

基于双阈值的 ANN-SNN 转换方法优化

何赞泽^{1,2}, 张天安¹, 邓堡元¹, 王洪金^{1,2}, 王耀南^{1,2}

(1. 湖南大学 电气与信息工程学院, 长沙 410082;

2. 机器人视觉感知与控制技术国家工程研究中心, 长沙 410082)

摘要: 脉冲神经网络作为第三代神经网络, 能够克服许多人工神经网络中所存在的问题, 如高功耗、鲁棒性较差等; 通过对预训练好的人工神经网络模型进行转换是获取深度脉冲神经网络模型的一种主要方法, 然而通过这种方法获取的脉冲神经网络的延迟较高, 无法满足实时性要求; 文章在双阈值转换方法的基础上, 采用阈值平衡技术对转换过程进行优化, 通过理论推导, 提出了一种对称阈值 LeakyReLU 激活函数, 并对人工神经网络到脉冲神经网络的转换流程进行了梳理; 此外, 采用了泄漏机制对转换后的脉冲神经网络模型结构进行了优化, 并通过脉冲时序依赖可塑性学习规则对该结构进行训练; 最终, 在 MNIST 数据集与 CIFAR-10 数据集上进行了实验, 结果表明, 优化后脉冲神经网络的收敛速度与鲁棒性得到了大幅提升。

关键词: ANN-SNN 转换; 双阈值; 阈值平衡; 脉冲时序依赖可塑性; 泄漏机制

Optimization of ANN-SNN Conversion Method with Double Threshold

HE Yunze^{1,2}, ZHANG Tianan¹, DENG Baoyuan¹, WANG Hongjin^{1,2}, WANG Yaonan^{1,2}

(1. School of Electrical and Information Engineering, Hunan University, Changsha 410082, China;

2. Robot Visual Perception and Control Technology of National Engineering Research Center, Changsha 410082, China)

Abstract: As a third-generation neural network, spiking neural network (SNN) can overcome many problems in artificial neural networks, such as high power consumption and poor robustness. Transforming the pre-trained artificial neural network (ANN) model is a primary method for obtaining deep spiking neural network model, but this spiking neural network has a high latency and cannot meet real-time requirements. On the basis of the double threshold conversion method, the threshold balance technology is used to optimize the transition process. Through theoretical derivation, a symmetric threshold LeakyReLU activation function is proposed, and the transition process from artificial neural network to spiking neural network is sorted out. In addition, the leakage mechanism is used to optimize the structure of the transformed spiking neural network model, and the structure is trained by the spiking-timing-dependent plasticity learning rule. Finally, the experiments are carried out on the MNIST dataset and the CIFAR-10 dataset, and the results show that the optimized spiking neural network greatly improves the convergence speed and robustness.

Keywords: ANN-SNN transition; double thresholds; threshold balance; spiking-timing-dependent plasticity; leakage mechanism

0 引言

随着图形处理器算力的不断增强, 人工神经网络 (ANN, artificial neural networks) 在图像识别、自然语言处理^[1-5]等方面都取得了显著的进展。然而, 强大的性能却是以高能耗为代价。脉冲神经网络 (SNN, spiking neural networks) 作为第三代神经网络, 相比于传统 ANN 模型, 利用稀疏的脉冲来传递信息, 类似于人脑的信息处理机制, 克服了人工神经网络中的高能耗问题。与人工神经网络相比, 脉冲神经网络可以将能耗降低几个数量级^[6], 对于移动端低功耗需求的场景有着重要的研究意义。近年来, 在

无人驾驶、医疗健康、工业检测等领域受到了广泛的关注^[7-10]。

获取 SNN 模型的方法包括直接训练与间接训练两种。其中, 直接训练又分为有监督学习以及无监督学习。由于 SNN 中的信号以离散的脉冲形式进行传递, 而不像 ANN 中的信号是连续可微的, 因此, 传统的误差反向传播算法不能直接应用于 SNN, 这使得早期许多有关 SNN 的研究都是基于无监督学习算法进行的^[11]。而由于脉冲信号离散且不可微的性质, SNN 的有监督学习算法也受到了国内外众多学者的关注, 目前已经有多种有监督的学习算法被提出^[12-15]。间接训练则是通过将预训练好的 ANN 模型转换为

收稿日期: 2024-04-19; 修回日期: 2024-05-29。

基金项目: 国家自然科学基金(62101184); 湖南省科技创新领军人才(2023RC1039); 湖南省自然科学基金重大项目(2021JC0004)。

作者简介: 何赞泽(1983-), 男, 教授, 博导。

王洪金(1986-), 女, 副教授, 博导。

王耀南(1957-), 男, 院士, 博导。

引用格式: 何赞泽, 张天安, 邓堡元, 等. 基于双阈值的 ANN-SNN 转换方法优化[J]. 计算机测量与控制, 2024, 32(11): 271-277.

SNN 模型来实现。这种做法的好处是，可以利用较为成熟的 ANN 训练算法训练一个深层神经网络模型，然后通过对预训练模型的转换，直接得到一个深层 SNN 模型。该方法主要关注的问题在于模型转换过程中所产生的误差，以及转换后 SNN 模型的收敛速度。目前，已有相当多的学者对其进行了研究。Cao 等^[16]最早提出了较为详细的 ANN 到 SNN 转换的方法，并在 DARPA 与 CIFAR-10 数据集上取得了不错的成绩。Yu 等^[17]基于 LeakyReLU 激活函数，提出了双阈值 SNN 的转换方法，并在 MNIST 与 CIFAR-10 的数据集上验证了双阈值方法的有效性，有效提高了 SNN 模型的准确率。尽管上述方法已经极大的降低了转换过程中所产生的误差，然而由于神经元的激活范围不同，会导致 SNN 模型的收敛需要很长的时间。Ding 等^[18]通过对 ANN 与 SNN 前向传播的表达式进行分析，提出了一种 RNL 结构来代替 ANN 中的激活函数对模型进行训练，提高了 SNN 模型的收敛速度。Deng 等^[19]结合阈值平衡与软重置机制，提出了一种阈值 ReLU 激活函数，使得 SNN 模型的收敛速度得到了进一步提升。

本文首先在双阈值 SNN 转换方法的基础上，基于阈值平衡技术，提出了一种对称阈值 LeakyReLU 激活函数 (sthLeakyReLU, symmetric threshold LeakyReLU)，在保证 SNN 模型准确率的同时，大幅加快了模型的收敛速度。其次，本文对 ANN 模型到 SNN 模型转换的具体流程进行了梳理，并基于 VGG 与 ResNet 骨干网络，在 MNIST 与 CIFAR-10 数据集上对转换后的 SNN 模型进行了实验，证明了所提出方法的有效性。最后，本文基于漏电积分放电 (LIF, leaky integer-and-fire) 神经元模型，使用泄漏机制对 SNN 模型的输出结构进行了优化，并使用脉冲时序依赖可塑性 (STDP, spike-timing-dependent plasticity) 学习规则对该结构进行训练，并在 CIFAR-10 数据集上进行了消融实验，对优化结构的有效性进行验证。

1 ANN-SNN 理论依据

Cao 等^[16]的实验结果表明，SNN 的神经元放电频率在足够长的时间步下，会逐渐逼近于 ANN 的神经元输出结果，通常使用 IF 神经元模型的发射频率去逼近 ReLU 激活函数的输出值。具体而言，对于使用 ReLU 作为激活函数的 ANN 模型了，第 l 层的神经元的输出值可以表示为：

$$x_l = \max(w_{l-1}x_{l-1} + b_{l-1}, 0) \quad (1)$$

式中， x_l 表示第 l 层神经元的输出值； w_{l-1} 和 b_{l-1} 分别表示第 l 层神经元与 $l-1$ 层神经元连接的权重与偏置。

对于 IF 神经元，使用软重置的方式对膜电位进行复位，则输入与输出的关系可以表示为：

$$m_l(t) = v_l(t-1) + W_{l-1}s_{l-1}(t) + b_{l-1}, \quad (2)$$

$$s_l(t) = \delta(m_l(t) - v_{th}), \quad (3)$$

$$v_l(t) = m_l(t) - s_l(t)v_{th}. \quad (4)$$

式中， $m_l(t)$ 表示第 l 层 t 时刻 IF 神经元膜电位的大小； $v_l(t-1)$ 则表示第 l 层在 $t-1$ 时刻积累的电压值； W_{l-1} 、 b_{l-1} 分别

表示 l 层与 $l-1$ 层神经元相连的权重与偏置大小； v_{th} 表示阈值电压； $\delta(\cdot)$ 表示一个脉冲函数。

式 (2)、(3) 描述了 SNN 中信号前向传播的过程。与 ANN 不同的是，SNN 通过记录 $t-1$ 时刻的电压积累量为网络引入了时序信息，通过膜电位 $m_l(t)$ 来表示当前的神经元电压大小，当超过阈值电压 v_{th} 时，则产生一个脉冲信号，将信息向后传播。神经元发射脉冲后，将会被复位，如式 (4) 所示。

由式 (2)、(4)，可以得出相邻两层神经元之间的电压关系为：

$$v_l(t) - v_l(t-1) = W_{l-1}s_{l-1}(t) + b_{l-1} - s_l(t)v_{th}. \quad (5)$$

假设 SNN 仿真的时间步长为 T ，将式 (5) 等式两边由 $t=1$ 到 T 进行累加，可以得出：

$$v_l(T) = W_{l-1} \sum_{t=1}^T s_{l-1}(t) + Tb_{l-1} - \sum_{t=1}^T s_l(t)v_{th}. \quad (6)$$

对于 SNN，神经元脉冲发射频率定义为：

$$r_l = \frac{\sum_{t=1}^T s_l(t)}{T}, \quad (7)$$

将式 (6) 两边同时除以 Tv_{th} ，当 T 足够大时，结合式 (7) 可以得出：

$$r_l = \text{clip}\left(\frac{W_{l-1}r_{l-1} + b_{l-1}}{v_{th}}, 0, 1\right) \quad (8)$$

式中， $\text{clip}(x, a, b)$ 输出值的范围在 $[a, b]$ 区间，当 $x \in [a, b]$ 时， $\text{clip}(x, a, b) = x$ ；当 $x \leq a$ 时， $\text{clip}(x, a, b) = a$ ；当 $x \geq b$ 时， $\text{clip}(x, a, b) = b$ 。

式 (1)、(8) 分别表示 ANN 与 SNN 信息前向传播的过程。显然，对于 $x > 0$ 的部分，ANN 与 SNN 的表述是不同的。因此，还需要对 ANN 模型进行一定转换，才能得到 SNN 中所需的权重，这种转换通常是对预训练好的 ANN 权重进行的。为了方便表述，这里设 W^{ANN} 为预训练的 ANN 权重， W^{SNN} 为转换后的 SNN 权重。

对于每一层神经元的输出，取所有输出中的最大值，此时，式 (1) 可以改写为：

$$x_l = \text{clip}(W_{l-1}^{\text{ANN}}x_{l-1} + b_{l-1}, 0, \max_l) \quad (9)$$

式中， \max_l 表示每一层神经元输出的最大值。

使用 \max_l 对第 l 层神经元的输出进行归一化处理，并带入式 (9)，可以得到：

$$y_{l,i} = \text{clip}\left(\frac{W_{l-1}^{\text{ANN}}y_{l-1,i}\max_{l-1} + b_{l-1}}{\max_l}, 0, 1\right), y_{l,i} \in [0, 1] \quad (10)$$

式 (10) 描述了一个与式 (8) SNN 前向传播过程等价的 ANN 信号传递公式，结合式 (8) 与式 (10)，可以得出 ANN 与 SNN 权重转换关系为：

$$\begin{cases} W_l^{\text{SNN}} = \frac{v_{th}\max_{l-1}}{\max_l} W_l^{\text{ANN}}, \\ b_l^{\text{SNN}} = \frac{v_{th}}{\max_l} b_l^{\text{ANN}} \end{cases} \quad (11)$$

2 转换方法优化

基于式 (11) 可以将一个训练好的 ANN 模型权重转换为 SNN 权重。然而, 该方法却存在着一定的局限性。首先, 在 ANN 模型中, 通常使用 ReLU 激活函数, 这导致 SNN 模型仅具有正脉冲, 从而限制了其表述能力。另一个主要问题是不同神经元的激活范围不同, 这导致激活具有高激活阈值的神经元需要很长的时间步, 从而导致模型的收敛速度大幅降低。基于上述分析, 本章提出了一种 sthLeakyReLU 激活函数, 在引入负脉冲的同时, 利用阈值平衡技术^[19], 使得 SNN 可以具有更高的模型精度以及更快的收敛速度。

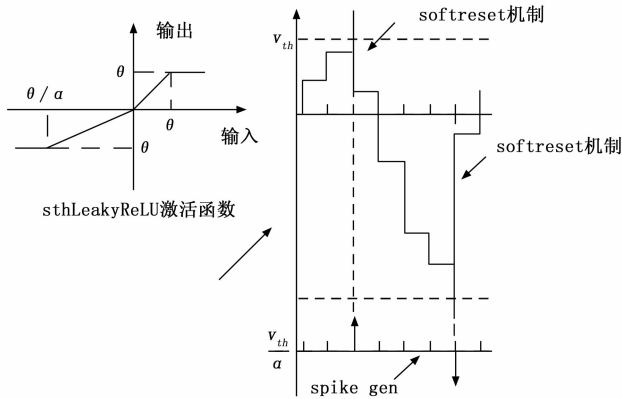


图 1 sthLeakyReLU 激活函数设计

由于负脉冲的引入, 式 (7) 中 SNN 神经元脉冲的发射频率的范围 $r_i \in [-1, 1]$, 此时, SNN 发射频率的前向传播公式为:

$$r_i = \begin{cases} \text{clip}(\frac{W_{i-1}^{\text{SNN}} r_{i-1} + b_{i-1}}{v_{th}^p}, 0, 1), & W_{i-1}^{\text{SNN}} r_{i-1} + b_{i-1} \geq 0, \\ \text{clip}(\frac{W_{i-1}^{\text{SNN}} r_{i-1} + b_{i-1}}{v_{th}^n}, -1, 0), & W_{i-1}^{\text{SNN}} r_{i-1} + b_{i-1} < 0 \end{cases} \quad (12)$$

此时, SNN 每一层神经元的输入与输出被限制在了 $[-1, 1]$ 之间, 而对于 ANN, 假设 LeakyReLU 激活函数输出结果的绝对值上限为 θ , 则有:

$$x_i = \begin{cases} \text{clip}(W_{i-1}^{\text{ANN}} x_{i-1} + b_{i-1}, 0, \theta), & W_{i-1}^{\text{ANN}} x_{i-1} + b_{i-1} \geq 0, \\ \text{clip}(\alpha(W_{i-1}^{\text{ANN}} x_{i-1} + b_{i-1}), -\theta, 0), & W_{i-1}^{\text{ANN}} x_{i-1} + b_{i-1} < 0 \end{cases} \quad (13)$$

式中, α 表示 LeakyReLU 激活函数在负半轴的斜率。

将式 (12) 与式 (13) 建立起等价关系, 则需要对 ANN 的激活函数进行归一化处理。由于 ANN 每层激活函数输出的范围是 $[-\theta, \theta]$, 则每一层神经元的输出可以归一化表示为:

$$x_i = y_i \theta. \quad (14)$$

将式 (14) 带入式 (13), 则有:

$$y_i = \begin{cases} \text{clip}(\frac{W_{i-1}^{\text{ANN}} y_{i-1} \theta + b_{i-1}}{\theta}, 0, 1), & W_{i-1}^{\text{ANN}} x_{i-1} + b_{i-1} \geq 0, \\ \text{clip}(\alpha(\frac{W_{i-1}^{\text{ANN}} y_{i-1} \theta + b_{i-1}}{\theta}), -1, 0), & W_{i-1}^{\text{ANN}} x_{i-1} + b_{i-1} < 0 \end{cases} \quad (15)$$

此时, 结合式 (15) 与式 (12), ANN 与 SNN 前向传播过程建立起了等价关系。结合两式可以得出 ANN 到 SNN 权重转换关系表达式为:

$$\begin{cases} W_i^{\text{SNN}} = v_{th}^p W_i^{\text{ANN}} = \alpha v_{th}^n W_i^{\text{ANN}}, \\ b_i^{\text{SNN}} = \frac{v_{th}^p}{\theta} b_i^{\text{ANN}} = \alpha \frac{v_{th}^n}{\theta} b_i^{\text{ANN}} \end{cases} \quad (16)$$

根据式 (16) 可以得出, SNN 正负阈值设定的比值 $v_{th}^p : v_{th}^n$ 即为 LeakyReLU 负半轴斜率 α 。sthLeakyReLU 激活函数的形式如图 1 所示。

3 模型转换流程

ANN 到 SNN 模型转化的整体流程如图 2 所示, 首先, 需要对原模型进行适当的修改, 使得模型能够满足 SNN 模型的需求。本文对模型所做的修改主要有以下两点:

- 1) 使用第 2 节中所设计的 sthLeakyReLU 激活函数替换原模型的激活函数。
- 2) 使用空间线性下采样替换原始模型中的最大值池化操作^[16]。

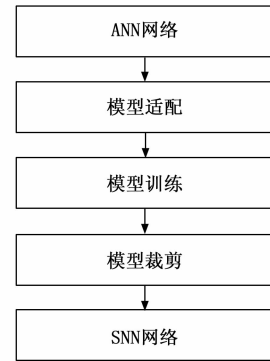


图 2 ANN 到 SNN 模型转换流程

转换后的 SNN 结构如图 3 所示, 虚线框中的部分表示由 ANN 训练得到的模型主体部分, 此外, 还需要对模型的输入层与输出层进行适当修改。由于 SNN 中, 输入的信号都是脉冲信号, 不能将 RGB 的图片信息直接进行输入, 因此, 需要对原输入图片进行脉冲编码, 以匹配 SNN 的输入要求。

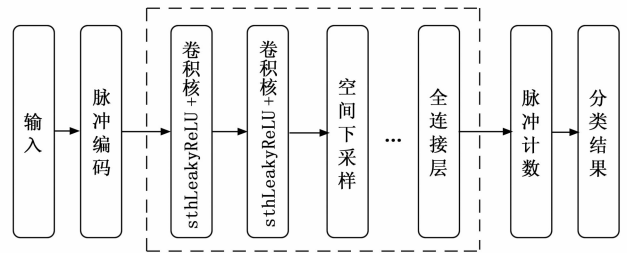


图 3 转换后的 SNN 模型结构

SNN 中对于输入图片常用的编码方式包括频率编码与时序编码两种, 频率编码是一种简单直观的脉冲编码方式, 它通过脉冲的频率来表示输入信号的强度。具体而言, 输入信号的强度越大, 对应时间步长内产生的脉冲数量就越多; 反之, 输入信号的强度越小, 对应时间步长内产生的

脉冲数量就越少。时序编码则是将输入信号的强度转化为脉冲发放时间的方式。具体而言，输入信号的强度越大，对应时间步长内产生的第一个脉冲发放时间越早；反之，输入信号的强度越小，对应时间步长内产生的第一个脉冲发放时间越晚。以一张大小为 28×28 的手写数字为例，假设所使用的时间窗口长度为 28，分别对其进行时序编码与频率编码，最终得到的编码结果如图 4 所示。对于图中较亮的部分，在经过时序编码后，所得到的脉冲发放时间更为提前，而对于频率编码，所得到的脉冲发放频率也会更高，反之亦然。

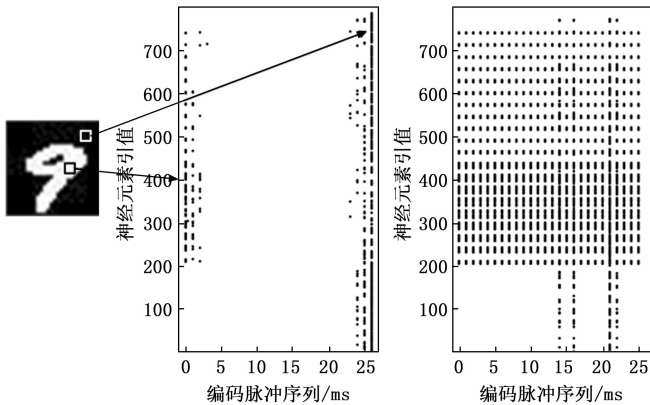


图 4 时序编码与频率编码示意图

频率编码与时序编码也分别存在着一定的局限性，对于频率编码，其编码的还原程度依赖于观察窗口的长度，观察窗口的长度越长，则其对于信号的还原程度也就越高，但较长的时间窗口可能会导致系统产生延迟。其次，高频率的脉冲会导致网络能耗的增加。而对于时序编码，由于每一个信号都只通过一个脉冲发放的相对时间来进行表示，因此该方法对于噪声比较敏感。其次，时序编码系统需要使用高精度的时钟同步来对信号进行还原，并且编码后的信号解码过程也较为复杂。为了提高 SNN 模型的鲁棒性，本文使用频率编码的方式对输入图片进行脉冲编码。对于输入的 RGB 图像进行频率编码的方式如下：

$$\begin{cases} R = \text{rand}(), R \in [0, 1] \\ s_{ijk} = \delta(c\text{Input}_{ijk} - R), k = 1, 2, 3 \end{cases} \quad (17)$$

式中， $\text{rand}()$ 是一个随机数生成函数，生成位于 $[0, 1]$ 区间服从均匀分布的随机数 R ；输入的图片乘以一个缩放因子 c 后与随机数 R 做差，传入一个脉冲函数 $\delta(\cdot)$ 中，从而生成一个符合频率编码要求的脉冲输入信号。

输出层使用脉冲计数模块，统计在设定的 SNN 仿真时间步长下，全连接层输出神经元脉冲发射频率，并根据脉冲发射频率的最大值产生预测结果，脉冲发射频率越高，表示该神经元对应的类别可能性越大。

4 输出结构优化

与 ANN 网络相比，SNN 通常具有更好的鲁棒性。其

原因在于，ANN 的神经元模型通常是 sigmoid 函数或者 ReLU 函数，这些函数都是连续函数。而 SNN 的神经元模型则是脉冲发放器，只有在达到一定电位值时才会发射脉冲。这种神经元的信息传递机制使得其对于输入信号没有那么敏感。此外，ANN 的推理只用到了当前时刻的输入信息，如果某一时刻输入的信号中存在较大的噪声，则会对网络的输出产生较大影响。SNN 的推理过程除空间信息外，还会使用到时序信息，根据神经元在一段时间内发射脉冲的数量，统计脉冲的发射频率，从而对结果进行预测。此时，如果在时序信息某一时刻的输入带有噪声，其对于输出结果的影响也会因其他时刻的正常输入信息而大幅减小，使得模型的鲁棒性得到提升。Wang 等^[20]的研究表明，通过在输出层引入泄露机制，可以进一步提升 SNN 模型的鲁棒性。因此，本文采用泄露机制，对转换后 SNN 的输出层结构进行了优化。

脉冲神经网络中的泄露机制是指神经元在产生脉冲后，其膜电位会随时间逐渐恢复到一个基准值。泄露机制可以通过自动调整神经元的膜电位，将噪声信号滤除或减小，从而提高模型对噪声的抑制能力。本章基于 LIF 模型，对泄露机制进行实现。对于 SNN 模型的输出，在脉冲计数模块之前，添加一层 LIF 神经元，用于接收 SNN 模型全连接层所输出的信息，对结果进行预测。LIF 神经元在 IF 神经元的基础上引入了一个泄露因子，使得膜电位的大小会随着时间的推移而逐渐衰减，具体而言，式 (2) 中膜电位更新的表达式变为：

$$m_i(t) = \gamma v_i(t-1) + W_{i-1} s_{i-1}(t) + b_{i-1} \quad (18)$$

该层 LIF 神经元采用全连接的方式与上一层 IF 神经元相连接，连接权重使用 STDP 学习规则进行无监督训练。STDP 算法是应用最为广泛的一种无监督学习算法，是对于 Hebb 理论的一个时间依赖性扩展。Hebb 理论认为，一起激活的神经元连接在一起。换句话说，如果前突触神经元和后突触神经元的激活是同时发生的，那么这种激活现象的产生会增强两个神经元之间的突触连接。STDP 算法则是在 Hebb 理论的基础上进一步考虑了神经元激活的时间顺序，根据神经元激活的相对时间来对突触的连接强度进行调整。当前突触脉冲先于后突触脉冲数毫秒抵达时，可触发长期增强 (LTP, long-term potentiation) 效应。相反，如果后突触脉冲先于前突触脉冲发生，将导致长期抑制 (LTD, long-term depression) 效应。

图 5 展示了前突触神经元和后突触神经元脉冲发放时间对于神经元连接强度变化的曲线。突触神经元用 j 来表示；后突触神经元用 i 来表示；神经元之间的连接强度定义为 W_{ij} 。假设前突触神经元产生的脉冲的时间为 t_f^j ，其中 $f = 1, 2, 3, \dots$ ；后突触神经元产生脉冲的时间为 t_f^i ，其中 $n = 1, 2, 3, \dots$ 。 f 与 n 分别表示前突触神经元和后突触神经元脉冲产生的序列。相连接的两个神经元产生脉冲的时间越接近，则其权重的变化就越明显，体现出了 Hebb 理论中“一起激

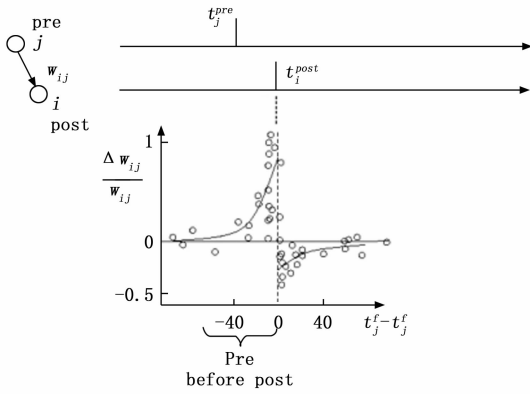


图 5 STDP 算法中的权重更新曲线^[21]

活的神经元连接在一起”这一准则。在此基础上, STDP 算法有引入了时间依赖性的概念, 前突触神经元产生脉冲的时间在后突触神经元产生脉冲的时间之前时, 此时认为这两个神经元产生的脉冲具有一种“因果关系”, 因此神经元之间的连接权重会增强, 即 LTP 效应。而如果将产生脉冲的顺序反过来, 后突触神经元在前突触神经元的脉冲到达之前就已经产生了脉冲, 便认为后突触神经元产生的脉冲与全突触神经元无关, 此时两个神经元之间的连接权重便会被减弱, 即 LTD 效应。根据以上描述可以得出, 在一段时间内, 神经元 j 与神经元 i 之间的权重变化量可以表示为:

$$\Delta W_{ij} = \sum_{j=1}^N \sum_{m=1}^N W(t_i^m - t_j^m) \quad (19)$$

式中, $W(x)$ 表示 STDP 函数。本文使用以下公式作为 STDP 函数:

$$\begin{cases} W(x) = A_+ \exp(-x/r_+), x > 0 \\ W(x) = A_- \exp(-x/r_-), x < 0 \end{cases} \quad (20)$$

如图 6 所示, 为最终所设计的转换后 SNN 的模型结构。垂直方向的箭头表示空间信息的传递方向, 而横向的箭头, 则表示时序信息的传递方向。

5 实验结果与分析

5.1 数据集与实验平台

MNIST 数据集是一个手写数字图像数据集, 由 60000 个训练样本和 10000 个测试样本组成。每个样本都是一个 28×28 像素的灰度图像, 表示 0 到 9 之间的一个数字。CIFAR-10 数据集是由 10 个不同类别的彩色图像组成, 每个类别有 6000 张图像, 总共有 50000 张训练图像和 10000 张测试图像。每个图像的尺寸为 32×32 像素, 并包含 3 个颜色通道 (红、绿、蓝)。CIFAR-10 数据集类别包括飞机、汽车、鸟类、猫、鹿、狗、青蛙、马、船和卡车。与 MNIST 数据集相比, CIFAR-10 数据集更加复杂, 因为它包含了彩色图像和多个类别。此外, 这两个数据集的格式有所不同。MNIST 数据集的图像以及对应的标签存储在单独的文件中, CIFAR-10 数据集以二进制格式存储, 需要使用特定的库或工具进行解析和处理。

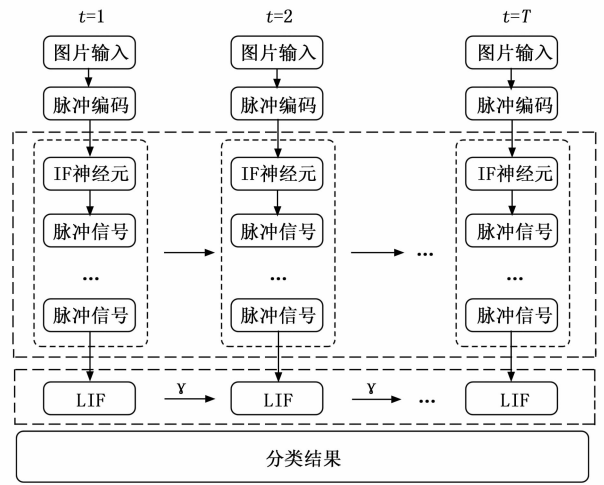


图 6 输出结构优化后的 SNN 模型结构

对于 MNIST 数据集, 本文使用了自定义的全连接网络以及卷积网络对结果进行测试, 网络结构如表 1 所示。表中, 784-1200-1200-10 表示一个包含两个隐藏层的全连接网络。该网络直接将输入图片展开成一维向量, 送入模型进行训练, 从而得到分类结果。12c5-p2-64c5-p2-10 则表示一个卷积网络。对于输入的 28×28 的图片, 12c5 表示一个卷积核大小为 5×5 , 通道数为 12 的卷积操作, p2 则表示一个 2×2 范围的平均值池化操作。最终, 同样通过一个全连接层输出模型的分类型结果。对于 CIFAR-10 数据集, 本文则分别使用了 VGG16 与 ResNet-20 两种模型进行了测试。本文将实验结果与其它现有的方法进行了对比, 验证了所提出方法的有效性。此外, 本文在不同程度噪声干扰下的 CIFAR-10 数据集上, 对优化后的输出结构进行了消融实验。

表 1 不同数据集所使用的网络结构定义

数据集	网络	结构
MNIST	Net1	784-1200-1200-10
	Net2	12c5-p2-64c5-p2-10
CIFAR-10	Net3	VGG16
	Net4	ResNet20

5.2 实验结果分析

如表 2 所示, 为 MNIST 数据集下, 不同转换方法所得到的 SNN 准确率对比。与使用标准 LeakyReLU 激活函数的 TerMapping、AugMapping 方法相比, 本文所提出的转换方法仅用了 8 个时间步便收敛到了相同水平的准确率, 收敛速度得到了大幅提高。在更加复杂的 CIFAR-10 数据集下, 本文使用了 VGG16 与 ResNet 两种不同的基础网络来进行实验。如表 3 所示, 随着网络层数与复杂度的提高, SNN 的收敛速度也随之降低, 然而与传统方法相比, 本文所提出的方法在收敛速度上, 依然保持巨大优势, 在 VGG16 网络结构下, 仅 16 个时间步 SNN 便收敛

到了与 ANN 相同的识别精度。与 thReLU 激活函数相比，由于负向脉冲的引入，使得模型中可以包含更加丰富的信息，在相同的时间步下，基于本文所设计的 sthLeakyReLU 激活函数训练得到的 SNN 模型识别的准确率得到了明显提升。

表 2 MNIST 数据集结果

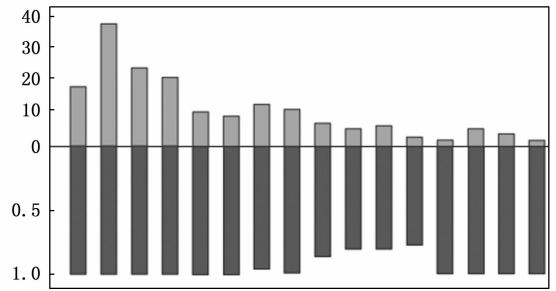
转化方法	ANN 网络	SNN 准确率/%	转换误差/%	延迟
DateNorm	Net2	99.10	0.04	200
ModelNorm	Net2	99.11	0.06	500
TerMapping	Net2	99.35	0.00	750
AugMapping	Net2	99.35	0.00	65
RNL	卷积网络	96.51	0.00	--
RNL	AlexNet	99.46	-0.04	--
本文	Net1	98.37	0.00	8
本文	Net2	99.32	0.03	8

表 3 CIFAR-10 数据集结果

转化方法	ANN 网络	SNN 准确率/%	转换误差/%	延迟
DateNorm	VGG16	91.76	0.40	--
TerMapping	VGG16	93.75	0.38	2800
AugMapping	VGG16	93.90	0.23	300
RNL	VGG16	92.86	-0.04	256
RNL	ResNet	93.45	-0.39	256
thReLU	VGG16	92.29	0.05	16
thReLU	ResNet	93.56	0.05	128
本文	VGG16	93.86	0.03	16
本文	ResNet	94.39	0.07	128

实验结果表明，本文所提出的转换方法在提高 SNN 推理速度方面具有显著效果。这主要得益于，在激活函数中，对于 ANN 神经元输出范围的限定，可以使得在 SNN 的转换过程中，模型每一层神经元的激活阈值更加接近。根据式 (19) 所表示的 ANN 到 SNN 的权重转换关系，当各层神经元输出的最大激活值差异较大时，会导致转换后 SNN 模型的激活阈值也存在较大差异，这会使得那些具有较小阈值的神经元容易陷入过度激活的状态，而那些具有较大阈值的神经元处于激活不足的状态。过度激活与激活不足神经元，都会造成网络的不稳定性和性能下降，因此在设计 SNN 时需要注意控制神经元的脉冲生成速率。如图 7 所示，为在 VGG16 网络中，每一层神经元输出的最大激活值分布情况。可以看出，由于阈值平衡技术的引入，sthLeakyReLU 激活函数能够有效减小各层神经元最大激活值的变化范围，从而使得 SNN 各层神经元的脉冲生成速率更加稳定，大幅提高了模型的收敛速度。

本章在 CIFAR-10 数据集上对使用 LIF 神经元优化的输出结构进行了消融实验。由于本文基于速率编码的方式对输入图片进行脉冲编码，因此这里采用在 CIFAR-10 数据集上添加椒盐噪声的方式来模拟外界干扰。通过使用不



(上) LeakyReLU (下) sthLeakyReLU

图 7 VGG16 网络中每一层神经元输出的最大值

同信噪比的椒盐噪声对输入图片进行处理，分别送入训练好的 ANN 与 SNN 网络，获取推理结果。如图 8 所示，分别为信噪比 0%、10%、20%、30% 的 CIFAR-10 数据集图像。

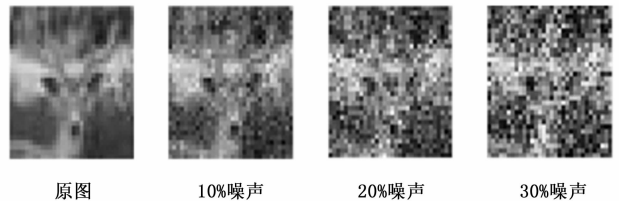


图 8 不同信噪比的 CIFAR10 数据集图像

对于 ANN 模型，直接将噪声图片输入模型即可，而对于 SNN 模型，由于模型的推理所需要的输入是一系列时序脉冲信号，因此在实验中，随机选取 5 个连续时间步中的 2 个输入，为其添加噪声信号进行输入。实验结果如图 9 所示，a)、b) 分别为不使用泄露机制与使用泄露机制时，不同信噪比的噪声环境下，模型准确率的变化趋势。可以看出，不论是否使用泄露机制，SNN 模型的鲁棒性均明显优于 ANN 模型。并且，随着噪声信号的增强，SNN 模型识别的准确率相比于原 ANN 模型的准确率提升幅度也越来越明显。这主要得益于 SNN 模型对于时序信息利用，即使在输入的一系列脉冲信号中，存在一些干扰，其推理结果也会被其他正常的输入信息所纠正，使其更倾向于输出正确的结果。同时，泄露机制会使得静息态神经元的膜电位随时间衰减。这意味着，当神经元在某些时刻产生了错误的分类结果，这个错误状态对于最终预测结果的影响会随着时间的推移而逐渐减小。因此，在引入泄露机制后，SNN 的鲁棒性得到了进一步提升。

6 结束语

将已训练好的 ANN 网络转换为 SNN 网络是获取 SNN 模型的一种常用方法。虽然目前主流方法已经可以做到几乎无损的转换，但转换后 SNN 的延迟普遍较高。本文在双阈值 SNN 模型的基础上，基于阈值平衡计数，提出了一种对称阈值 LeakyReLU 激活函数，并对其可行性进行了详细

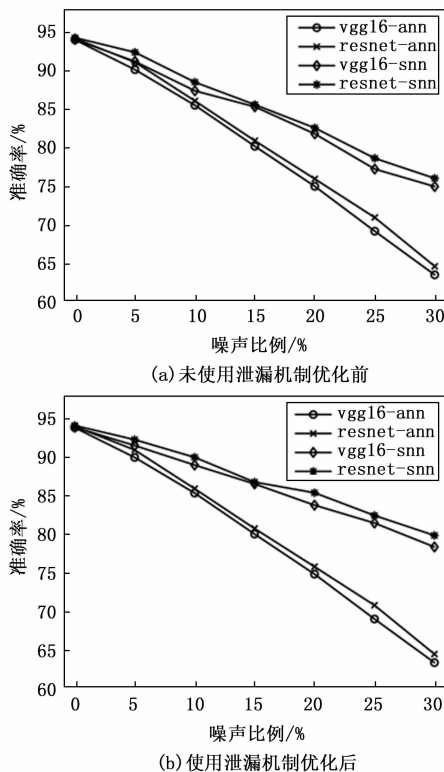


图 9 模型准确率随噪声影响的变化趋势

的论证。对照实验表明, 基于本文所提出的方法, SNN 的收敛速度得到了大幅提高。其次, 本文详细介绍了 ANN 到 SNN 模型转换的主要流程, 方便了后续其他相关研究工作的展开。最后, 本文 LIF 神经元模型, 使用泄露机制对 SNN 模型进行了优化, 进一步提高了 SNN 模型的鲁棒性。通过本文的研究, 对 SNN 网络的激活函数和模型结构做出了一定的改进, 为神经网络在不同场景下的应用提供了更为可靠和有效的解决方案。

参考文献:

- [1] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection [C] // Las Vegas, NV, USA; IEEE, 2016: 779 - 788.
- [2] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [J]. Advances in Neural Information Processing Systems, 2017, 30.
- [3] REDMON J, FARHADI A. YOLO9000: better, faster, stronger [C] // Honolulu, HI; IEEE, 2017: 6517 - 6525.
- [4] REDMON J, FARHADI A. Yolov3: an incremental improvement [J]. Arxiv Preprint Arxiv: 1804.02767, 2018.
- [5] SANDLER M, HOWARD A, ZHU M, et al. MobileNetV2: inverted residuals and linear bottlenecks [C] // Salt Lake City, UT; IEEE, 2018: 4510 - 4520.
- [6] KIM S, PARK S, NA B, et al. Spiking-YOLO: spiking neural network for energy-efficient object detection [J]. Proceedings of

the AAAI Conference on Artificial Intelligence, 2020, 34 (7): 11270 - 11277.

- [7] 董峻妃, 姜润皓, 燕锐, 等. 基于动态视觉运动特征的脉冲神经网络识别方法 [J]. 电子与信息学报, 2023, 45 (8): 2731 - 2738.
- [8] 刘伟, 李文娟, 高晋, 等. 基于对偶误差的脉冲神经网络目标检测方法 [J]. 电子与信息学报, 2023, 45 (12): 4469 - 4476.
- [9] 张季伦, 朱毅, 李颖, 等. 基于多层脉冲神经网络的非接触液位检测方法 [J]. 电子与信息学报, 2023, 45 (8): 2759 - 2769.
- [10] 周雅, 栗心怡, 武喜艳, 等. 基于 DT-LIF 神经元与 SSD 的脉冲神经网络目标检测方法 [J]. 电子与信息学报, 2023, 45 (8): 2722 - 2730.
- [11] KHERADPISHEH S R, GANJTABESH M, THORPS S J, et al. STDP-based spiking deep convolutional neural networks for object recognition [J]. Neural Networks, 2018, 99: 56 - 67.
- [12] XU Y, ZENG X, HAN L, et al. A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks [J]. Neural Networks, 2013, 43: 99 - 113.
- [13] BOHTE S M, KOK J N, LA POUTRÉ H. Error-backpropagation in temporally encoded networks of spiking neurons [J]. Neurocomputing, 2002, 48 (1-4): 17 - 37.
- [14] MOSTAFA H. Supervised learning based on temporal coding in spiking neural networks [J]. IEEE Transactions on Neural Networks and Learning Systems, 2017: 1 - 9.
- [15] LEE J H, DELBRUCK T, PFEIFFER M. Training deep spiking neural networks using backpropagation [J]. Frontiers in neuroscience, 2016, 10: 228000.
- [16] CAO Y, CHEN Y, KHOSLA D. Spiking deep convolutional neural networks for energy-efficient object recognition [J]. International Journal of Computer Vision, 2015, 113 (1): 54 - 66.
- [17] YU Q, MA C, SONG S, et al. Constructing accurate and efficient deep spiking neural networks with double-threshold and augmented schemes [J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33 (4): 1714 - 1726.
- [18] DING J, YU Z, TIAN Y, et al. Optimal ann-snn conversion for fast and accurate inference in deep spiking neural networks [J]. Arxiv Preprint Arxiv: 2105.11654, 2021.
- [19] DENG S, GU S. Optimal conversion of conventional artificial neural networks to spiking neural networks [J]. Arxiv Preprint Arxiv: 2103.00476, 2021.
- [20] WANG B, CAO J, CHEN J, et al. A new ANN-SNN conversion method with high accuracy, low latency and good robustness [C] // Macau, SAR China; International Joint Conferences on Artificial Intelligence Organization, 2023: 3067 - 3075.
- [21] SJÖSTRÖM J, GERSTNER W. Spike-timing dependent plasticity [J]. Scholarpedia, 2010, 5 (2): 1362.