

# 《Training Deep Spiking Neural Networks Using Backpropagation》阅读笔记

人工智能学院 刘子昊

## 一、研究背景与意义

在人工智能领域，深度神经网络（DNNs）凭借其强大的特征学习与数据拟合能力，在图像识别、自然语言处理等众多任务中取得了卓越成果。然而，传统深度神经网络在实时交互场景中面临着 latency（延迟）和 energy efficiency（能量效率）的瓶颈——其需要对大量网络单元进行重复且往往冗余的参数更新，这使得其在边缘设备、实时机器人交互等对速度和功耗敏感的应用中难以高效部署。

脉冲神经网络（Spiking Neural Networks, SNNs）作为一种模拟生物神经元信息处理方式的网络模型，为解决这一问题提供了新的方向。SNNs 基于事件驱动（event-based）的计算模式，仅在神经元产生脉冲（spike）事件时才进行状态更新，能够显著减少冗余计算，降低能量消耗。尤其当 SNNs 与动态视觉传感器（Dynamic Vision Sensor, DVS）等事件驱动型传感器结合时，可进一步通过异步事件处理减少输入信息的冗余性，在实时性与能效方面展现出巨大潜力。

从理论上讲，Maass 和 Markram 在 2004 年的研究已证明 SNNs 与传统人工神经网络（Artificial Neural Networks, ANNs）具有同等的计算能力。但在实际应用中，SNNs 在传统机器学习任务（如手写数字识别）中的精度始终无法与 ANNs 媲美，核心瓶颈在于“训练算法的缺失”。传统深度神经网络的训练依赖误差反向传播（backpropagation）算法，而该算法的核心是基于可微的激活函数计算梯度；但 SNNs 中的脉冲事件是离散的——神经元仅在膜电位（membrane potential）超过阈值时产生脉冲，这种离散性导致脉冲信号不可微，使得直接应用反向传播算法训练深度 SNNs 成为难题。

此前，研究人员提出了“间接训练 + 网络转换”的解决方案：先训练一个传统 ANN，再通过转换算法将 ANN 的权重映射到 SNN 中。例如，O'Connor 等人（2013）基于 Siegart 事件率近似模型训练脉冲深度信念网络，Hunsberger 和 Eliasmith（2015）使用软化率模型（将 LIF 神经元的硬阈值替换为连续可微函数）训练 ANN 后转换为 SNN。然而，这类方法存在显著缺陷：它们通过连续信号（如平均发放率）近似脉冲序列的统计特性，无法精确捕捉实际事件驱动传感器产生的脉冲序列中超出理想速率模型的细节（如脉冲同步性、时间依赖关系），导致转换后的 SNN 精度损失。

此外，另一类方法尝试直接基于脉冲序列训练，但仅能实现单层的监督学习或依赖无监督学习规则（如脉冲时序依赖可塑性 STDP），多层网络的精度会因误差传递困难而大幅下降。O'Connor 和 Welling（2016）虽提出一种基于脉冲学习的 SNN 训练方法，但需要维护统计信息以近似 ANN 的随机梯度下降（SGD）更新，仍未摆脱对传统 ANN 模型的依赖。

在这一背景下，Lee 等人于 2016 年发表的《Training Deep Spiking Neural Networks Using Backpropagation》一文具有里程碑意义。该研究提出了一“直接基于脉冲信号和膜电

位训练深度 SNNs 的反向传播算法”，通过将膜电位视为可微信号、脉冲时刻的不连续性视为噪声，突破了脉冲离散性带来的梯度计算难题。这一方法不仅能够精确捕捉脉冲序列的统计特性，还能直接处理事件驱动传感器的输入，为缩小 SNNs 与 ANNs 的精度差距、推动 SNNs 在实际场景中的应用奠定了基础。

## 二、核心方法与技术细节

### 2.1 脉冲神经网络基础模型

文章首先明确了研究中使用的 SNN 基本结构，包括神经元模型和竞争机制，为后续训练算法的设计提供基础。

#### 2.1.1 泄漏积分发放 (Leaky Integrate-and-Fire, LIF) 神经元模型

LIF 神经元是 SNN 中最常用的简化神经元模型，其核心特性是膜电位随输入脉冲积分并随时间泄漏，当膜电位超过阈值时产生输出脉冲并重置。文章对传统 LIF 模型进行了改进，引入“动态权重 (dynamic weight)”以控制不应期 (refractory period)，具体公式与机制如下：

##### (1) 膜电位更新规则

对于第  $p$  个输入脉冲，神经元的膜电位  $V_{mp}$  更新公式为：

$$V_{mp}(t_p) = V_{mp}(t_{p-1}) \cdot e^{-(t_p - t_{p-1})/\tau_{mp}} + w_{dyn} \cdot w_i^{(p)}$$

其中：

$\tau_{mp}$  为膜时间常数 (membrane time constant)，控制膜电位泄漏速度；

$t_p$  和  $t_{p-1}$  分别为当前和前一个输入脉冲的时间；

$w_i^{(p)}$  为第  $i$  个突触的权重；

$w_{dyn}$  为动态权重，用于调节不应期内输入脉冲的影响。

##### (2) 动态权重设计

动态权重  $w_{dyn}$  的作用是在神经元产生输出脉冲后，短期内抑制输入脉冲对膜电位的贡献，模拟生物神经元的不应期特性，其计算公式为：

$$w_{dyn} = \begin{cases} (\Delta_t / T_{ref})^2, & \text{if } \Delta_t < T_{ref} \text{ and } w_{dyn} < 1 \\ 1, & \text{otherwise} \end{cases}$$

其中：

$T_{ref}$  为最大不应期时长；

$\Delta_t = t_{out} - t_p$ ， $t_{out}$  为神经元最近一次输出脉冲的时间。

与传统“直接阻断输出脉冲”的不应期机制相比，动态权重的优势在于：可通过外部机制（如后续介绍的 WTA 电路）灵活控制不应期状态，例如在竞争电路中使所有参与竞争的神经元同时进入不应期，确保获胜神经元有再次竞争的机会。

### (3) 脉冲产生与膜电位重置

当膜电位  $V_{mp}$  超过阈值  $V_{th}$  时，神经元产生输出脉冲，同时膜电位按以下规则重置：

$$V_{mp}(t_p^+) = V_{mp}(t_p) - V_{th}$$

其中  $t_p^+$  为重置后的时刻。此外，文章设置膜电位的下限为  $V_{th}$ ，当膜电位低于该值时进行截断（clipping），避免神经元因膜电位过低而长期“沉默”，保证训练过程中神经元的参与度平衡。

## 2.1.2 赢家通吃（Winner-Take-All, WTA）电路

为提升 SNN 的特征提取能力和稀疏性，文章在部分网络层引入 WTA 电路。WTA 电路由多个神经元组成，神经元间通过侧向抑制（lateral inhibition）连接，当任一神经元产生输出脉冲时，会抑制其他神经元的膜电位，阻止其产生脉冲，从而实现“竞争获胜”的机制。

### (1) 侧向抑制机制

WTA 电路中，侧向抑制对神经元膜电位的影响与该神经元的阈值  $V_{th}$  成正比，具体表现为：对阈值小的神经元抑制弱，对阈值大的神经元抑制强。这种设计的优势在于：结合后续的阈值正则化（threshold regularization），活性高的神经元会具有更大的阈值，从而受到更强的抑制，平衡不同神经元的活性，避免部分神经元过度激活。

### (2) 计算成本优化

文章将所有侧向连接的强度设置为相同值，无需为每个连接学习独立权重，显著降低了 WTA 电路的内存占用和计算开销。同时，侧向抑制信号会触发所有被抑制神经元的动态权重  $w_{dyn}$  进入不应期，进一步强化竞争机制的稳定性。

## 2.2 适用于 SNN 的反向传播算法

传统反向传播算法依赖可微的激活函数，而 SNN 中的脉冲离散性导致梯度无法直接计算。文章的核心创新在于：将膜电位的低通滤波信号视为连续可微信号，脉冲时刻的不连续性视为噪声，从而推导适用于 SNN 的反向传播规则。

### 2.2.1 反向传播算法回顾

传统深度 ANN 的反向传播流程可概括为以下步骤：

(1) **前向传播**：计算各层的预激活值  $z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$  和激活值  $a^{(l)} = f^{(l)}(z^{(l)})$ （ $f$  为可微激活函数）；

- (2) 输出层误差计算:  $\delta^{(n_l)} = (a^{(n_l)} - y) \odot f'(z^{(n_l)})$ ,  $y$  为标签,  $\odot$  为元素-wise 乘法;
- (3) 误差反向传播: 对于隐藏层  $l = n_l - 1, \dots, 2$ ,  $\delta^{(l)} = (W^{(l+1)^T} \delta^{(l+1)}) \odot f'(z^{(l)})$ ;
- (4) 参数更新: 权重  $W^{(l)}$  和偏置  $b^{(l)}$  的梯度分别为  $\partial L / \partial W^{(l)} = \delta^{(l)} a^{(l-1)^T}$  和  $\partial L / \partial b^{(l)} = \delta^{(l)}$ , 再通过 SGD 更新参数。

### 2.2.2 SNN 中的传递函数与梯度推导

为将反向传播应用于 SNN, 文章首先定义了“连续可微的传递函数”, 核心是通过低通滤波将离散脉冲信号转换为连续活性值。

#### (1) 活性值定义

对于第  $k$  活性突触, 其对目标神经元膜电位的累积贡献 (归一化后) 定义为  $x_k(t)$ ; 对于神经元  $i$ , 其输出脉冲对自身膜电位的重置贡献 (归一化后) 定义为  $a_i(t)$ 。两者均表示为指数衰减项的和:

$$x_k(t) = \sum_{t_p < t} w_k e^{-(t-t_p)/\tau_{mp}},$$

$$a_i(t) = \sum_{t_q < t} V_{th} e^{-(t-t_q)/\tau_{mp}}.$$

其中  $t_p$  为输入脉冲时间,  $t_q$  出脉冲时间。  $x_k(t)$  和  $a_i(t)$  是连续信号, 仅在脉冲时刻存在跳变 (视为噪声), 因此可用于梯度计算。

#### (2) WTA 电路中的膜电位方程

考虑侧向抑制后, WTA 电路中第  $(i)$  个活性神经元的膜电位 (归一化后) 可表示为:

$$0 = x_i(t) - a_i(t) - \sigma V_{th,i} \sum_{j \neq i} \kappa_{ij} a_j(t)$$

其中:

右侧三项分别为输入贡献、重置贡献和侧向抑制贡献;

$\kappa_{ij}$  为神经元  $j$  对  $i$  的侧向抑制强度  $-1 \leq \kappa_{ij} \leq 0$ ;

$\sigma$  为侧向抑制的有效系数 (需小于 1, 确保膜电位不低于  $-V_{th}$ , 实验中取  $\sigma \approx 0.5$ )。

当  $\kappa_{ij} = 0$  时, 该方程退化为非 WTA 层的膜电位关系, 因此该模型可统一描述全连接和 WTA 层的 SNN。

#### (3) 梯度计算

忽略脉冲时刻的跳变 (视为噪声), 对上述膜电位方程求解  $a_i$  关于  $x_k$  的偏导数, 即可得到传递函数的梯度。当所有侧向抑制强度相同时 ( $\kappa_{ij} = \mu$ ), 梯度可简化为:

$$\frac{\partial a_i}{\partial x_k} \approx \frac{1}{\mu\sigma(n-1)+1} \cdot \frac{w_{ik}}{V_{th,i}}$$

其中  $n$  为 WTA 电路中的神经元数量。该梯度公式将离散脉冲的影响转化为连续的梯度信号，从而可将传统反向传播的误差传递规则适配到 SNN 中。

### 2.2.3 权重初始化与误差归一化

深度 SNN 训练中易出现梯度爆炸或消失问题, 文章提出了针对性的权重初始化方法和误差归一化策略:

#### (1) 权重初始化

第  $l$  层神经元的权重和阈值初始化公式为:

$$W^{(l)} \sim U\left[-\frac{\alpha}{\sqrt{M^{(l)}}}, \frac{\alpha}{\sqrt{M^{(l)}}}\right], \quad V_{th}^{(l)} \sim U[\alpha, 2\alpha]$$

其中:

$U[a, b]$  为区间  $[a, b]$  上的均匀分布;

$M^{(l)}$  为该层每个神经元的突触数量;

$\alpha$  为调节常数 (实验中取 3~10), 需根据层规模调整: 神经元数量少的层取较小  $\alpha$ , 以保证足够的脉冲发放率。

该初始化方法满足  $E\left[\sum_k \left(w_k^{(l)}\right)^2\right] = \alpha^2$ , 为后续误差归一化提供基础。

#### (2) 反向传播误差归一化

为平衡各层参数更新的幅度, 文章对反向传播的误差  $\delta^{(l)}$  进行归一化:

$$\delta_i^{(l)} = \frac{g_i^{(l)}}{\overline{g^{(l)}}} \cdot \delta_i^{(l, \text{raw})}$$

其中:

$g_i^{(l)} = 1/V_{th,i}^{(l)}$ , 与神经元阈值成反比;

$\overline{g^{(l)}} = \sqrt{\frac{1}{n^{(l)}} \sum_{i=1}^{n^{(l)}} \left(g_i^{(l)}\right)^2}$ , 为  $g_i^{(l)}$  的均方根;

归一化后可保证  $E\left[\sum_i \left(\delta_i^{(l)}\right)^2\right] = 1$ , 避免梯度爆炸或消失。

### (3) 参数更新规则

权重和阈值的更新公式为：

$$W^{(l)} \leftarrow W^{(l)} - \eta_w \cdot \delta^{(l)} \widehat{x^{(l-1)T}},$$

$$V_{th}^{(l)} \leftarrow V_{th}^{(l)} - \eta_{th} \cdot \delta^{(l)} \widehat{a^{(l)T}}$$

其中：

$\eta_w$  和  $\eta_{th}$  分别为权重和阈值的学习率（实验中  $\eta_{th} = 0.1\eta_w$ ，避免阈值过小导致脉冲爆炸）；

$\hat{x}$  和  $\hat{a}$  为有效活性值，已融入侧向抑制的影响。

## 2.3 正则化技术

为提升 SNN 的泛化能力和训练稳定性，文章提出了\*\*权重正则化\*\*和\*\*阈值正则化\*\*两种策略，分别解决权重漂移和神经元“死亡”问题。

### 2.3.1 权重正则化（Exponential Regularization）

传统 L2 正则化会导致权重先快速增长后持续下降，无法维持初始化时的权重分布；max-norm 正则化虽能限制权重上限，但泛化能力有限。文章提出“指数正则化”，其损失函数为：

$$L_w(l, i) = \beta \cdot e^{\lambda(\sum_k (w_k^{(l,i)})^2 - \alpha^2)}$$

其中：

$\beta$  和  $\lambda$  为正则化参数（实验中  $\beta = 10$ ,  $\lambda = 0.002 \sim 0.04$ ）；

$\alpha^2$  为初始化时权重平方和的期望。

该正则化的优势在于：

- 当权重平方和超过  $\alpha^2$  时，指数项快速增长，强抑制权重增大（类似 max-norm）；
- 当权重平方和低于  $\alpha^2$  时，指数项衰减，弱抑制权重减小（类似 L2）；
- 始终维持权重在合理范围内，保证神经元的脉冲发放稳定性。

### 2.3.2 阈值正则化（Threshold Regularization）

SNN 初始化时阈值较大，易导致部分神经元因膜电位难以达到阈值而长期“沉默”（dead neurons），尤其在 WTA 电路中更为严重。文章提出的阈值正则化机制如下：

#### (1) 阈值调整规则

对于接收同一输入的  $N$  个神经元，若有  $N_w$  个神经元产生脉冲，则：

- 产生脉冲的神经元：阈值增加  $(\rho N)$  ( $\rho$  为调整系数, 实验中取  $0.00004 \sim 0.0002$ );
- 未产生脉冲的神经元：阈值减少  $\rho N_w$ 。

这种调整使得活性高的神经元阈值升高(降低后续活性), 活性低的神经元阈值降低(提升后续活性), 平衡所有神经元的参与度。

## (2) 阈值下限约束

为避免阈值过小导致脉冲爆炸, 设置阈值的下限(实验中取初始阈值的  $1/5$ )。当阈值更新后低于下限时, 不降低阈值, 而是按比例增加该神经元的所有突触权重, 确保膜电位的累积速度与阈值匹配。

## 2.4 目标函数与训练流程

### 2.4.1 目标函数

文章采用均方误差 (MSE) 作为损失函数, 并融入权重正则化项:

$$L = \frac{1}{2}|o - y|^2 + \sum_{l \in \text{hidden}} \sum_i L_w(l, i)$$

其中:

- $o$  为输出层的归一化脉冲数, ( $o_i = \text{spike}_i / \max_j(\text{spike}_j)$ ) ( $\text{spike}_i$  为第  $i$  个输出神经元的脉冲数);
- 归一化采用最大值而非求和, 避免输出层权重正则化的需求, 简化计算。

### 2.4.2 训练流程

整个训练过程基于事件驱动, 无需离散时间步长, 具体步骤如下:

(1) **事件生成**: 对于输入样本(如 MNIST 图像), 生成脉冲事件流(如基于像素强度的泊松脉冲);

#### (2) 前向传播:

- 按 LIF 模型更新神经元膜电位, 产生输出脉冲;
- 计算活性值  $x_k(t)$  和  $a_i(t)$  (仅用于训练);
- 执行阈值正则化, 调整神经元阈值;

(3) **误差计算**: 统计输出层脉冲数, 计算归一化输出  $o$  和损失  $L$ ;

(4) **反向传播**: 基于活性值的梯度, 计算各层误差  $\delta^{(l)}$ , 更新权重和阈值;

(5) **迭代优化**: 重复上述步骤, 直至训练收敛(实验中训练 150~200 个 epoch)。

### 三、实验设计与结果分析

为验证所提方法的有效性,文章在三个经典基准任务上进行实验:排列不变 MNIST (PI MNIST)、标准 MNIST 和神经形态 MNIST (N-MNIST),分别测试全连接 SNN 和卷积 SNN 的性能,并与现有方法对比。

#### 3.1 实验设置

##### 3.1.1 数据集与事件生成

- PI MNIST: 对 MNIST 图像的像素进行随机排列,破坏空间结构,仅测试全连接网络的分类能力;输入事件为泊松脉冲,总事件率归一化为 5k eps (每非零像素平均 ~43 eps),训练时输入时长 50ms,测试时 1s (确保结果稳定)。
- 标准 MNIST: 保留图像空间结构,用于测试卷积 SNN;采用弹性畸变(elastic distortion)进行数据增强,提升泛化能力。
- N-MNIST: 通过 DVS 传感器拍摄移动的 MNIST 图像生成,包含动态时空事件流;每个样本时长 300ms (分 3 个 100ms 的扫视阶段),输入层尺寸为 34×34×2 (2 个通道分别对应亮度增加/减少事件)。

##### 3.1.2 网络结构与参数

表 1 实验中使用的网络结构与关键参数

任务	网络类型	结构细节	关键参数
PI MNIST	全连接 SNN	输入层 784 神经元, 1~2 个隐藏层 (100~800 神经元), 输出层 10 神经元; 隐藏层和输出层采用 WTA 电路	$\tau_{mp} = 20 \text{ ms}$ , $T_{ref} = 1 \text{ ms}$ , $\alpha = 3 \sim 10$ , $\eta_w = 0.002 \sim 0.004$
标准 MNIST	卷积 SNN	2 个卷积层 (5×5 核, 分别 20/50 特征图) + 2×2 求和池化, 1 个全连接隐藏层 (200 神经元), 输出层 10 神经元	同 PI MNIST, 新增弹性畸变数据增强
N-MNIST	全连接 SNN	输入层 34×34×2=2312 神经元, 1 个隐藏层 (800 神经元), 输出层 10 神经元	$\tau_{mp} = 200 \text{ ms}$ , 训练时随机选取 300 个事件, 测试时使用全事件流

##### 3.1.3 对比方法

实验对比了传统 ANN、现有 SNN 方法与所提方法, 主要包括:

- ANN: Dropout、DropConnect、Maxout等;
- 现有 SNN: 脉冲深度信念网络 ANN 转换权重阈值平衡转换 CNN 转换。

#### 3.2 PI MNIST 实验结果



PI MNIST 用于验证全连接 SNN 的性能，核心是测试方法对非空间结构数据的拟合能力。

### 3.2.1 WTA 电路的影响

文章首先分析了侧向抑制强度对 accuracy 的影响,结果如图 1 所示。从图中可观察到:

- 当侧向抑制强度为  $-0.4$  时，不同隐藏层规模（100、200、300 神经元）的 SNN 均达到最高 accuracy;
- 输出层的最优侧向抑制强度为  $-1.0$ ，这是因为输出层需要更强烈的竞争以明确分类结果。

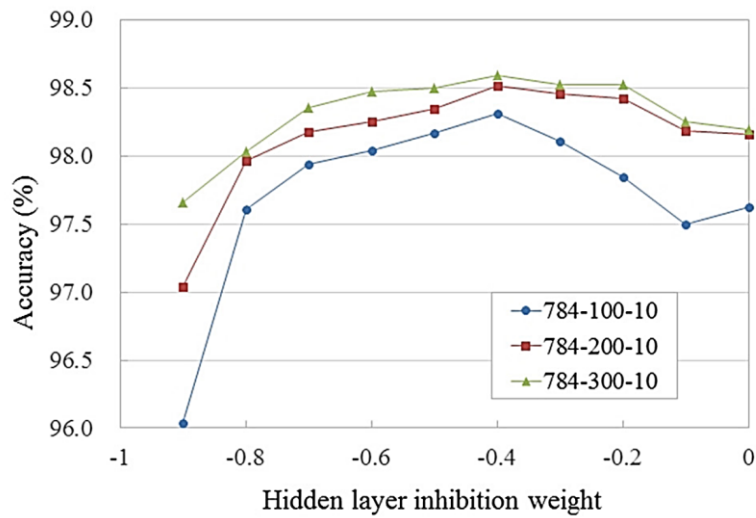


图 1 隐藏层中横向抑制强度与准确率的关系，针对 PI MNIST。网络使用相同的初始权重进行训练。数值为第[131, 150]轮的平均值。

此外，图 2 展示了 784-100-10 SNN 隐藏层的训练后权重与阈值，可见 WTA 电路使权重分布更稀疏，提升了特征的判别性。

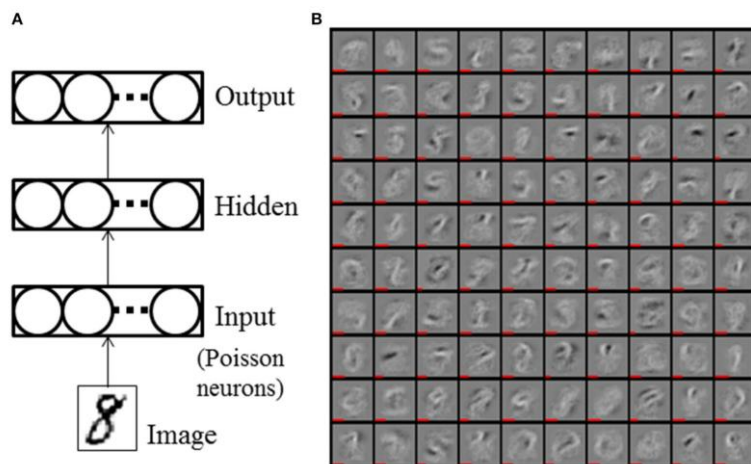


图 2 (A) 单隐藏层 SNN。(B) 784-100-10 SNN 隐藏层的训练后权重值，带有 WTA 电路。

权重示意图下红色条的长度表示神经元的阈值。

### 3.2.2 不同网络结构的性能对比

表 2 对比了不同方法在 PI MNIST 上的测试 accuracy，核心结果如下：

- 单隐藏层 SNN：使用 SGD 训练时，最高 accuracy 达 98.64%，与 Diehl et al. (2015) 的现有 SNN 最佳结果持平，且超过传统单隐藏层 ANN (98.4%)；
- 双隐藏层 SNN：使用 SGD 时 accuracy 为 98.7%，采用 ADAM 优化器后进一步提升至 98.77%，接近使用 Dropout 或 DropConnect 的深度 ANN (98.8%~98.99%)；
- 优势：所提方法无需无监督预训练或数据增强，仅通过直接训练脉冲信号即达到同等精度，证明了其对脉冲统计特性的捕捉能力。

表 2 不同模型在 PI MNIST 上的准确率比较。

模型	隐藏层单元数	测试精度 (%)
人工神经网络 (Srivastava 等人, 2014)	800	98.4
人工神经网络 (Srivastava 等人, 2014), 含 Drop-out 机制	4096-4096	98.99
人工神经网络 (Wan 等人, 2013), 含 Drop-connect 机制	800-800	98.8
人工神经网络 (Goodfellow 等人, 2013), 含 maxout 机制	240×5-240×5	99.06
脉冲神经网络 (O'Connor 等人, 2013)	500-500	94.09
脉冲神经网络 (Hunsberger 和 Eliasmith, 2015)	500-300	98.6
脉冲神经网络 (Diehl 等人, 2015)	1200-1200	98.64
脉冲神经网络 (O'Connor 和 Welling, 2016)	200-200	97.8
脉冲神经网络 (本文方法, SGD 优化)	800	[98.56, 98.64, 98.71]
脉冲神经网络 (本文方法, SGD 优化)	500-500	[98.63, 98.70, 98.76]
脉冲神经网络 (本文方法, ADAM 优化)	300-300	[98.71, 98.77, 98.88]

### 3.3 标准 MNIST 实验结果

标准 MNIST 用于验证卷积 SNN 的性能，利用空间结构提升分类精度。

3.3.1 卷积 SNN 的设计

文章设计的卷积 SNN 结构为：输入层（28×28）→ 卷积层 1（5×5 核，20 特征图）→ 池化层 1（2×2 求和池化）→ 卷积层 2（5×5 核，50 特征图）→ 池化层 2（2×2 求和池化）→ 全连接层（200 神经元）→ 输出层（10 神经元）。其中，池化层通过“权重=1、阈值=0.8”的 LIF 神经元实现近似求和池化。

3.3.2 性能对比

表 3 对比了不同卷积模型在标准 MNIST 上的 accuracy，结果显示：

- 所提卷积 SNN 在使用弹性畸变数据增强后，测试 accuracy 达 99.31%，超过 Diehlet al. (2015) 的 SNN (99.1%) 和 Garbin et al. (2014) 的传统 CNN (98.3%)；
- 虽低于 Esser et al. (2015) 基于 64 个网络集成的结果 (99.42%)，但所提方法仅使用单个网络，且无需复杂的预处理，在效率上更具优势。

表 3 MNIST 基准测试中脉冲卷积神经网络（CNN）模型的精度对比

网络模型	预处理方式	集成数量	测试精度 (%)
卷积神经网络（Garbin 等人，2014）	无	1	98.3
卷积神经网络 (Diehl 等人，2015)	无	1	99.1
稀疏连接网络 (Esser 等人，2015)	仿射变换	64	99.42
卷积神经网络（本文方法）	弹性畸变	1	99.31

3.4 N-MNIST 实验结果

N-MNIST 是最具挑战性的任务，需处理动态时空事件流，直接测试 SNN 对事件驱动传感器输入的处理能力。

3.4.1 事件流处理特点

N-MNIST 与传统 MNIST 的核心区别在于：

- 事件流非泊松分布，而是由 DVS 传感器的亮度变化产生；
- 数字位置随扫视运动变化，静态累积事件会导致图像模糊，需利用时空信息；
- 输入包含“on”（亮度增加）和“off”（亮度减少）两种事件，需双通道处理。

3.4.2 性能对比

表 4 对比了不同方法在 N-MNIST 上测试精度，核心结果如下：

- 精度优势：所提方法的全连接 SNN (2312-800-10) 测试 accuracy 达 98.66% (误差率 1.34%)，远超现有 SNN 方法：Neil et al. (2016) 的卷积 SNN (95.72%)、Cohen et al. (2016) 的大隐藏层 SNN (92.87%)；同时超过 Neil et al. (2016) 的传统 ANN (98.3%)，成为当时 N-MNIST 任务的最佳结果；
- 效率优势：现有方法需对事件流进行“中心化”预处理（消除位置变化），而所提方法直接使用原始未预处理事件流，且隐藏层仅 800 神经元，远少于 Cohen et al. (2016) 的 10k 神经元。

表 4 N-MNIST 数据集上不同模型的精度对比

网络模型	隐藏层单元数	是否进行中心 化预处理	测试精度 (%)
人工神经网络 (Neil 和 Liu, 2016)	卷积神经网络 (无具体单元数，以结构代指)	是	98.3
脉冲神经网络 (Neil 和 Liu, 2016)	卷积神经网络 (无具体单元数，以结构代指)	是	95.72
脉冲神经网络 (Cohen 等人, 2016)	10,000	否	92.87
脉冲神经网络 (本文方法)	800	否	[98.56, 98.66, 98.74]

3.4.3 时间动态特性与计算成本

文章进一步分析了 SNN 的时间动态特性和计算成本：

- 时间动态精度：图 2 展示了 SNN 随时间累积事件的误差率变化，可见：
  - [1] 误差率随事件累积快速下降，第一个扫视阶段 (0~100ms) 即可实现低误差；
  - [2] 每次扫视阶段的事件率峰值对应误差率的大幅下降，证明 SNN 能有效利用时空事件的动态信息。
- 计算成本对比：图 3 对比了 SNN 与 ANN 的计算成本，其中：
  - [1] ANN 需将事件流累积为 30ms 帧，通过 10 帧集成实现 2.2% 的误差率，总计算量 (MAC 操作) 约为 5M；
  - [2] SNN 直接处理事件流，达到同等误差率时的计算量 (突触操作) 仅约 1M，是 ANN 的 1/5，证明了 SNN 在事件驱动场景下的能效优势。

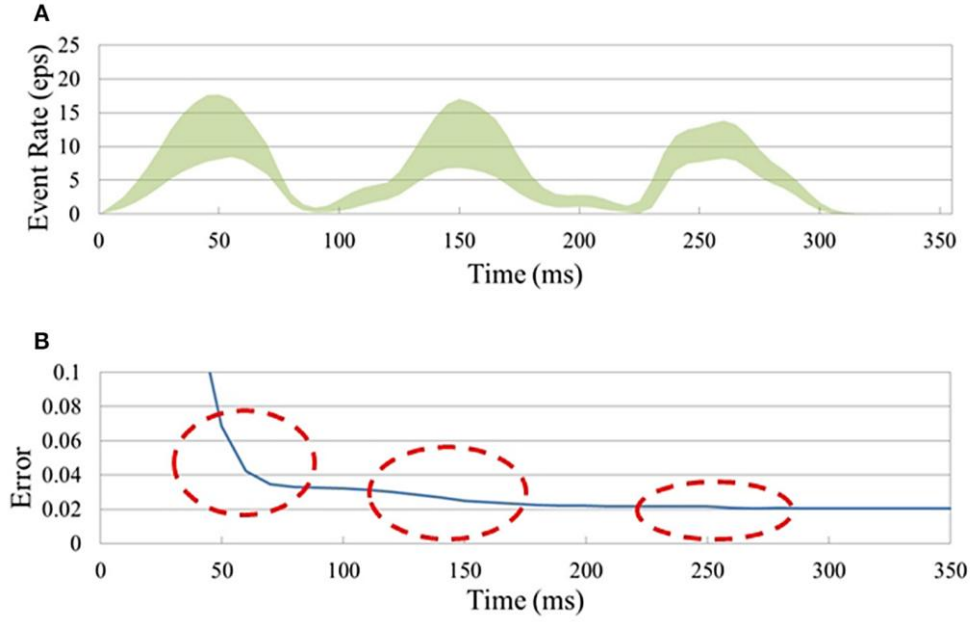


图 5 N-MNIST 神经形态数据集的分类。(A) 每像素的瞬时输入事件率（即，总事件率除以  $34 \times 34 \times 2$ ）（ $\pm$  标准差），平均取自 10,000 个 N-MNIST 测试样本。(B) 脉冲神经网络（SNN）在整合先前和当前推理结果时会随时间自然提升其准确性。当输入事件率最高时（见 A）， $(34 \times 34 \times 2)$ -200-10 SNN 的准确率出现了显著跃升（红圈标示）。

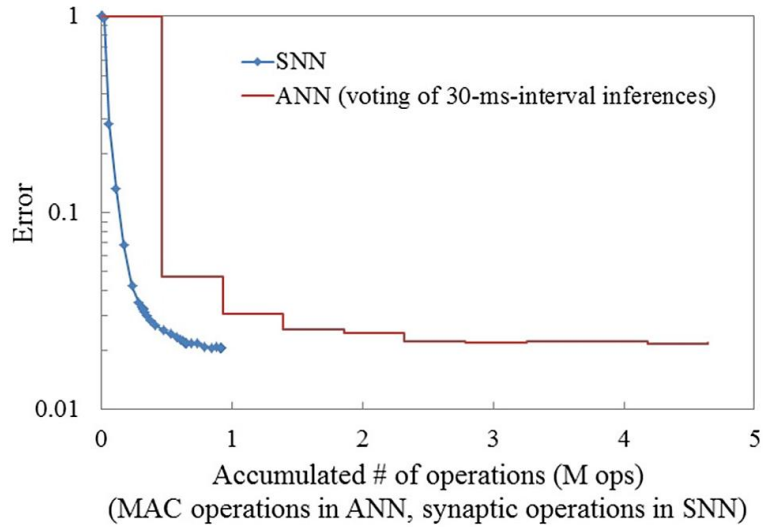


图 6 在 N-MNIST 任务中，ANN 与 SNN 在推理过程中计算成本的比较（ANN 的 MAC 操作数量，SNN 的突触操作数量）。SNN 和 ANN 具有相同的架构： $(34 \times 34 \times 2)$ -200-10。为了解决 ANN 情况下数字移动的问题，输入脉冲流累积了 30 毫秒并转化为帧。随后，每一帧的推理结果会随时间进行整合以提高准确性。SNN 远早于 ANN 达到其最佳准确率，而 ANN 则需要整合多帧才能达到相近的准确率。

## 四、讨论与展望

### 4.1 方法的核心优势

文章提出的深度 SNN 训练方法在理论和实践上均具有重要突破,核心优势可总结为三点:

- 直接训练脉冲信号:与“ANN 转换”方法不同,该方法直接基于脉冲序列和膜电位训练,无需通过连续信号近似脉冲统计特性,能够精确捕捉事件驱动传感器产生的脉冲时序信息(如同步性、动态变化),因此在 N-MNIST 等动态任务中表现更优。
- 通用性强:该框架可统一处理全连接 SNN、卷积 SNN 和 WTA 电路,且兼容 LIF 神经元的泄漏和不应期特性,无需针对不同网络结构设计专用训练规则。此外,通过 ADAM 等优化器的适配,可进一步提升深度网络的训练稳定性和精度。
- 能效优势显著:SNN 的事件驱动特性在 N-MNIST 任务中得到充分体现——达到与 ANN 同等精度时,计算量仅为 ANN 的 1/5。这一优势在边缘设备、实时机器人等能效敏感场景中具有重要应用价值,尤其随着 neuromorphic 硬件(如 IBM TrueNorth)的发展,SNN 的能效潜力将进一步释放。

## 4.2 现有局限性

尽管实验结果优异,该方法仍存在一些局限性,需在后续研究中改进:

- 数据集规模有限:实验主要基于 MNIST 及其变体(PIMNIST、N-MNIST),这些数据集的样本量(10k 测试样本)和复杂度远低于 ImageNet 等大规模数据集。作者也指出,MNIST 并非 SNN 的理想基准(无真实传感器脉冲),未来需在更大规模、更复杂的事件驱动数据集(如 DVS 录制的 ImageNet)上验证方法的扩展性。
- 训练效率问题:当前训练依赖 GPU 模拟事件驱动的前向传播,而脉冲事件的异步性导致 GPU 并行计算效率较低(GPU 更适合同步的批量计算)。此外,反向传播过程中需存储活性值 $x_k(t)$ 和 $a_i(t)$ ,增加了内存开销。
- neuromorphic 硬件适配挑战:现有 neuromorphic 硬件(如 SpiNNaker)的数值精度较低(通常为 8~16 位),且难以实时测量脉冲统计特性以支持反向传播。如何将该训练方法适配到专用硬件,实现“硬件上的端到端训练”,仍是未来研究的难点。

## 4.3 未来研究方向

基于文章的局限性和 SNN 领域的发展趋势,未来可从以下方向展开研究:

- 扩展到更大规模任务:将方法应用于 ImageNet 级别的图像识别、视频分析等复杂任务,需设计更深的 SNN 结构(如残差连接、注意力机制),并解决大规模事件流的高效处理问题(如稀疏卷积、事件采样)。
- 硬件加速训练:针对 SNN 事件驱动的特性,设计专用硬件架构(如基于 FPGA 或 ASIC),优化前向传播的并行性和反向传播的梯度计算效率。例如,利用 SpiNNaker 等 neuromorphic 平台的分布式计算能力,实现大规模 SNN 的实时训练。
- 融合递归结构:文章仅研究了前馈 SNN,未来可将方法扩展到递归脉冲神经网络(RNN-SNN),利用 RNN 对时序信息的建模能力,处理语音识别、动态场景理解等更复杂的时序任务。

- 改进神经元模型：当前使用的 LIF 模型简化了生物神经元的许多特性（如 STDP、多 compartments），未来可结合更复杂的神经元模型（如 Izhikevich 模型），进一步提升 SNN 的生物合理性和计算能力。

## 五、总结

Lee 等人在《Training Deep Spiking Neural Networks Using Backpropagation》一文中，针对深度 SNN 训练的核心瓶颈（脉冲离散性导致反向传播不可用），提出了一种基于“膜电位可微化”的创新训练方法。该方法通过将膜电位的低通滤波信号视为连续可微信号、脉冲时刻的不连续性视为噪声，成功推导了适用于 SNN 的反向传播规则，并结合权重正则化、阈值正则化和误差归一化等技术，实现了深度 SNN 的稳定训练。

实验结果表明，该方法在 PIMNIST、标准 MNIST 和 N-MNIST 任务上均取得了优异性能：在 PIMNIST 上达到 98.77% 的精度，接近深度 ANN；在标准 MNIST 上卷积 SNN 精度达 99.31%；在 N-MNIST 上更是以 98.66% 的精度刷新当时最佳结果，且计算量仅为 ANN 的 1/5。这些成果不仅缩小了 SNN 与 ANNs 的精度差距，还为 SNN 在事件驱动、能效敏感场景中的应用奠定了基础。

尽管方法在数据集规模、训练效率和硬件适配方面仍有改进空间，但该研究无疑为深度 SNN 的训练提供了一个通用、高效的框架，对后续 SNN 理论研究和实际应用具有重要的指导意义。对于脉冲神经网络领域的研究者而言，该文章不仅是学习深度 SNN 训练技术的经典文献，更是理解“如何将传统深度学习方法与生物神经元特性结合”的重要参考。