

# 基于脉冲神经网络的时空交互图像分类<sup>①</sup>

曲海成, 李竹媛, 刘万军

(辽宁工程技术大学 软件学院, 葫芦岛 125105)

通信作者: 李竹媛, E-mail: 1715742681@qq.com



**摘要:** 脉冲神经网络作为人工智能发展的重要方向之一, 在神经形态工程和类脑计算领域得到了广泛的关注. 为解决脉冲神经网络泛化性差、内存和时间消耗较大等问题, 本文提出了一种基于脉冲神经网络的时空交互图像分类方法. 首先引入时间有效训练算法弥补梯度下降过程中的动能损失; 其次融合空间随时间学习算法, 提高网络对信息的高效处理能力; 最后添加空间注意力机制, 增强网络对空间维度上重要特征的捕捉能力. 实验结果表明, 改进后的方法在 CIFAR10、DVS Gesture、CIFAR10-DVS 这 3 个数据集上的训练内存占用分别减少了 46.68%、48.52%、10.46%, 训练速度分别提升了 2.80 倍、1.31 倍、2.76 倍, 在保证精度的情况下, 网络性能得到有效提升.

**关键词:** 脉冲神经网络; 时间有效训练; 空间随时间学习; 空间注意力机制; 人工智能

引用格式: 曲海成, 李竹媛, 刘万军. 基于脉冲神经网络的时空交互图像分类. 计算机系统应用, 2024, 33(5): 162-169. <http://www.c-s-a.org.cn/1003-3254/9505.html>

## Spatio-temporal Interactive Image Classification Based on Spiking Neural Network

QU Hai-Cheng, LI Zhu-Yuan, LIU Wan-Jun

(Software College, Liaoning Technical University, Huludao 125105, China)

**Abstract:** As one of the important development directions of artificial intelligence, spiking neural networks have received extensive attention in the fields of neuromorphic engineering and brain-inspired computing. To solve the problems of poor generalization as well as large memory and time consumption in spiking neural networks, this study proposes a classification method based on spiking neural networks for spatio-temporal interactive images. Specifically, a temporal efficient training algorithm is introduced to compensate for the kinetic energy loss in the gradient descent process. Then, the spatial learning through time algorithms are integrated to improve the ability of the network to process information efficiently. Finally, the spatial attention mechanism is added to enable the network to better capture important features in the spatial dimension. The experimental results show that the training memory occupation on the three datasets of CIFAR10, DVS Gesture, and CIFAR10-DVS are reduced by 46.68%, 48.52%, and 10.46%, respectively, and the training speed is increased by 2.80 times, 1.31 times, and 2.76 times, respectively. These results indicate that the proposed method improves network performance effectively on the premise of maintaining accuracy.

**Key words:** spiking neural network; temporal efficient training; spatial learning through time; spatial attention mechanism; artificial intelligence

## 1 引言

自 20 世纪人工智能迎来了飞速发展, 类脑智能研究也得到了人们广泛的关注. 脉冲神经网络 (spiking

neural network, SNN)<sup>[1]</sup>作为类脑智能研究中的重点, 通过模仿高效生物网络<sup>[2]</sup>的神经动力学机制, 将信息以脉冲信号的形式在神经元之间通过突触进行传播计算,

① 基金项目: 国家自然科学基金面上项目 (42271409)

收稿时间: 2023-12-02; 修改时间: 2023-12-29; 采用时间: 2024-01-12; csa 在线出版时间: 2024-04-01

CNKI 网络首发时间: 2024-04-03

体现出较好地处理时间序列数据的能力. 研究表明, 作为第3代神经网络, SNN在芯片和交叉学科等方面中取得了较好的表现, 根据脉冲神经元只有在发射脉冲时才会活跃的特性, 将SNN应用到TrueNorth<sup>[3]</sup>和Loihi<sup>[4]</sup>等神经形态硬件时, 可以实现节能和快速推理计算. 但这种特性也导致训练深层SNN<sup>[5]</sup>比较困难, 因为脉冲的不可微性阻碍了反向传播.

目前, 深层SNN训练主要有两种学习算法. 一种是以转换SNN(ANN-converted SNN)<sup>[6-8]</sup>为代表的间接监督性学习算法, 该算法可以训练较深层的脉冲神经网络并取得一系列具有竞争性的结果, 但它需要消耗大量的时间步长来保证编码分辨率<sup>[9]</sup>, 同时转换SNN不能利用时域(temporal domain, TD)信息, 这给神经形态数据集的训练带来困难. 另一种是以时空反向传播为代表的直接监督学习算法<sup>[10-13]</sup>, 这也是本文采用的算法. 直接监督学习在充分利用时空信息对网络进行训练方面具有很大的潜力, 并且可以减少对时间步长的需求. 虽然直接监督学习算法在一定程度上缓解了转换SNN算法存在的问题, 但仍存在一些不足. 基于直接监督学习的标准化直接训练(standard direct training, SDT)方法采用替代梯度(surrogate gradient, SG)<sup>[14]</sup>来缓解SNN的不可微性<sup>[15]</sup>. 然而, SG的梯度下降趋势与SNN中的损失范围不匹配, 易使训练结果陷入局部极小值, 导致网络泛化能力较差. 另一个不足之处是脉冲神经网络通过基于梯度的时间反向传播(backpropagation through time, BPTT)框架进行训练, 在这样的框架下, 梯度经过逐层的空间域和时间域进行反向传播, 会消耗大量的内存和训练时间, 降低模型的训练性能. 对于一些大规模数据集, 训练成本将显著增加.

针对上述问题, 本文提出了一种基于脉冲神经网络的时空交互图像分类方法. 利用时间有效训练(temporal efficient training, TET)算法<sup>[16]</sup>, 优化每个时刻的突触前输入, 避免了由于局部极小值产生的误差. 融合空间随时间学习(spatial learning through time, SLTT)算法<sup>[17]</sup>, 删除计算图中不重要的路径进行反向传播, 并在每个时间步长瞬时计算梯度, 无需存储其他步长信息, 进而降低训练成本. 最后添加空间注意力机制(spatial attention mechanism, SAM)<sup>[18]</sup>, 增强网络对空间维度上重要特征的捕捉能力, 扩大空间感受野, 从而实现高效率、低能耗的图像分类方法. 相关实验结果及性能对比证明了本文方法的有效性.

## 2 相关工作

### 2.1 基础架构

残差网络(ResNet)作为应用最广泛的基本块之一, 在深度网络中取得了巨大的成就, 但由于脉冲的不可微性和二进制脉冲信号的空间表达能力弱等问题, 使得ResNet存在梯度消失的情况. 因此, 本文使用休眠抑制残差网络(dormant-suppressed residual network, DS-ResNet)作为网络的基础架构, 如图1所示.

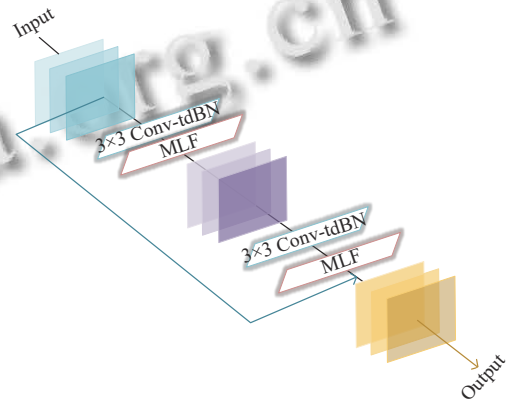


图1 DS-ResNet结构图

相对于传统残差网络(residual network, ResNet), DS-ResNet以依赖阈值的批归一化方法(threshold-dependent batch normalization, tdBN)和多层次发射(multi-level firing, MLF)<sup>[19]</sup>模块代替批归一化(batch normalization, BN)<sup>[20]</sup>和ReLU激活函数, 其中tdBN用于协调分布差并将输入分布归一化, 可以解决梯度消失或爆炸的问题, 同时调节网络整体的脉冲发射率, 并将MLF激活设置在快捷连接方式之前, 减少休眠单元的产生, 从而在非常深的SNN中实现更有效的梯度传播和更有效的离散脉冲信息映射.

### 2.2 层次发射单元

MLF单元包含多个具有不同水平阈值的泄漏整合发放(leaky integrate-and-fire, LIF)<sup>[21]</sup>神经元, 其通过分配每个层级的近似导数的覆盖范围来扩展近似导数的非零区域, 从而减少休眠单元, 并且, MLF单元在激活输入时能够产生具有不同阈值的脉冲, 可以提高其表达能力, 使梯度传播在深度SNN中更有效. MLF单元结构如图2所示.

LIF神经元的正向传播<sup>[22]</sup>过程为:

$$x_i^{t+1,n} = \sum_{j=1}^{l(n-1)} w_{ij}^n o_j^{t+1,n-1} + b_i^n \quad (1)$$

$$u_i^{t+1,n} = k_\tau \quad (2)$$

$$o_i^{t+1,n} = f(u_i^{t+1,n} - V_{th}) = \begin{cases} 1, & u_i^{t+1,n} \geq V_{th} \\ 0, & u_i^{t+1,n} < V_{th} \end{cases} \quad (3)$$

其中,  $n$ 和 $l(n-1)$ 分别表示第 $n$ 层和第 $(n-1)$ 层的神经元个数.  $w_{ij}^n$ 是第 $(n-1)$ 层中第 $j$ 个神经元到第 $n$ 层中第 $i$ 个神经元的突触权值,  $T$ 是时间指标.  $u_i^{t,n}$ 和 $o_i^{t,n}$ 分别为第 $n$ 层第 $i$ 个神经元在 $t$ 时刻的膜电位和输出.  $b_i^n$ 是偏置.  $k_\tau$ 是衰变因子.  $f(\cdot)$ 是激活函数.  $v$ 是触发阈值. 当膜电位超过放电阈值时, 神经元将发射一个尖峰, 膜电位被重置为0.

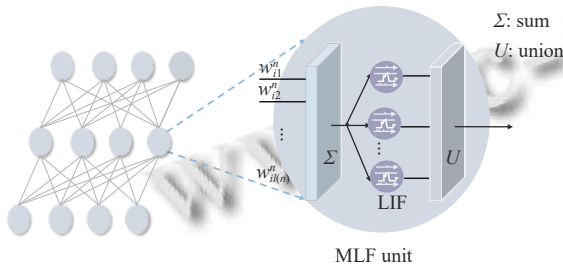


图2 MLF 单元结构图

MLF 单元正向传播过程可以描述为:

$$u_i^{t+1,n} = k_\tau u_i^{t,n} \odot (1 - o_i^{t,n}) + x_i^{t+1,n} \quad (4)$$

$$o_i^{t+1,n} = f(u_i^{t+1,n} - V_{th}) \quad (5)$$

$$\delta_i^{t+1,n} = s(o_i^{t+1,n}) \quad (6)$$

其中,  $u_i^{t,n} = (u_{i,1}^{t,n}, u_{i,2}^{t,n}, \dots, u_{i,k}^{t,n}, \dots, u_{i,K}^{t,n})$ 和 $o_i^{t,n} = (o_{i,1}^{t,n}, o_{i,2}^{t,n}, \dots, o_{i,k}^{t,n}, \dots, o_{i,K}^{t,n})$ 分别表示在时间 $t$ 的第 $n$ 层中的第 $i$ 个 MLF 单元的膜电位矢量和输出矢量,  $\odot$ 表示 Hadamard 乘积.  $k$ 和 $K$ 分别表示第 $k$ 级别和级别的数目.  $V_m = (V_{th1}, V_{th2}, \dots, V_{thk}, \dots, V_{thK})$ 为阈值向量. 为了便于计算突触前输入  $x_i^{t,n}$ , 将脉冲编码器定义为:  $s(o_i^{t,n}) = o_{i,1}^{t,n} + o_{i,2}^{t,n} + \dots + o_{i,k}^{t,n}$ ,  $\delta_i^{t,n} = s(o_i^{t,n})$ 是第 $n$ 层中的第 $i$ 个 MLF 单元在时间 $t$ 的最终输出. 然后,  $x_i^{t,n}$ 可以通过式(1)来计算, 其中  $o_i^{t,n}$ 被替换为 $\delta_i^{t,n}$ .

对比式(2)–式(6)可以看出, MLF 单元并没有给网络引入额外的可训练参数, 并且, 利用多尖峰的结合, MLF 单元可以区分一些大值的尖锐特征和小值的非尖锐特征.

### 2.3 标准化直接训练

SDT 用  $O(t)$ 表示输出层的突触前输入  $I(t)$ , 并计算

交叉熵损失. 式(7)为标准化直接训练  $L_{SDT}$  的损失函数式, 其中  $T$ 为模拟总时间,  $L_{CE}$ 为交叉熵损失,  $y$ 为目标符号.

$$L_{SDT} = L_{CE} \left( \frac{1}{T} \sum_{t=1}^T O(t), y \right) \quad (7)$$

根据链式法则, 使用  $SoftmaxS(\cdot)$ 推理函数得到权值梯度:

$$\frac{\partial L_{SDT}}{\partial W} = \frac{1}{T} \sum_{t=1}^T [S(O_{(t)}) - \hat{y}] \frac{\partial O(t)}{\partial W} \quad (8)$$

其中,  $O_{mean}$ 表示输出  $O_{(t)}$ 随时间的平均值,  $\hat{y}$ 表示  $y$ 的单个编码.

在 SDT 的情况下, 梯度由两部分组成, 误差项  $(S(O_{mean}) - \hat{y})$ 和输出的偏导数  $\partial O(t)/\partial W$ . 当训练过程达到局部最小值附近时, 对于所有  $t = 1, \dots, T$ 项  $(S(O_{mean}) - \hat{y})$ 近似为0, 忽略了  $\partial O(t)/\partial W$ . 对于传统的人工神经网络, 累积动量可能有助于摆脱局部极小值, 但这通常意味着泛化性较差. 然而, 当 SNN 使用替代梯度训练时, 考虑到梯度和损失的不匹配, 累积动量可能非常小. 激活函数是阶跃函数, 而 SG 受函数约束. 这种不匹配消散了局部最小值周围的动量, 并阻止了 SDT 搜索可能意味着更好的泛化性的更平坦的最小值.

### 2.4 时间反向传播

BPTT 是训练循环网络的常用技术之一<sup>[22]</sup>. BPTT 通过创建循环单元的多个副本来及时“展开”神经网络, 可以将其视为具有绑定权重的(深层)前馈网络. 完成此操作后, 使用前向传播算法来评估整个输入序列上的网络适应性, 而反向传播算法可用于评估损失标准的偏导数相对于所有网络参数. 这种方法虽然计算效率高, 但在内存使用方面也相当密集.

BPTT 方法可通过式(9)计算具有  $T$ 个时间步长的 SNN 中的权重梯度:

$$\nabla_w^l L = \sum_{t=1}^T \left( \frac{\partial L}{\partial u^l[t]} \right)^T s^{l-1}[t]^T, l = L, L-1, \dots, 1 \quad (9)$$

其中,  $w$ 是权重,  $l$ 是层数索引,  $L$ 是损失函数,  $u^l[t]$ 是  $t$ 个时间步长  $l$ 层的膜电位,  $s^l$ 是第  $l$ 层的输出脉冲序列.

$$e^l[t] \triangleq \frac{\partial u^l[t+1]}{\partial u^l[t]} + \frac{\partial u^l[t+1]}{\partial s^l[t]} \frac{\partial s^l[t]}{\partial u^l[t]} \quad (10)$$

定义  $e^l[t]$ 为  $u^l[t+1]$ 对  $u^l[t]$ 的灵敏度, 然后利用式(10)可以进一步递归地计算出式(9)中的  $\partial L/\partial u^l[t]$ . 特别地,

对于输出层:

$$\frac{\partial L}{\partial u^L[t]} = \frac{\partial L}{\partial s^L[t]} \frac{\partial s^L[t]}{\partial u^L[t]} + \sum_{t'=t+1}^T \frac{\partial L}{\partial s^L[t']} \frac{\partial s^L[t']}{\partial u^L[t']} \prod_{t''=1}^{t'-t} \epsilon^L[t'-t''] \quad (11)$$

对于中间层  $l = L-1, \dots, 1$  有:

$$\frac{\partial L}{\partial u^l[t]} = \frac{\partial L}{\partial u^{l+1}[t]} \frac{\partial u^{l+1}[t]}{\partial s^l[t]} \frac{\partial s^l[t]}{\partial u^l[t]} + \sum_{t'=t+1}^T \frac{\partial L}{\partial u^{l+1}[t']} \frac{\partial u^{l+1}[t']}{\partial s^l[t']} \frac{\partial s^l[t']}{\partial u^l[t']} \prod_{t''=1}^{t'-t} \epsilon^l[t'-t''] \quad (12)$$

其中, BPTT 方法中计算的膜电位导数可以分解为空间分量和时间分量, 而空间分量在计算的导数中占主导地位。

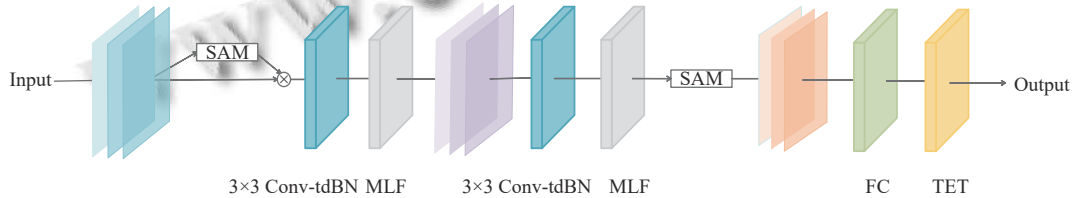


图3 基于脉冲神经网络的时空交互图像分类方法

为了得到更广义的 SNN, 修改优化目标调整每个矩的输出分布. TET 的工作流如图 4 所示. 它约束每个时刻的输出 (突触前输入) 接近目标分布. 其损失函数  $L_{TET}$  可描述为:

$$L_{TET} = \frac{1}{T} \sum_{t=1}^T L_{CE}[O(t), y] \quad (13)$$

重新计算损失函数  $L_{TET}$  下的权值梯度, 得到:

$$\frac{\partial L_{TET}}{\partial W} = \frac{1}{T} \sum_{t=1}^T [S(O(t)) - \hat{y}] \frac{\partial O(t)}{\partial W} \quad (14)$$

在 TET 的情况下, 通过重新加权  $\partial O(t)/\partial W$  的贡献, 可以缓解 SDT 中梯度和损失不匹配的问题. 由于训练集的早期输出精度不可能达到 100%, 第 1 项  $S(O(t) - \hat{y})$  不可能在 SNN 的每个时刻都为 0. 因此, TET 需要第 2 项  $\partial O(t)/\partial W$  趋近于 0 才能使  $L_{TET}$  收敛. 该机制增加了局部极小值附近的梯度范数, 并驱动 TET 寻找权值干扰不会引起  $O(t)$  巨大变化的平坦局部极小值。

### 3.3 空间随时间学习算法

为了减少训练时间和内存成本, 在原有的用替代梯度进行时间反向传播基础上, 引入空间随时间学习

## 3 基于脉冲神经网络的时空交互图像分类方法

### 3.1 分类方法的网络结构

基于脉冲神经网络的时空交互图像分类方法的网络结构主要是由主干网络、两个 SAM 模块和 TET 这 3 个主要部分组成. 主干网络 DS-ResNet 包括  $3 \times 3$  Conv-tdBN 层、MLF 层和全连接层 (fully connected layer, FC). 具体结构设计如图 3 所示。

### 3.2 时间有效训练

为了弥补替代梯度的直接训练方法导致 SNN 泛化性差的现象, 引入时间有效训练 (temporal efficient training, TET) 算法来弥补梯度下降过程中的动能损失, 使训练过程收敛到更平坦的最小值, 具有更好的泛化性. 同时, TET 提高了 SNN 的时间可扩展性, 并诱导了一个时间可继承的加速训练。

SLTT 算法, 替换 BPTT, 构成了一种基于 SG 的 SLTT 框架结构。

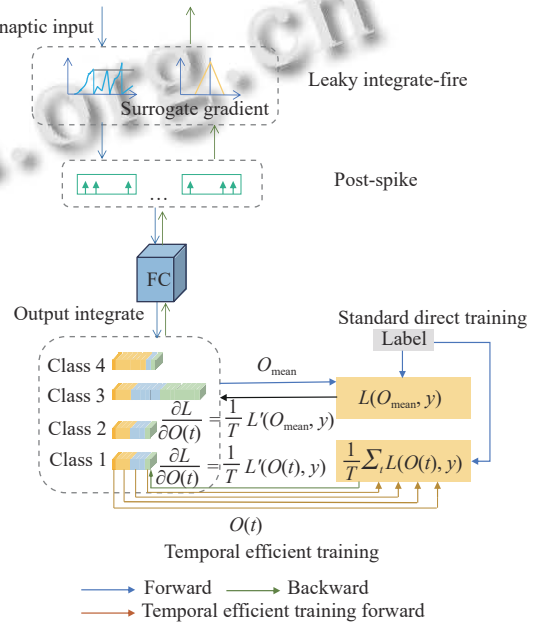


图4 时间效率训练 workflow

SLTT 删除了 BPTT 原本计算图中不重要的路径, 仅利用空间分量来进行更有效的反向传播, 减少标量

乘法运算的次数,进而减少训练时间. SLTT 还通过在每个时间步长即时计算梯度来实现在线训练,无需存储其他时间步长的信息,内存占用与总时间步长无关,避免了 BPTT 的显著训练内存成本. 相应权重梯度的公式定义为:

$$\nabla w^l L = \sum_{t=1}^T e^l w[t], e^l w[t] = e_u^l[t]^T s^{l-1}[t]^T \quad (15)$$

其中,行向量  $e_u^l[t]$  公式为:

$$e_u^l[t] = \begin{cases} \frac{\partial L}{\partial s^L[t]} \frac{\partial s^L[t]}{\partial u^L[t]}, & l = L \\ e_u^{l+1}[t] \frac{\partial u^{l+1}[t]}{\partial s^l[t]} \frac{\partial s^l[t]}{\partial u^l[t]}, & l < L \end{cases} \quad (16)$$

与式 (9)、式 (11) 相比,式 (15) 中所需的标量乘法次数由  $\Omega(T^2)$  减少至  $\Omega(T)$ , 时间复杂度明显降低,根据式 (15), 在没有来自其他时间步长的信息的情况下,模型可以独立地计算每个时间步长的误差信号  $e^l w^{[l]}$  和  $e_u^l[t]$ . 因此,如果可以在时间步  $t$  瞬时计算出  $\partial L / \partial s^L[t]$ , 则也可以在时间步  $t$  瞬时计算  $e^l w^{[l]}$  和  $e_u^l[t]$ . 那么就没有必要存储整个时间范围的中间状态. 为了实现  $\partial L / \partial s^L[t]$  的瞬时计算,可以采用损失函数:

$$L = \frac{1}{T} \sum_{t=1}^T \ell(o(t), y) \quad (17)$$

其中,  $y$  是标签,  $\ell$  可以是交叉熵函数,网络的最终输出是  $o[t] = W^o s^L[t]$ ,  $W^o$  是分类器的参数.

综上所述,时间步长  $t$  处的所有中间项,如  $e_u^l[t]$ 、 $s^l[t]$ 、 $\partial u^{l+1}[t] / \partial s^l[t]$ 、 $\partial s^l[t] / \partial u^l[t]$  从未在其他时间步长中使用,因此 SLTT 所需的内存开销与时间步长  $t$  的总数无关. 而 BPTT 具有与存储所有时间步长的所有中间状态相关联的存储器成本. 所以,SLTT 相较于 BPTT 的内存开销较小,运行速率明显提升.

### 3.4 空间注意力机制

为了减少信息弥散和增强网络的空间跨维度交互能力,加强网络在空间维度上对重要特征的捕捉能力,本文引入了空间注意力模块,可以利用特征的空间关系生成空间注意力图来获取数据中更重要的信息,中间特征图通过空间注意力机制 (spatial attention mechanism, SAM)<sup>[23]</sup> 在深度网络的每个卷积块上自适应地细化,其结构如图 5 所示.

为了获取空间信息,使用平均池化和最大池化两

次操作对特征图的通道信息进行聚合,生成两个 2D 特征图:  $F_{\text{avg}}^s \in R^{1 \times H \times W}$  和  $F_{\text{max}}^s \in R^{1 \times H \times W}$  分别表示跨通道的平均池化特征和最大池化特征. 然后将它们通过标准卷积层进行连接和卷积,生成二维空间的注意力图. 具体的计算过程为式 (18):

$$\begin{aligned} M_s(F) &= \sigma(f^{7 \times 7}([\text{AvgPool}(F); \text{MaxPool}(F)])) \\ &= \sigma(f^{7 \times 7}([F_{\text{avg}}^s; F_{\text{max}}^s])) \end{aligned} \quad (18)$$

其中,  $\sigma$  表示 Sigmoid 函数,  $f^{7 \times 7}$  表示滤波器大小为  $7 \times 7$  的卷积运算.

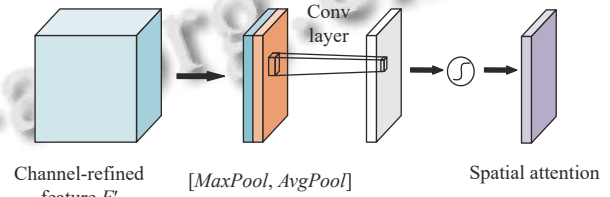


图 5 空间注意力机制

## 4 实验结果与分析

### 4.1 实验环境

实验环境为 Linux 操作系统, CPU 为 AMD EPYC 7642, 显卡为 NVIDIA GeForce RTX 3090, 显存为 24 GB, 采用 PyTorch 1.13.1 和 CUDA 11.7 环境运行代码. 在 CIFAR10、DVS Gesture、CIFAR10-DVS 这 3 个数据集设置训练初始学习率为 0.1, batchsize 分别为 64、16、32, 采用 SGD 优化器进行优化, 3 级阈值  $V_{th}$  为 0.6、1.6、2.6, 权重衰减分别为 0.000 1、0.000 1、0.001<sup>[19]</sup>.

### 4.2 实验数据集

#### 4.2.1 CIFAR10 数据集

CIFAR10 数据集<sup>[24]</sup> 是一个广泛用于识别通用物体的彩色图像数据集. 它包含 5 万张训练图像和 1 万张测试图像, 分为 10 个类别, 大小为  $32 \times 32$ . 对于 CIFAR10 的数据预处理, 遵循标准的数据增强策略. 首先对原始图像进行随机裁剪和翻转, 然后通过减去像素强度的全局平均值并除以沿每个 RGB 通道的全局标准方差进行归一化.

#### 4.2.2 DVS Gesture 数据集

DVS Gesture 数据集<sup>[25]</sup> 是通过捕获 29 个受试者在 3 种照明条件下的不同手势而获得的神经形态视觉数据集. 它共有 1324 个实例, 23 个类别的 1176 个实例作为训练集, 6 个类别的 288 个实例作为测试集, 大小为  $128 \times 128$ . DVS Gesture 数据集的数据预处理为将

原始事件流下采样到  $32 \times 32$ , 每 30 ms 采样一次. 在每个时间步长中, 并且设置训练和测试时间步为 40.

#### 4.2.3 CIFAR10-DVS 数据集

CIFAR10-DVS 数据集<sup>[26]</sup>是使用 DVS 摄像机从 CIFAR10 数据集转换而来的神经形态视觉数据集. 它由 10 个类别的 10 000 个实例组成, 大小为  $128 \times 128$ , 随机选择 9 000 个实例用于训练, 其余的用于测试. CIFAR10-DVS 数据集的数据预处理为将原始事件流下采样到  $42 \times 42$ , 每 10 ms 采样一次, 并且设置训练和测试时间步为 10.

#### 4.3 评价指标

使用精确度  $Acc$  (Accuracy), 参数量 Params, 整体时长 Time, 内存 Memory, 每轮训练时长 (Time/Epoch), 作为综合评价指标, 来测评分类效果.  $TP$  (true positive) 为正确分类图像的数目,  $FP$  (false positive) 为错误分类图像的数目.  $Acc$  的计算公式为:

$$Acc = \frac{TP}{TP + FP} \quad (19)$$

#### 4.4 消融实验

为了证明本文所提出方法的有效性, 以 DS-ResNet 为基础网络结构, 在 CIFAR10、DVS Gesture、CIFAR10-DVS 上进行了评估, 以证明其在训练成本和准确性方面的优越性能. 为公平比较实验结果, 所有参数保持一致, 消融实验结果见表 1.

表 1 CIFAR10 数据集消融实验

Method	Memory (GB)	Time/Epoch (s)	Acc (%)
DS-ResNet	9.64	339	94.25
DS-ResNet+TET	2.58	291	94.46
DS-ResNet+SLTT	5.06	277	94.28
DS-ResNet+SAM	9.56	315	94.42
DS-ResNet+TET+SLTT+SAM	5.14	121	94.71

第 1 组为改进前的 CIFAR10 数据集模型实验结果, 作为后几组实验的参考基准, 每一轮训练内存占用为 9.64 GB, 训练测试时长 Time/Epoch 为 339 s, 精度  $Acc$  为 94.25%.

第 2 组由于将 TET 模块引入模型中, 缓解了梯度和损失不匹配的问题, 使得训练内存占用减少到 2.58 GB, Time/Epoch 减少到 291 s,  $Acc$  值也提高了 0.21 个百分点.

第 3 组用 SLTT 算法替换传统网络中的 BPTT, 删除计算图中不重要的路径, 虽然  $Acc$  值增长不明显, 但训练内存占用和 Time/Epoch 得到显著降低.

第 4 组将 SAM 融入到 DS-ResNet 模型中, 使模型能够自适应的学习不同区域的注意力权重, 更加关注

重要的图像区域, 训练内存占用和 Time/Epoch 得到一定程度降低,  $Acc$  也提升了 0.17 个百分点.

第 5 组是本文整体改进方法实验, 与改进前的模型相比, 内存占用降低了 46.68%, Time/Epoch 仅为改进前的 35.7%, 同时,  $Acc$  升高了 0.46 个百分点, 证实了各方法对于模型的有效性.

本文对比了 CIFAR10 数据集每轮训练时的损失, 从图 6 可以看出改进前后的模型在 40 轮前一直呈现波动状态, 改进前的模型在 46 轮损失下降至 0.221 7, 改进后的模型损失下降至 0.260 2, 损失差距变得明显, 之后一直保持平稳. 实验结果表明, 本文使用的方法可使网络训练损失达到更低的稳定数值; 另外从图 7 所示的混淆矩阵可以看出改进后的模型在 CIFAR10 数据集 10 个类别中的 7 个类别上的准确率相较于原模型均有提升, 进一步证明了网络精度的提高.

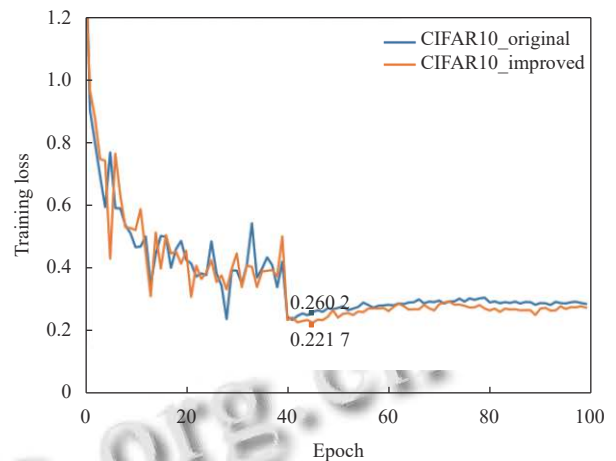


图 6 CIFAR10 数据集训练损失图

#### 4.5 对比实验

表 2 为本文整体改进方法与改进前分别在 CIFAR10、DVS Gesture、CIFAR10-DVS 数据集上的实验结果对比, 改进后的方法能够在保证精度  $Acc$  的情况下, 内存和整体时长均取得明显降低, 大大减少了内存占用和时间需求. 由此可见, 方法在模型的内存消耗与时间需求上有明显优势.

为进一步验证改进后的方法性能, 分别在 CIFAR10、DVS Gesture、CIFAR10-DVS 数据集上与主流方法进行对比, 每个方法实验环境、训练集、测试集均相同. 从表 3 结果可以看出, 本文提出的改进方法在各个数据集上均达到了最佳精度, 同时, 参数量与目前最小参数量相比并无明显增加, 方法性能在精度和参数量方面均占优势.

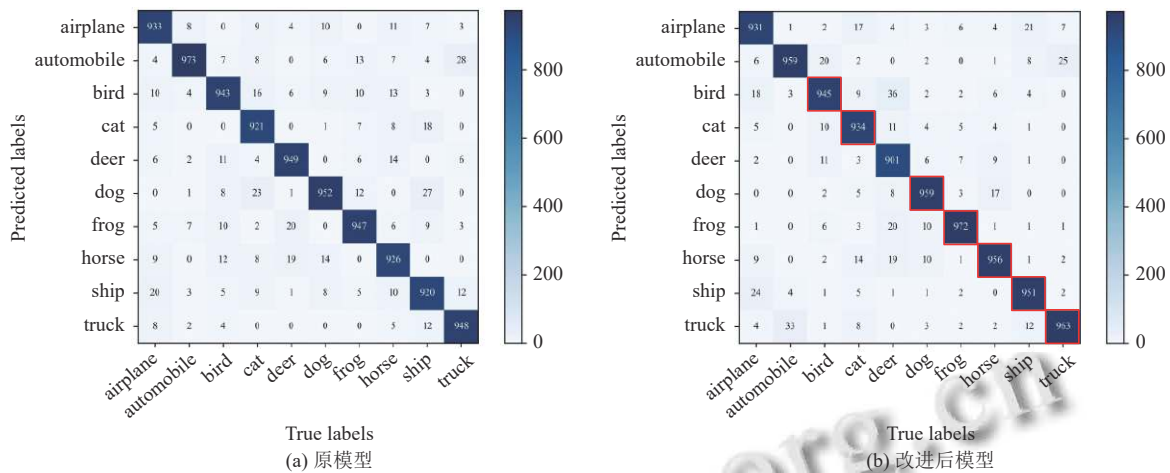


图7 CIFAR10数据集混淆矩阵

表2 各数据集对比实验

Dataset	Method	Memory (G)	Time (h)	Acc (%)
CIFAR10	DS-ResNet <sup>[19]</sup>	9.64	9.42	94.25
	DS-ResNet+TET+SLTT+SAM	<b>5.14</b>	<b>3.36</b>	<b>94.71</b>
DVS Gesture	DS-ResNet <sup>[19]</sup>	1.69	3.50	97.29
	DS-ResNet+TET+SLTT+SAM	<b>0.87</b>	<b>2.67</b>	<b>97.56</b>
CIFAR10-DVS	DS-ResNet <sup>[19]</sup>	19.70	22.8	70.36
	DS-ResNet+TET+SLTT+SAM	<b>17.64</b>	<b>8.25</b>	<b>71.23</b>

表3 各模型对比实验

Dataset	Method	Acc (%)	Params (M)
CIFAR10	STBP <sup>[10]</sup>	90.53	44.99
	Spike-based <sup>[11]</sup>	90.95	18.2
	ASF-BP <sup>[27]</sup>	91.35	8.81
	STBP+tdBN <sup>[13]</sup>	93.16	15.10
	ANN-SNN <sup>[28]</sup>	94.16	9.33
	MLF+DS-ResNet <sup>[19]</sup>	94.25	<b>4.32</b>
	SLTT+ DS-ResNet (ours)	<b>94.71</b>	4.33
DVS Gesture	STBP <sup>[29]</sup>	93.40	2.32
	ANN-SNN <sup>[30]</sup>	95.56	0.80
	STBP+tdBN <sup>[13]</sup>	96.87	0.35
	MLF+DS-ResNet <sup>[19]</sup>	97.29	<b>0.27</b>
	SLTT+ DS-ResNet (ours)	<b>97.56</b>	<b>0.2</b>
CIFAR10-DVS	STBP <sup>[10]</sup>	60.50	26.82
	ANN-SNN <sup>[30]</sup>	65.61	<b>0.50</b>
	ASF-BP <sup>[27]</sup>	62.50	26.78
	STBP+tdBN <sup>[13]</sup>	67.80	16.27
	MLF+DS-ResNet <sup>[19]</sup>	70.36	0.69
	SLTT+ DS-ResNet (ours)	<b>71.23</b>	0.70

### 5 结论与展望

针对 SNN 替代梯度方法中梯度下降趋势与 SNN 之间损失分布不匹配的问题, 本文融入了 TET 算法, 优化每个时刻的突触前输入, 弥补梯度下降过程中的

动能损失; 引入 SLTT 算法利用空间分量来进行更有效的反向传播, 减少网络的内存消耗, 缩短训练时长; 添加 SAM 提升网络在空间维度上重要特征的捕捉能力. 实验结果表明, 本文使用的方法在保证精度提升的条件下, 显著降低了模型的内存消耗和训练时长. 对于 CIFAR10、DVS Gesture、CIFAR10-DVS 这 3 个数据集的内存消耗分别降低了 46.68%、48.52%、10.46%, 训练速度分别提升了 2.80 倍、1.31 倍、2.76 倍, 为使用 SNN 实现高效率、低能耗的分类训练提供了可能. 在后续研究中, 将重点关注影响反向传播的因素, 进一步完善时空交互图像分类模型, 提升 SNN 图像分类的判断能力, 为将来在复杂环境下部署低功耗设备打下坚实基础.

### 参考文献

- 胡一凡, 李国齐, 吴郁杰, 等. 脉冲神经网络研究进展综述. 控制与决策, 2021, 36(1): 1–26. [doi: 10.13195/j.kzyjc.2020.1006]
- 王恩旺, 王恩达. 改进的帧差法在空间运动目标检测中的应用. 天文研究与技术, 2016, 13(3): 333–339.
- Debole MV, Taba B, Amir A, et al. TrueNorth: Accelerating from zero to 64 million neurons in 10 years. Computer, 2019, 52(5): 20–29. [doi: 10.1109/MC.2019.2903009]
- Davies M, Srinivasa N, Lin TH, et al. Loihi: A neuromorphic manycore processor with on-chip learning. IEEE Micro, 2018, 38(1): 82–99. [doi: 10.1109/MM.2018.112130359]
- 张铁林, 徐波. 脉冲神经网络研究现状及展望. 计算机学报, 2021, 44(9): 1767–1785.
- Sengupta A, Ye YT, Wang R, et al. Going deeper in spiking neural networks: VGG and residual architectures. Frontiers in Neuroscience, 2019, 13: 95. [doi: 10.3389/fnins.2019.000

- 95]
- 7 Yan ZL, Zhou J, Wong WF. Energy efficient ECG classification with spiking neural network. *Biomedical Signal Processing and Control*, 2021, 63: 102170. [doi: [10.1016/j.bspc.2020.102170](https://doi.org/10.1016/j.bspc.2020.102170)]
  - 8 Hu YF, Tang HJ, Pan G. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2023, 34(8): 5200–5205. [doi: [10.1109/TNNLS.2021.3119238](https://doi.org/10.1109/TNNLS.2021.3119238)]
  - 9 Gautam A, Singh V. CLR-based deep convolutional spiking neural network with validation based stopping for time series classification. *Applied Intelligence*, 2020, 50(3): 830–848. [doi: [10.1007/s10489-019-01552-y](https://doi.org/10.1007/s10489-019-01552-y)]
  - 10 Wu YJ, Deng L, Li GQ, *et al.* Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 2018, 12: 331. [doi: [10.3389/fnins.2018.00331](https://doi.org/10.3389/fnins.2018.00331)]
  - 11 Gu PJ, Xiao R, Pan G, *et al.* STCA: Spatio-temporal credit assignment with delayed feedback in deep spiking neural networks. *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. Macao: ijcai.org, 2019. 1366–1372.
  - 12 Liu QH, Xing D, Tang HJ, *et al.* Event-based action recognition using motion information and spiking neural networks. *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. Montreal: ijcai.org, 2021. 1743–1749.
  - 13 Zheng HL, Wu YJ, Deng L, *et al.* Going deeper with directly-trained larger spiking neural networks. *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. Washington: AAAI Press, 2021. 11062–11070.
  - 14 Cramer B, Billaudelle S, Kanya S, *et al.* Training spiking multi-layer networks with surrogate gradients on an analog neuromorphic substrate. *arXiv:2006.07239*, 2020.
  - 15 Lee JH, Delbruck T, Pfeiffer M. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience*, 2016, 10: 508.
  - 16 Deng SK, Li YH, Zhang SH, *et al.* Temporal efficient training of spiking neural network via gradient re-weighting. *Proceedings of the 10th International Conference on Learning Representations*. OpenReview.net, 2022.
  - 17 Meng QY, Xiao MQ, Yan S, *et al.* Towards memory- and time-efficient backpropagation for training spiking neural networks. *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Paris: IEEE, 2023. 6143–6153.
  - 18 Gao CL, Ye HL, Cao FL, *et al.* Multiscale fused network with additive channel-spatial attention for image segmentation. *Knowledge-based Systems*, 2021, 214: 106754. [doi: [10.1016/j.knsys.2021.106754](https://doi.org/10.1016/j.knsys.2021.106754)]
  - 19 Feng L, Liu QH, Tang HJ, *et al.* Multi-level firing with spiking DS-ResNet: Enabling better and deeper directly-trained spiking neural networks. *Proceedings of the 31st International Joint Conference on Artificial Intelligence*. Vienna: ijcai.org, 2022. 2471–2477.
  - 20 Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on International Conference on Machine Learning*. Lille: JMLR.org, 2015. 448–456.
  - 21 Fang W, Yu ZF, Chen YQ, *et al.* Incorporating learnable membrane time constant to enhance learning of spiking neural networks. *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal: IEEE, 2021. 2641–2651.
  - 22 Yin B, Corradi F, Bohtë SM. Accurate online training of dynamical spiking neural networks through forward propagation through time. *Nature Machine Intelligence*, 2023, 5(5): 518–527. [doi: [10.1038/s42256-023-00650-4](https://doi.org/10.1038/s42256-023-00650-4)]
  - 23 Bauer FC, Lenz G, Haghghatshoar S, *et al.* EXODUS: Stable and efficient training of spiking neural networks. *Frontiers in Neuroscience*, 2023, 17: 1110444. [doi: [10.3389/fnins.2023.1110444](https://doi.org/10.3389/fnins.2023.1110444)]
  - 24 Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 2009, 1(4): 1–60.
  - 25 Amir A, Taba B, Berg D, *et al.* A low power, fully event-based gesture recognition system. *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu: IEEE, 2017. 7388–7397.
  - 26 Li HM, Liu HC, Ji XY, *et al.* CIFAR10-DVS: An event-stream dataset for object classification. *Frontiers in Neuroscience*, 2017, 11: 309.
  - 27 Wu H, Zhang YY, Weng WM, *et al.* Training spiking neural networks with accumulated spiking flow. *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. Washington: AAAI Press, 2021. 10320–10328.
  - 28 Yan ZL, Zhou J, Wong WF. Near lossless transfer learning for spiking neural networks. *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. Washington: AAAI Press, 2021. 10577–10584.
  - 29 He WH, Wu YJ, Deng L, *et al.* Comparing SNNs and RNNs on neuromorphic vision datasets: Similarities and differences. *Neural Networks*, 2020, 132: 108–120. [doi: [10.1016/j.neunet.2020.08.001](https://doi.org/10.1016/j.neunet.2020.08.001)]
  - 30 Kugele A, Pfeil T, Pfeiffer M, *et al.* Efficient processing of spatio-temporal data streams with spiking neural networks. *Frontiers in Neuroscience*, 2020, 14: 439. [doi: [10.3389/fnins.2020.00439](https://doi.org/10.3389/fnins.2020.00439)]

(校对责编: 孙君艳)