadmap on neuromorphic computing and engineering [J]. *Neuromorphic Computing and Engineering*, 2022, 2(2): 022501.
- [195] Koopman Roel, Yousefzadeh Amirreza, Shahsavari Mahyar, et al. Overcoming the Limitations of Layer Synchronization in Spiking Neural Networks [J]. arXiv preprint arXiv:240805098, 2024.
- [196] Zhou Zhenglong, Firestone Chaz. Humans can decipher adversarial images [J]. *Nature communications*, 2019, 10(1): 1334.
- [197] Zhang Tielin, Cheng Xiang, Jia Shuncheng, et al. A brain-inspired algorithm that mitigates catastrophic forgetting of artificial and spiking neural networks with low computational cost [J]. *Science Advances*, 2023, 9(34): eadi2947.
- [198] Xiao Mingqing, Meng Qingyan, Zhang Zongpeng, et al. Hebbian Learning based Orthogonal Projection for Continual Learning of Spiking Neural Networks; proceedings of the The Twelfth International Conference on Learning Representations, F, 2024 [C].
- [199] Hammouamri Ilyass, Masquelier Timothe, Wilson Dennis George. Mitigating Catastrophic Forgetting in Spiking Neural Networks through Threshold Modulation [J]. *Transactions on Machine Learning Research*, 2022.
- [200] Zhang Tielin, Cheng Xiang, Jia Shuncheng, et al. Self-backpropagation of synaptic modifications elevates the efficiency of spiking and artificial neural networks [J]. *Science advances*, 2021, 7(43): eabh0146.
- [201] Sharmin Saima, Rathin Nitin, Panda Priyadarshini, et al. Inherent Adversarial Robustness of Deep Spiking Neural Networks: Effects of Discrete Input Encoding and Non-linear Activations, Cham, F, 2020 [C]. Springer International Publishing.
- [202] Loshchilov Ilya, Hutter Frank. SGDR: Stochastic Gradient Descent with Warm Restarts; proceedings of the International Conference on Learning Representations, F, 2017 [C].
- [203] Smith Leslie N, Topin Nicholas. Super-convergence: Very fast training of neural networks using large learning rates; proceedings of the Artificial

intelligence and machine learning for multi-domain operations applications, F, 2019 [C]. SPIE.

附录

1 可复现性

本文的实验代码、训练日志可以从如下网址获取,方便读者自行复现或进一步研究:

2 实验细节

2.1 突触操作数统计

突触操作数(SOP)的统计是在整个测试集上完成的,汇报的结果为平均到每个样本的突触操作数.需要注意的是,当突触处理非脉冲的整数类型输入时,例如神经形态数据集中的事件积分得到的帧,或由 SEW 残差连接导致的脉冲之和,突触操作数按照输入整数的数值计算.例如,当输入为脉冲[0,1,0,1]则突触操作数为 2;而输入为非负整数[0,2,0,3]则突触操作数按 5 计算.

2.2 CIFAR 分类实验细节

网络结构:本文使用 Fang 等^[70]的网络结构,记卷积层为 Conv,批量标准化层为 BN,脉冲神经元层为 SN,平均池化层为 AP,全连接层为 FC,则网络结构为:

(Conv-BN-SN)-(Conv-BN-SN)-(Conv-BN-SN)-AP-(Conv-BN-SN)-(Conv-BN-SN)-(Conv-BN-SN)-AP-FC-SN-FC,

其中卷积层的通道数固定为 128.对于静态 CIFAR10 分类任务,使用 2 维卷积,第一个 FC 层输出特征数为 2048;对于序列 CIFAR10 分类任务,使用 1 维卷积,第一个 FC 层输出特征数为 256.两个任务的网络结构中第二个 FC 层输出特征均为 10,与总类别数一致.

数据增强:使用了 Mixup($p=1, \alpha=0.2$)、Cutmix($p=1, \alpha=1$)、随机擦除($p=0.1$)、自动数据增强(Trivial Augment)、标签平滑($p=0.1$)、数据标准化.

训练超参数:在不做出特殊说明的情况下,默认批量大小为 128、训练轮数 256、使用动量大小为 0.9 的 SGD 优化器、学习率 0.1、混合精度训练、周期为训练轮数的余弦退火学习率调节器^[202].对于静态 CIFAR10 分类任务,时间步数为 4;对于序列 CIFAR10 分类任务,时间步数与图像宽度相同,为 32.

对于不同方法,额外调节的超参数通常是通过网格搜索确定的,具体如下:

响应蒸馏:蒸馏损失的强度为 0.001,温度为 4.作为教师的 ANN 是将 SNN 中脉冲神经元换成 ReLU 激活函数,并使用默认超参数训练出来的.

特征蒸馏:蒸馏损失的强度为 0.1,温度为 1.作为教师的

ANN 与响应蒸馏中使用的是同一个 ANN.

Tandem:默认学习率性能较差,使用学习率为 0.001 时效果最好.

PSN 家族:静态 CIFAR10 分类任务使用 PSN,序列 CIFAR10 分类任务使用 $k=4$ 的 Sliding PSN.

Block ALIF:分块大小为 1.使用更大的分块大小,则性能会剧烈下降.

2.3 SHD 分类实验细节

本文使用 Ilyass 等^[100]的开源代码和网络结构.对于神经形态的 SHD 语音数据集,使用 SpikingJelly 框架^[49]中的固定时间间隔积分方式,每 20ms 积分为一帧,产生帧数为 88~126、特征数为 140 的帧.使用 3 层全连接网络,具体结构为:

(FC256-BN-SN-DP)-(FC256-BN-SN-DP)-FC20-LIF,

其中 FC256 表示输出特征数量为 256 的全连接层,FC20 表示输出特征数量为 20 的全连接层,DP 表示 Dropout 层,且丢弃率均为 0.4.SN 表示脉冲神经元层,根据不同方法使用不同的脉冲神经元,而最后输出层固定为 LIF 神经元层,且阈值设置为无穷大,网络的输出是其所有时间步的膜电位.

默认的训练超参数包括,训练 150 轮,对于 LIF 神经元和其他有膜时间常数的神经元均初始化膜时间常数为 1.005,使用 $\alpha=5$ 的 ArcTan 替代函数,使用 Adam 优化器,学习率 0.001,权重衰减 0.00001,使用最大学习率为 0.005、周期为训练轮数的 OneCycle 学习率调节器^[203].

对于不同方法,额外调节的超参数如下:

Sliding PSN:使用 $k=3$.

TEBN:由于实验观测到 IF 神经元表现反而比 LIF 神经元好,故使用 IF 神经元配合 TEBN.

BlockALIF:分块大小为 1.

需要注意的是,TEBN 和 BlockALIF 神经元要求固定时间步数,不能处理变长输入,而本实验中使用的固定时间间隔积分方式会产生长度不固定的输入帧.因而,在实验中将它们的时间步数都设置为帧的最长长度 126,并在计算时将长度不足 126 的输入进行 0 填充,输出再重新去除填充部分.

2.4 Gen1 目标检测实验细节

本文使用 Fan 等^[159]的开源代码和网络结构,使用 Spiking DenseNet121-16 作为检测的骨干网络,首先在神经形态的 NCAR^[161]数据集上预训练分类任务.分类任务的默认超参数为:数据集积分到 5 帧(故时间步数为 5),批量大小为 64,使用 AdamW 优化器,学习率为 0.005,余弦退火学习率调节器(周期和训练轮数一致,最小学习率 0.00001),训练轮数 30 并设置了早停机制(Early Stop,当连续 5 轮训练的最小损失没有改善则退出训练),使用混合精度训练.网络结构为标准的 Spiking DenseNet121-16.

分类任务预训练完成后,卷积层部分权重作为目标检测任务的骨架(backbone)网络的初始权重,并在 Gen1 数据集上进行训练,默认超参数为:数据集积分到 5 帧(故时间步数为 5),训练轮数为 50,使用 AdamW 优化器,学习率 0.001,权重衰减 0.0001,余弦退火学习率调节器(周期和训练轮数一致,最小学习率 0.00001),使用混合精度训练.需要注意的是,BlockALIF 神经元的参数是逐神经元的,而该任务中,预训练分类任务和其后的目标检测任务,输入尺寸不一致.因而加载预训练权重时,卷积层可以正常加载,而 BlockALIF 神经元则由于参数尺寸不匹配无法加载,参数按照默认方式初始化.

对于不同方法,需要额外调节的超参数如下:

CLIF 神经元: 学习率 0.0001.

Sliding PSN: 使用 $k = 3$.

TEBN: 实验发现 LIF 神经元好于 IF 神经元,故使用 LIF 神经元配合 TEBN.

OSR: 学习率 0.0001.

BlockALIF: 学习率 0.0001.



Wei Fang received his B.S. degree from Department of Automation, Tsinghua University, China in 2019 and Ph.D. degree from School of Computer Science, Peking University in 2024. He is currently the Research Assistant Professor in School of Electronic and Computer Engineering,

Shenzhen Graduate School, Peking University. His research interests include the learning and network structure of Spiking Neural Networks. He has published 13 articles in journals such as Science Advances/Nature Communications, Neural Networks and conferences such as NeurIPS, ICML, ICLR, ICCV, IJCAI.

Yonghong Tian is currently dean of school of electronic and computer engineering, a Boya Distinguished Professor with the Department of Computer Science and Technology, Peking University, China, and is also the deputy director of Artificial Intelligence Research Center, PengCheng Laboratory, Shenzhen, China. His research interests include neuromorphic vision, brain-inspired computation and multimedia big data. He has co-authored over 200 technical articles in refereed journals such as Science Advances/Nature Communications/Scientific Data, IEEE TPAMI, TNNLS, TIP, TMM, TCSVT, TKDE, TPDS, TCYB, ACM CSUR, TOIS, TOMM and conferences such as NeurIPS, ICML, ICLR, CVPR, ICCV, ECCV, AAAI, IJCAI.

Background

Artificial Neural Networks (ANNs) monopolize the current Artificial Intelligence (AI) systems for their higher performance than other computational models. However, the floating activation and intensive computation of ANNs cause high energy consumption. Spiking Neural Networks(SNNs), the third generation of neural network models, are the potential alternatives of ANNs for up to hundreds of times of power efficiency. Modules in SNNs communicate by asynchronous spikes as the human brain, which introduces sparse activations, event-driven computations, and consequently low power consumption.

However, there is still a huge performance gap between SNNs and ANNs, which restricts the practical values of SNNs. Complex temporal dynamics and non-differentiable firing mechanisms make it challenging to design learning methods for SNNs. Traditional bio-inspired learning methods such as the Hebbian rule and the Spike Timing Dependent Plasticity rule are unsupervised algorithms and can only solve simple learning tasks such as classifying the MNIST dataset. Primitive supervised learning methods including SpikeProp, Tempotron, and ReSuMe are limited to train SNNs with a single layer or single spike. Recently, deep learning methods have been introduced into SNNs and overwhelmed previous algorithms, growing into the booming spiking deep learning research community.

The ANN to SNN conversion and surrogate learning methods are two mainstream methods in spiking deep learning. The former is based on rate coding and approximates the activations in ANNs by firing rates in SNNs. However, it requires the SNNs to run many time steps and causes high energy consumption and long latency. It cannot solve temporal tasks because the time dimension is already occupied to represent rates. On the contrary, the surrogate learning methods are more flexible. It re-defines the gradient of the discrete Heaviside function used in spike generation by that of a smooth surrogate function and then is capable of training SNNs directly. It is not based on rate coding and can fully utilize neural dynamics to process temporal tasks such as classifying the neuromorphic data. It is not restricted to rate coding and requires much fewer time steps than the conversion methods.

This survey reviews the latest research advancements of the surrogate learning methods in spiking deep learning. The basic concepts, components, and benchmarks of SNNs are first

introduced. Then learning methods are systemically divided into different categories and illustrated. A comprehensive experiment is conducted to compare these methods fairly. The advantages and shortcomings of each category are then presented. Lastly, the future research directions are discussed.

This work is partially supported by the National Natural Science Foundation of China under contracts No.62425101, No.62332002, No.62027804, and No.62088102.