

FF-STDP: 一种基于前向-前向与STDP融合的 低功耗在线学习SNN加速器

徐长卿^{1,2*}, 莫桂棋¹, 刘毅², 李裕科¹, 陈梓鑫¹, 杨银堂²

(1. 西安电子科技大学广州研究院, 广东广州 510555; 2. 西安电子科技大学集成电路学部, 陕西西安 710071)

摘 要: 脉冲神经网络(Spiking Neural Network, SNN)作为第三代人工神经网络(Artificial Neural Network, ANN),通过离散脉冲事件进行信息编码与传递,具有事件驱动、时空信息融合以及低功耗计算等特点,在边缘智能、类脑计算和神经形态芯片等领域展现出广阔的应用前景。随着智能终端和物联网设备的快速发展,设备对环境自适应和持续学习能力的需求不断提升,在线学习逐渐成为边缘智能系统的重要研究方向。然而,现有在线学习方法主要分为两类:一类基于反向传播(BackPropagation, BP)机制,能够利用全局监督信息获得较高精度,但需存储大量中间状态和误差信息,导致计算复杂度、存储开销和能耗显著增加;另一类基于脉冲时序依赖可塑性(Spike-Timing-Dependent Plasticity, STDP)机制,具有局部更新和事件驱动等特点,易于硬件实现,但由于缺乏有效的监督信息,其学习能力和任务性能通常受到限制。针对上述问题,本文提出一种融合前向-前向(Forward-Forward, FF)学习范式与STDP机制的在线学习算法FF-STDP。该算法借鉴FF逐层局部优化思想,将教师信号引入脉冲学习过程,通过构建基于神经元脉冲发放率的局部优度(Goodness)评价机制,实现无需误差BP的半监督学习。在保留STDP局部权值更新特性的同时,引入FF提供的监督信息,使学习过程兼具较强的任务驱动能力与硬件友好性。针对传统FF算法在SNN中计算复杂、硬件实现成本较高的问题,本文采用脉冲计数近似梯度计算,并结合局部导数近似策略简化权值更新过程,使突触更新仅依赖于前后神经元的脉冲统计信息,显著降低计算复杂度和存储开销。基于所提出的FF-STDP算法,本文设计了一种面向边缘智能应用的事件驱动SNN加速器架构。该架构采用地址事件表示(Address Event Representation, AER)通信机制,结合并行突触后神经元处理、双口存储器和流水线执行策略,实现推理与学习的高效协同执行。为进一步降低硬件复杂度,设计了基于查找表(LookUp Table, LUT)的权值更新单元,实现低开销、高吞吐率的在线学习过程。为验证算法与硬件架构的有效性,本文在Xilinx ZYNQ-7010平台上完成系统实现,并采用混合国家标准与技术研究院数据库(Modified National Institute of Standards and Technology database, MNIST)和时尚-MNIST(Fashion-MNIST)数据集进行实验评估。实验结果表明:所提出的FF-STDP算法在MNIST和Fashion-MNIST数据集上的分类准确率分别达到90.43%和80.53%。在100 MHz工作频率下,加速器核心动态功耗仅为15 mW,推理速度达每秒1 887帧。与现有BP或传统STDP在线学习SNN硬件相比,所提方法在保持竞争性分类性能的同时,降低了硬件资源消耗和功耗开销,展现出良好的能效优势和边缘部署潜力。

关键词: 脉冲神经网络(SNN);前向-前向(FF)学习;在线学习;现场可编程门阵列(FPGA);硬件加速器;软硬件协同设计

基金项目: 广东省自然科学基金(No.2025A1515012058);陕西省自然科学基金(No.2025JCYBQN822);抗辐射应用技术创新中心创新基金(No.KFZC2025010203);深圳市科技重大专项项目(No.KJZD20240903100506009);国家自然科学基金(No.62090043, No.61934006);国家重点研发计划(No.2022YFB4401301)

中图分类号: TP183;TN791

文献标识码: A

文章编号: 0372-2112(2026)06-2443-11

电子学报URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20260284

FF-STDP: A Forward-Forward STDP-Based Energy-Efficient SNN Accelerator with On-Chip Learning

XU Changqing^{1,2*}, MO Guiqi¹, LIU Yi², LI Yuke¹, CHEN Zixin¹, YANG Yintang²

(1. Guangzhou Institute of Technology, Xidian University, Guangzhou, Guangdong 510555, China;

2. School of Microelectronics, Xidian University, Xi'an, Shaanxi 710071, China)

Abstract: Spiking neural networks (SNNs), regarded as the third generation of artificial neural networks (ANNs), encode and transmit information through discrete spike events. Owing to their event-driven nature, spatiotemporal information processing capability, and low-power characteristics, SNNs have shown great potential in edge intelligence, brain-inspired

computing, and neuromorphic hardware. With the rapid development of intelligent terminals and internet-of-things devices, the demand for environmental adaptation and continual learning is increasing, making online learning an important research direction for edge intelligence systems. Existing online learning methods can generally be categorized into two groups. Backpropagation (BP)-based methods achieve high accuracy by utilizing global supervisory information, but require the storage of a large number of intermediate states and error signals, resulting in considerable computational complexity, memory overhead, and power consumption. In contrast, spike-timing-dependent plasticity (STDP)-based methods adopt local and event-driven synaptic updates, making them hardware-friendly; however, their learning capability and task performance are often limited due to the lack of effective supervisory information. To address these challenges, this paper proposes an on-line learning algorithm, termed forward-forward (FF)-STDP, which combines the FF learning paradigm with the STDP mechanism. Inspired by the layer-wise local optimization strategy of FF, the proposed algorithm introduces teacher signals into the spike-based learning process and constructs a local goodness metric based on neuronal firing rates, thereby enabling supervised learning without error backpropagation. By preserving the local weight-update property of STDP while incorporating supervisory information provided by FF, the proposed method achieves both task-oriented learning capability and hardware efficiency. Furthermore, to reduce the computational complexity and hardware implementation cost of conventional FF algorithms in SNNs, spike-count-based gradient approximation and local derivative approximation are adopted, such that synaptic updates depend only on the spike statistics of pre-and post-synaptic neurons, significantly reducing computational and memory overhead. Based on the proposed FF-STDP algorithm, an event-driven SNN accelerator architecture for edge intelligence applications is developed. The architecture adopts an address event representation (AER) communication scheme and integrates parallel post-synaptic neuron processing, dual-port memory, and pipelined execution strategies to enable efficient collaborative execution of inference and learning. To further reduce hardware complexity, a lookup-table (LUT)-based weight update unit is designed, achieving low-overhead and high-throughput online learning. To validate the effectiveness of the proposed algorithm and hardware architecture, the system is implemented on a Xilinx ZYNQ-7010 platform and evaluated using the modified national institute of standards and technology database (MNIST) and Fashion-MNIST datasets. Experimental results demonstrate that the proposed FF-STDP algorithm achieves classification accuracies of 90.43% and 80.53% on MNIST and Fashion-MNIST, respectively. Operating at 100 MHz, the accelerator core consumes only 15 mW of dynamic power and achieves an inference throughput of 1 887 frames per second. Compared with existing BP-based and conventional STDP-based online learning SNN hardware, the proposed approach significantly reduces hardware resource consumption and power overhead while maintaining competitive classification performance, demonstrating excellent energy efficiency and potential for edge deployment.

Keywords: spiking neural network (SNN); forward-forward (FF) learning; online learning; field programmable gate array (FPGA); hardware accelerator; hardware-software co-design

Foundation Item(s): Guangdong Basic and Applied Basic Research Foundation (No.2025A1515012058); Natural Science Basic Research Program of Shaanxi (No.2025JCYBQN822); Innovation Center of Radiation Application (No. KFZC2025010203); Shenzhen Science and Technology Program (No. KJZD20240903100506009); National Natural Science Foundation of China (No.62090043, No.61934006); National Key Research and Development Program of China (No.2022YFB4401301)

0 引言

脉冲神经网络(Spiking Neural Network, SNN)受生物神经系统启发,被认为是具有重要发展潜力的第三代神经网络模型^[1]。不同于传统人工神经网络(Artificial Neural Network, ANN), SNN通过离散脉冲事件进行信息编码,并利用时间动态特性完成计算。这种事件驱动的处理范式天然具有稀疏激活和较低开关翻转次数的特点,使其在低功耗及类脑硬件实现方面具备显著优势^[2]。随着智能边缘设备的广泛部署,在线学习能力成为设备适应新数据分布的关键技术。大量研究表明:SNN凭借其持续学习与增量学习特

性,成为实现在线学习最具潜力的模型之一^[3]。近年来,部分采用专用集成电路实现并支持在线学习的类脑处理器已展现出较高的能量效率^[4];同时,基于现场可编程门阵列(Field Programmable Gate Array, FPGA)的实现方案也受到广泛关注。FPGA作为一种灵活、可重构的平台,有利于快速设计空间探索以及算法与硬件的协同优化^[5-11]。

现有SNN在线学习算法大致可分为两类。第一类基于反向传播(BackPropagation, BP)或受BP启发的机制,在训练过程中需进行逐层误差传播^[12]。为提升硬件友好性与执行效率,相关研究通常通过算法简化或模型压缩等方式降低计算复杂度并提高并行

度,如文献[13-14]提出的方法。尽管此类方法通常能够取得较优性能,但随着网络层数增加,其在硬件资源占用、存储开销、训练与推理时延以及功耗方面的代价显著上升^[9,11,15]。第二类方法基于脉冲时序依赖可塑性(Spike-Timing-Dependent Plasticity, STDP),通过输入与输出脉冲之间的时间相关性实现局部突触权值更新。由于其更新规则具备良好的局部性与事件驱动特性,该类算法在硬件实现上具有天然优势,并已在低功耗与低存储开销方向上得到充分优化。然而,与BP类方法相比,基于STDP的算法在推理速度、学习能力及任务性能方面通常仍存在一定差距^[4,6-7]。

近年来,前向-前向(Forward-Forward, FF)传播算法因其具备局部学习特性与单向计算流结构,被认为是一种硬件友好的学习方法^[16]。针对BP类方法与STDP类方法存在的上述局限性,本文受FF与STDP机制启发,提出一种新型在线学习算法——FF-STDP。基于该算法,在FPGA平台上实现了一种在线训练处理器,在保持片上竞争性精度的同时,显著降低了硬件资源占用与功耗。本文的主要贡献如下:

(1) 提出一种融合 Forward-Forward 范式与 STDP 机制的 FF-STDP 在线学习算法。通过引入教师信号与局部优化学习规则,实现完全局部化的逐层学习,消除误差 BP 过程及中间变量存储需求,从而具备良好的硬件实现特性。

(2) 基于所提 FF-STDP 算法,设计了一种面向边缘智能应用的事件驱动、高并行度 SNN 处理核心。

该架构在保持与现有先进方法相当精度的同时,实现了较低的资源利用率与功耗水平。

(3) 通过 FPGA 原型系统验证,实验结果表明:所提 FF-STDP 方法在分类精度方面具有竞争力,并在硬件开销与功耗效率方面达到先进水平,验证了其在资源受限边缘场景中的应用潜力。

1 算法模型

1.1 FF学习算法

不同于传统 BP 算法,FF 学习算法通过两次连续前向传播替代标准的前向-反向训练流程,从而消除了误差 BP 的需求^[16]。该范式避免了为 BP 而存储大规模计算图及中间状态信息,这对于引入额外时间维度的 SNN 而言尤为重要。如图 1 所示,FF 算法通过分别向网络输入正样本与负样本,实现对比学习机制。各层以相对独立的方式进行训练,层间仅传递激活值信息。每一层的学习目标是在正样本输入时增强神经元响应强度,而在负样本输入时抑制其激活水平。对应的第 l 层损失函数 $\mathcal{L}^{(l)}$ 定义如下:

$$G_{\text{ANN}}^{(l)} = \frac{1}{N} \sum_{i=0}^N (h_i^{(l)})^2 \quad (1)$$

$$\mathcal{L}^{(l)} = \ln(1 + e^{\theta - G_+^{(l)}}) + \ln(1 + e^{(G_-^{(l)} - \theta)}) \quad (2)$$

其中: $G_{\text{ANN}}^{(l)}$ 为 ANN 模型的优度值; N 为神经元的数量; $h_i^{(l)}$ 为第 l 层中神经元 i 的激活; $G_+^{(l)}$ 和 $G_-^{(l)}$ 对应于正样本和负样本的优度; θ 为损失函数的超参数阈值,用于将优度值限制在一定范围内。

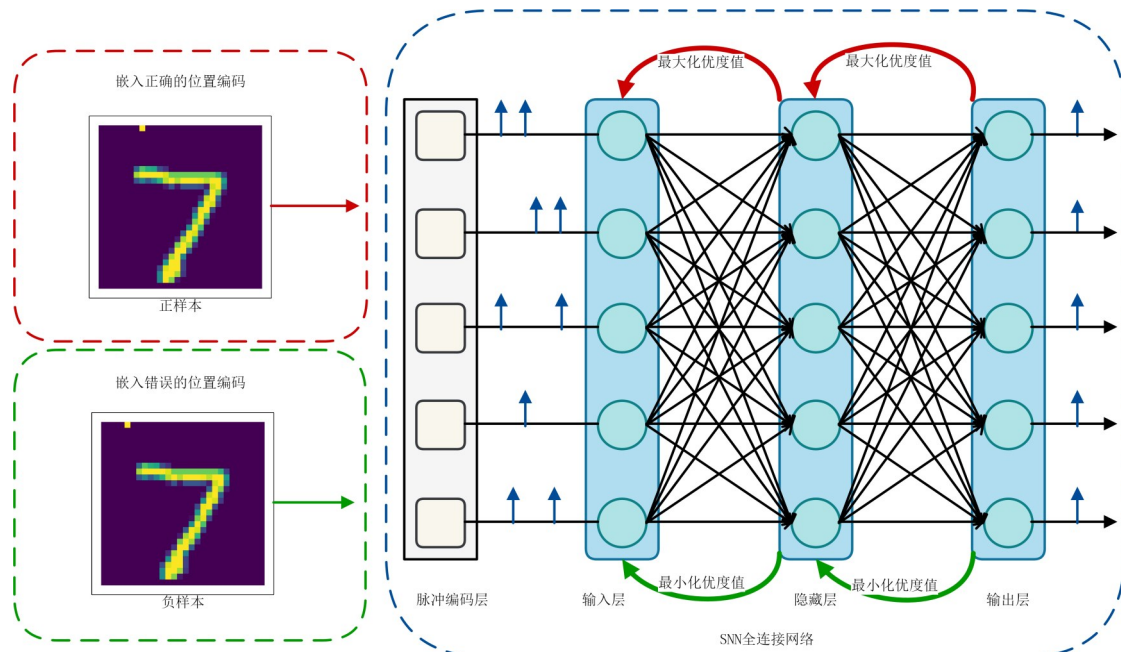


图1 FF算法示意图

Figure 1 Forward-Forward algorithm schematic

1.2 提出 FF-STDP 算法

FF 算法与 SNN 相结合的关键挑战之一在于如何定义合适的优度值函数。本文提出采用神经元脉冲发放率平方和来刻画神经元活动强度,从而构建适用于在线学习且具有良好硬件实现特性的评价指标。由于在固定时间步长条件下,脉冲发放率取值为离散且有界数值,这一特性有利于在数字电路中进行数值表示与计算,从而降低实现复杂度。然而,在基于 SpikeJelly 平台^[17]的实验仿真中发现,若直接采用发放率平方和作为优度值度量,则在训练过程中会导致梯度分布不均衡,进而影响模型收敛稳定性。为缓解该问题,本文对优度值乘上时间步总数进行缩放处理,使梯度分布更加均匀,从而提升训练稳定性。改进后的优度值表达式为

$$G_{\text{SNN}}^{(l)} = \frac{T}{N} \sum_{i=0}^N g_i = \frac{T}{N} \sum_{i=0}^N f_i^2 = \frac{T}{N} \sum_{i=0}^N \left(\frac{1}{T} \sum_{t=0}^T (s_i^{(l)}[t]) \right)^2 \quad (3)$$

其中, T 为总时间步长, g_i 和 f_i 分别为单个输出神经元 i 的优度和放电率, $s_i^{(l)}[T]$ 为第 l 层中 SNN 神经元 i 在 T 时刻的尖峰。由于 FF 范式具有逐层独立学习特性,因此产生的权重梯度计算图等效于单层 BP。本文采用了具有硬复位的集成与发射 (Integrate Fire, IF) 模型,膜电位的更新式为

$$V_i[t] = V_i[t-1] + \sum_{j=0}^P s_j[t] \omega_{ij} \quad (4)$$

其中, $V_i[t]$ 为时间步长 t 处神经元 i 的膜电压, ω_{ij} 为突触前 j 和突触后 i 之间的权重, P 为突触前的数量。突触权重相对于损失函数的偏导数可以推导为

$$\frac{\partial \mathcal{L}}{\partial \omega_{ij}} = \frac{\partial \mathcal{L}}{\partial G} \frac{\partial G}{\partial s_i} \frac{\partial s_i}{\partial \omega_{ij}} = -\xi S(\xi(\theta - G)) \frac{2f_i}{N} \frac{\partial s_i}{\partial \omega_{ij}} \quad (5)$$

其中, ξ 为教师信号,当输入为正样本时 ξ 为 1,当输入为负样本时 ξ 为 -1, $S(x) = \frac{1}{1 + \exp(-x)}$ 。当给出样本时,

突触权重相对于尖峰 $\frac{\partial s_i}{\partial \omega_{ij}}$ 的偏导数可以推导为

$$\frac{\partial s_i}{\partial \omega_{ij}} = \sum_{t=0}^T \frac{\partial s_i(t)}{\partial \omega_{ij}} = \sum_{t=0}^T \sum_{\tau=0}^t s_j[\tau] \prod_{a=\tau}^t \delta(V_i[a]) \quad (6)$$

其中, $\delta(x)$ 为尖峰偏导数相对于膜电压 $\frac{\partial s_i}{\partial V_i}$ 的替代梯度函数。根据式 (5) 和式 (6), 突触权重相对于损失函数的偏导数可以表示为

$$\frac{\partial \mathcal{L}}{\partial \omega_{ij}} = -\xi S(\xi(\theta - G)) \frac{2f_i}{N} \left(\sum_{t=0}^T \sum_{\tau=0}^t s_j[\tau] \prod_{a=\tau}^t \delta(V_i[a]) \right) \quad (7)$$

受 BP-STDP 方法^[18]的启发,本文对权值更新规则进行了进一步简化。具体而言,在式 (7) 中去除了替代梯度形式的输入脉冲累积项,并以输入脉冲计数对其进行近似,从而得到式 (8) 所示的简化权值更新

表达式。同时,在优度值评估方面,不再采用逐层统计形式,而改用基于单神经元的局部优度值度量,并引入教师信号以调制权值更新策略,为

$$\Delta \omega_{ij} \propto -\frac{\partial \mathcal{L}}{\partial \omega_{ij}} \rightarrow \Delta \omega_{ij} \approx \xi \alpha S(\xi(g_i - \theta)) 2f_i \sum_{t=0}^T s_j[t] \quad (8)$$

其中, α 为学习率。通过在时间域与空间域同时采用局部导数近似,所提方法在显著简化学习动态过程的同时,有效降低了存储访问需求。具体而言,每次突触权值更新仅依赖于前、后神经元的脉冲计数信息,从而使学习规则具备较高的计算效率,并更加适用于硬件实现。

2 硬件框架

2.1 总体框架

如图 2 所示,所提 SNN 加速器的整体核心结构以事件驱动表示 (Address-Event Representation, AER)^[4],能够实现最多 $N_{\text{in}} = 784$ 个输入神经元与 $N_{\text{out}} = 256$ 个输出神经元的全连接层。输入与输出之间设置双先入先出 (First-In First-Out, FIFO) 队列以保证事件的顺畅传递。数据流以 AER 事件开始,经调度器传输,控制器协调神经元核心与突触核心执行不同的工作流程。每个核心包含最多 $M = 8$ 个执行单元,以支持并行加速。脉冲发放连接至优度值计算模块进行优度累加,同时被编码为输出 AER 事件。

特别地,设计中使用三类 RAM 来存储关键的神经元状态变量和突触权值参数,以支持灵活开发。神经元核心与突触核心共同管理 RAM 的读写操作。内存使用情况将在 2.2 节与 2.3 节中详细说明。通过时分复用,核心依次完成膜电位累加、神经元激活、突触权值更新以及 AER 生成。值得注意的是,所有模块均参数化设计,以实现灵活配置与扩展。

2.2 神经元核心设计

由于 IF 神经元的机制与 ReLU 激活高度等效^[19],而 ReLU 已在 ANN 的 FF 算法中应用^[16],本文采用 IF 神经元模型实现神经元核心。如图 3 所示,每个 IF 神经元从多个突触权重 RAM 中访问 8 bit 权值,从突触后神经元 RAM 中读取 16 bit 神经元状态,总计内存开销分别为 200 KB 与 512 B。具体地,一个 16 bit 的神经元状态由 12 bit 膜电位和 4 bit 脉冲累积组成。IF 神经元由控制器提供的事件类型驱动,其时序如图 4 所示。定义了三种事件类型:神经元激活、时间步和推理,构成单个输入样本的完整处理流程。

神经元激活:由 AER 触发,输入端的前突触神经元将访问所有后突触神经元,通过累加突触权值更新膜电位,该过程包含三个阶段。使用双端口 RAM 实现流水线以进一步提升吞吐量,如图 4 所示。由于后

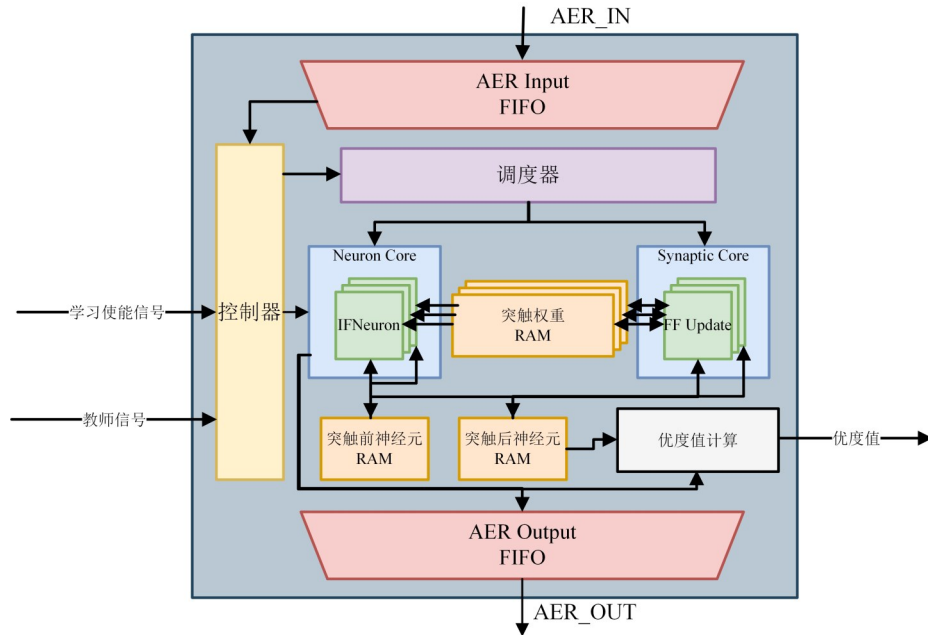


图2 硬件框架示意图

Figure 2 Hardware overall

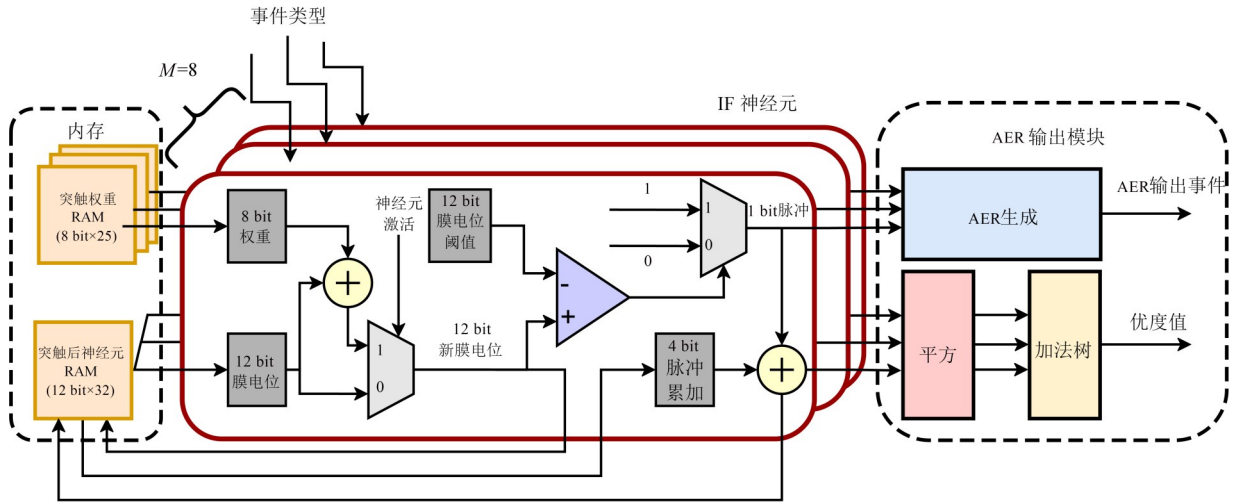


图3 神经元核心单元的设计示意图

Figure 3 The design of neuron core unit

突触神经元单元的并行设计,每个事件仅需 $\frac{N_{out}}{M} + 2 = 34$ 个时钟周期处理完毕。

时间步: 在一个时间步内所有输入激活事件处理完成后,神经元按核心顺序比较膜电位 V_m 与阈值 V_{th} ,生成脉冲或保持状态,其处理周期与激活事件相同。同时,输出脉冲被记录用于推理与学习。

推理: 最终,突触核心遍历所有突触权值,根据前后突触状态更新权值,同时计算优度值并重置脉冲计数器。整个过程共需 $\frac{N_{in} N_{out}}{M} + 4 = 25\,092$ 个时

钟周期。

综上所述,基于FF-STDP算法的处理器核心支持并行学习与推理,无需额外的误差计算延迟或大量中间变量缓冲。此外,采用双端口分区RAM的流水线架构有效隐藏了数据访问开销,进一步提升整体吞吐量。

2.3 FF-STDP学习单元设计

FF-STDP更新单元电路如图5所示。根据式(8),权值梯度依赖于前突触和后突触的脉冲计数,而其解析表达式涉及相对复杂的非线性Sigmoid函数。为解决此问题,所有可能的梯度值通过实验仿真预先计

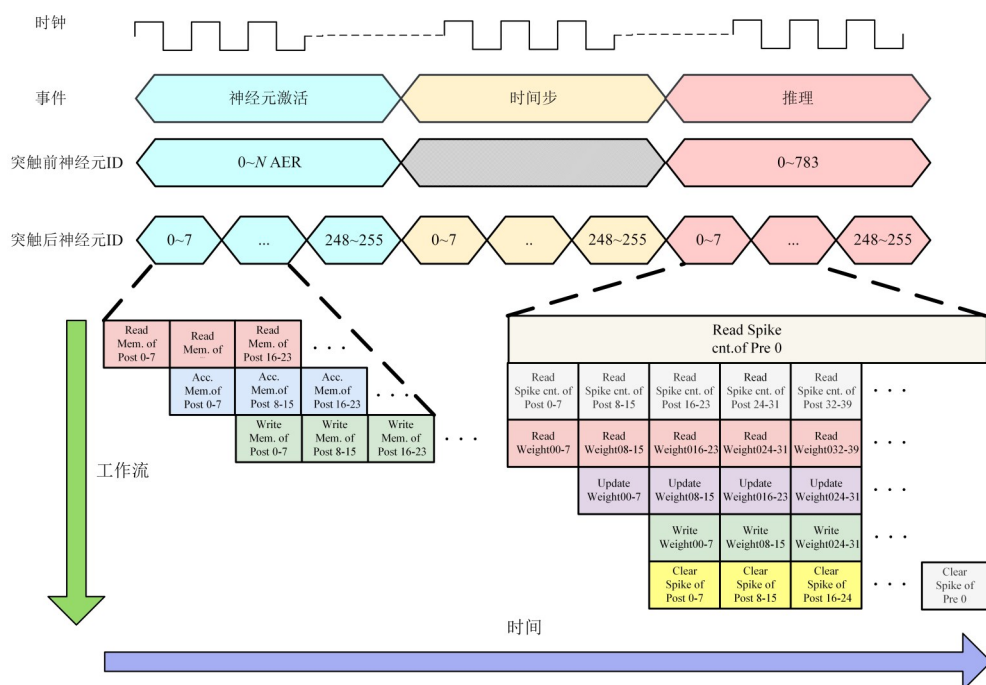


图4 一次单样本推理学习期间事件和工作流的时序图

Figure 4 Sequence diagram of events and workflow

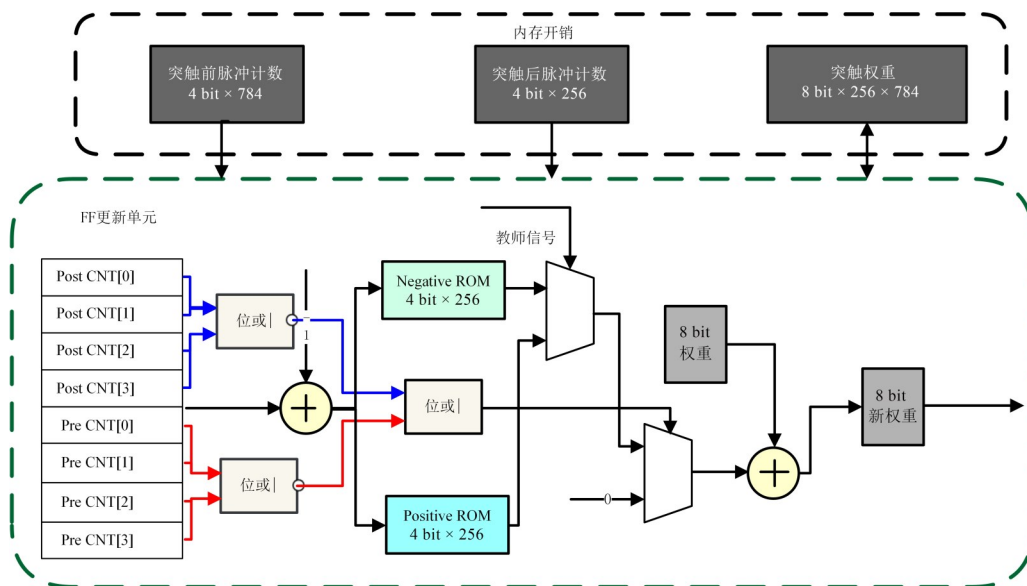


图5 FF-STDP更新单元电路

Figure 5 Circuit of FF-STDP update unit

算,并存储在基ROM的查找表(LookUp Table, LUT)中, LUT通过脉冲计数的编码表示访问。如图5所示,低位对应后突触脉冲计数减一,高位对应前突触脉冲计数。当任一脉冲计数为零(如蓝色和红色箭头所示)时,权重更新被偏置为零,不执行更新。外部教师信号用于选择对应的正负梯度值。得益于该设计,学习单元高效简洁,逻辑资源开销最小。以并行度 $M=8$ 计算,权重更新的总内存开销为 $256M+392+128=2568$ 。

3 软件仿真实验

本节展示FF-STDP算法在软件仿真实验结果。为了检验式(8)的合理性,在仿真过程中,对比分析了采用基于式(7)的自动微分方法计算梯度,以及基于式(8)的脉冲计数近似方法计算梯度,在负样本与正样本训练过程中,分别绘制了两个梯度向量的余弦相似度变化曲线,如图6所示,余弦相似度衡量了两个向量的方向相似性,越接近1代表方向

越一致。在训练初期,由于FF算法的逐层训练特性,深层的学习方向存在一定的迟滞性,但随着迭代轮次的增加,基于FF-STDP各层的梯度方向与自动微分的梯度方向保持高度一致,以确保模型收敛成功。

表1展示了基于FF-STDP算法所训练的SNN模

型在不同数据集软件仿真的实验结果,均为3层网络,可见本文提出的算法均在不同数据集上成功收敛,并在准确度上展现出一定的竞争力。此外,在基于脉冲频率评估优度值的FF算法收敛下,SNN模型的整体网络的脉冲发放频率保持在0.1~0.25,仍具备较好的稀疏度特性。

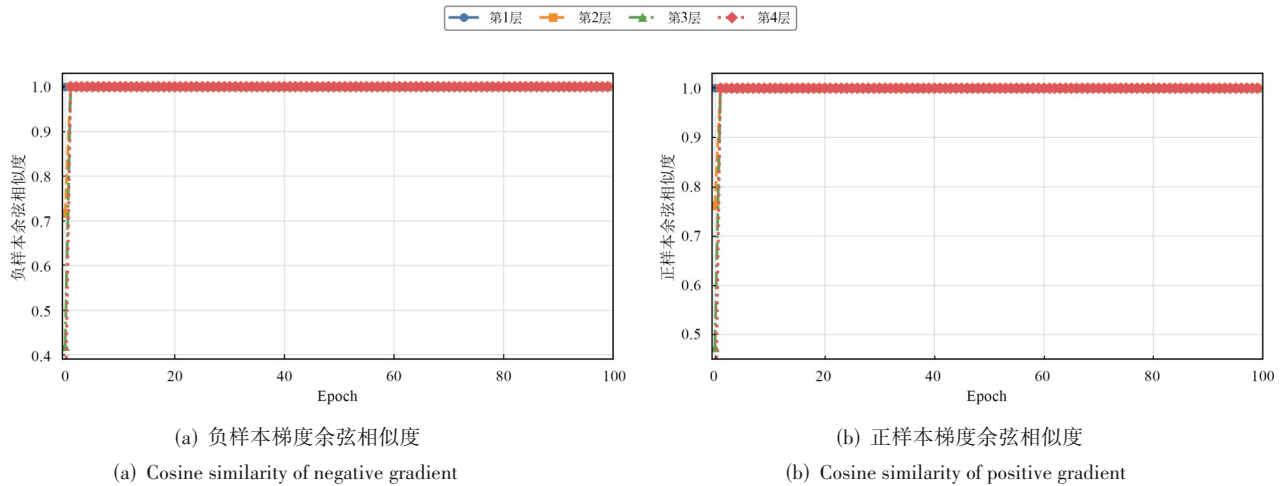


图6 近似梯度与自动微分梯度余弦相似度随Epoch变化曲线图

Figure 6 Cosine similarity between approximate gradient and autograd gradient vs. epoch

表1 软件仿真实验数据

Table 1 Experimental data of software simulation

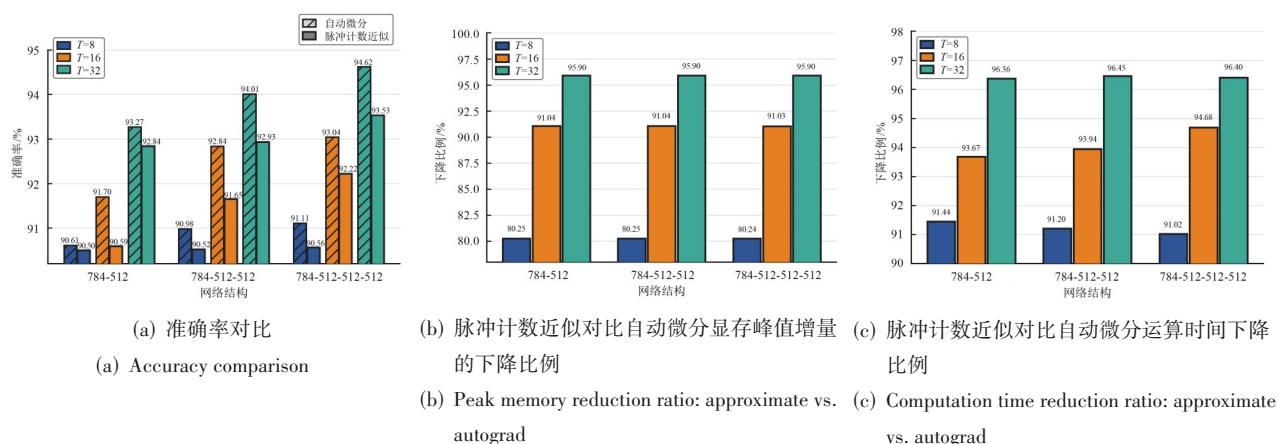
数据集	网络结构	损失阈值	时间步	epoch	发放频率	准确率/%
MNIST ^[20]	784-512-512-512	0.25	32	150	0.173	93.20
FashionMNIST ^[21]	784-512-512-512	0.60	16	200	0.233	82.37
NMNIST ^[22]	2312-1024-512-256	0.25	32	200	0.178	88.42
DVS128-Gesture ^[23]	32768-1024-512*	2.50	64	400	0.119	72.71
CIFAR10 ^[24]	3072-1024-1024-1024	0.90	64	500	0.108	45.31

注:*为依据文献[25]的复现结果。

为了进一步研究本文所提出的FF-STDP算法对准确率、内存开销、计算效率的影响,对比采用自动微分的方法,开展了不同网络结构、不同时间步下的消融实验。如图7所示,本文统计了相同实验超参数设置下,各组别的模型准确率、梯度计算时的平均显存峰值增量以及平均计算时间,数据集为MNIST。如图7(a)所示,对比自动微分,基于式(8)脉冲计数近似的方法在准确度上带来了0.1%~1%的损失;但如图7(b)和图7(c)所示,随着时间步步数的增加,FF-STDP对于内存开销以及计算时间的减少效果越明显,下降比例至多超90%。尽管本文所提出的FF-STDP硬件友好算法对模型的准确度造成了精度损失,却大幅降低了计算的复杂度,在内存开销与计算效率上带来了可观的效益,尤其是随着时间步步数的扩大,这对于具有时间维度的SNN模型尤为重要。

4 FPGA部署与实验结果

本文提出的设计在Xilinx-ZYNQ-7010器件上实现,以100 MHz时钟频率验证本文算法的性能。实验使用混合国家标准与技术研究院数据库(Modified National Institute of Standards and Technology database, MNIST)^[20]和时尚-MNIST(Fashion-MNIST)^[21]数据集,每个数据集包含50 000个训练样本和10 000个测试样本,用于评估模型性能。所有样本均采用频率编码方法并存储在SD卡中。如图8(a)所示,本文利用处理器系统(Processor System, PS)构建评估系统,并通过UART总线将结果显示在PC上位机上。PS充当激励源,CPU将SD卡中的输入数据写入双倍数据率同步动态随机存取存储器(Double Data Rate synchronous dynamic random access memory, DDR)中,由逻辑系统(Processor Logic, PL)端的高速接口总线(Ad-



注:下降比例计算方式为: $100\% \times (1 - \text{脉冲计数近似} / \text{自动微分})$ 。

图7 不同网络结构与时间步下的消融实验结果

Figure 7 Ablation experiment results under different network structures and time steps

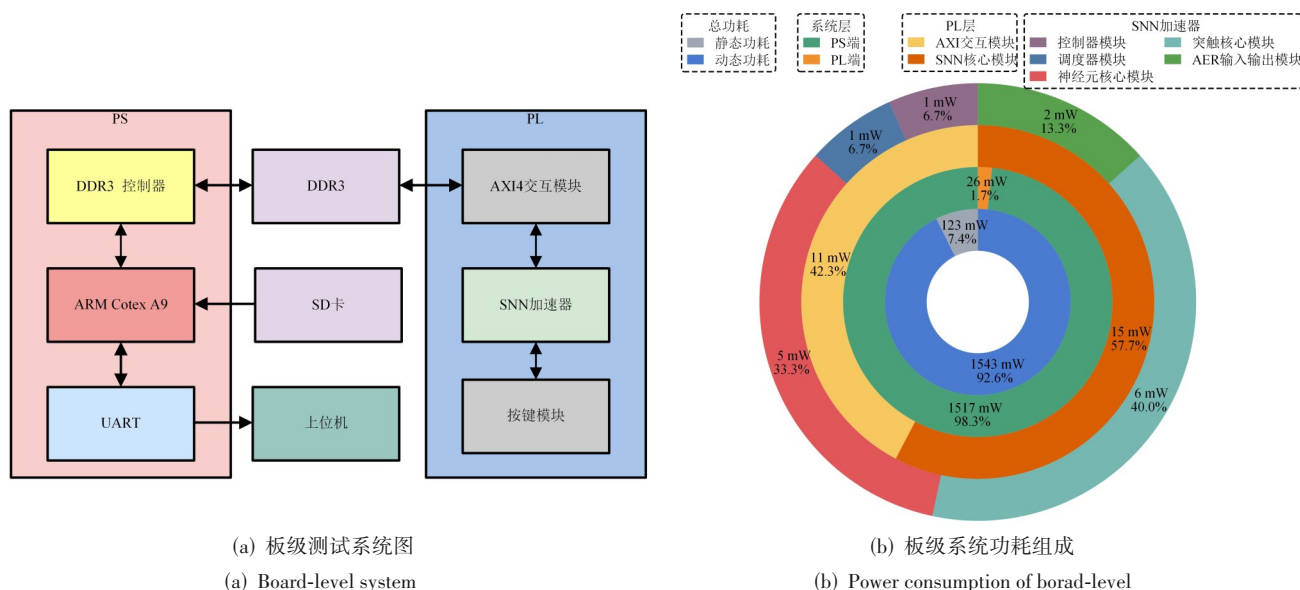


图8 FPGA板级部署实验图

Figure 8 FPGA board-level deployment experiment figure

vanced eXtensible Interface, AXI)交互模块发起读写请求,并向处理器核心发送脉冲信号。实验中在资源受限硬件上部署了单层全连接网络,并从仿真波形和综合报告中获取资源利用率、功耗和推理吞吐量等指标。实验参数设置如下:膜电位阈值 V_{th} 为 1.2,学习率 α 为 0.015 625,损失阈值 θ 分别为 MNIST 的 0.25 和 Fashion-MNIST 的 0.6,总时间步 T 分别为 MNIST 的 8 和 Fashion-MNIST 的 16。图 8(b)展示了本实验板级系统功耗组成分析,数据由接受 MNIST 数据集在真实仿真场景下,导出波形文件经 Vivado 功耗分析得到,整体实验板功耗约为 1.6 W。板级静态功耗为 123 mW,动态功耗中 PS 端占主要成分,PL 端中的 AXI 交互模块负责搬运所设计 SNN 核心的输入输出结果,本设

计的硬件动态功耗为 15 mW,其中神经元核心模块与突触核心模块包含片上存储单元,访存功耗占据主要功耗开销。

如表 2 所示,本文选取了若干基于 BP 或 STDP 方法的最新研究作为对照方法,算法分为有监督和无监督两种。实验结果显示,本文的设计在 MNIST 上实现了峰值测试准确率 90.43%,在 Fashion-MNIST 上为 80.53%。本文提出的 FF-STDP 属于无监督算法,一般而言有监督算法准确度优于无监督算法,由于 FF 算法缺乏全局误差信息,相较于 BP 算法会造成一定的精度损失^[26],因此本文方法的准确率略低于部分 BP 方法^[9, 11]。然而,在逻辑资源和功耗方面,本文的设计具有明显优势,这对于边缘设备尤为关键。尤其是推理与学习可以同时进行,

表 2 其他基于FPGA的在线学习SNN硬件实现的对比工作

Table 2 Other comparative works on FPGA-based online learning SNN hardware implementation

工作	FPGA 平台	Slice LUTs	Slice LUTs/权重	Slice Registers	Slice Registers/权重	DSPs	DSPs/权重	BRAMs	BRAMs/权重	时钟频率/MHz	功耗/W	网络结构	时间步	量化方式	MNIST /%	Fashion-MNIST/%	算法	方式	学习速度 (FPS)	推理速度 (FPS)	能耗比
文献 [8]	ZYNQ-7045	16 813	9.29×10^{-3}	7 559	4.18×10^{-3}	8	4.41×10^{-6}	129	71.3×10^{-6}	200.0	1.030*	784-2048-100	200/150	INT8	93.00	78.50	TripletR-STDP	有监督	22.50	30.00	45.78 mJ/frame
文献 [9]	VC707	75 996	0.32	77 580	0.33	0	0	—	—	100.0	1.260*	784-300-10	25	INT13	95.30	—	BP-STDP	有监督	819.00	3 703.00	1.53 mJ/frame
文献 [11]	ZYNQ-7010	9 331	8.41×10^{-3}	5 548	5.00×10^{-3}	0	0	26	23.4×10^{-6}	100.0	0.111	1024-1024-60	—	INT8	94.93	—	BP-STDP	有监督	87.00	89.00	1.28 mJ/frame
文献 [5]	Virtex-6	97 287	0.16	58 826	0.09	0	0	34	54.2×10^{-6}	120.0	0.080	784-800	—	—	89.10	—	STDP	无监督	0.06	0.12	1.33 J/frame
文献 [6]	ZYNQ-7030	78 586	1.00	23 498	0.30	433	5.52×10^{-3}	0	0	301.8	0.754	784-100	250	INT8	83.40	—	STDP-CORDIC-based	无监督	163.90	—	4.60 mJ/frame
文献 [10]	ZYNQ-7045	6 033	30.06×10^{-3}	1 747	8.70×10^{-3}	0	0	57	0.28×10^{-3}	250.0	0.422	784-256	120	INT8	90.19	—	PLR	无监督	1 009.00	2 143.00	0.42 mJ/frame
本文	ZYNQ-7010	2 071	10.32×10^{-3}	2 226	11.1×10^{-3}	0	0	49	0.24×10^{-3}	100.0	0.015	784-256	8	INT8	90.43	80.53	FF-STDP	无监督	1 887.00	1 887.00	7.95 μ J/frame

注:功耗和FPS指标是通过在MNIST数据集下测试的仿真信号翻转率获得的,*所标注文献数据未明确是否加速器动态功耗。

而无需大量内存访问或计算开销。因此,与相关工作相比,处理器核心在 100 MHz 时钟下实现了极低功耗 15 mW,相当于降低了约 1~2 个数量级。

此外,逻辑资源的大幅降低也表明 FF-STDP 的硬件友好性明显优于以往在类似网络结构上的工作,而这些工作通常需要约 2 000 LUT 和寄存器^[6,9-10]。值得注意的是,相对较高的 BRAM 使用量主要是为了在芯片上存储单层网络的全部权值以实现快速原型设计。得益于并行处理和流水线架构,推理与学习的吞吐量可达 1 887 FPS。此外,该算法在时间步数量较少的情况下也能实现满意性能,将时间步从数百或几十降低至 8,相较于其他方法显著减少^[8-10]。

本文设计的核心具有可拓展性^[27],可复用实例化实现多层 SNN 网络,表 3 展示了不同网络结构的硬件资源开销与功耗数据,工作频率均为 100 MHz。以 INT8 量化方式部署模型,200 704 参数量需 49 块 BRAM,随着网络层数增加,访存功耗随之增加,实验结果显示,本文设计在实现 3 层全连接 SNN 的动态功耗高达 57 mW。

表 3 不同网络结构的硬件设计实验数据

Table 3 Experimental data on hardware design of different network structures

网络结构	参数量	LUTs	Registers	BRAMs	动态功耗/mW
784-256	200 704	2 071	2 226	49	15
784-512	401 408	2 404	2 347	98	22
784-512-512	663 552	3 948	3 084	162	37
784-512-512-512	925 696	5 939	4 093	226	57

5 总结与展望

本文提出了一种基于所提出 FF-STDP 算法的能效型 SNN 加速器,实现了片上在线学习。该方法摒弃了 BP 思想,使学习过程完全局部化且硬件友好。受到 BP-STDP 的启发,引入了教师信号和局部导数,以在保持精度的同时进一步简化 FF 算法。所设计的 FPGA 架构采用后突触并行处理与流水线技术,可支持低延迟的并行推理与学习。基于脉冲计数的优度值评估和查找表驱动的权利更新显著降低了计算开销、内存访问和功耗。在 MNIST 和 Fashion-MNIST 上的实验结果表明:该加速器在保持竞争性准确率的同时,功率仅消耗 15 mW,并在 100 MHz 时钟下实现 1 887 FPS 的吞吐量,充分体现了其在资源受限的边缘智能应用中的适用性。

参考文献

- [1] Maas W. Networks of spiking neurons: The third generation of neural network models[J]. Transactions of the Society for Computer Simulation International, 1997, 14(4): 1659-1671.
- [2] Han Song, Pool J, Tran J, et al. Learning both weights and connections for efficient neural networks[C]//Proceedings of the 29th International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2015: 1135-1143.
- [3] Lobo J L, Del Ser J, Bifet A, et al. Spiking neural networks and online learning: An overview and perspectives[J]. Neural Networks, 2020, 121: 88-100.
- [4] Frenkel C, Lefebvre M, Legat J D, et al. A 0.086-mm² 12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS[J]. IEEE Transactions on Biomedical Circuits and Systems, 2019, 13(1): 145-158.
- [5] Wang Qian, Li Youjie, Shao Botang, et al. Energy efficient parallel neuromorphic architectures with approximate arithmetic on FPGA[J]. Neurocomputing, 2017, 221: 146-158.
- [6] Wu Jiajun, Zhan Yi, Peng Zixuan, et al. Efficient design of spiking neural network with STDP learning based on fast CORDIC[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2021, 68(6): 2522-2534.
- [7] Li Sixu, Zhang Zhaomin, Mao Ruixin, et al. A fast and energy-efficient SNN processor with adaptive clock/event-driven computation scheme and online learning[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2021, 68(4): 1543-1552.
- [8] He Zhen, Shi Cong, Wang Tengxiao, et al. A low-cost FPGA implementation of spiking extreme learning machine with on-chip reward-modulated STDP learning[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2022, 69(3): 1657-1661.
- [9] Zhang Jian, Wang Ran, Pei Xudong, et al. A fast spiking neural network accelerator based on BP-STDP algorithm and weighted neuron model[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2022, 69(4): 2271-2275.
- [10] Liu Wei, Xiao Shanlin, Liu Yue, et al. SC-PLR: An approximate spiking neural network accelerator with on-chip predictive learning rule[J]. IEEE Transactions on Biomedical Circuits and Systems, 2024, 18(5): 1156-1165.
- [11] Shi Cong, Fu Xiang, Wang Haibing, et al. Ghost reservoir: A memory-efficient low-power and real-time neuromorphic processor of liquid state machine with on-chip learning[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2024, 71(10): 4526-4530.
- [12] Xu Changqing, Liu Yi, Chen Dongdong, et al. Direct training via backpropagation for ultra-low-latency spiking neural networks with multi-threshold[J]. Symmetry, 2022, 14(9): 1933.
- [13] Xu Changqing, Zhang Wenrui, Liu Yu, et al. Boosting throughput and efficiency of hardware spiking neural accelerators using time compression supporting multiple spike codes[J]. Frontiers in Neuroscience, 2020, 14: 104.
- [14] Xu Changqing, Liu Yi, Yang Yintang. STCSNN: High energy efficiency spike-train level spiking neural net-

- works with spatio-temporal conversion[J]. Neurocomputing, 2024, 607: 128364.
- [15] Lee J, Zhang Renqian, Zhang Wenrui, et al. Spike-train level direct feedback alignment: Sidestepping backpropagation for on-chip training of spiking neural nets[J]. Frontiers in Neuroscience, 2020, 14: 143.
- [16] Hinton G. The forward-forward algorithm: Some preliminary investigations[PP/OL]. V1. arXiv (2022-12-27)[2026-03-01]. <https://arxiv.org/abs/2212.13345>.
- [17] 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.
- [18] Tavanaei A, Maida A. BP-STDP: Approximating back-propagation using spike timing dependent plasticity[J]. Neurocomputing, 2019, 330: 39-47.
- [19] Bu Tong, Fang Wei, Ding Jianhao, et al. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks[PP/OL]. V1. arXiv (2023-03-08)[2026-03-01]. <https://arxiv.org/abs/2303.04347>.
- [20] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [21] Xiao Han, Rasul K, Vollgraf R. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms[PP/OL]. V2. arXiv (2017-09-15) [2026-03-01]. <https://arxiv.org/abs/1708.07747>.
- [22] Orchard G, Jayawant A, Cohen G K, et al. Converting static image datasets to spiking neuromorphic datasets using saccades[J]. Frontiers in Neuroscience, 2015, 9: 437.
- [23] Amir A, Taba B, Berg D, et al. A low power, fully event-based gesture recognition system[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2017: 7388-7397.
- [24] Krizhevsky A. Learning multiple layers of features from tiny images[R]. Toronto: University of Toronto, 2009.
- [25] Cheng Xi, Cao Shu, Wang Shangmei, et al. An FPGA-based event-driven SNN accelerator for DVS applications with structured sparsity and early-stop[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2025, 72(7): 3298-3310.
- [26] Ma Jingxiao, Panda P, Reda S. FF-INT8: Efficient forward-forward DNN training on edge devices with INT8 precision[C]//Proceedings of the 62nd ACM/IEEE Design Automation Conference (DAC). Piscataway: IEEE, 2025: 1-7.
- [27] 王莹, 高岚, 张哲, 等. 基于非一般类算子融合方法及硬件架构设计[J]. 电子学报, 2025, 53(9): 3299-3309.
- Wang Ying, Gao Lan, Zhang Zhe, et al. Operator fusion method and hardware architecture design based on non-standard operators[J]. Acta Electronica Sinica, 2025, 53(9): 3299-3309. (in Chinese)

作者简介



徐长卿 男, 1989年10月出生于河南省焦作市。现为西安电子科技大学副教授, 硕士生导师。主持国家自然科学基金等多个项目, 在国内外发表学术论文42篇。主要研究方向为类脑计算。

E-mail: cqxu@xidian.edu.cn



李裕科 男, 2003年8月出生于广东省茂名市。现为西安电子科技大学硕士研究生。主要研究方向为脉冲神经网络硬件加速。

E-mail: lixituo@outlook.com



莫桂棋 男, 2002年10月出生于广东省广州市。现为西安电子科技大学广州研究院硕士研究生。主要研究方向为脉冲神经网络算法及硬件加速。

E-mail: mogreatch@icloud.com



陈梓鑫 男, 2003年10月出生于广东省汕头市。现为西安电子科技大学卓越工程学院学院硕士研究生。主要研究方向为类脑人工智能与集成电路仿真技术。

E-mail: 25181214328@stu.xidian.edu.cn



刘毅 男, 1971年4月出生于陕西省榆林市。现为西安电子科技大学教授, 博士生导师。主持过国家自然科学基金、研究所以及企业横向课题等科研项目二十余项。主要研究方向为数字集成电路效应仿真技术、NoC关键技术及片上网络柔性电子系统应用等。

E-mail: yiliu@mail.xidian.edu.cn



杨银堂 男, 1962年10月出生于河北省邯郸市。现为西安电子科技大学教授, 博士生导师。全国模范教师, 中国青年科技奖获得者, 入选“教育部跨世纪优秀人才计划”、国家“百千万人才工程”及新世纪“百千万人才工程”。主要研究方向为SOC与混合信号集成电路设计。中国电子学会会员编号: E190011537F。

E-mail: ytyang@xidian.edu.cn