

学校代码: 10284

分类号: TP181

密 级: 公开

U D C: 004.8

学 号: 522022330086



南京大學

硕士学位论文

论文题目 基于转换的脉冲神经

网络训练算法研究

作者姓名 杨洪朝

专业学位类别(领域) 计算机技术

研究方向 脉冲神经网络

导师姓名 申富饶教授

2025年5月15日

答辩委员会主席 武港山 教授

评 阅 人 武港山 教授

徐明华 教授

论文答辩日期 2025年5月16日

研究生签名:

导师签名:

Conversion-based Training Algorithm of Spiking Neural Network

by

Yang Hong-Chao

Supervised by

Professor Shen Fu-Rao

A dissertation submitted to
the graduate school of Nanjing University
in partial fulfilment of the requirements for the degree of

MASTER

in

Computer Technology



School of Computer Science

Nanjing University

May 15, 2025

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：基于转换的脉冲神经网络训练算法研究

计算机技术 专业 2022 级硕士生姓名：杨洪朝

指导教师（姓名、职称）：申富饶 教授

摘 要

脉冲神经网络作为一种生物启发的神经计算模型，因其生物学可解释性和低功耗特性在深度学习、类脑计算领域展现出广阔的应用前景。然而，由于脉冲信号的离散性，直接构建高性能脉冲神经网络模型依然面临显著挑战。为了充分利用现有人工神经网络的成熟训练框架与高效性能，基于人工神经网络到脉冲神经网络（ANN-SNN）的转换方法成为近年来的重要研究方向。本文提出了基于转换的脉冲卷积神经网络和脉冲 Transformer 的训练方法，旨在优化人工神经网络到脉冲神经网络的转换性能，提升脉冲神经网络模型的特征表达能力与实用性。

首先，针对现有 ANN-SNN 转换方法中的 QCFS 函数局限性，本文提出了通道自适应阈值和 Softplus 变换相结合的 CS-QCFS 函数。通过为每个通道独立训练阈值，有效适配了通道间的特性差异，同时引入 Softplus 变换来确保阈值始终为正，拉近了脉冲神经网络与人工神经网络在行为特性和性能上的一致性。实验结果表明，改进后的 CS-QCFS 函数在 CIFAR 数据集上的表现超越了传统方法，为 SNN 的实际应用提供了更高效的解决方案。

其次，为适配 Transformer 模型中的多头注意力机制和 MLP 的差异性问题，本文设计了多头 QCFS 函数和神经元 QCFS 函数，并结合两者提出了 HN-QCFS 激活函数。该方法通过为每个注意力头和全连接层神经元独立训练阈值，有效增强了脉冲 Transformer 的表达能力和适应性。实验结果表明，基于 HN-QCFS 的转换方法能够显著缩小 SNN 与 ANN 之间的性能差距，并在多个任务中取得优异成绩。

此外，本文构建了一个基于转换的脉冲神经网络训练平台。该平台集成了 CS-QCFS 和 HN-QCFS 方法，为非脉冲神经网络领域的专业人员提供了便捷的

训练工具，支持脉冲卷积神经网络和脉冲 Transformer 的高效训练与测试。

本文的研究成果不仅进一步缩小了 ANN 与 SNN 的性能差距，还为脉冲神经网络的研究和实际应用提供了新的理论基础和技术支持。未来，将进一步探索所提方法在更广泛任务和场景中的应用潜力。

关键词：脉冲神经网络；人工神经网络；ANN-SNN 转换；激活函数；卷积神经网络；Transformer

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Conversion-based Training Algorithm of Spiking Neural Network

SPECIALIZATION: Computer Technology

POSTGRADUATE: Yang Hong-Chao

MENTOR: Professor Shen Fu-Rao

ABSTRACT

Spiking Neural Networks(SNNs), as biologically inspired computational models, have demonstrated significant potential in deep learning and neuromorphic computing due to their biological interpretability and low-power characteristics. However, the discrete nature of spiking signals poses significant challenges in directly constructing high-performance SNN models. To leverage the mature training frameworks and high performance of existing Artificial Neural Networks (ANNs), ANN-to-SNN (ANN-SNN) conversion methods have emerged as a prominent research direction in recent years. This study proposes conversion-based training methods for Spiking Convolutional Neural Networks and Spiking Transformers to optimize the performance of ANN-SNN conversion and enhance the feature representation capabilities and practicality of SNN models.

Firstly, addressing the limitations of the QCFS function in existing ANN-SNN conversion methods, this study introduces the Channel-wise Softplus QCFS (CS-QCFS) function, which combines channel-wise thresholds with Softplus transformation. By independently training thresholds for each channel, the function effectively adapts to inter-channel differences. Meanwhile, the introduction of the Softplus transformation ensures that the thresholds are always positive, thereby narrowing the gap between the behavioral characteristics and performance of SNNs and ANNs. Experimental results show that the improved CS-QCFS function outperforms traditional methods on the CIFAR dataset, providing a more efficient solution for practical SNN applications.

Secondly, to adapt the differences in the multi-head attention mechanisms and the MLP in Transformer models, this study designs the Head-wise QCFS (H-QCFS) func-

tion and Neuron-wise QCFS (N-QCFS) function and integrates them into the HN-QCFS activation function. By independently training thresholds for each attention head and neuron in fully connected layer, the proposed method effectively enhances the expressive power and adaptability of Spiking Transformers. Experimental results demonstrate that the HN-QCFS-based conversion method significantly reduces the performance gap between SNNs and ANNs, achieving excellent results across multiple tasks.

Additionally, this study develops a conversion-based training platform for SNNs. This platform integrates the CS-QCFS and HN-QCFS methods, providing a convenient training tool for professionals outside the spiking neural network field, supporting the efficient training and testing of Spiking CNNs and Spiking Transformers.

The findings of this study not only narrow the performance gap between ANNs and SNNs but also offer new theoretical foundations and technical support for the research and practical application of spiking neural networks. Future work will explore the potential of the proposed methods in broader tasks and scenarios.

KEYWORDS: Spiking Neural Network; Artificial Neural Network; ANN-SNN; Activation Function; Convolutional Neural Network; Transformer

目 录

中文摘要	I
ABSTRACT	III
目 录	V
插图目录	IX
表格目录	XI
第一章 绪论	1
1.1 研究背景与意义	1
1.2 研究现状	3
1.2.1 基于转换的训练算法研究现状	3
1.2.2 脉冲卷积神经网络研究现状	4
1.2.3 脉冲 Transformer 研究现状	5
1.3 研究内容与贡献	6
1.4 本文组织结构	7
第二章 相关工作	9
2.1 脉冲神经元模型	9
2.1.1 HH 神经元模型	9
2.1.2 IF/LIF 神经元模型	11
2.2 神经网络架构	13
2.2.1 卷积神经网络	13
2.2.2 Transformer	14
2.3 人工神经网络转脉冲神经网络	16

2.3.1	转换原理	16
2.3.2	转换误差分析	17
2.3.3	消除误差	18
2.4	深度脉冲神经网络	19
2.4.1	深度脉冲卷积神经网络	19
2.4.2	脉冲 Transformer	21
2.5	本章小结	22
第三章 基于转换的脉冲卷积神经网络训练方法		25
3.1	问题分析	25
3.1.1	神经元阈值的限制	25
3.1.2	层级别阈值的限制	27
3.2	基于可训练阈值的脉冲卷积神经网络训练方法	28
3.2.1	正阈值特性	28
3.2.2	通道差异性	30
3.2.3	CS-QCFS	30
3.3	实验与分析	33
3.3.1	数据集与模型	33
3.3.2	实验设置	33
3.3.3	CIFAR 数据集对比实验	34
3.3.4	消融实验	37
3.3.5	理论支持实验	40
3.4	本章小结	42
第四章 基于转换的脉冲 Transformer 训练方法		45
4.1	问题分析	45
4.1.1	Transformer 与 CNN 的差异	45
4.1.2	QCFS 应用在多头注意力层的限制	46
4.1.3	QCFS 应用在全连接层的限制	47
4.2	基于转换的脉冲 Transformer 训练方法	48
4.2.1	多头级别 QCFS	48

4.2.2	神经元级别 QCFS	50
4.2.3	HN-QCFS	53
4.3	实验与分析	53
4.3.1	数据集与模型	54
4.3.2	实验设置	54
4.3.3	CIFAR 数据集对比实验	55
4.3.4	消融实验	56
4.3.5	理论支持实验	58
4.4	本章小结	60
第五章	基于转换的脉冲神经网络训练系统	61
5.1	系统研发背景	61
5.2	需求分析	62
5.3	系统设计	63
5.4	系统实现	64
5.4.1	系统开发环境	64
5.4.2	界面展示和使用流程	65
5.4.3	ANN 预训练模块	66
5.4.4	QANN 训练模块	66
5.4.5	测试模块	68
5.5	本章小结	69
第六章	总结与展望	71
	参考文献	73
	致 谢	81
	简历和科研成果	83

插图目录

1-1	ANN 神经元 (上) 和 SNN 神经元 (下) ^[13]	1
1-2	全文结构	8
2-1	HH 模型	10
2-2	卷积操作示意图 ^[46]	13
2-3	注意力机制 (左); 多头注意力机制 (右) ^[21]	14
2-4	Vision Transformer 模型概述 ^[12]	15
2-5	ANN-SNN 转换误差分析	17
2-6	ANN-SNN 转换流程	21
2-7	SpikeZIP-TF 模型架构 ^[51]	21
3-1	QCFS、ReLU、IF 反应模式	25
3-2	VGG-16 每层不同通道的平均激活值	27
3-3	人工神经网络精度比较	35
3-4	不同量化步长的影响	40
3-5	训练过程中的负阈值	42
4-1	ViT 架构 (左); CNN 架构 (右)	46
4-2	量化多头自注意力层 (左); 量化缩放-点积注意力 (右)	49
4-3	全连接运算与卷积运算	51
4-4	QCFS 量化 FFN (左); N-QCFS 量化 FFN (右)	52
4-5	应用 HN-QCFS 的编码器	53
5-1	平台整体架构	63
5-2	登录页面	65
5-3	首页	65

5-4	发起 ANN 训练	66
5-5	ANN 预训练记录	67
5-6	ANN 训练日志	67
5-7	发起 QANN 训练	68
5-8	QANN 训练记录	68
5-9	QANN 训练日志	69
5-10	测试 SNN	69
5-11	测试记录	70

表格目录

3-1	实验参数设置	34
3-2	CIFAR-10 数据集实验结果	36
3-3	CIFAR-100 数据集实验结果	37
3-4	通道级别阈值和 Softplus 变换的消融实验结果	37
3-5	两阶段训练消融实验结果	38
3-6	通道级别阈值的理论验证实验	41
4-1	实验参数设置	54
4-2	CIFAR-10 数据集实验结果	55
4-3	CIFAR-100 数据集实验结果	55
4-4	多头级别阈值和神经元级别阈值消融实验结果	56
4-5	多头级别阈值和神经元级别阈值验证实验结果	59

第一章 绪论

1.1 研究背景与意义

近年来，深度学习技术的飞速发展推动了人工智能（Artificial Intelligence, AI）^[1-2]在诸多领域的广泛应用，包括计算机视觉（Computer Vision, CV）^[3-5]、自然语言处理（Natural Language Processing, NLP）^[6-8]和语音识别（Speech Recognition, SR）^[9-10]等。随着大规模数据集和高性能硬件的支持，深度人工神经网络（Artificial Neural Networks, ANNs）不断向更深、更宽的方向发展，其性能持续提升，甚至在某些任务中超越了人类。然而，这种性能提升是以高昂的计算成本为代价的。大规模深度学习模型不仅需要巨大的存储空间和计算能力，其训练和推理过程也消耗了大量的能源，尤其是当前大规模深度学习模型（如 GPT-4^[11]、Vision Transformer^[12]等）的训练和推理过程，依赖于高性能计算硬件，其功耗与资源消耗呈指数级增长。因此寻找低功耗的高效计算模型成为当前人工智能领域的重要研究方向之一。

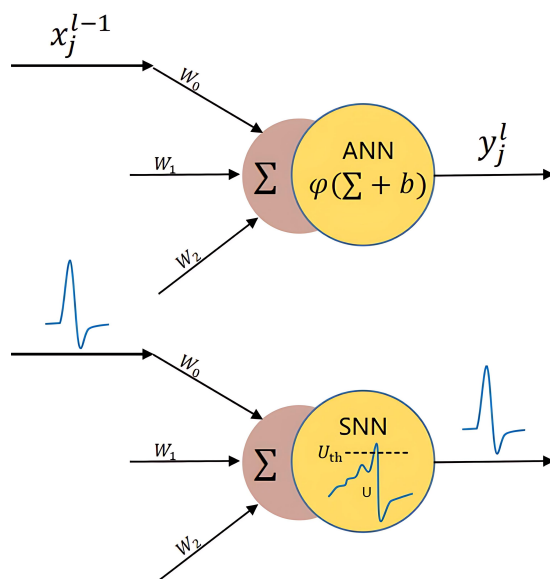


图 1-1 ANN 神经元（上）和 SNN 神经元（下）^[13]

与传统计算模型不同，类脑计算（Brain-inspired Computing）^[14-15]从人脑的工

作机制中汲取灵感，为低功耗智能计算提供了一条新的路径。人类大脑以极低的能耗（约 20 瓦）进行复杂的感知、决策和学习任务，这一特点启发了研究者探索仿生计算模型的可能性。脉冲神经网络（Spiking Neural Networks, SNNs）^[16]作为第三代人工神经网络，通过模拟生物神经元的发放脉冲机制，以事件驱动（Event-driven）的方式完成信息传递和计算。如图 1-1所示，与传统人工神经网络中的连续值信号不同，脉冲神经网络通过离散的脉冲（Spike）信号进行信息传递，神经元仅在膜电位超过阈值时发射脉冲。这种事件驱动的计算方式不仅大幅降低了冗余计算，还在理论上具备较高的能效比和生物学可解释性。这使得脉冲神经网络成为解决高能耗深度学习问题的重要研究方向之一，同时也被认为是下一代人工智能的重要基础。

尽管脉冲神经网络在理论上具有高效性，其实际应用仍面临诸多挑战。由于脉冲神经网络的离散脉冲信号特性，传统深度神经网络中基于梯度下降的反向传播算法难以直接应用，带来了显著的训练困难。为了解决脉冲神经网络的训练难题，研究者提出了多种方法，包括基于脉冲梯度逼近的优化方法^[17]、基于转换的间接训练方法^[18]以及基于生物启发的训练方法^[19]。脉冲梯度逼近通过替代梯度技术使反向传播算法适用于脉冲神经网络，但存在梯度信息不准确的问题；生物启发方法模拟生物学习机制，尽管具有较好的生物学可解释性，但优化效率和精度相对较低。所以本文重点关注了基于转换（Conversion-based）的间接训练方法，这种方法的核心思想是利用人工神经网络的高效训练框架，将训练完成后的人工神经网络模型参数映射到脉冲神经网络中。具体而言，先通过常规的梯度优化方法进行人工神经网络模型训练，随后通过特定的转换规则将人工神经网络的参数和结构映射为等效的脉冲神经网络模型中。

基于转换的训练方法通过利用传统人工神经网络的高效训练机制，绕过了脉冲神经网络因不可微分性带来的优化难题，不仅降低了训练的计算复杂度，还能够继承人工神经网络成熟模型的性能优势，为脉冲神经网络提供了可靠的性能保障。此外，这种方法在理论上构建了人工神经网络与脉冲神经网络之间的联系，在实践上表现出对不同网络架构的广泛适用性，尤其在复杂网络结构（如卷积神经网络^[20]和 Transformer^[21]）中展现出巨大潜力。卷积神经网络以其强大的局部特征提取能力成为计算机视觉任务的核心模型，广泛应用于图像分类、目标检测和语义分割等任务中。视觉 Transformer 则因其强大的全局建模能力在计

计算机视觉领域迅速崛起，尤其在处理复杂全局依赖关系和高维特征时展现了卓越性能。然而，这两类网络模型的高计算成本限制了其在能耗敏感场景中的应用。通过将网络脉冲化并结合转换训练方法，可以在保持其优异性能的同时，大幅降低推理过程中的计算能耗，为高效低能耗智能系统的实现提供了一条切实可行的路径。这项研究不仅能够推动脉冲神经网络的发展，还为下一代人工智能技术在可持续发展中的应用奠定了重要基础。

1.2 研究现状

1.2.1 基于转换的训练算法研究现状

由于脉冲神经网络的复杂时间动态特性和脉冲生成的离散性，直接训练脉冲神经网络往往面临显著挑战，这包括开发能够有效处理脉冲生成过程中不可微特性的专用学习算法。为了解决这个问题，近年来提出了人工神经网络到脉冲神经网络的转换方法（ANN-SNN），将预训练的人工神经网络模型参数映射到脉冲神经网络模型中。这种方法利用了人工神经网络已经成熟且高效的训练方法，显著简化了脉冲神经网络的训练过程，具有显著优势^[22]。

Cao 等人^[18]率先提出了一种方法，先使用 ReLU 激活函数训练人工神经网络，然后将激活层替换为脉冲神经元，以实现人工神经网络到脉冲神经网络的转换。为了缓解人工神经网络到脉冲神经网络转换过程中信息丢失的问题，Rueckauer 等人^[23]提出了一种软复位机制，通过减法方式重置膜电位。此外，为了缩小脉冲发放率与 ReLU 激活值之间的差距，一系列参数归一化方法应运而生。例如，Diehl 等人^[24]提出了一种基于数据的归一化方法，通过计算人工神经网络中每一层的最大激活值来确定初始归一化比例，从而提高深度脉冲神经网络的性能。Rueckauer 等人进一步引入了 p 分位数法作为归一化比例，用于过滤极端异常值，通常将 p 设为 99 或 99.9。Kim 等人^[25]则在 Spiking-YOLO 中提出了通道级的基于数据的归一化方法。

与参数归一化的思想类似，许多研究集中于动态调整脉冲阈值。Han 等人^[26]通过比例缩放脉冲神经网络的脉冲阈值，提出了一种阈值平衡技术，从而改善推理延迟。Ding 等人^[27]提出了 Rate Norm Layer (RNL) 替代人工神经网络中的 ReLU 激活函数，RNL 在训练过程中动态确定阈值。Ho 等人^[28]引入了一

种可训练的裁剪层技术，用于在训练过程中寻找最优的数据归一化因子。Deng 等人^[29]则对转换误差进行了理论分析，并通过在人工神经网络中使用阈值 ReLU 代替传统 ReLU 以及引入额外偏置，有效降低了逐层误差。Li 等人^[30]进一步利用最小化均方误差 (MMSE) 方法来获得自适应阈值，并通过量化微调校准脉冲神经网络参数。类似于^[29]和^[30]，Bu 等人^[31]深入研究了转换误差，提出了量化裁剪-向下舍入-偏移 (quantization clip-floor-shift, QCFS) 激活函数，使人工神经网络的激活更贴合脉冲发放率的有限性和离散性。他们还证明，将初始膜电位设置为阈值的一半是最优的，因为这一设置可以使期望转换误差达到零。由于 Bu 等人描述的不均匀性误差的存在，人工神经网络和脉冲神经网络之间仍然存在一定的性能差距。针对这一问题，Hao 等人^[32]通过分析人工神经网络与脉冲神经网络输出的差异，将不均匀性误差分为四种情况，并提出了一种基于优化残余膜电位的策略来缓解该误差。在后续研究中^[33]，他们进一步引入了偏移脉冲的概念，以量化脉冲神经网络实际脉冲发放率与期望发放率之间的差异，并提出了一种通过调整初始膜电位来优化转换误差的方法。

尽管这些研究在低延迟的条件下实现了不错的性能，但没有考虑通道异质性或多头异质性的问题，从而未能充分发挥脉冲神经网络的潜力。此外，这些方法也忽略了 ANN-SNN 转换的基本要求之一，即确保脉冲阈值始终为正。在本文中，我们将针对这些问题展开研究，结合现有方法的理论，提出更高精度和超低延迟的脉冲神经网络解决方案。

1.2.2 脉冲卷积神经网络研究现状

Cao 等人^[18]提出了一种通过调整 CNN 架构来适配脉冲神经网络的方法，以应对 CNN 到脉冲神经网络转换过程中可能导致的精度下降问题。该研究指出转换的主要挑战包括：CNN 中负值输出在脉冲神经网络中的表示困难、偏置项在脉冲神经网络中的难以映射以及最大池化需要通过复杂的两层脉冲网络实现。为此，论文提出了三项关键改进，包括通过加入绝对值函数层和将激活函数替换为 ReLU 确保所有层的输出为正，移除偏置项以简化脉冲神经网络实现，以及用空间线性下采样替代最大池化以降低复杂性和适配脉冲域。由于脉冲神经网络中的神经元激活是二值化的，直接使用最大池化会导致下一层显著的信息丢失，因此 Diehl 等人^[24]采用了空间平均池化 (spatial-averaging) 作为替代方案，以更好

地保留信息。Rueckauer 等人^[23]对 CNN 结构提出了一系列改进,从而优化从人工神经网络到脉冲神经网络的转换精度。他们针对批归一化 (Batch-Normalization) 提出了参数融合方法,将训练过程中生成的参数直接整合到前一层的权重中,从而实现无损转换,并通过将模拟输入视为常量电流,避免了传统的泊松脉冲编码带来的噪声。针对分类过程,论文提出了两种改进的脉冲化 Softmax 方法,解决了传统方法在没有脉冲输出时的分类问题。此外,研究还提出了一种新的脉冲最大池化机制,通过门控函数仅允许最大激活神经元的脉冲通过,从而有效保留了特征图的空间信息。Sengupta 等人^[34]提出了一种新的算法技术,用于构建深层的脉冲神经网络,并在 CIFAR-10 和 ImageNet 等复杂视觉识别任务上验证了其效果。该研究特别针对 VGG 和 ResNet 架构,提出了一种名为 Spike-Norm 的 ANN-SNN 转换技术,显著提高了转换后的脉冲神经网络的分类准确性。

1.2.3 脉冲 Transformer 研究现状

脉冲 Transformer 的研究旨在通过利用脉冲神经网络的事件驱动计算模式,减少冗余计算和能量消耗,同时保留 Transformer 的强大特征提取能力。目前,针对脉冲 Transformer 的研究大多集中在模型架构设计、训练方法改进等方面。

Zhou 等人^[35]提出了 Spikformer 模型,结合了脉冲神经网络和自注意力机制。Spikformer 通过使用脉冲形式的 Query (Q)、Key (K) 和 Value (V) 来模拟稀疏的视觉特征,避免了传统 Transformer 中的 Softmax 操作,从而显著降低了计算能耗。Zhou 等人^[36]提出了一种硬件友好的脉冲驱动残差学习架构 Spikingformer,这种架构基于残差设计,开发出了一个纯脉冲驱动的基于 Transformer 的脉冲神经网络,以避免非脉冲计算。Yao 等人^[37]提出了新的模型架构 Spike-driven Transformer,该模型重新设计了一种脉冲驱动自注意力 (Spike-Driven Self-Attention, SDSA) 算子,将 Q (Query)、K (Key)、V (Value) 矩阵之间的运算转换为了掩码 (mask) 和加法,从而得到线性复杂度的注意力计算,大幅降低了自注意力机制的计算成本。Wang 等人^[38]提出了掩码脉冲 Transformer (Masked Spiking Transformer, MST)。MST 通过引入一种专门针对脉冲神经网络设计的随机脉冲掩码 (Random Spike Masking, RSM) 方法来剪枝输入脉冲,显著减少了计算过程中涉及的脉冲数量。这种方法不仅减少了 Transformer 中自注意力模块和 MLP 模块的能耗,而且保持了模型性能。Shi 等人^[39]提出了一种新的脉冲自注

注意力机制，称为双脉冲自注意力 (Dual Spike Self-Attention, DSSA)。DSSA 通过合理的缩放方法，能够处理任意尺寸的特征图。在 DSSA 的基础上，他们提出了 SpikingResformer，结合了基于 ResNet 的多阶段结构，在减少参数的同时提高了性能和能量效率。Zhou 等人^[40]在 Spikformer 的基础上进行了优化，提出了 Spikformer V2。Spikformer V2 引入了自监督预训练，使得 Spikformer V2 在 ImageNet 上的性能取得了重大突破，超过了 80% 的高准确率瓶颈。此外，Yao 等人^[41]将 Spike-driven Transformer 扩展为了 Meta-SpikeFormer。Meta-SpikeFormer 是第一个可以同时执行分类、检测和分割任务的直接训练脉冲神经网络主干网络，为脉冲神经网络作为通用视觉主干网络铺平了道路，并可以启发未来基于 Transformer 的神经形态芯片设计。

1.3 研究内容与贡献

本文主要研究基于转换的脉冲神经网络训练方法，深入分析了现有基于转换的脉冲神经网络算法存在的问题：在卷积神经网络中，层级别统一阈值策略难以适配通道间激活分布的显著差异，且现有量化方法可能产生负阈值，导致脉冲神经元与人工神经元行为不一致。在 Transformer 架构中，多头注意力机制的特征空间异质性与 MLP 模块的神经元响应多样性，使得现有量化方法难以实现精准的脉冲行为映射。针对上述问题，本文分别提出了基于转换的脉冲卷积神经网络和脉冲 Transformer 训练方法，进一步缩小了人工神经网络与脉冲神经网络之间的性能差距。本文的主要研究内容概述如下：

1. 本文提出了一种基于转换的脉冲卷积神经网络训练方法，主要针对 QCFS 函数应用在卷积神经网络的两大局限性提出了创新性解决方案。首先，本文分析了 QCFS 函数的两大局限性，即通道差异性和负阈值问题，QCFS 函数统一的层级别阈值无法有效捕捉通道间的差异，并且负阈值的存在与 ANN-SNN 转换的核心思想相违背。针对上述问题，本文提出了通道自适应阈值的改进方法，通过为每个通道独立训练阈值，精准适配通道间的特性差异。同时，引入了 Softplus 变换来确保阈值始终为正，拉近了脉冲神经网络与人工神经网络在行为特性和性能上的一致性。实验验证表明，改进后的算法在 CIFAR 数据集上的表现超越了现有方法，并为脉冲神经网络

的实际应用提供了新的解决方案。

2. 本文提出了一种基于转换的脉冲 Transformer 训练方法。通过对 Transformer 中关键组件的深入分析，我们发现多头注意力机制和 MLP 中存在多头差异性和神经元差异性。针对上述问题，本文提出了多头 QCFS 函数和神经元 QCFS 函数，并结合这两种函数设计了 HN-QCFS 激活函数。该方法通过为每个 Attention 模块的注意力头和 MLP 的全连接层神经元独立训练阈值，确保了模型能够更好地捕捉到不同子空间特征，同时提高了特征变换的细致程度，从而增强了脉冲 Transformer 的自适应性和表达能力。实验结果表明，HN-QCFS 能够有效缩小脉冲神经网络与人工神经网络之间的性能差距。
3. 本文搭建了一个基于转换的脉冲神经网络训练平台。该系统结合了本文提出的基于转换的脉冲卷积神经网络和脉冲 Transformer 的训练方法，提供了一个便捷的训练平台，尤其适用于非脉冲神经网络领域的专业人员。通过详细的需求分析与系统架构设计，本系统包含 ANN 预训练、QANN 训练和测试模块，通过不同模块的协作，成功支持了脉冲 CNN 和脉冲 Transformer 的训练与测试，为用户提供了高效的 ANN-SNN 转换与训练流程。

1.4 本文组织结构

本文主要研究基于转换的脉冲神经网络训练方法，分别提出了基于转换的脉冲卷积神经网络和脉冲 Transformer 训练方法，并基于这两种算法搭建了一个基于转换的脉冲神经网络训练平台。

全文一共分为六章，总体结构如图所示1-2所示。第一章为绪论，介绍了脉冲神经网络的研究背景与意义，接着概述了 ANN-SNN 转换、脉冲卷积神经网络以及脉冲 Transformer 的研究现状；第二章为相关工作，主要介绍了基于转换的脉冲神经网络训练方法所需的基础知识及关键技术，并概述了在此领域内一些具有里程碑意义的研究成果；第三章介绍了一种基于转换的脉冲卷积神经网络训练方法，分析了 QCFS 函数的局限性，并提出了 CS-QCFS 函数进行相应的改进；第四章介绍了一种基于转换的脉冲 Transformer 训练方法，分别针对 Attention 模块全连接模块设计了新的激活函数，并结合这两种函数设计了 HN-QCFS 激活函

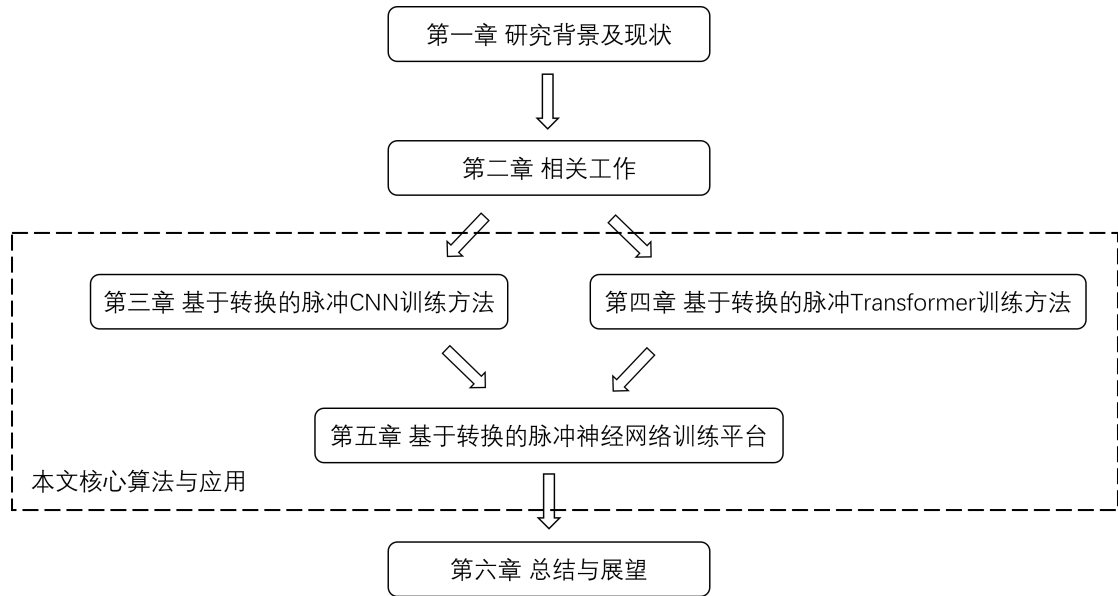


图 1-2 全文结构

数；第五章介绍了基于转换的脉冲神经网络训练平台，集成了本文提出的基于转换的脉冲卷积神经网络和脉冲 Transformer 训练方法；第六章为总结与展望，概述了本文的研究内容与贡献，并对未来工作进行了展望。

第二章 相关工作

本章节主要介绍实施基于转换的脉冲神经网络训练方法所需的基础知识及关键技术，并概述了在此领域内一些具有里程碑意义的研究成果。

2.1 脉冲神经元模型

在脉冲神经网络的研究中，脉冲神经元模型是模拟神经元动态行为的核心部分，直接影响网络的计算方式和信息处理能力。与传统人工神经网络中使用连续激活函数的神经元不同，脉冲神经网络中的神经元以脉冲 (Spike) 形式传递信息，模拟了生物神经元在处理信息时的激活机制。脉冲神经元模型的设计是脉冲神经网络研究的基础，不同的神经元模型在实现计算功能和模拟生物学特性方面存在显著差异。本文将重点介绍两类经典的脉冲神经元模型——HH (Hodgkin-Huxley) 神经元模型^[42]和 IF (Integrate-and-Fire)^[43]/LIF (Leaky Integrate-and-Fire) 神经元模型^[44]。

2.1.1 HH 神经元模型

HH 模型是由生物学家 Alan Hodgkin 和 Andrew Huxley 在 1952 年基于生物学神经元的电流和电压动态方程而提出的经典模型，该模型描述了神经元膜电位的变化过程，并模拟了生物神经元的行为^[42]。HH 模型通过一组非线性微分方程来模拟神经元膜的电活动，成功解释了神经冲动的产生机制，被认为是生物神经元建模的经典模型。

如图2-1所示，HH 模型将神经元膜视为一个电学系统，由电容和离子通道构成。在该模型中，膜电位 (V_m) 的变化受控于膜电容、电流、以及三种主要离子通道（钠离子通道、钾离子通道和泄露通道）的电流共同作用。根据电学分析，

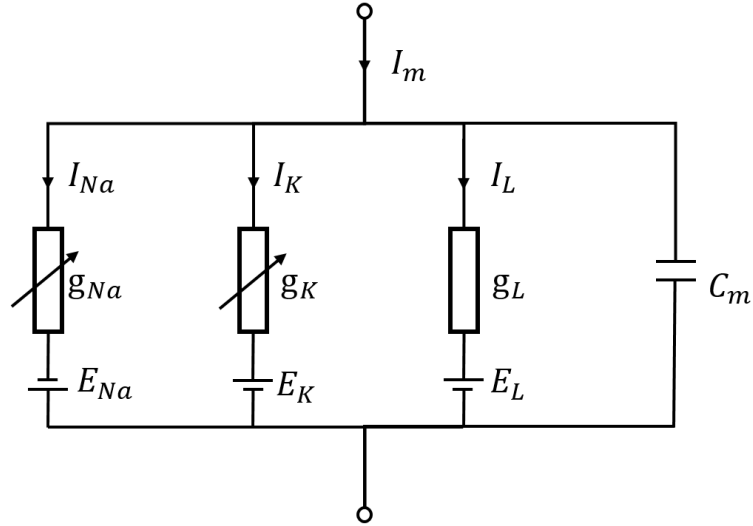


图 2-1 HH 模型

膜电位的变化可描述为以下方程：

$$\begin{aligned}
 C_m \frac{dV_m}{dt} &= I_{\text{ext}} - I_{\text{Na}} - I_{\text{K}} - I_{\text{L}} \\
 &= I_{\text{ext}} - g_{\text{Na}} \cdot m^3 \cdot h \cdot (V_m - E_{\text{Na}}) - g_{\text{K}} \cdot n^4 \cdot (V_m - E_{\text{K}}) - g_{\text{L}} \cdot (V_m - E_{\text{L}}),
 \end{aligned}
 \tag{2-1}$$

其中 C_m 是膜电容，用于表示膜的电容特性； I_{ext} 是外部电流输入，通常是刺激电流； I_{Na} 、 I_{K} 、 I_{L} 分别表示钠离子、钾离子和泄露通道的电流； g_{Na} 、 g_{K} 、 g_{L} 分别是钠离子、钾离子、泄露通道电流的最大导通率； E_{Na} 、 E_{K} 、 E_{L} 分别是钠离子、钾离子和泄露通道的反向电位； m 、 h 、 n 是电压依赖的“激活”和“失活”变量，这些变量决定了 Na、K 离子通道的开闭状态。变量 m 、 h 、 n 的动态变化由以下微分方程描述：

$$\frac{dx}{dt} = \alpha_x \cdot (1 - x) - \beta_x \cdot x, \quad x \in \{m, h, n\}
 \tag{2-2}$$

其中 α_x 、 β_x 是与电压 V_m 相关的速率常数，通过实验测定。HH 模型通过这些方程系统地描述了生物神经元的膜电位变化。

HH 模型因其生物学准确性，常用于对神经元生物学机制进行深入研究。但 HH 模型计算复杂度较高，且在大规模神经网络中的计算效率较低，因此 HH 模型一般用于小规模神经元的模拟或神经科学领域的精细研究，而在实际应用中

较少被直接用于大规模脉冲神经网络的训练与推理。

2.1.2 IF/LIF 神经元模型

为了克服 HH 模型计算复杂度较高的问题, 研究人员提出了“积分-点火”(IF) 神经元模型。IF 模型是一种简化的脉冲神经元模型, 主要通过一个简单的电压积分方程来描述神经元的动态行为^[43]。IF 模型认为神经元的膜电位会随着时间的推移逐渐增加, 当外部刺激输入时, 神经元的膜电位会根据输入的电流进行积分。若膜电位超过设定的阈值, 神经元会触发一个脉冲并将膜电位重置为初始值。IF 模型主要有两种常用变体: 简单积分-点火模型 (IF 模型) 和泄露-积分-点火模型 (LIF 模型)。其中, LIF 模型在 IF 模型的基础上加入了膜电位的衰减项, 进一步提升了模型的生物合理性。

在简单的 IF 模型中, 膜电位 $V(t)$ 受输入电流 $I(t)$ 的积累影响, 可以描述为以下形式:

$$\frac{dV(t)}{dt} = \frac{I(t)}{C_m}, \quad (2-3)$$

其中 C_m 是神经元的膜电容; $I(t)$ 是外部输入电流。当 $V(t)$ 达到阈值 V_{th} 时, 神经元发放一个脉冲, 并且膜电位被重置到复位电位 V_{reset} 。这一过程可描述为:

$$\text{if } V(t) \geq V_{th}, \quad V(t) \rightarrow V_{reset}. \quad (2-4)$$

IF 模型的主要优点是计算简单且容易实现, 非常适合用于大规模的脉冲神经网络模拟。IF 模型广泛应用于神经科学研究、脑机接口以及机器人控制等领域。

LIF 模型在简单 IF 模型的基础上加入了一个泄漏项, 以模拟膜电位在无输入时的自然衰减, 即当没有外部输入时, 膜电位会逐渐衰退至静息电位。这一特性使得 LIF 模型能够更好地模拟生物神经元在没有刺激时的静息状态。LIF 模型的方程为:

$$\tau_m \frac{dV(t)}{dt} = -(V(t) - E_L) + R \cdot I(t), \quad (2-5)$$

其中, $V(t)$ 为神经元膜电位, τ_m 为膜电位时间常数, E_L 为静息电位, R 为膜电阻, $I(t)$ 为输入电流。当膜电位 $V(t)$ 超过阈值 V_{th} 时, 神经元会放电并恢复到初

始状态。LIF 模型的电压会逐渐衰减，模拟了生物神经元在无输入时的“休眠”状态。这种泄漏机制使得 LIF 模型在计算上更加接近生物神经元的行为，并且计算复杂度较低，适合用于大规模脉冲神经网络的训练。LIF 神经元模型因其较为简单且高效，在脉冲神经网络中得到了广泛应用。

为了在计算机中模拟 LIF 模型，需要将时间连续的微分方程离散化。在离散化过程中，时间被划分为离散的时间步 $t_n = n \cdot \Delta t$ ，其中 Δt 是时间步长。通过欧拉方法（Euler's method）将时间导数离散化，得到如下的差分方程：

$$\frac{dV(t)}{dt} \approx \frac{V(t_{n+1}) - V(t_n)}{\Delta t}. \quad (2-6)$$

将其代入 LIF 模型的方程中，得到离散化的 LIF 方程：

$$V(t_{n+1}) = V(t_n) + \frac{\Delta t}{\tau_m} [-(V(t_n) - E_L) + R \cdot I(t_n)]. \quad (2-7)$$

在深度学习中，我们通常需要将 LIF 神经元的模型进行适当修改，以便其适应神经网络的训练和推理过程。具体而言，LIF 模型被离散化为如下形式：

$$V_{mem}^l(t) = \lambda V_{mem}^l(t-1) + z^l(t) - V_{th}^l S^l(t), \quad (2-8)$$

$$z^l(t) = W^l S^{l-1}(t) V_{th}^{l-1}, \quad (2-9)$$

$$S^l(t) = \Theta(V_{mem}^l(t) - V_{th}^l). \quad (2-10)$$

其中， $V_{mem}^l(t)$ 、 $z^l(t)$ 和 $S^l(t)$ 分别表示第 l 层神经元在时刻 t 时的膜电位、输入和脉冲。 λ 是泄漏系数，用于控制膜电位的衰减。每层神经元都有一个特定的阈值电压 V_{th}^l ，它决定了神经元的放电行为。输入 $z^l(t)$ 是来自上一层（第 $l-1$ 层）神经元在时刻 t 发放的脉冲 $S^{l-1}(t)$ 的加权和， W^l 为第 l 层的权重矩阵。当膜电位 V_{mem}^l 超过阈值电压 V_{th}^l 时，神经元会出发脉冲 $S^l(t)$ ，这一过程通过 Heaviside 阶跃函数 $\Theta(\cdot)$ 来描述^[45]。

2.2 神经网络架构

近年来，神经网络作为深度学习的核心技术，已经在计算机视觉、自然语言处理和强化学习等领域取得了显著成就。其中，卷积神经网络（Convolutional Neural Networks, CNNs）和 Transformer 架构作为代表性模型，被广泛应用于不同任务中。

2.2.1 卷积神经网络

卷积神经网络（CNN）是一种专为处理结构化数据（如图像）的深度学习模型。通过局部感知和参数共享的方式，CNN 能够高效提取数据的空间特征，同时极大减少模型参数，提高计算效率^[20]。CNN 的基本模块包括卷积层、池化层和全连接层，这些模块相互配合，逐层提取输入数据的空间和语义特征。其中卷积层负责特征提取，池化层则通过降维操作减少计算量和防止过拟合，全连接层则用于对高层特征进行分类或回归等任务。

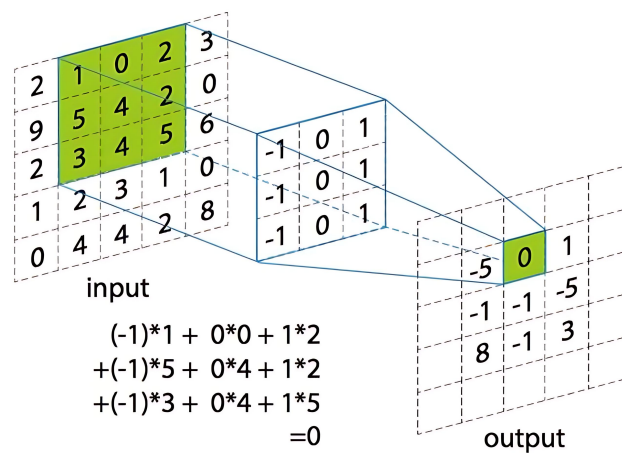


图 2-2 卷积操作示意图^[46]

卷积层是卷积神经网络的核心，通过卷积核（或滤波器）对输入张量进行特征提取。如图 2-2 所示，卷积操作的过程中，卷积核在输入特征图上滑动，通过局部感知方式提取局部区域内的特征，同时保持图像的空间结构。给定输入张量 $\mathbf{X} \in \mathbb{R}^{H \times W \times C_{in}}$ 和卷积核 $\mathbf{K} \in \mathbb{R}^{k \times k \times C_{in} \times C_{out}}$ ，输出特征图 $\mathbf{Y} \in \mathbb{R}^{H' \times W' \times C_{out}}$ 由

以下公式定义：

$$\mathbf{Y}_{h,w,c_{out}} = \sum_{i=1}^k \sum_{j=1}^k \sum_{c_{in}=1}^{C_{in}} \mathbf{X}_{h+i,w+j,c_{in}} \cdot \mathbf{K}_{i,j,c_{in},c_{out}}, \quad (2-11)$$

其中, H, W 分别是输入特征图的高度和宽度, C_{in} 和 C_{out} 分别是输入和输出的通道数, k 是卷积核的大小。

在卷积神经网络中, 通道是特征的不同维度, 用于描述数据中蕴含的信息。例如, 对于彩色图像的输入, 通道通常为 RGB 三个维度, 每个通道分别表示红、绿、蓝三种颜色成分。卷积神经网络中不同通道专注于捕捉不同类型的特征, 即使在同一层中, 不同通道之间也存在差异性, 它们对特征的表达方式和重要性有所不同^[47]。某些通道可能专注于捕捉某种特定的局部特征, 而其他通道则可能关注更全局或抽象的特征, 这种差异性使得网络能够从多个维度进行特征学习, 从而形成更加丰富的表示。

2.2.2 Transformer

Transformer 最初由 Vaswani 等人在 2017 年提出^[21], 已成为自然语言处理和视觉任务中表现突出的通用架构。其核心思想是通过自注意力机制 (Self-Attention Mechanism) 建模数据中不同位置之间的长距离依赖关系。

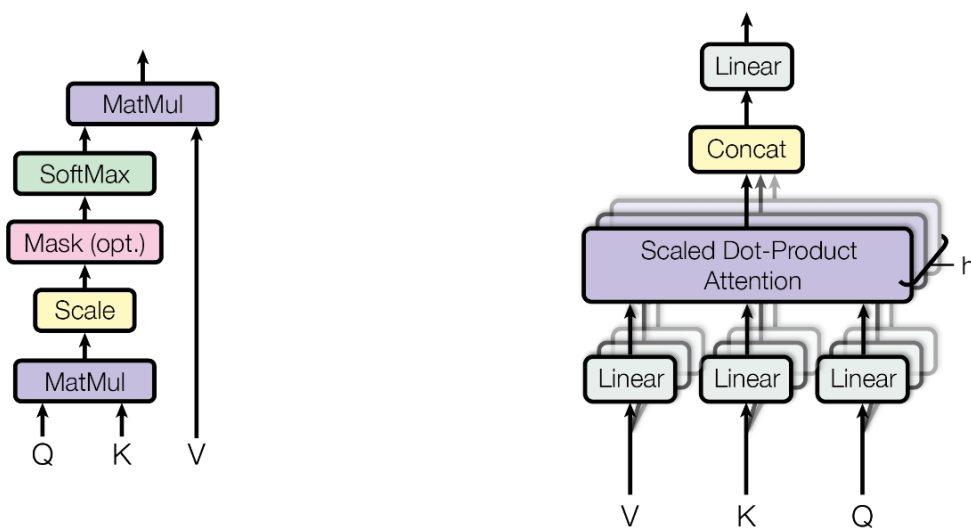


图 2-3 注意力机制 (左); 多头注意力机制 (右)^[21]

自注意力机制是 Transformer 的基础模块, 通过计算查询 (query)、键 (key)

和值 (value) 三者的加权关系, 实现对输入序列的全局信息建模。其计算公式为:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \quad (2-12)$$

其中, \mathbf{Q} 、 \mathbf{K} 和 \mathbf{V} 是通过线性变换获得的特征, d_k 是特征维度。注意力机制通过计算查询与键的相似度分数生成注意力权重, 从而对输入序列中的重要信息进行突出表示。

为了提升模型对复杂关系的表达能力, Transformer 引入了多头注意力机制 (Multi-Head Attention)。多头注意力机制通过将输入特征投影到多个低维子空间中并行计算注意力, 每个头独立捕捉序列中不同层次或模式的信息。在多头注意力中, 假设有 h 个头, 每个头独立计算其注意力:

$$\text{head}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i), \quad i = 1, 2, \dots, h, \quad (2-13)$$

最后将所有头的结果拼接并线性变换:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)\mathbf{W}^O, \quad (2-14)$$

其中 \mathbf{W}^O 是输出投影矩阵。通过这种方式, 多头注意力能够同时从多个子空间捕捉不同层次的信息, 并有效增强模型对复杂关系的表达能力。多头注意力机制的引入, 不仅使得模型可以在不同的关系模式下进行并行学习, 还通过信息的融合, 提供了更加全面和细致的特征表示。

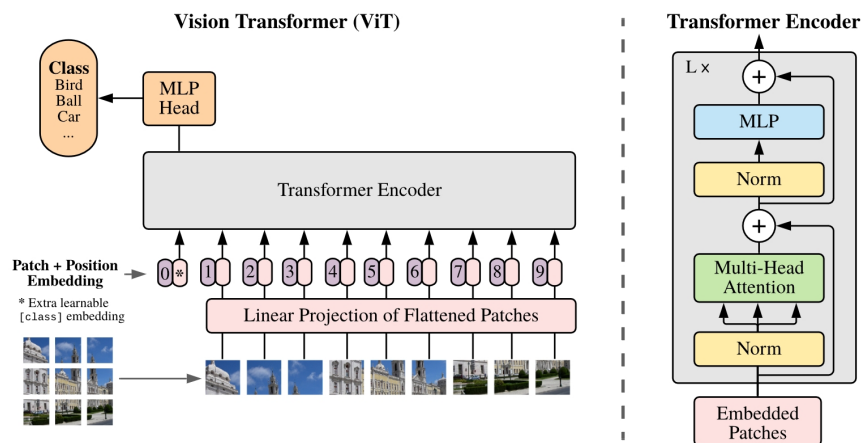


图 2-4 Vision Transformer 模型概述^[12]

近年来, Transformer 的应用逐渐从自然语言处理扩展到计算机视觉领域, 取得了显著成效。Vision Transformer (ViT)^[12] 是这一领域的典型代表。如图 2-4 所示, ViT 首次将 Transformer 引入视觉任务, 通过将输入图像划分为固定大小的图块 (patches), 并将这些图块线性嵌入为序列特征, 送入 Transformer 进行全局建模。与 CNN 相比, ViT 的优势在于其擅长捕捉全局上下文信息, 而不仅局限于局部特征。ViT 的全局建模能力使其在大型数据集上的表现尤为突出, 例如 ImageNet 上的分类任务。然而, 与此同时, ViT 也存在计算复杂度较高的问题, 尤其是在高分辨率图像任务中。此外, ViT 对数据规模和预训练模型的依赖较大, 在数据不足的情况下, 其表现可能不如 CNN。

2.3 人工神经网络转脉冲神经网络

2.3.1 转换原理

ANN-SNN 的基本概念是将预训练的人工神经网络模型参数迁移到对应的脉冲神经网络模型中^[32]。这个过程通过脉冲神经网络中的 IF 神经元的平均脉冲发射率 (或平均突触后电位) 来近似人工神经网络中 ReLU 函数的连续激活值^[48]。

积分-点火 (IF) 神经元^[49] 是广泛使用的脉冲神经元, 具体原理已通过 (2-8)-(2-10) 进行了介绍。通过将方程 (2-8) 从 $t = 1$ 求和到 $t = T$, 并将方程 (2-9) 中的变量 $z^l(t)$ 代入方程 (2-8), 我们得到:

$$V_{mem}^l(T) - V_{mem}^l(0) = W^l V_{th}^{l-1} \sum_{i=1}^T S^{l-1}(i) - V_{th}^l \sum_{i=1}^T S^l(i). \quad (2-15)$$

然后将两边同时除以 T , 并指定 $r^l(T) = \sum_{i=1}^T S^l(i)/T$, 我们得到:

$$V_{th}^l r^l(T) = W^l V_{th}^{l-1} r^{l-1}(T) - \frac{V_{mem}^l(T) - V_{mem}^l(0)}{T}. \quad (2-16)$$

平均脉冲发射率和平均突触后电位之间的关系是 $\phi^l(T) = r^l(T) V_{th}^l$, 那么我们得到:

$$\phi^l(T) = W^l \phi^{l-1}(T) - \frac{V_{mem}^l(T) - V_{mem}^l(0)}{T}, \quad (2-17)$$

其中 $\phi^l(T)$ 代表平均突触后电位，在我们的推导上下文中始终是非负的。值得注意的是，在 T 接近无穷大的情况下，Eq. (2-17) 与人工神经网络中的前向传播非常相似。具体来说，在 T 接近无穷大时，Eq. (2-17) 可以表述为：

$$\phi^l(T) = W^l \phi^{l-1}(T), \quad (2-18)$$

而在人工神经网络中，前向传播可以描述为：

$$a^l = \text{ReLU}(W^l a^{l-1}), \quad (2-19)$$

其中 a^l 表示层 l 的激活。

2.3.2 转换误差分析

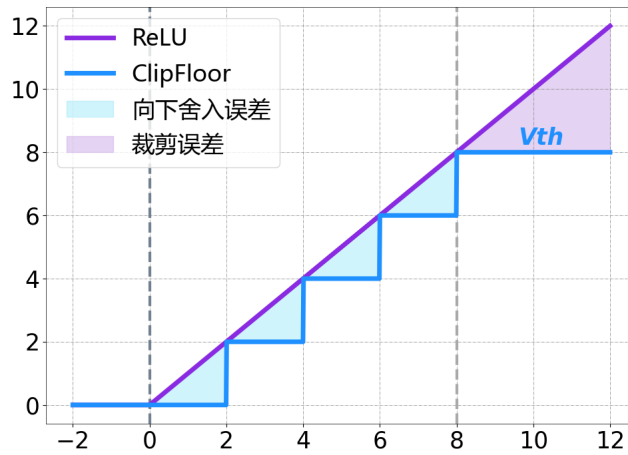


图 2-5 ANN-SNN 转换误差分析

为了了解 IF 神经元和基于 ReLU 的神经元之间的差异，我们从理论上分析了基于 ReLU 的神经元和 IF 神经元的输出之间的转换误差。在 IF 神经元中，平均突触后电位为：

$$\phi^l(T) = r^l(T) V_{th}^l = V_{th}^l \left(\sum_{i=1}^T S^l(i) \right) / T. \quad (2-20)$$

由于 IF 神经元的平均脉冲发射率在 $[0, 1]$ 的范围内，因此相应的平均突触后电位也在 0 到 V_{th} 之间。在每个时间步长，脉冲 $S(i)$ 可以是 0 或 1，因此平均突触

后电位 $\phi(T)$ 的量化分辨率为 V_{th}^l/T 。基于上述几点，我们可以将方程 (2-20) 重新表述为：

$$\begin{aligned}\phi^l(T) &= \frac{V_{th}^l}{T} \left(\sum_{i=1}^T S^l(i) \right) = \frac{V_{th}^l}{T} \lfloor \frac{T}{V_{th}^l} W^l \phi^{l-1}(T) \rfloor \\ &= \text{clip} \left(\frac{V_{th}^l}{T} \lfloor \frac{T}{V_{th}^l} W^l \phi^{l-1}(T) \rfloor, 0, V_{th}^l \right),\end{aligned}\quad (2-21)$$

其中，clip 函数将上限确定为 V_{th}^l 并将下限设置为 0。floor 函数表示为 $\lfloor x \rfloor$ ，产生不超过 x 的最大整数。在上一层的脉冲神经网络和人工神经网络输入相同的情况下，即 $a^{l-1} = \phi^{l-1}(T)$ ， l 层的人工神经网络和脉冲神经网络之间的转换误差为：

$$\begin{aligned}\text{error}^l &= \phi^l(T) - a^l \\ &= \text{clip} \left(\frac{V_{th}^l}{T} \lfloor \frac{T}{V_{th}^l} W^l a^{l-1} \rfloor, 0, V_{th}^l \right) - \text{ReLU}(W^l a^{l-1}).\end{aligned}\quad (2-22)$$

根据 Eq. (2-22)，转换误差来自两个方面，即向下舍入误差和裁剪误差^[31]。如图 2-5 所示，向下舍入误差源于 floor 函数引入的近似值，这导致人工神经网络中的连续值与脉冲神经网络中的量化值不匹配。另一方面，裁剪误差与 clip 函数设置的边界相关联，它将脉冲神经元的输出范围限制在 0 到 V_{th} 之间。相比之下，人工神经元的输出范围从 0 扩展到无穷大。因此，这种差异会导致不匹配，因为人工神经元中超过 V_{th} 的值在脉冲神经元中缺乏等效的表示。

2.3.3 消除误差

通过修改源人工神经网络的激活函数，可以消除向下舍入误差和裁剪误差。具体来说，QCFS 激活函数^[31]被提出来取代人工神经网络中的 ReLU 激活函数，其定义为：

$$a^l = \text{QCFS}(z^l) = \lambda^l \text{clip} \left(\frac{1}{L} \lfloor \frac{z^l L}{\lambda^l} + \varphi \rfloor, 0, 1 \right),\quad (2-23)$$

其中 z^l 第 l 层的输入， L 表示量化步长， λ^l 是特定于第 l 层的可训练阈值， φ 控制激活函数的偏移。当人工神经网络使用 QCFS 作为激活函数时，Eq. (2-22) 的

转换误差可以重写为:

$$\begin{aligned} error^l &= \phi^l(T) - a^l \\ &= \text{clip}\left(\frac{V_{th}^l}{T} \lfloor \frac{T}{V_{th}^l} z^l \rfloor, 0, V_{th}^l\right) - \lambda^l \text{clip}\left(\frac{1}{L} \lfloor \frac{z^l L}{\lambda^l} + \varphi \rfloor, 0, 1\right). \end{aligned} \quad (2-24)$$

如果 $T = L, V_{th}^l = \lambda^l$, 则有:

$$error^l = \phi^l(T) - a^l = 0. \quad (2-25)$$

QCFS 激活函数是神经网络的一个与脉冲神经网络兼容的激活函数, 能够消除 ANN-SNN 转换中存在的裁剪误差和向下舍入误差, 它还可以在训练过程中调整 λ^l , 映射到脉冲神经元中的阈值 V_{th}^l 。

2.4 深度脉冲神经网络

2.4.1 深度脉冲卷积神经网络

深度卷积神经网络由多个功能层堆叠构成, 每个功能层通常包括卷积层、归一化层、激活层和池化层。在将深度卷积神经网络转换为深度脉冲卷积神经网络时, 需要针对每个层的特点进行相应的改造, 以适应脉冲神经元的事件驱动机制和时序处理特性。

卷积层 (Convolution)。卷积层的转换相对简单, 通常不需要对卷积操作本身做任何改变。具体来说, 卷积层的权重可以直接从神经网络复制到脉冲神经网络中。这是因为卷积操作本质上是一种线性运算, 而在脉冲神经网络中, 卷积层的输出是基于时序脉冲的累积和, 脉冲的频率和时间编码了特征信息。只要输入信号以适当的脉冲编码形式提供, 卷积核的权重在脉冲神经网络中依然有效。

批归一化层 (Batch Normalization, BN)。批归一化在神经网络中用于减小内部协变量偏移 (internal covariate shift), 从而加速训练过程^[50]。BN 在每

一层引入仿射变换，对输入进行零均值和单位方差的归一化，计算公式为：

$$\text{BN}[x] = \gamma \frac{x - \mu}{\sigma} + \beta \quad (2-26)$$

其中, x 是输入特征； μ 和 σ 分别为均值和标准差； γ 和 β 是可训练参数。在训练过程中， μ 和 σ 通常通过指数滑动平均估计，而 γ 和 β 通过梯度下降学习。在脉冲神经网络中，由于时间维度的引入，BN 层在空间和时间两个维度上计算统计量变得困难。因此，Rueckauer 等人^[23]提出了通过参数折叠将 BN 层的功能合并到前一层的权重和偏置中的方法，从而在推理阶段消除显式的 BN 计算。具体地，利用 BN 层的运行统计量 (μ, σ) 和变换参数 (γ, β) ，调整前一层的权重和偏置：

$$\tilde{W}_{ij} = \frac{\gamma}{\sigma} W_{ij}, \quad \tilde{b} = \frac{\gamma}{\sigma}(b - \mu) + \beta \quad (2-27)$$

其中， W 和 b 分别为前一层的权重和偏置； \tilde{W} 和 \tilde{b} 是经过 BN 参数折叠后的新权重和偏置。

池化层 (Pooling)。深度卷积神经网络架构通常由中间池化层组成，以减少卷积输出图的大小。两种流行的池化机制选择是最大池化 (池化窗口上的最大神经元输出) 和空间平均池化 (在池化窗口上进行二维平均池化操作)。由于脉冲神经网络中神经元的激活值是二进制的，而不是模拟值，执行最大池化将导致下一层的显著信息丢失。因此，Diehl 等人^[24]提出使用空间平均池化代替空间最大池化。空间平均池化函数可以很容易地转换到脉冲域。

激活层 (Activation)。在脉冲神经网络中，传统卷积神经网络的负值输出难以通过脉冲神经元精确表示。因此，在将卷积神经网络转换为脉冲神经网络时，需要确保激活层的输出为非负。其中，最关键的改变是将 CNN 中的 tanh 函数等可能产生负值的激活函数替换为 ReLU 函数 ($\text{ReLU}(x) = \max(x, 0)$)。ReLU 不仅能保证非负性，还具有线性特性 (当输出为正值时)，这在脉冲神经网络中可以减少转换时的精度损失。此外，ReLU 作为非饱和非线性函数，加快了训练过程，是 CNN 到脉冲神经网络转换的理想选择。

2.4.2 脉冲 Transformer

You 等人^[51]提出了一种 ANN-SNN 转换方法 SpikeZIP-TF，该方法通过引入脉冲等效自注意力（spike-equivalent self-attention, SESA）、脉冲 Softmax 和脉冲 LayerNorm，确保了人工神经网络和转换后的脉冲神经网络之间的精确等效，从而在不降低准确性的情况下进行转换。

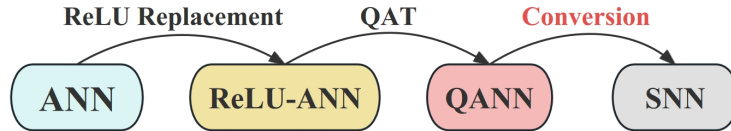


图 2-6 ANN-SNN 转换流程

如图 2-6 所示，SpikeZIP-TF 遵循了先前的 ANN-SNN 方法^{[31][30]}所建立的转换流程。首先将人工神经网络（ANN）中的激活函数替换为 ReLU 激活函数。随后，应用量化感知训练（QAT）以获得低位宽且高精度的量化人工神经网络（QANN）。最后，通过将量化激活函数（神经元）替换为特定的脉冲神经元，将 QANN 转换为脉冲神经网络（SNN），并确保精度不受影响。

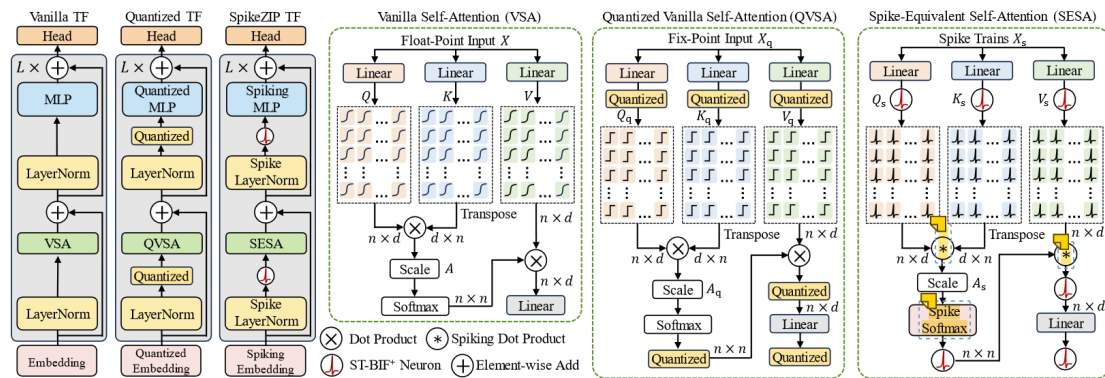


图 2-7 SpikeZIP-TF 模型架构^[51]

如图 2-7 所示，SpikeZIP-TF 生成的脉冲 Transformer 的架构与 Transformer 几乎完全相同，它由一个嵌入层、多个 Transformer 编码器和一个浅层分类头组成。给定一个目标的基于 Transformer 的人工神经网络，为了得到对应的脉冲 Transformer，需要进行如下步骤：

1. 在矩阵乘法操作符前后插入量化器，以获得量化的 Transformer。

2. 使用量化感知训练 (QAT) 训练具有量化功能的模型, 然后将量化函数替换为脉冲神经元, 确保矩阵乘法的输入和输出以脉冲序列的形式存在。
3. 将量化 Transformer 中的脉冲神经网络不友好操作符 (例如 Softmax、Layer-Norm 和点积) 替换为脉冲神经网络友好操作符 (例如 Spike-Softmax、Spike-LayerNorm 和脉冲点积)。

脉冲等效自注意力模块的计算过程可以表述为:

$$\begin{aligned}
 Q_{spk}[t] &= \text{IF}(X[t] * W_Q), \\
 K_{spk}[t] &= \text{IF}(X[t] * W_K), \\
 V_{spk}[t] &= \text{IF}(X[t] * W_V),
 \end{aligned} \tag{2-28}$$

其中 Q_{spk} , K_{spk} , V_{spk} 分别表示查询、键、值在时间步 t 的脉冲矩阵, $\text{IF}(\cdot)$ 是 IF 神经元函数, W_q , W_k , W_v 表示相应的权重矩阵。注意力分数定义为:

$$\text{Attn}_{spk}[t] = \text{IF}\left(\text{softmax}\left(\frac{Q_{spk}[t] * K_{spk}^T[t]}{\sqrt{d}}\right)\right), \tag{2-29}$$

其中 $\text{Attn}_{spk}[t]$ 表示由查询脉冲矩阵和键脉冲矩阵的点积计算得到的注意力分数的脉冲矩阵, d 是缩放因子, 等于给定注意力头的特征维度。最终的注意力输出为:

$$\text{SESA}(Q_{spk}, K_{spk}, V_{spk}) = \text{IF}(\text{Attn}_{spk}[t] * V_{spk}[t]). \tag{2-30}$$

2.5 本章小结

本章首先介绍了常用的脉冲神经元模型, 包括 HH 神经元模型和 IF/LIF 神经元模型, 并讨论了它们的生物学背景和在神经网络中的应用。然后回顾了神经网络架构中的两大经典模型: 卷积神经网络 (CNN) 和 Transformer, 对这两种架构的分析为我们提出脉冲神经网络结构的改进和转换方法提供了关键指导。在人工神经网络到脉冲神经网络的转换部分, 我们详细介绍了 ANN-SNN 的基本原理, 分析了转换过程中可能出现的误差, 并介绍了目前主流的误差消除方案。最后, 我们探讨了深度脉冲神经网络的最新进展, 重点介绍了深度脉冲卷积

神经网络和脉冲 Transformer 的主流研究进展。

第三章 基于转换的脉冲卷积神经网络训练方法

尽管现有的 ANN-SNN 研究在低延迟的条件下实现了不错的性能，但没有考虑 ANN-SNN 转换的基本要求之一，即确保脉冲阈值始终为正。此外，这些方法也忽略了通道异质性，从而未能充分发挥脉冲神经网络的潜力。在本节中，我们将针对这些问题展开研究，提出更高精度和超低延迟的脉冲神经网络解决方案。

3.1 问题分析

3.1.1 神经元阈值的限制

在脉冲神经网络中，神经元的阈值电位具有重要的意义，它决定了神经元发出脉冲的触发点。我们分析了脉冲神经网络中 IF 神经元的行为，并探讨了在人工神经网络中，使用 ReLU 和 QCFS 激活函数时神经元的表现。图 3-1 系统地说明了这些神经元的在不同条件下的反应模式。

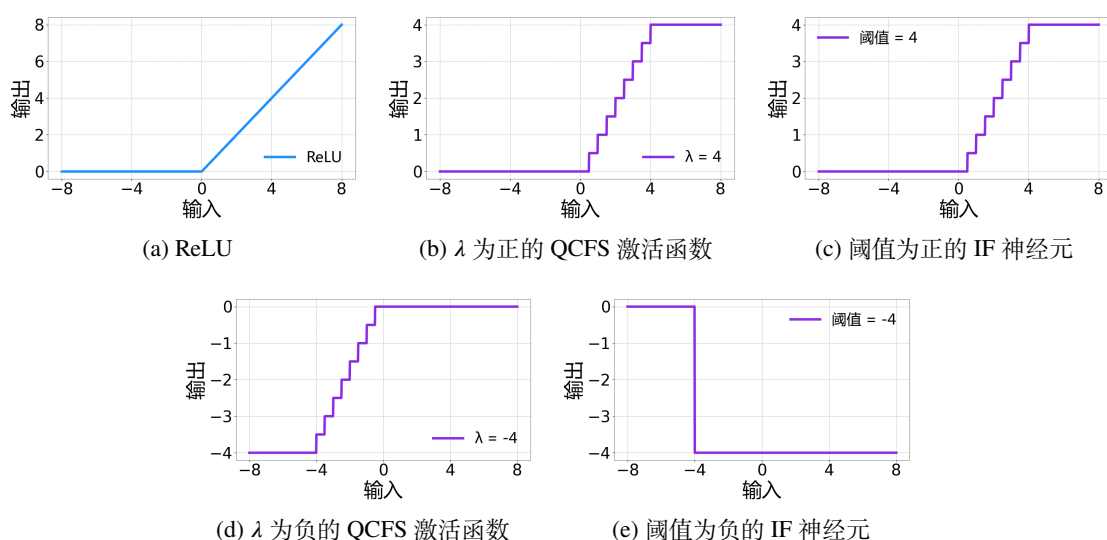


图 3-1 QCFS、ReLU、IF 反应模式

在将人工神经网络转换为脉冲神经网络时，必须设置正阈值以确保脉冲神

神经网络在功能上与人工神经网络保持一致，特别是当人工神经网络使用 ReLU 或 QCFS 作为其激活函数时。如图 3-1(a)-3-1(c) 所示，这种对齐不仅确保脉冲神经网络神经元反映人工神经网络神经元的活动模式，而且符合 ANN-SNN 转换的基本原理。

相反，负阈值会导致脉冲神经网络与人工神经网络在行为上的差异。当人工神经网络使用 ReLU 激活函数时，如图 3-1(a) 所示，只有当神经元的输入大于零时，神经元才会被激活。也就是说，对于正输入，神经元对网络输出有贡献，而对于负输入，它保持非活动状态。在这种情况下，如果脉冲神经网络中的对应神经元具有负阈值，它可能会错误地响应某些负输入并触发脉冲。这种偏差在图 3-1(e) 中有清晰的说明。这一差异突显了脉冲神经元可以在人工神经元未被激活的情况下激活。同样，对于使用 QCFS 激活函数的人工神经网络，情况也类似。如图 3-1(d) 所示，如果可训练阈值 λ 为负，则神经元的输出对于负输入可能为负，对于正输入为零。在映射此人工神经网络行为的脉冲神经网络中，负阈值会导致神经元可能对某些负输入触发脉冲，而对正输入则几乎肯定会触发。这种行为在图 3-1(e) 中再次展示，显示了脉冲神经网络模型行为的偏差。

为了更深入地理解负阈值的影响，本文通过数学形式化描述因使用负阈值所导致的误差。其中，误差指代人工神经元和脉冲神经元输出之间的差异。首先定义激活函数输出的误差为：

$$error^l = QCFS(z^l) - \phi^l(T). \quad (3-1)$$

当阈值 λ^l 为正时，误差为零：

$$error^l_{\lambda^l > 0} = 0. \quad (3-2)$$

然而，当阈值 λ^l 为负时，误差可描述如下：

$$error^l_{\lambda^l < 0} = \begin{cases} -\lambda^l & z^l < \lambda^l, \\ \frac{\lambda^l}{L} \lfloor \frac{z^l L}{\lambda^l} \rfloor & \lambda^l \leq z^l < 0, \\ \lambda^l & z^l \geq 0. \end{cases} \quad (3-3)$$

上述方程表明，负阈值的存在会在激活函数输出中引入偏差，从而导致行为偏离预期。这一发现突显了在脉冲神经网络中保持正阈值对于与人工神经网络计算和功能一致性的关键重要性。然而，QCFS 函数中对 λ 的训练过程并不能内在保证其为正值，因为其无界且受梯度优化技术的影响。因此，根据数据分布、网络结构和学习动态的不同， λ 在迭代训练过程中可能会取负值。

3.1.2 层级别阈值的限制

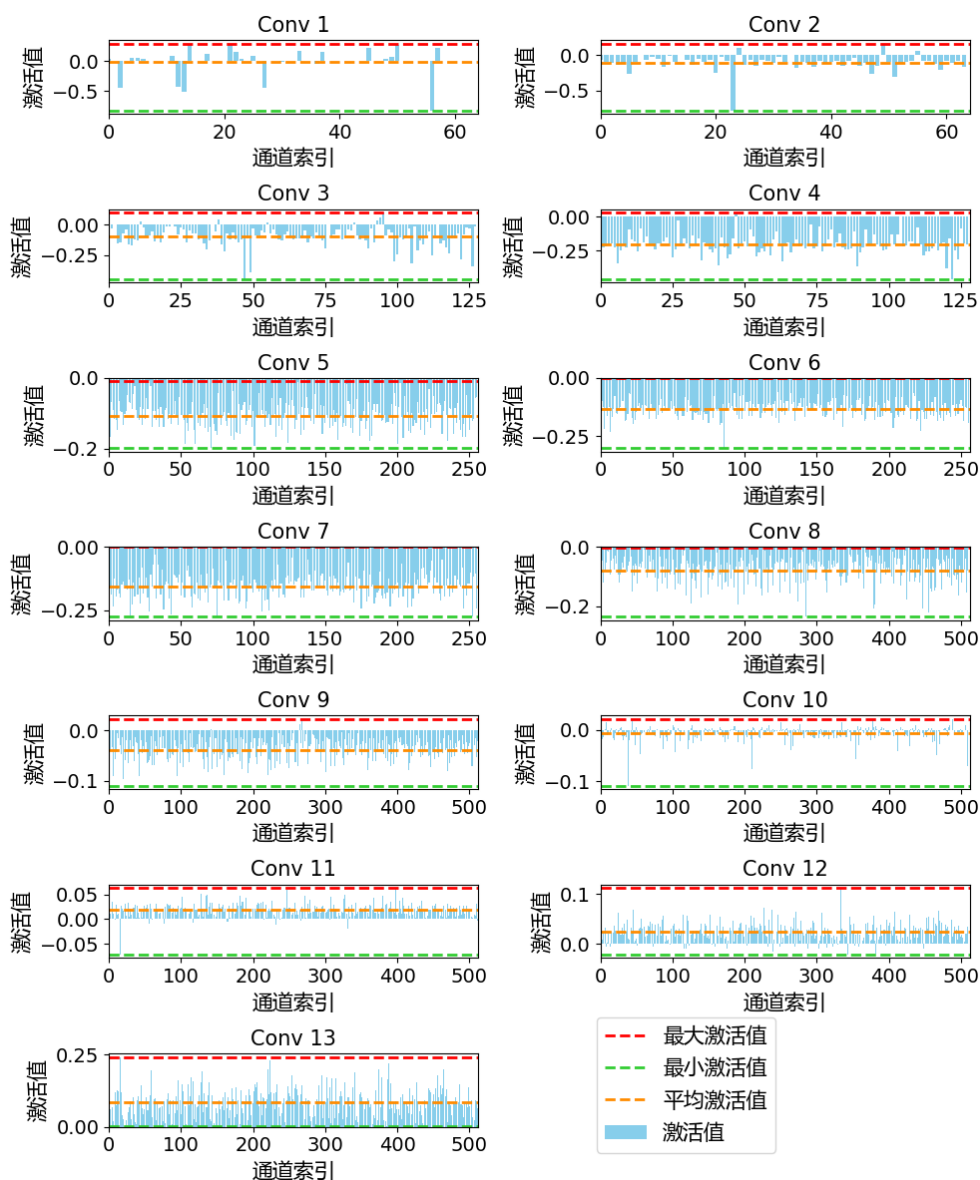


图 3-2 VGG-16 每层不同通道的平均激活值

尽管 QCFS 激活函数在消除向下舍入误差和量化误差方面表现出色，但当以层级方式应用时，仍存在显著局限性。在传统的深度学习架构中，同一层内的

不同通道通常捕获不同的特征或模式^[52]。这一现象通过各种可视化方法得到了验证，显示了不同通道能够捕获独特的特征和模式^[53-55]。然而，当为整个网络层的神经元仅分配一个 λ 时，未能顾及不同通道之间多样化的激活分布。这种同质化可能导致一些通道的表达能力未被充分利用，同时对其他通道造成过度惩罚。因此，层级阈值可能会限制脉冲神经网络的效率和潜在性能。

为了验证不同通道间激活分布的多样性，我们在 CIFAR-10 数据集上训练了一个 VGG-16 模型，并计算了模型中各层不同通道的平均激活值。详细结果如图3-2所示。在某个特定的卷积层内，不同通道的激活值存在显著差异。以 Conv1 层为例，诸如通道 14、21 和 45 的归一化最大激活值约为 0.2，而通道 2、12、13、27 和 56 的激活值则接近-0.5。在其他卷积层中也可以观察到类似的趋势，进一步证实了不同通道之间存在显著的激活差异。

3.2 基于可训练阈值的脉冲卷积神经网络训练方法

3.2.1 正阈值特性

Softplus 变换

为了确保阈值为非负，本文提出用 Softplus 函数对 λ 进行变换。Softplus 函数定义如下：

$$\text{Softplus}(x) = \log(e^x + 1), \quad (3-4)$$

该函数确保了其对于所有实值输入的输出始终为正值。从数学上看，Softplus 函数在其定义域的不同区域具有不同的行为：对于较大的正值 λ 它表现出线性特性，实际效果接近恒等函数，这使其适用于需要线性响应的大输入场景。另一方面，对于较小或负值的 λ ，Softplus 函数平滑地趋近于零，避免了负输出，同时提供了一个柔性阈值，这对稳定的神经计算至关重要。

Softplus 函数的另一个重要特性是其导数——逻辑函数 (Logistic Function)。逻辑函数的值始终为正，并且在 0 到 1 之间变化。这种稳定的梯度对基于梯度的学习算法非常有利，能够避免学习过程中的剧烈变化^[56]。此外，Softplus 函数还可以看作是 ReLU 函数的一种平滑近似。与 ReLU 在从零输出到线性增长的过程中出现的陡峭过渡不同，Softplus 提供了一种平滑且连续的过渡。这种特性结合

了 ReLU 的优点（例如促进稀疏激活的简单性和有效性），并额外提供了稳定梯度的优势。

综上所述，Softplus 函数凭借其平滑过渡、正值输出和稳定梯度等特性，在脉冲神经网络中表现出卓越的适用性。其在处理不同输入范围时保持计算稳定性的能力，对于实现有效且具有生理合理性的神经网络学习至关重要。

结合 Softplus 变换后的 QCFS 激活函数定义如下：

$$\begin{aligned} a_i^l &= \text{S-QCFS}(z^l) \\ &= \text{Softplus}(\lambda^l) \times \text{clip} \left(\frac{1}{L} \lfloor \frac{z^l L}{\text{Softplus}(\lambda^l)} + \phi \rfloor, 0, 1 \right). \end{aligned} \quad (3-5)$$

其中，关键的改进在于对可训练阈值 λ^l 应用了 Softplus 函数。这一变换对于确保有效阈值始终为正值至关重要，确保了脉冲神经元能够表现出类似于人工神经元的响应行为，从而保持两种神经网络之间的基本操作一致性。

总之，将 Softplus 函数集成到 QCFS 框架中，巧妙地保留了可训练阈值的灵活性和适应性，同时确保这些阈值始终为正值。因此，这种方法不仅保留了 QCFS 函数的特性，还使脉冲神经网络的操作动态与人工神经网络更加一致。

为什么使用 Softplus

尽管指数函数 ($\text{Exp}(x) = e^x$) 同样可以确保 λ 为正值，但在脉冲神经网络中，Softplus 函数通常更受青睐，因为它更符合神经计算的需求。Softplus 函数提供了平滑且逐渐的增长，使其非常适合在阈值中产生受控的响应。相比之下，指数函数的增长速度过快，对于较大的正输入会变得过于敏感且不稳定。

此外，Softplus 具有稳定的梯度，这对有效的反向传播至关重要，能够避免诸如梯度爆炸等问题。这种稳定性还体现在 Softplus 通常与 λ 的数量级相近，从而确保参数和激活值保持在相似的尺度范围内。

总之，在脉冲神经网络中，Softplus 因其渐进的非线性、计算稳定性以及与 λ 保持相似尺度的特性而优于指数函数。这些特性使其在神经网络中实现精细的阈值调整方面特别适用，不仅能确保阈值为正，还能促进高效且有效的学习动态。

3.2.2 通道差异性

通道级别阈值

为了解决层级阈值限制，我们的方法从层级 QCFS 函数转向通道级 QCFS 函数。在该方法中，层 l 内的每个通道 i 都配备了其独立的可训练阈值 λ_i^l 。通道级 QCFS (C-QCFS) 函数定义如下：

$$a_i^l = \text{C-QCFS}(z^l) = \lambda_i^l \text{clip}\left(\frac{1}{L} \left\lfloor \frac{z^l L}{\lambda_i^l} + \phi \right\rfloor, 0, 1\right), \quad (3-6)$$

其中， λ_i^l 是层 l 中通道 i 的可训练阈值。对于层 l 中的每个通道 i ，其激活值 a_i^l 由对应的 λ_i^l 计算得出，从而能够基于输入分布提供更具适应性的响应。

引入通道级阈值显著增强了脉冲神经网络的表征能力。通过让每个通道能够独立调整其阈值以适应所处理的输入，这种方法实现了更细致的特征提取和改进的学习动态。

总的来说，通道级 QCFS 激活函数代表了脉冲神经网络设计和功能的一项重要进步。通过为层内的每个通道配备独立且可训练的阈值，该方法为更高级、适应性更强且更高效的神经计算铺平了道路。它使脉冲神经网络更接近于生物神经处理的复杂特性，从而扩展了脉冲神经网络在各种应用中的潜力。

3.2.3 CS-QCFS

为了解决 QCFS 中层级阈值的局限性并确保阈值为正值，我们提出了通道级 Softplus QCFS 激活函数。这一新激活函数结合了通道级阈值适应的优势和 Softplus 变换确保正阈值的特性，其定义如下：

$$\begin{aligned} a_i^l &= \text{CS-QCFS}(z^l) \\ &= \text{Softplus}(\lambda_i^l) \times \text{clip}\left(\frac{1}{L} \left\lfloor \frac{z^l L}{\text{Softplus}(\lambda_i^l)} + \phi \right\rfloor, 0, 1\right). \end{aligned} \quad (3-7)$$

QCFS 函数的所有改进，包括通道级阈值和 Softplus 函数的引入，都仅在人工神经网络训练阶段进行。这种策略不会影响随后的人工神经网络到脉冲神经网络的转换过程，也不会改变脉冲神经网络的运行动态，从而保持了脉冲神经网络

固有的低能耗特性。总体而言，CS-QCFS 激活函数在脉冲神经网络的设计与功能上实现了重要突破。

求导规则

在推导 CS-QCFS 函数的导数规则时，我们基于原始 QCFS 函数的原则，同时引入了通道级阈值和 Softplus 变换。该方法使用了直通估计器 (straight-through estimator)^[57]来处理取整函数的导数，其定义为 $\frac{d\lfloor x \rfloor}{dx} = 1$ 。CS-QCFS 函数的详细导数规则如下所示 (Eq.(3-8)和 Eq. (3-9)):

$$\frac{\partial f_{ij}(z^l)}{\partial z_{ij}^l} = \begin{cases} 1 & -\frac{\lambda_i^l}{2L} < z_{ij}^l < \lambda_i^l - \frac{\lambda_i^l}{2L}, \\ 0 & \text{otherwise.} \end{cases} \quad (3-8)$$

$$\frac{\partial f_{ij}(z^l)}{\partial \lambda_i^l} = \begin{cases} \frac{\text{Sigmoid}(\lambda_i^l)f(z^l) - z^l e^{\lambda_i^l}}{\text{Softplus}(\lambda_i^l)} & -\frac{\lambda_i^l}{2L} \leq z_{ij}^l < \lambda_i^l - \frac{\lambda_i^l}{2L}, \\ 0 & z_{ij}^l < -\frac{\lambda_i^l}{2L}, \\ 1 & z_{ij}^l \geq \lambda_i^l - \frac{\lambda_i^l}{2L}. \end{cases} \quad (3-9)$$

在上述公式中， f 表示 CS-QCFS 函数， z_{ij}^l 表示 z^l 中第 i 个通道的第 j 个元素。Eq. (3-8)提供了一种简单而有效的方式，用于计算神经元输出相对于输入的偏导数，从而促进梯度在网络中的传播。Eq. (3-9)则表示神经元输出相对于通道级阈值 λ_i^l 的偏导数。

借助这些导数规则，使用 CS-QCFS 激活函数的人工神经网络可以使用随机梯度下降算法 (SGD) 有效训练。这一训练方法允许网络在学习过程中调整通道特定的阈值 λ_i^l ，从而增强脉冲神经网络的表征能力和适应性。这种方法在开发更细致、更高效的脉冲神经网络方面迈出了重要一步，不仅更贴合生物神经处理的原则，也与人工神经网络的计算范式高度一致。

CS-QCFS 的训练

直接训练通道级阈值而缺乏先验知识可能导致收敛效果不佳或陷入不理想的局部最优点。其主要原因在于扩展参数空间后对初始值的敏感性。当为每个

算法 1 CS-QCFS 的 ANN-SNN 转换流程**输入:** 使用权重 W 初始化的 ANN 模型 $M_{\text{ANN}}(x; W)$; 数据集 D **参数:** 量化步长 L ; 初始化阈值 λ_0 ; 学习率 ϵ ; 微调学习率 ϵ_{ft} **输出:** $M_{\text{SNN}}(x; \hat{W})$

```

1: for  $l = 1$  to  $M_{\text{ANN}}.\text{layers}$  do
2:   if 是 ReLU 激活函数 then
3:     使用 QCFS( $x; L, \lambda_0$ ) (Eq. (2-23)) 替换 ReLU( $x$ )
4:   end if
5: end for
6: 使用学习率为  $\epsilon$  的随机梯度下降法 (SGD) 训练层级别的 QCFS 阈值  $\lambda^l$ 
7: for  $l = 1$  to  $M_{\text{ANN}}.\text{layers}$  do
8:   if 是 QCFS 激活函数 then
9:     通过 CS-QCFS( $x; L, \lambda^l$ ) (Eq. (3-7)) 替换 QCFS( $x; L, \lambda_0$ )
10:  end if
11: end for
12: 使用学习率为  $\epsilon_{ft}$  的随机梯度下降法 (SGD) 训练通道级别的 CS-QCFS 阈值

13: for  $l = 1$  to  $M_{\text{ANN}}.\text{layers}$  do
14:    $M_{\text{SNN}}.\hat{W}^l \leftarrow M_{\text{ANN}}.W^l$   $\triangleright$  将权重从 ANN 传输到 SNN
15:   for  $c = 1$  to  $M_{\text{ANN}}.l_i.\text{channels}$  do
16:      $M_{\text{SNN}}.V_i^l \leftarrow M_{\text{ANN}}.\lambda_i^l$   $\triangleright$  设置 SNN 的阈值
17:      $M_{\text{SNN}}.v_i^l(0) \leftarrow M_{\text{SNN}}.V_i^l/2$   $\triangleright$  初始化 SNN 的膜电压
18:   end for
19: end for
20: return  $M_{\text{SNN}}$ 

```

通道训练独立的阈值时，参数空间会随之扩展，这可能会在训练初期阶段引发不稳定。此外，对不同层的所有通道采用统一的初始阈值无法从一开始就考虑各层间的特性差异。如果没有层级训练提供的上下文信息，这种统一的方法难以有效地指导通道级阈值的优化演化。

为解决上述问题，采用了一种逐步训练策略：首先，使用层级 QCFS 激活函数训练模型，为整个层获取一个通用的近似阈值。随后，将得到的层级阈值 λ^l 用作 CS-QCFS 激活函数中通道级阈值 λ_i^l 的起点，网络随后进入微调阶段。这种方法确保每个通道在微调阶段开始时具有一个由层级训练提供上下文信息的阈值。算法1展示了人工神经网络到脉冲神经网络的转换过程。

通过在逐步优化中整合层级训练的先验信息，这种粗到细的优化策略为每个通道提供了更加平滑且信息丰富的优化空间，从而加快了收敛速度并增强了收敛的稳健性。这种集成的方法预计能够实现从人工神经网络到脉冲神经网络更加稳健且高效的转换。

3.3 实验与分析

在本节中，我们通过图像分类数据集 CIFAR-10 和 CIFAR-100^[58]上的实验，验证了本文方法的有效性和可扩展性。实验中使用的网络架构包括 ResNet-18、ResNet-20、ResNet-50、ResNet-101^[59]以及 VGG-16^[60]。此外，我们将本文的方法与当前最先进的 ANN-SNN 转换方法进行对比，包括 QCFS^[31]、SRP^[32]和 COS^[33]。

3.3.1 数据集与模型

数据集： CIFAR-10 和 CIFAR-100 是由加拿大多伦多大学提出的广泛使用的图像分类数据集^[58]，主要用于评估图像分类模型的性能。CIFAR-10 包含 60,000 张 32×32 的彩色图像，其中 50,000 张用于训练，10,000 张用于测试。数据集中包含 10 个类别，每个类别均匀分布有 6,000 张图像，类别包括飞机 (airplane)、汽车 (automobile)、鸟 (bird)、猫 (cat)、鹿 (deer)、狗 (dog)、青蛙 (frog)、马 (horse)、船 (ship) 和卡车 (truck)。由于其规模适中且分类任务相对简单，CIFAR-10 常被用作深度学习模型的基础评测任务。CIFAR-100 与 CIFAR-10 类似，但具有更细粒度的类别划分，包含 100 个类别，每个类别有 600 张图像，其中 500 张用于训练，100 张用于测试。相比 CIFAR-10，CIFAR-100 分类任务更具挑战性，因为类别数量增加且图像类别间的相似度更高。

模型： 在实验中，我们采用了经典的 ResNet 和 VGG 网络架构来验证方法的有效性和通用性。ResNet 系列（包括 ResNet-18、ResNet-20、ResNet-50 和 ResNet-101）以其引入残差模块缓解梯度消失问题著称，能够在保持高效训练的同时支持更深层的网络设计；而 VGG 系列（以 VGG-16 为代表）通过堆叠简单的 3×3 卷积核实现高效特征提取，以其模块化设计和良好的迁移性能广泛应用。这些模型覆盖了从浅层到深层、从轻量到复杂的多种网络结构，为验证方法在多样化架构中的表现提供了坚实的基础。

3.3.2 实验设置

训练与优化： 在人工神经网络模型的训练中，我们采用带有 0.9 动量参数的随机梯度下降 (SGD) 优化器^[61]。训练分为两个阶段：首先，对于 CIFAR-10 数

数据集，模型以 0.1 的基础学习率训练 300 个轮次；对于 CIFAR-100 数据集，基础学习率为 0.05。在训练 ResNet-50 和 ResNet-101 时，CIFAR-10 和 CIFAR-100 数据集的基础学习率均设置为 0.01。随后，在微调阶段将学习率降至初始学习率的 1/10，并继续训练 100 个轮次。学习率采用余弦退火调度器^[62]进行动态调整。为了正则化训练过程，在两个数据集上均采用 5×10^{-4} 的权重衰减。

数据增强: 为提升模型性能^[63]，在所有数据集上，我们使用了随机裁剪、Cutout^[64]和 AutoAugment^[65]等数据增强技术。

CS-QCFS 参数: 在实验中，我们为每个数据集校准了量化步长 L 。具体而言，对于 CIFAR-10 数据集，我们在所有网络架构（包括 ResNet-18、ResNet-20、ResNet-50、ResNet-101 和 VGG-16）中均设置 $L = 4$ ；对于 CIFAR-100 数据集， L 在 VGG-16、ResNet-50 和 ResNet-101 中设置为 4，而在 ResNet-20 中设置为 8。此外，在 CIFAR-10 数据集的实验中，初始阈值设置为 4；对于 CIFAR-100 数据集的 ResNet-50、ResNet-101 和 VGG-16 实验，初始阈值也为 4，而在 CIFAR-100 的 ResNet-20 实验中，初始阈值设置为 8。

表 3-1 实验参数设置

数据集	模型	优化器	量化步长 L	初始阈值 λ	学习率
CIFAR10	ResNet-18	SGD	4	4	0.1
	ResNet-20				0.1
	VGG-16				0.1
	ResNet-50				0.01
	ResNet-101				0.01
CIFAR100	ResNet-18	SGD	4	4	0.05
	ResNet-20		8	8	0.05
	VGG-16		4	4	0.01
	ResNet-50		4	4	0.01

3.3.3 CIFAR 数据集对比实验

我们首先评估了集成了原始 QCFS 激活函数、本文所提出的 CS-QCFS 激活函数，以及标准 ReLU 激活函数的人工神经网络的性能。图3-3分别展示了 VGG-16 在 CIFAR-10 上的结果、ResNet-20 在 CIFAR-10 上的结果、VGG-16 在 CIFAR-100 上的结果以及 ResNet-20 在 CIFAR-100 上的结果。

可以观察到，使用 CS-QCFS 激活函数的人工神经网络与使用原始 QCFS 激活函数的人工神经网络表现相当。值得注意的是，无论是原始的 QCFS 激活函数

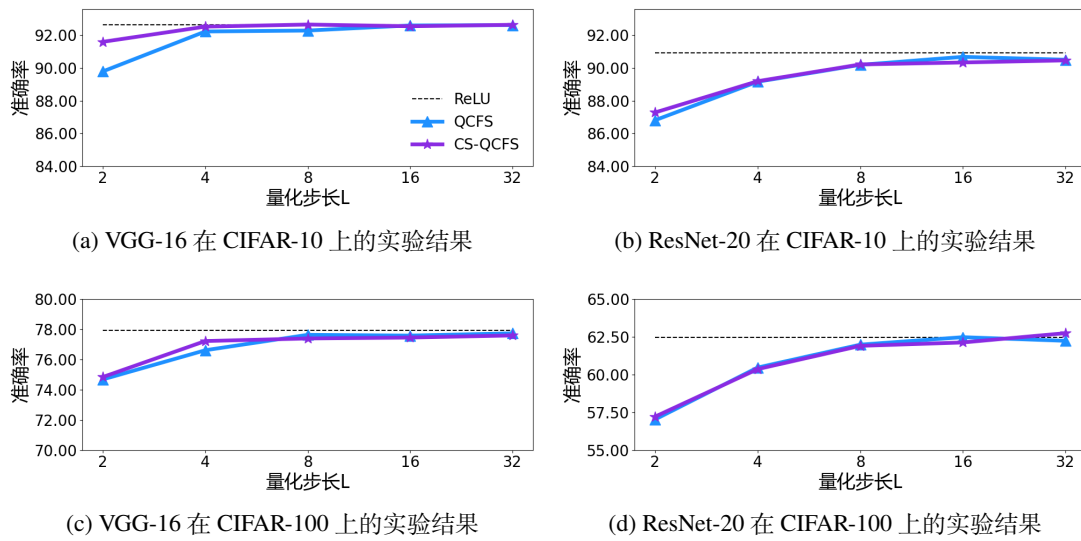


图 3-3 神经网络精度比较

还是本文的 CS-QCFS 激活函数，当量化步长 $L > 4$ 时，都表现出与传统 ReLU 激活函数神经网络相近的性能。这表明集成 CS-QCFS 激活函数并不会削弱神经网络的性能。因此，CS-QCFS 激活函数是 ANN-SNN 转换中的更优选择。它不仅能够保持神经网络的性能，还在保留原始 QCFS 激活函数特性的同时，使脉冲神经网络的行为更贴近神经网络，从而更好地实现两者的转换。

然后我们将所提出的方法与当前最先进的 ANN-SNN 转换方法进行对比，包括 QCFS、SRP 和 COS，以验证其有效性。此外，我们还利用偏移脉冲 (offset spikes)^[33]，进一步减少转换误差。

表3-2中展示了这些方法在 CIFAR-10 数据集上不同时间步下的测试准确率。对于 ResNet-18，本文的方法在 $T = 1$ 时达到 95.86% 的准确率，超越了所有其他方法在 $T = 32$ 时的表现。在 $T = 32$ 时，本文方法的准确率进一步提升至 96.51%。对于 VGG-16，本文的方法在 $T = 4$ 时始终优于其他方法，准确率达 95.70%，显著优于其他方法在 $T = 32$ 时的表现。对于 ResNet-20，本文的方法在 $T = 1$ 时达到 91.24% 的准确率，超过了 COS (89.88%) 和 SRP (86.37%)。到 $T = 32$ 时，准确率提升至 92.41%，同样优于 COS (92.16%) 和 SRP (91.72%)。对于 ResNet-50，本文的方法在 $T = 1$ 时达到 96.72%，在 $T = 32$ 时进一步提升至 96.90%，始终领先。对于 ResNet-101，本文的方法在 $T = 1$ 时达到 96.66%，并在 $T = 32$ 时保持这一表现，在每个时间步都优于 QCFS、SRP 和 COS。这些

表 3-2 CIFAR-10 数据集实验结果

网络架构	方法	ANN*	T=1	T=2	T=4	T=8	T=16	T=32
ResNet-18	QCFS ^[31]	95.64%	-	91.75%	93.83%	95.04%	95.56%	95.67%
	SRP ^[32]	95.67%	94.59%	95.06%	95.25%	95.60%	95.55%	95.55%
	COS ^[33]	95.64%	95.25%	95.45%	95.46%	95.66%	95.68%	95.68%
	Ours	96.48%	95.86%	96.24%	96.34%	96.41%	96.47%	96.51%
VGG-16	QCFS ^[31]	95.52%	-	91.18%	93.96%	94.95%	95.40%	95.54%
	SRP ^[32]	95.52%	93.80%	94.47%	95.32%	95.52%	95.44%	95.42%
	COS ^[33]	95.51%	94.90%	95.36%	95.46%	95.51%	95.57%	95.61%
	Ours	95.67%	94.93%	95.47%	95.70%	95.78%	95.93%	95.88%
ResNet-20	QCFS ^[31]	91.77%	-	73.20%	83.75%	89.55%	91.62%	92.24%
	SRP ^[32]	91.77%	86.37%	88.73%	90.51%	91.37%	91.64%	91.72%
	COS ^[33]	91.77%	89.88%	91.26%	91.68%	91.86%	92.20%	92.16%
	Ours	91.80%	91.24%	91.66%	91.74%	92.00%	92.29%	92.41%
ResNet-50	QCFS ^[31]	96.81%	69.07%	84.29%	91.20%	94.30%	95.99%	96.58%
	SRP ^[32]	96.81%	81.01%	92.24%	95.34%	95.91%	96.13%	96.22%
	COS ^[33]	96.81%	95.99%	96.55%	96.58%	96.58%	96.65%	96.70%
	Ours	96.86%	96.72%	96.62%	96.70%	96.76%	96.82%	96.90%
ResNet-101	QCFS ^[31]	96.74%	32.17%	59.32%	84.02%	91.75%	94.33%	95.80%
	SRP ^[32]	96.74%	50.54%	82.14%	91.48%	94.09%	94.89%	95.19%
	COS ^[33]	96.74%	96.58%	96.59%	96.61%	96.55%	96.55%	96.60%
	Ours	96.92%	96.66%	96.81%	96.83%	96.77%	96.67%	96.66%

ANN* 是用 QCFS/CS-QCFS 替换 ReLU 后的准确率。

结果突出了本文方法的效率和效果，展示了其在更少时间步内实现高准确率的能力，并在不同网络架构中保持一致性能。

我们进一步扩展实验，评估本文方法在 CIFAR-100 数据集上的泛化能力，结果如表3-3所示。对于 VGG-16，本文的方法在所有时间步上均优于其他方法，在 $T = 4$ 时达到 76.87% 的准确率，在 $T = 32$ 时达到 77.64%，比最近的竞争者 QCFS 提高了 0.6 对于 ResNet-20，本文的方法在 $T = 1$ 时达到 62.47% 的准确率，显著高于 SRP 的 48.33% 和 COS 的 59.43%，并在随后的时间步中保持领先。对于 ResNet-50，本文的方法在 $T = 1$ 时达到 78.25% 的准确率，超越 QCFS 和 SRP 在更高时间步时的表现。在 $T = 32$ 时，准确率达到 81.60%，为所有方法中的最高值。对于 ResNet-101，本文的方法在 $T = 1$ 时达到 78.28% 的准确率，优于 QCFS 和 SRP。尽管 COS 在 $T = 32$ 时略微超过本文的方法，但总体来看，本文的方法表现依然具有竞争力。

因此，CIFAR-100 数据集的实验结果进一步验证了本文方法的泛化能力，以及其在超低延迟条件下表现出的性能优势和一致性。

表 3-3 CIFAR-100 数据集实验结果

网络架构	方法	ANN	T=1	T=2	T=4	T=8	T=16	T=32
VGG-16	QCFS ^[31]	76.28%	-	63.79%	69.62%	73.96%	76.24%	77.01%
	SRP ^[32]	76.28%	71.52%	74.31%	75.42%	76.25%	76.42%	76.45%
	COS ^[33]	76.28%	74.24%	76.03%	76.26%	76.52%	76.77%	76.96%
	Ours	76.84%	74.83%	76.56%	76.87%	77.26%	77.66%	77.64%
ResNet-20	QCFS ^{*[31]}	68.30%	14.37%	20.91%	35.21%	55.47%	66.18%	68.52%
	SRP ^{*[32]}	68.30%	48.33%	55.48%	60.28%	63.20%	64.81%	65.26%
	COS ^{*[33]}	68.30%	59.43%	63.62%	64.70%	66.17%	67.86%	68.74%
	Ours	68.54%	62.47%	64.91%	65.35%	66.61%	68.28%	68.94%
ResNet-50	QCFS ^[31]	80.82%	38.55%	48.47%	58.75%	69.48%	76.57%	79.88%
	SRP ^[32]	80.82%	55.98%	71.21%	73.81%	75.35%	76.25%	76.65%
	COS ^[33]	80.82%	77.02%	80.27%	80.53%	80.71%	80.94%	81.10%
	Ours	81.10%	78.25%	80.78%	80.80%	81.07%	81.38%	81.60%
ResNet-101	QCFS ^[31]	80.53%	19.41%	32.06%	44.43%	58.72%	70.11%	76.81%
	SRP ^[32]	80.53%	7.14%	53.54%	61.23%	65.80%	68.49%	69.82%
	COS ^[33]	80.53%	77.89%	80.13%	80.23%	80.34%	80.22%	80.29%
	Ours	80.66%	78.28%	80.11%	80.23%	80.44%	80.17%	80.15%

标有 * 的结果是我们根据作者提供的代码复制的结果，实验设置与本文的方法相同

表 3-4 通道级别阈值和 Softplus 变换的消融实验结果

网络架构		ANN	T=1	T=2	T=4	T=8	T=16	T=32
VGG-16	w/o channel	95.37%	94.68%	95.20%	95.33%	95.42%	95.50%	95.57%
	w/o Softplus	95.36%	94.91%	95.16%	95.31%	95.45%	95.41%	95.44%
	Full	95.67%	94.93%	95.47%	95.70%	95.78%	95.93%	95.88%
ResNet-20	w/o channel	91.48%	90.04%	91.06%	91.29%	91.68%	92.10%	92.13%
	w/o Softplus	91.35%	90.69%	91.05%	91.25%	91.49%	91.80%	92.01%
	Full	91.80%	91.24%	91.66%	91.74%	92.00%	92.29%	92.41%

3.3.4 消融实验

在本节中，我们评估了方法中关键组件的单独影响：通道阈值机制、Softplus 函数的引入，以及两阶段训练策略的作用。

通道级别阈值和 Softplus 函数

为了测试通道阈值机制和 Softplus 函数在 QCFS 激活函数中的潜力，我们对比了以下几种配置：标准 QCFS 激活函数、仅使用通道阈值的 QCFS 激活函数 (w/o Softplus)、仅使用 Softplus 的 QCFS 激活函数 (w/o channel)，以及完整模型 (Full)。实验结果如表3-4所示。

实验结果表明，移除通道阈值机制 (w/o channel) 后，与完整模型 (Full) 相比，在所有时间步下的性能均略有下降。以 VGG-16 为例，当时间步 $T = 32$ 时，完整模型的精度为 95.88%，而移除通道阈值后的精度下降至 95.57%。类似地，在 ResNet-20 中，当时间步 $T = 1$ 时，完整模型的精度为 91.24%，移除通道阈值

后的精度为 90.04%。这些结果表明，通道阈值机制能够更好地捕获不同通道间的特性差异，优化激活阈值的分布，从而提升网络的性能。

另一方面，移除 Softplus 函数（w/o Softplus）对模型的性能也有一定影响。以 VGG-16 为例，当时间步 $T = 32$ 时，移除 Softplus 函数后的精度为 95.44%，低于完整模型的 95.88%。在 ResNet-20 中，移除 Softplus 函数后的精度为 92.01%，同样低于完整模型的 92.41%。这表明，Softplus 函数在优化过程中通过其非负性约束提高了激活阈值的稳定性，确保了脉冲神经网络的训练与推理性能。

综合来看，通道阈值机制和 Softplus 函数在提升 QCFS 激活函数性能方面各自发挥了关键作用。通道阈值机制通过引入通道级别的灵活性，有效提升了网络对复杂特征的表达能力。而 Softplus 函数通过提供非负约束，确保了激活阈值的合理性与稳定性。两者的结合（完整模型）能够进一步发挥协同作用，显著提升神经网络到脉冲神经网络的转换精度和鲁棒性。

两阶段训练

在本部分中，我们还研究了两阶段训练的概念，旨在回答一个基本问题：性能的提升是来自通道级策略，还是仅仅由于使用了两阶段训练过程？

为了解答这一问题，我们对三种不同的训练策略进行了比较分析。第一种策略是层级单阶段训练，在单个阶段中对使用层级 QCFS 激活函数的网络直接进行训练。第二种策略是层级两阶段训练，先对使用层级 QCFS 激活函数的网络进行标准训练，然后在较低的学习率下进行微调。最后一种策略是通道级两阶段训练，首先训练使用层级 QCFS 激活函数的网络，然后切换到使用通道级 QCFS 激活函数的网络进行微调。

表 3-5 两阶段训练消融实验结果

数据集	网络架构	ANN	T=1	T=2	T=4	T=8	T=16	T=32	
CIFAR-10	VGG-16	origin	95.30%	94.49%	95.01%	95.26%	95.33%	95.47%	95.48%
		layer-wise	95.50%	94.67%	95.24%	95.47%	95.52%	95.62%	95.63%
		channel-wise	95.43%	95.01%	95.30%	95.35%	95.47%	95.51%	95.48%
	ResNet-20	origin	91.41%	89.82%	90.95%	91.27%	91.58%	92.00%	92.17%
		layer-wise	91.85%	90.03%	91.31%	91.65%	92.02%	92.36%	92.41%
		channel-wise	91.53%	90.87%	91.36%	91.47%	91.65%	92.02%	92.21%
CIFAR-100	VGG-16	origin	76.57%	74.67%	76.16%	76.52%	76.90%	77.11%	77.17%
		layer-wise	76.46%	74.83%	76.04%	76.43%	76.68%	77.01%	77.12%
		channel-wise	76.52%	75.53%	76.24%	76.46%	76.70%	77.02%	77.07%
	ResNet-20	origin	68.82%	59.88%	64.15%	65.38%	66.60%	68.56%	69.19%
		layer-wise	69.18%	59.89%	64.12%	65.47%	66.94%	68.86%	69.60%
		channel-wise	68.77%	62.46%	65.13%	65.75%	66.86%	68.44%	69.20%

如表3-5所示，“origin”、“layer-wise”和“channel-wise”分别表示层级单阶段训练、层级两阶段训练和通道级两阶段训练。CIFAR-10 的实验结果显示，层级两阶段训练在确实提升了人工神经网络准确率的同时，对脉冲神经网络准确率的提升却只有微小的效果。值得注意的是，通道级两阶段训练在低延迟场景下表现出更显著的改进。具体来说，对于 ResNet-20，虽然在 $T = 1$ 时，使用层级两阶段训练的脉冲神经网络准确率从 89.82% 提高到了 90.03%，但通道级两阶段训练进一步将脉冲神经网络准确率提高到了 90.87%。这一结论在 CIFAR-100 的结果中也得到了验证，进一步强化了上述结论。

综上所述，实验结果表明，虽然两阶段训练在一定程度上带来了提升，但在低延迟场景下，性能改进的主要驱动因素是通道级微调策略。通道级微调使得每个通道能够进行更细粒度的调整，有效捕获其特定的特征，从而最终构建出性能更强的神经网络模型。

量化步长 L

与 QCFS 函数类似，在本文提出的 CS-QCFS 方法中，参数 L 表示量化步数的超参数。它在量化过程中起着关键作用，直接影响转换后的脉冲神经网络的准确性和性能。为深入研究不同量化步数 L 值的影响并确定最佳设置，我们设计了一系列实验。这些实验将 CS-QCFS 激活函数分别应用于 VGG-16 和 ResNet-20 网络，量化步数 L 的取值范围包括 2、4、8、16 和 32。训练完成后，这些网络被转换为脉冲神经网络，以评估不同量化级别对网络在脉冲域中的表现影响，从而为优化脉冲神经网络量化策略提供重要的见解。

为了在脉冲神经网络中实现低延迟与高准确率的最佳平衡，我们从两个关键方面进行评估。首先，考察脉冲神经网络在超低延迟（尤其是 4 个时间步内）下的准确率；其次，评估脉冲神经网络能够达到的峰值准确率。实验结果如图3-4所示，展示了转换后的脉冲神经网络在 CIFAR-10/100 数据集上的性能。

实验结果显示出一个明确的趋势：当量化步数 L 增大时，脉冲神经网络在超低延迟下的准确率会下降。相反，过小的 L 会限制模型的能力，从而导致准确率降低。值得注意的是，当 $L = 2$ 时，脉冲神经网络最佳准确率与原始人工神经网络之间的差距显著。然而，当 L 增加到 4 以上时，脉冲神经网络的最佳准确率更接近于源人工神经网络的表现。从这些观察中可以看出， L 的最佳选择在

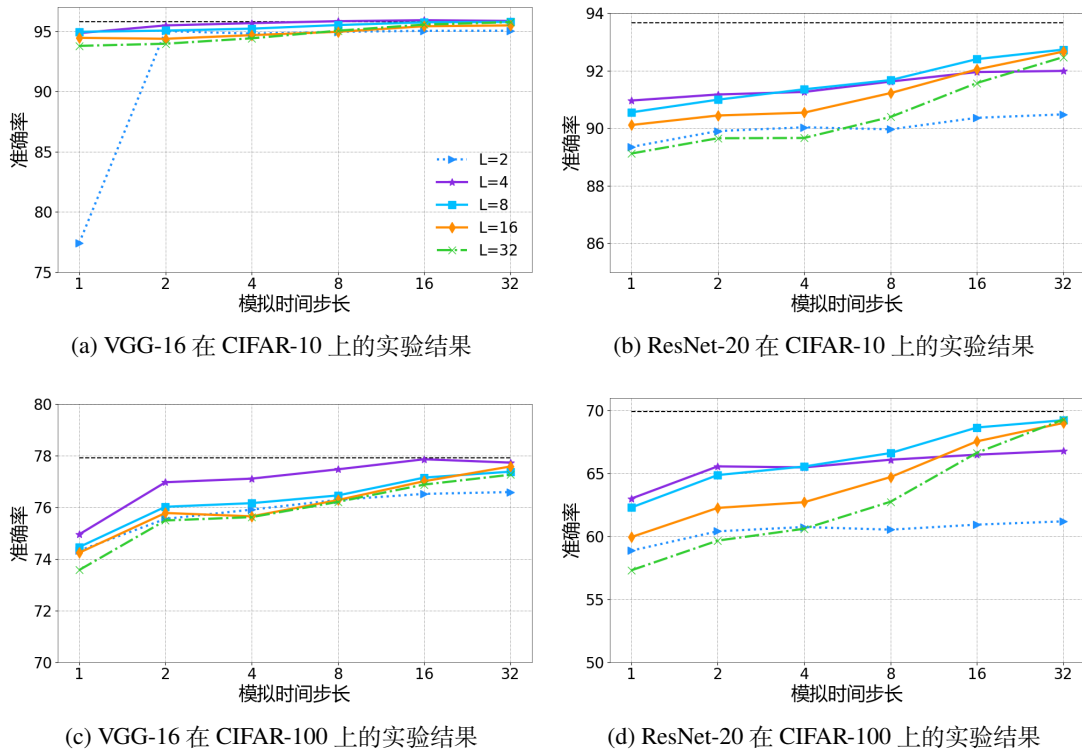


图 3-4 不同量化步长的影响

很大程度上依赖于低延迟和最大化准确率之间的权衡。基于本文的分析，建议将 L 设置为 4 或 8，这些值能够使转换后的脉冲神经网络在低时间步和高时间步下都表现优异。

综上所述，本文的消融实验阐明了通道级调整、Softplus 引入以及两阶段训练策略的优势。这些改进共同解决了传统 QCFS 方法中的局限性，为开发更强大、更高效的脉冲神经网络实现铺平了道路。

3.3.5 理论支持实验

为探究本文对 QCFS 函数的改进方法的理论合理性，我们设计了多维度实证实验，旨在为通道级阈值与 Softplus 变换的改进机制提供统计一致性证据。

通道阈值理论支持性实验

引入通道级阈值的主要目的是应对不同通道之间可能存在的差异性。为了确定通道级阈值的理论合理性，我们重点量化网络各层中不同通道阈值的差异性。实验将 C-QCFS 和 CS-QCFS 作为激活函数集成到 VGG-16 和 ResNet-20 架构

表 3-6 通道级别阈值的理论验证实验

激活函数	数据集	网络架构	<i>MaxDiff</i>	<i>AvgDiff</i>
C-QCFS	CIFAR-10	VGG-16	339.7%	153.4%
		ResNet-20	155.3%	68.2%
	CIFAR-100	VGG-16	246.2%	99.5
		ResNet-20	98.4%	45.5%
CS-QCFS	CIFAR-10	VGG-16	55.3%	10.5%
		ResNet-20	38.2%	16.1%
	CIFAR-100	VGG-16	65.8%	14.0%
		ResNet-20	44.4%	16.9%

中，并在 CIFAR-10 和 CIFAR-100 数据集上进行评估。实验结果汇总于表3-6中。其中，指标 *MaxDiff* 定义为模型所有层中 *Diff* 值的最大值，而 *Diff* 表示单层内最高和最低阈值的比例差，计算公式为 $Diff = (MaxThre - MinThre)/MinThre$ 。其中，*MaxThre* 是某一层通道中最大阈值，*MinThre* 是最小阈值。类似地，*AvgDiff* 指标计算模型所有层中 *Diff* 值的平均值。

在所有模型和数据集中，C-QCFS 和 CS-QCFS 的 *MaxDiff* 和 *AvgDiff* 均呈现出较大的值。例如，在 CIFAR-10 数据集上，使用 C-QCFS 的 VGG-16 模型的 *MaxDiff* 达到 339.7%，而 *AvgDiff* 为 153.4%。即使在 CS-QCFS 中引入了 Softplus 函数后，仍然存在显著的差异，例如在 CIFAR-100 数据集上，VGG-16 模型的 *MaxDiff* 为 65.8%，*AvgDiff* 为 14.0%。

Softplus 函数通过其对输入变化的平滑非线性响应，促进了通道间更平衡的阈值调整。其确保阈值为正的特性适度缩小了阈值范围，从而减小了极端差异。然而，无论是 C-QCFS 还是 CS-QCFS，持续存在的大 *MaxDiff* 和 *AvgDiff* 值表明通道间的固有差异性。这一观察突显了统一阈值可能限制网络学习能力和表征能力的事实。

因此，这进一步验证了采用通道级阈值的合理性。这一策略通过适应不同通道多样化的特征处理需求，能够充分挖掘网络的潜能。

负阈值理论支持性实验

引入 Softplus 函数旨在解决 QCFS 函数中可能出现的负阈值问题。为了探讨其合理性，我们分析了未经 Softplus 调整的网络中得到的阈值。

在实验中，我们分别在 CIFAR-10 和 CIFAR-100 数据集上训练 ResNet-20 模型，发现特别是在初始阈值较低的情况下，出现了负阈值的现象。这一现象如

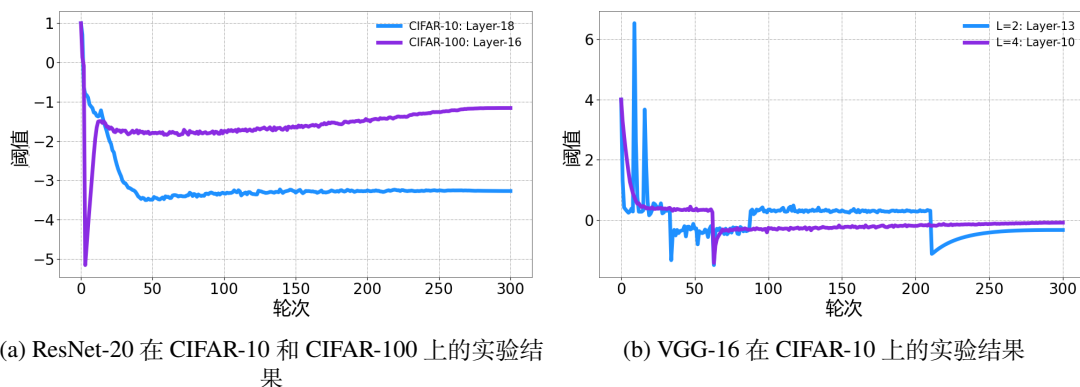


图 3-5 训练过程中的负阈值

图3-5(a)所示。此外，在使用 QCFS 建议的初始阈值时，我们在 CIFAR-10 数据集上训练 VGG-16 模型时，也在多种超参数 L 设置下遇到了负阈值的情况，如图3-5(b)所示。这一发现至关重要，因为它表明在没有任何修正措施的情况下，训练过程中确实可能产生不仅次优且与 ANN-SNN 转换原理根本不兼容的阈值。

无论是数据集还是模型架构，负阈值的持续出现都证实了 Softplus 函数的必要性。通过将阈值 λ 重新参数化为 $\text{Softplus}(\lambda)$ ，Softplus 函数始终能够保证阈值为正，从而与 ANN-SNN 转换的核心理念保持一致。这种修正不仅是预防性的增强，更是对一个根本性缺陷的关键修复，否则该缺陷可能会破坏脉冲神经网络在实际应用中的可靠性和适用性。

综上所述，我们的理论支持实验证实了通道级调整和 Softplus 函数不仅仅是功能上的增强，而是确保脉冲神经网络有效性和一致性运行的关键要素。

3.4 本章小结

在本章中，我们针对 QCFS 函数的两大局限性提出了创新性解决方案，为 ANN-SNN 转换领域做出了重要贡献。首先，我们深入分析了脉冲神经网络中通道差异性和负阈值问题的核心挑战，发现传统方法中统一的阈值设定无法有效捕捉通道间的差异，并且负阈值的存在可能导致脉冲神经网络性能的不稳定性，且与 ANN-SNN 转换的核心思想相违背。

为应对这些问题，我们提出了通道自适应阈值的改进方法，通过为每个通道独立训练阈值，精准适配通道间的特性差异，从而显著提升了网络的特征提取能力和表征能力。同时，我们引入 Softplus 变换，确保训练得到的阈值始终为

正。该方法不仅解决了负值阈值的问题，还进一步拉近了脉冲神经网络与人工神经网络在行为特性和性能上的一致性。

在实验评估中，我们改进后的 CS-QCFS 函数在 CIFAR 数据集上取得了显著的性能提升，设定了新的基准值。这些结果表明，改进方法有效缩小了脉冲神经网络与人工神经网络之间的性能差距，同时保留了脉冲神经网络独有的生物启发建模特性，为实际应用中的神经形态计算提供了更高效、更适应性的解决方案。

尽管 CS-QCFS 目前已成功应用于基于 ReLU 激活函数的网络结构，未来的工作将探索其在更多模型（如 Transformer）和更广泛任务（如目标检测）中的适用性。这一扩展将进一步推动脉冲神经网络在多样化计算场景中的发展与应用。

第四章 基于转换的脉冲 Transformer 训练方法

在脉冲神经网络领域，基于转换的脉冲卷积神经网络训练方法已取得显著进展，如本文第三章提出的 CS-QCFS 方法，通过通道自适应阈值和 Softplus 变换有效解决了卷积层中的特征差异性问题。然而，随着 Transformer 在视觉和自然语言处理任务中的广泛应用，如何将 Transformer 的全局注意力建模能力与脉冲神经元的事件驱动特性结合，成为一个重要的研究方向。Transformer 与 CNN 在特征建模范式上存在本质差异：CNN 依赖局部感受野的卷积核进行层次化特征提取，Transformer 则通过自注意力机制显式建模任意位置间的全局关系。这种架构差异导致现有基于卷积优化的 ANN-SNN 转换策略难以直接应用到 Transformer 中来。因此，亟需构建面向 Transformer 架构特性的脉冲转换新范式，以实现脉冲计算与全局特征建模的有机统一。

4.1 问题分析

在本节中，我们将对 ViT 和 CNN 的结构特点及计算差异进行分析，以明确基于转换的脉冲 Transformer 训练方法需要解决的问题。

4.1.1 Transformer 与 CNN 的差异

如图4-1所示，ViT 和 CNN 在图像建模方法上存在显著差异。ViT 以 Transformer 编码器为基础，通过 Multi-head Attention 模块建模图像块 (Patch) 之间的全局依赖关系。Attention 机制通过计算 Query、Key 和 Value 的加权相关性，显式捕获输入序列任意位置间的长距离依赖。相比之下，CNN 则依赖卷积操作通过局部感受野逐层提取图像局部特征，再通过堆叠多层卷积隐式捕获全局依赖。Attention 的全局特性使 ViT 在处理复杂结构数据时更具优势，而 CNN 则更注重局部特征提取的高效性，适合规则化数据。

此外，ViT 在每个 Transformer Block 中广泛使用多层感知机 (Multi-Layer Per-

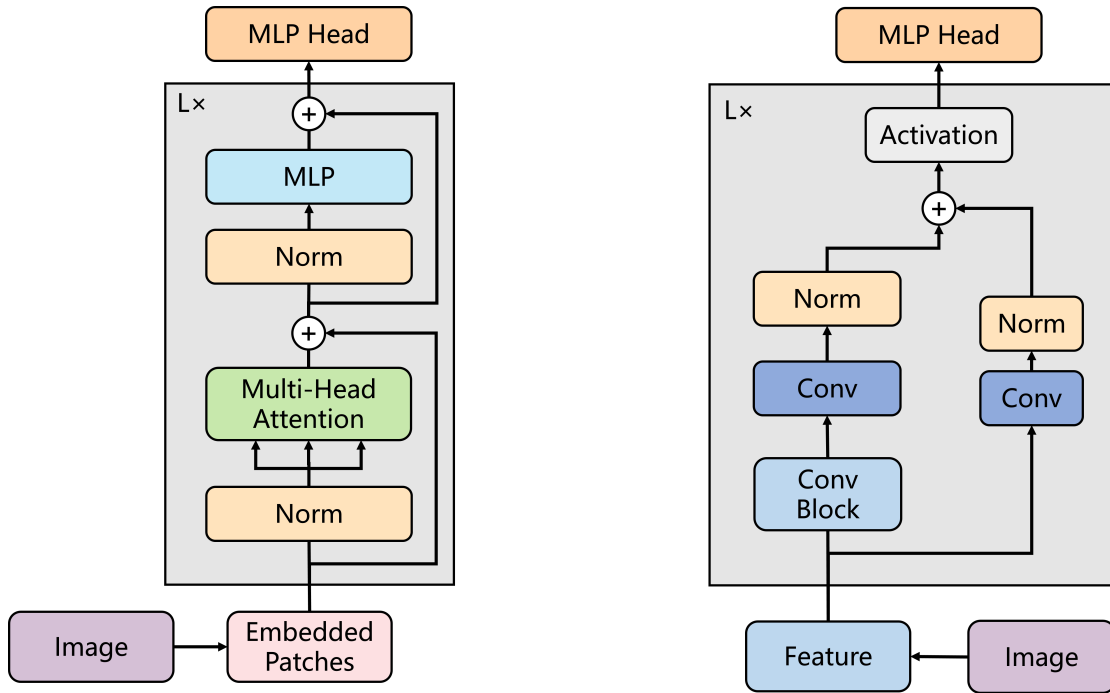


图 4-1 ViT 架构 (左); CNN 架构 (右)

ception, MLP) 模块。MLP 主要用于 Attention 后的特征非线性变换与重构，通常由两层全连接层和激活函数 (如 GELU) 组成，能够有效增强特征表达能力。这使得 ViT 的 MLP 模块贯穿整个网络架构，作为特征提取的重要组成部分。相比之下，CNN 中的 MLP 通常只出现在网络的末端 (即分类头部分)，用于将卷积提取的特征映射到最终的类别标签，在特征提取阶段的作用较小。

综上所述，ViT 强调通过 Attention 和 MLP 进行全局特征建模，适合捕获复杂输入数据的长距离依赖；CNN 则依赖卷积操作高效提取局部特征，适合规则化图像数据。在设计基于转换的脉冲 CNN 训练方法时，我们重点关注了卷积层并进行了相应的优化。同样，在设计基于转换的脉冲 Transformer 训练方法时，需要将重点放在 Attention 和 MLP 的特性上，以更好地适应其架构需求。

4.1.2 QCFS 应用在中头注意力层的限制

多头注意力机制 (Multi-head Attention) 是 Transformer 架构的核心组件，其通过将输入特征分解为多个子空间并分别进行注意力计算，显著提升了模型的特征表达能力。多头机制的设计目标是让不同的头聚焦于输入数据的不同特征模式或关系，从而捕获更加多样化的特征信息。不同头之间的差异主要体现在投影矩阵的独立性、注意力权重的分布特性以及其专注的特征模式上。

首先，多头注意力机制中，每个头均使用独立的查询 (Query)、键 (Key) 和值 (Value) 的线性投影矩阵。这种独立性使得每个头在特征空间中分布各异，能够从不同的角度关注输入数据。例如，有些头可能更关注局部特征的关系，而另一些头则偏向于建模全局依赖关系。此外，多头注意力机制生成的注意力权重矩阵通常具有显著的分布差异。Vaswani 等人^[21]中指出，某些头的注意力权重可能集中于局部区域（即局部注意力），而其他头可能更倾向于捕获远距离的长程关系（即全局注意力）。

在视觉任务中，不同头还会对输入图像块关注不同的特征模式或区域。根据 Dosovitskiy 等人^[12]的研究，有些头可能专注于捕捉图像的边缘信息，有些头提取纹理特征，而另一些头则关注特定的语义模式。这种多样性使得 Vision Transformer 能够更好地建模图像的复杂特征分布。进一步的研究对多头注意力机制的差异性和重要性进行了深入分析。例如，Michel 等人^[66]的研究中指出，并非所有的头在模型性能中都起到了显著作用。一些头具有较高的注意力权重集中度，对特征建模有重要贡献，而另一些头则可能存在冗余。

因此，当将 QCFS 应用到 Transformer 中时则可能面临类似 QCFS 在 CNN 应用中的局限性，即如果对所有头强行使用统一的阈值，可能会忽视这些头之间的多样性需求。统一阈值会导致模型无法适应不同头的激活分布，某些头可能因此受到过度限制，从而无法充分表达其独特的特征建模能力；而其他头则可能因为阈值的设置过于宽松而产生冗余。这种同质化处理可能对模型整体性能产生负面影响，限制脉冲神经网络在 Transformer 框架下的效率和潜在能力。

4.1.3 QCFS 应用在全连接层的限制

全连接层 (Fully Connected Layer, FC Layer) 作为神经网络中最基础的层之一，其特点是对输入的所有特征进行全局线性组合，并将输出传递到下一层。在 ViT 架构中，全连接层不仅存在于分类头中，也存在于每个编码器的多层感知机中，承担着重要的特征映射和非线性变换的任务。

在将 QCFS 激活函数应用于全连接层时，其局限性在 Transformer 框架中尤为突出。首先，全连接层中的不同神经元通常会对输入特征的不同方面进行建模。例如，有些神经元可能对全局信息更为敏感，而另一些则可能更加关注局部特征变化。这种差异性会影响神经元之间的激活分布。然而，当对所有神经元应

用统一的阈值时，无法充分考虑每个神经元对特定特征模式的响应差异。这种同质化处理可能抑制 Transformer 中各个层之间信息的有效传递，进而影响模型的整体性能。

其次，由于 Transformer 中每个编码器和解码器层都包含一个多层感知机，全连接层在网络中的层级结构上占据了大量比重。全连接层的频繁出现进一步放大了其在激活分布多样性上的潜在问题。虽然全连接层不像卷积层或多头注意力机制那样显式地区分通道或子空间的结构差异，但其内部的神经元依然呈现出显著的差异性。在这种情况下，如果对所有层级或所有全连接层使用相同的阈值设置，会限制网络对不同特征的灵活建模能力，从而导致 Transformer 在特征提取和信息传递过程中的能力受限。

因此，在全连接层中使用 QCFS 时，需要针对 Transformer 的结构特点，充分考虑神经元之间的差异性以及全连接层的层级分布特性，以避免统一阈值带来的模型性能损失。

4.2 基于转换的脉冲 Transformer 训练方法

4.2.1 多头级别 QCFS

为了解决统一阈值在多头注意力机制中的局限性，我们的方法从全局统一的阈值设计转向多头级别的 QCFS 激活函数。在该方法中，每个注意力头 (head) 都配备了其独立的可训练阈值 λ_h 。在 Transformer 中，多头注意力机制的核心计算公式见公式(2-12)-公式(2-14)。通过多头机制，输入特征被分解到 H 个独立的子空间，每个头学习不同的特征表示。

针对多头注意力机制的差异性，H-QCFS (Head-wise QCFS) 函数被定义为：

$$a_h = \text{H-QCFS}(z) = \lambda_h \cdot \text{clip} \left(\frac{1}{L} \lfloor \frac{zL}{\lambda_h} + \phi \rfloor, 0, 1 \right), \quad (4-1)$$

其中， λ_h 为注意力头 h 的可训练阈值， z 为输入到该头的特征。对于每个注意力头 h ，其输出激活值 a_h 是基于独立的阈值 λ_h 计算的。这样，每个头可以根据自身处理的输入分布动态调整其响应，从而更好地捕获输入特征中的多样性和层次性关系。

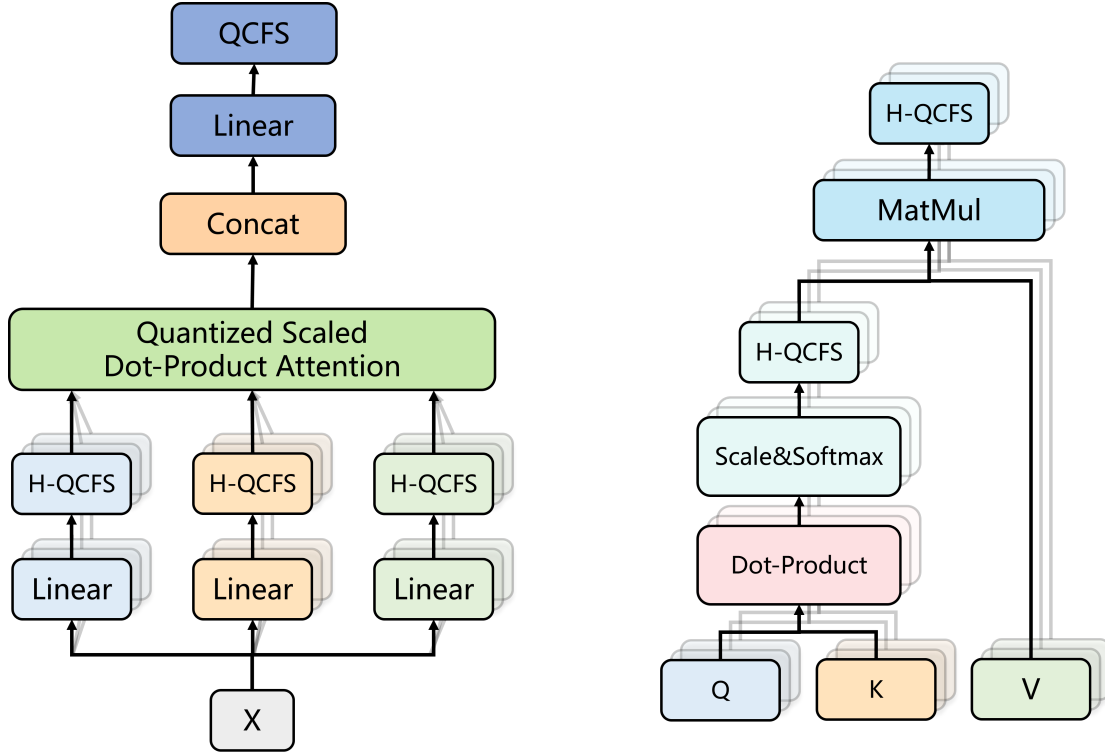


图 4-2 量化多头自注意力层 (左); 量化缩放-点积注意力 (右)

如图4-2所示, 使用了HQCFS的多头注意力可以重写为以下, 首先对 \mathbf{Q} , \mathbf{K} , \mathbf{V} 进行 H-QCFS 量化:

$$\begin{aligned}\mathbf{Q}_{\text{hqdfs}} &= \text{H-QCFS}(\mathbf{Q}), \\ \mathbf{K}_{\text{hqdfs}} &= \text{H-QCFS}(\mathbf{K}), \\ \mathbf{V}_{\text{hqdfs}} &= \text{H-QCFS}(\mathbf{V}),\end{aligned}\quad (4-2)$$

然后根据量化后的 $\mathbf{Q}_{\text{hqdfs}}$, $\mathbf{K}_{\text{hqdfs}}$ 计算注意力矩阵:

$$\text{Attention}_{\text{hqdfs}} = \text{H-QCFS}\left(\text{softmax}\left(\frac{\mathbf{Q}_{\text{hqdfs}}\mathbf{K}_{\text{hqdfs}}^T}{\sqrt{d}}\right)\right),\quad (4-3)$$

再根据量化的 $\text{Attention}_{\text{hqdfs}}$, $\mathbf{V}_{\text{hqdfs}}$ 计算注意力输出:

$$\text{head}_{\text{hqdfs}}^i = \text{H-QCFS}(\text{Attention}_{\text{hqdfs}}(\mathbf{Q}_i, \mathbf{K}_i)\mathbf{V}_{\text{hqdfs}}), \quad i = 1, 2, \dots, h,\quad (4-4)$$

最后将所有头的结果拼接并线性变换：

$$\text{MultiHead}_{\text{hqdfs}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{QCFS}(\text{Concat}(\text{head}_{\text{hqdfs}}^1, \text{head}_{\text{hqdfs}}^2, \dots, \text{head}_{\text{hqdfs}}^h) \mathbf{W}^O). \quad (4-5)$$

引入 H-QCFS 激活函数具有以下优势：

1. **提升多头的特征表达能力。** 通过为每个注意力头独立分配阈值，H-QCFS 能够更精确地建模输入特征的多样性。例如，不同的头可以分别专注于局部注意力、全局关系、边缘特征或语义模式。
2. **增强适应性与动态性。** 由于每个头的输入特征分布可能具有较大的差异性，H-QCFS 提供了更高的灵活性，允许每个头根据自身的特定任务需求调整其激活策略。
3. **改进学习动态。** 独立阈值的设计减少了头与头之间的干扰，避免了统一阈值可能导致的过度约束或宽松，从而提升了注意力机制的有效性和稳定性。

H-QCFS 的引入扩展了脉冲神经网络在多头注意力机制中的应用场景，使其能够更好地适应 Transformer 架构的特点，同时提供更细粒度的特征处理能力。这种方法为构建更强大、更高效的脉冲 Transformer 模型奠定了基础。

4.2.2 神经元级别 QCFS

全连接层与卷积层

全连接层可以看作是卷积层的特殊形式，其特殊性体现在卷积核的大小和权重的组织方式上。从数学上，全连接层的操作完全等价于特定形式的卷积操作。

如图4-3所示，假设输入特征图 \mathbf{X} 的大小为 $2 \times 2 \times 1$ ，全连接层将其展平为一个向量 \mathbf{X}' （维度为 4×1 ），并通过权重矩阵 \mathbf{W} （大小为 4×4 ）与其相乘，计算输出向量 $\mathbf{Y} = [y_1, y_2, y_3, y_4]^T$ 。这一过程等价于将输入特征图 \mathbf{X} 视为单通道输入，对其应用一个步长为 1、无填充的卷积操作。具体而言，权重矩阵 \mathbf{W} 的每一行可以视为一个大小为 2×2 的卷积核（等同于输入特征图的空间维度）。这种卷积核完全覆盖输入特征图，使每个输出值由所有输入值共同作用得到。卷积操作的结果是一个 $1 \times 1 \times 4$ 的输出特征图，正好对应全连接层输出向量的维度。

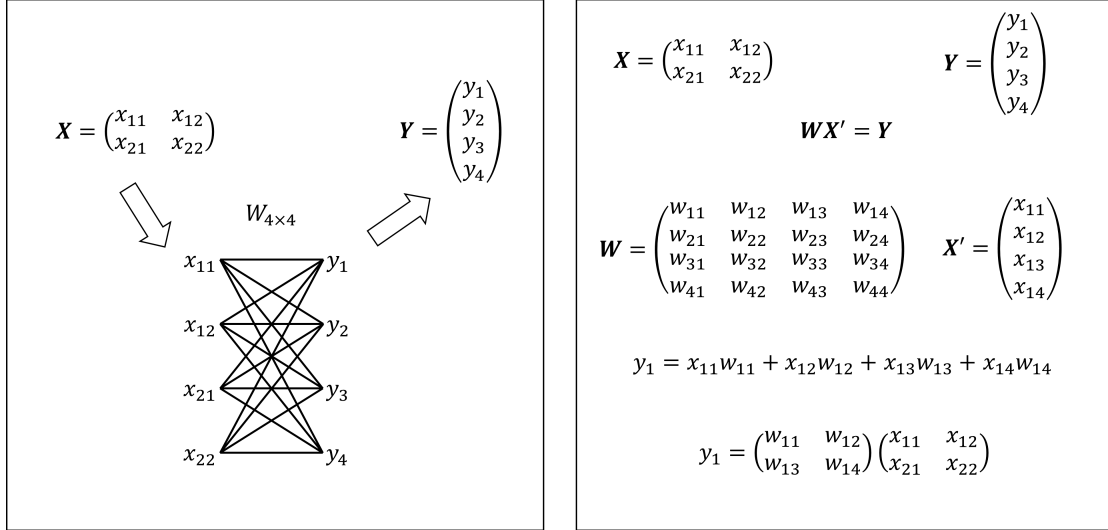


图 4-3 全连接运算与卷积运算

因此，全连接层可以理解为卷积核大小等于输入特征图大小、卷积输出通道数等于全连接层的输出维度的一种特殊卷积层。

N-QCFS

由于全连接层可以理解为一种特殊的卷积层。因此，在设计适用于全连接层的 QCFS 激活函数时，可以借鉴卷积层的通道级 QCFS 方法。我们引入了神经元级别的 QCFS 激活函数（Neuron-wise QCFS，简称 N-QCFS）。在该方法中，全连接层内的每个神经元均配备一个独立的可训练阈值 λ_j 。其激活函数定义如下：

$$a_j = \text{N-QCFS}(z) = \lambda_j \cdot \text{clip} \left(\frac{1}{L} \left[\frac{zL}{\lambda_j} + \phi \right], 0, 1 \right), \quad (4-6)$$

其中， λ_j 为神经元 j 的可训练阈值， z 为输入特征。通过为每个神经元独立分配阈值，N-QCFS 激活函数允许神经元根据其任务需求和输入分布动态调整响应，从而提升全连接层的特征建模能力。

使用 QCFS 和 N-QCFS 的 FFN 如图4-4所示，QCFS 为全连接层的所有神经元共同训练一个阈值，而 N-QCFS 则会对全连接层的每个输出单独训练一个阈值，其可以重写如下：

$$\text{FFN}_{\text{nqcfs}}(\mathbf{X}) = \text{N-QCFS}(\text{N-QCFS}(\mathbf{X}\mathbf{W}_1 + b_1)\mathbf{W}_2 + b_2). \quad (4-7)$$

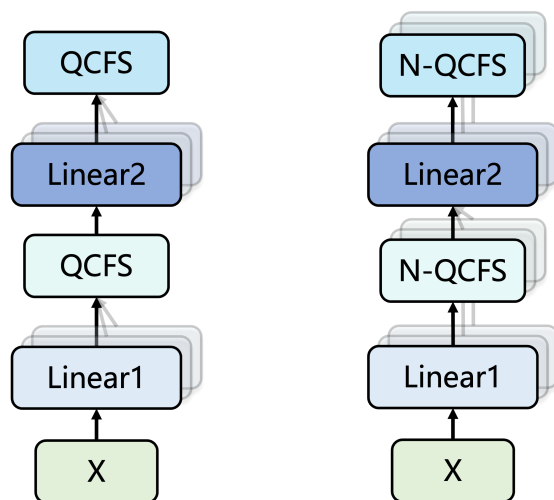


图 4-4 QCFS 量化 FFN (左); N-QCFS 量化 FFN (右)

其中， W_1 ， W_2 分别是第一层和第二层全连接的权重矩阵， b_1 ， b_2 是对应的偏置。

神经元级别 QCFS 的特性与优势如下：

1. **提升神经元的表达能力。**全连接层的每个神经元对输入的所有特征进行加权求和后计算激活值。这些神经元的功能和任务具有显著的独立性。借鉴通道级 QCFS 的思想，为每个神经元设计独立的阈值，可以更灵活地适应输入特征分布，提高每个神经元的表达能力。
2. **适应 Transformer 中全连接层的广泛应用。**Transformer 的编码器和解码器中每层多层感知机模块均包含两个全连接层，而 Transformer 结构又是多层堆叠的。因此，全连接层在 Transformer 中频繁出现，其重要性不容忽视。N-QCFS 的引入能够显著改善 Transformer 各层全连接层的激活响应，提升网络整体性能。
3. **改进信息流动。**由于全连接层在整个网络中起到信息整合和传递的作用，不同神经元的激活动态性对于信息流的连续性和效率至关重要。N-QCFS 提供的细粒度调节能力，有助于减少神经元间的干扰并改善特征传递。

全连接层作为卷积层的特例，其神经元的独立性类似于卷积通道的独立性。因此全连接层的 N-QCFS 和卷积层的 C-QCFS 在机制上类似，均通过为每个计算单元（神经元或通道）引入独立的阈值来增强特征建模能力。神经元 QCFS 的设计可被看作是通道级 QCFS 的进一步扩展，针对全连接层的特殊性进行了优化和调整。这种关联进一步说明，不同结构的层需要适配不同的 QCFS 设计，才能充分挖掘其潜力。神经元 QCFS 为脉冲神经网络的设计提供了一种通用且高

效的激活方案，使其能够在复杂模型中实现更精细的特征建模和高效的计算。

4.2.3 HN-QCFS

在 Transformer 中，多头注意力机制和全连接层共同构成了其核心结构。为提升神经元激活的自适应性与量化性能，我们提出了 HN-QCFS (QCFS with Head-wise and Neuron-wise)。该方法在 Transformer 的多头注意力部分应用多头 QCFS (H-QCFS)，在全连接层部分应用神经元级别 QCFS (N-QCFS)，如图4-5所示。

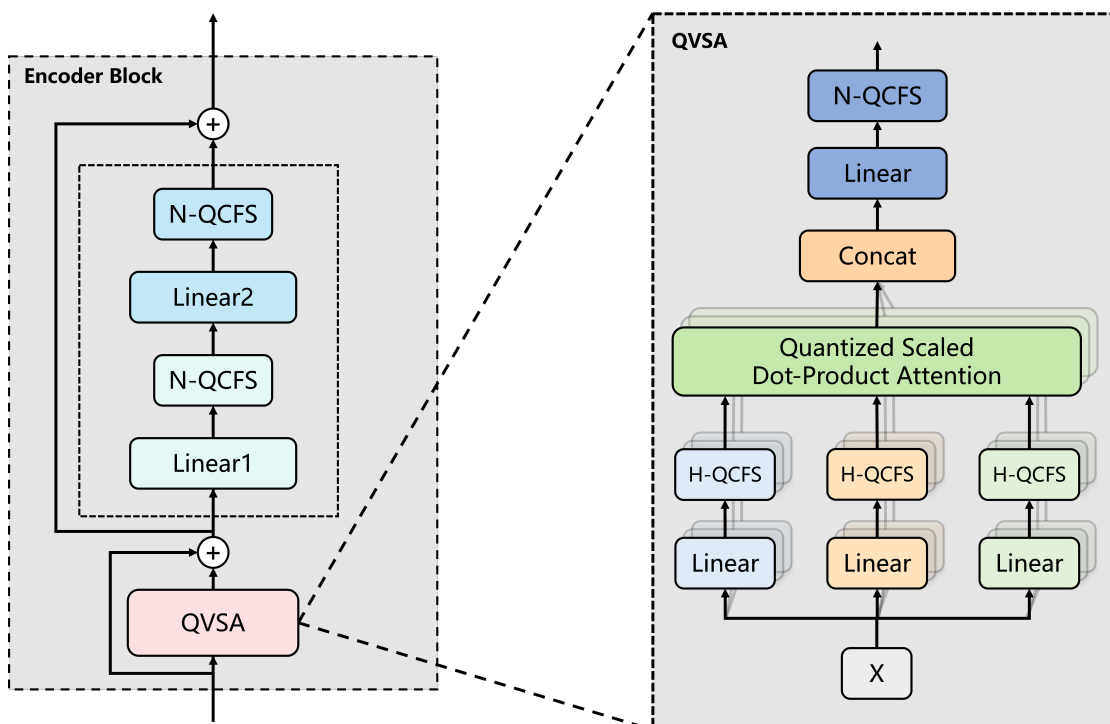


图 4-5 应用 HN-QCFS 的编码器

HN-QCFS 不仅继承了 H-QCFS 和 N-QCFS 各自的优点，还通过其协同作用进一步提升了 Transformer 的性能：在多头注意力机制中，H-QCFS 增强了注意力头的独立性，使模型能够更高效地捕捉不同子空间特征。在全连接层中，N-QCFS 提高了特征变换的细致程度，优化了 Transformer 的学习动态。整体上，HN-QCFS 显著提升了 Transformer 的自适应性与表达能力。

4.3 实验与分析

在本节中，我们通过在图像分类数据集 CIFAR-10 和 CIFAR-100^[58]上的实验，验证了本文方法的有效性和可扩展性。实验中使用的网络架构包括 ViT^[12]以

及 Swin-Transformer^[67]。此外，我们将本文的方法与目前的 ANN-SNN 转换方法进行了对比，包括 MST^[38]。

4.3.1 数据集与模型

数据集：实验主要在 CIFAR-10 和 CIFAR-100 数据集上进行，以验证本文方法的有效性和可扩展性。在预处理阶段，将原始 32×32 分辨率的图像统一调整为 224×224，以匹配模型的输入维度。采用随机裁剪、水平翻转等数据增强技术，增加训练数据的多样性，提高模型的泛化能力。此外，对输入图像进行数据归一化，以确保其具有零均值和单位方差。

模型：在实验中，我们选择了两个基于 Transformer 的网络架构：ViT 和 Swin-Transformer。ViT 是一种开创性的视觉模型，首次将 Transformer 应用于计算机视觉任务。它通过将输入图像分割为固定大小的图像块，并将其展平后映射为嵌入向量，引入标准的 Transformer 编码器进行特征提取。ViT 展现了在大规模数据集上超越卷积神经网络的潜力，为视觉领域提供了一种新的设计范式；相比之下，Swin Transformer 在此基础上进一步优化了计算效率与全局建模能力。通过引入滑动窗口的局部注意力机制，Swin Transformer 在降低计算复杂度的同时保留了跨区域的特征建模能力。此外，其分层架构允许模型高效处理不同分辨率的特征，适用于多种视觉任务。

4.3.2 实验设置

表 4-1 实验参数设置

数据集	模型	优化器	量化步长 L	初始阈值 λ	学习率
CIFAR10	ViT	AdamW	16	3	0.0001
	Swin-T				
CIFAR100	ViT	AdamW	16	3	0.0001
	Swin-T				

训练与优化：在神经网络预训练过程中，采用 AdamW 优化器^[68]对模型进行训练，总共分为两个阶段。第一阶段使用初始学习率 1×10^{-4} ，训练 300 个 epoch；第二阶段为微调阶段，将学习率调整为 1×10^{-3} ，并继续训练 100 个 epoch。整个训练过程结合了 warmup 策略和余弦学习率调度策略（Cosine LR Scheduler）^[62]。其中，在训练的前 10 个 epoch 中，学习率逐渐从 0 增长至基础学

习率 1×10^{-4} ，随后按照余弦函数逐步减小，以实现更平稳的优化过程。此外，为了正则化训练并提升模型的泛化能力，对 CIFAR-10 和 CIFAR-100 两个数据集均施加权重衰减，衰减系数设置为 0.05，从而有效抑制过拟合现象。

HN-QCFS 参数: 在实验中，所有模型的量化步长 L 均设置 16，初始阈值 λ 设置为 3。

4.3.3 CIFAR 数据集对比实验

我们将所提出的方法与 MST 进行了对比，以验证其有效性。此外，我们还利用偏移脉冲 (offset spikes)^[33]，进一步减少转换误差。

表 4-2 CIFAR-10 数据集实验结果

网络架构	方法	ANN*	T=1	T=2	T=4	T=8	T=16	T=32
ViT	MST ^[38]	92.47%	37.09%	66.01%	82.21%	83.91%	87.81%	90.70%
	Ours	92.83%	71.96%	88.21%	89.54%	89.81%	90.99%	91.96%
Swin-T	MST ^[38]	96.11%	87.42%	92.18%	92.22%	92.74%	93.74%	94.7%
	Ours	96.30%	93.56%	93.55%	93.64%	93.97%	94.56%	95.20%

ANN* 是用 QCFS/CS-QCFS 替换 ReLU 后的准确率。

表4-2中展示了这些方法在 CIFAR-10 数据集上不同时间步下的测试准确率。在 CIFAR-10 数据集上的实验结果表明，本文提出的 HN-QCFS 激活函数相较于原始 QCFS 在人工神经网络到脉冲神经网络的转换过程中显著提升了模型性能。在 ViT 架构下，HN-QCFS 在低时间步数（如 T=1）时的准确率达到 71.96%，较 QCFS 提升了 34.87 个百分点；在高时间步数（如 T=32）时，准确率为 91.96%，接近人工神经网络的 92.83%。在 Swin-Transformer 架构下，HN-QCFS 在大多数时间步数上均优于 QCFS，尤其在 T=1 时准确率达到 93.56%，较 QCFS 提升了 6.14 个百分点。综合来看，HN-QCFS 通过对 Transformer 中多头注意力机制和全连接层的每个部分分别训练阈值，使得模型在低时间步数下显著减少转换误差，同时在中高时间步数范围内保持了更高的鲁棒性和精度。

表 4-3 CIFAR-100 数据集实验结果

网络架构	方法	ANN*	T=1	T=2	T=4	T=8	T=16	T=32
ViT	MST ^[38]	74.53%	13.32%	49.72%	56.15%	58.64%	63.62%	68.51%
	Ours	74.29%	46.48%	64.92%	65.75%	67.01%	69.29%	71.19%
Swin-T	MST ^[38]	79.92%	63.91%	71.84%	73.35%	74.58%	76.40%	77.56%
	Ours	80.42%	71.02%	73.98%	74.70%	75.46%	77.32%	78.53%

我们进一步扩展实验，评估本文方法在 CIFAR-100 数据集上的泛化能力，结

果如表4-3所示。无论是在 ViT 还是 Swin-T 架构下，采用 HN-QCFS 的转换方法在各个时间步数上均优于基于 QCFS 的 MST 方法。

在 ViT 架构中，HN-QCFS 在低时间步数 ($T=1$) 时表现尤为突出，准确率达到 46.48%，相比 MST 的 13.32% 提升了 33.16 个百分点。随着时间步数的增加，HN-QCFS 的性能进一步提升。例如，在 $T=4$ 和 $T=8$ 时，准确率分别为 65.75% 和 67.01%，均明显高于 MST 的 56.15% 和 58.64%。在高时间步数 ($T=32$) 时，HN-QCFS 的准确率达到 71.19%，接近人工神经网络基准性能 74.29%，显示了其在高时间步数范围内的稳定性和收敛性能。在 Swin-T 架构中，HN-QCFS 同样展现了全面的性能优势。在 $T=1$ 时，准确率达到 71.02%，相比 MST 的 63.91% 提升了 7.11 个百分点，表明改进的激活函数在初始阶段能够有效减少信息传递过程中的转换误差。在中间时间步数 ($T=4$ 到 $T=16$) 时，HN-QCFS 继续保持小幅度但稳定的优势。例如，在 $T=16$ 时，HN-QCFS 的准确率为 77.32%，相比 MST 的 76.40% 仍有显著提升。值得注意的是，在 $T=32$ 的高时间步数下，HN-QCFS 的准确率为 78.53%，不仅优于 MST 的 77.56%，而且更接近人工神经网络的 80.42%，表明改进方法能够有效保留人工神经网络的表达能力。

通过对两种数据集和架构的实验结果进行比较可以发现，HN-QCFS 显著提升了人工神经网络到脉冲神经网络转换在低时间步数下的响应能力，同时在中高时间步数范围内保持了更高的稳定性和逼近性能。这得益于 HN-QCFS 激活函数在人工神经网络训练过程中对 Transformer 中多头注意力和全连接层的每个部分分别训练阈值，使得转换后的脉冲神经网络能够更精准地表达人工神经网络的特征。特别是在 CIFAR-100 数据集这种更具挑战性的任务中，HN-QCFS 进一步体现了其泛化能力，有效减少了因数据复杂性增加导致的性能损失。

4.3.4 消融实验

表 4-4 多头级别阈值和神经元级别阈值消融实验结果

数据集		ANN	T=1	T=2	T=4	T=8	T=16	T=32
CIFAR-10	w/o neuron	92.47%	66.39%	87.03%	88.83%	89.4%	90.63%	91.68%
	w/o head	92.57%	67.26%	86.03%	88.01%	88.56%	90.08%	91.13%
	Full	92.83%	71.96%	88.21%	89.54%	89.81%	90.99%	91.96%
CIFAR-100	w/o neuron	74.25%	26.81%	62.99%	64.83%	66.19%	68.8%	70.99%
	w/o head	74.44%	42.46%	61.29%	62.49%	63.71%	66.69%	69.37%
	Full	74.29%	46.48%	64.92%	65.75%	67.01%	69.29%	71.19%

为了评估多头级别阈值和神经元级别阈值对脉冲 Transformer 模型性能的影响

响，我们在 CIFAR-10 和 CIFAR-100 数据集上进行了消融实验。本文所提出的 HN-QCFS 激活函数对多头注意力机制中的每个头和全连接层的每个输出神经元分别训练了阈值。消融实验包括以下三种配置：

1. **w/o neuron**: 移除神经元级别阈值，仅使用多头级别阈值，为多头注意力机制中的每个头分别训练阈值。
2. **w/o head**: 移除多头级别阈值，仅使用神经元级别阈值，为全连接层的每个输出神经元分别训练阈值。
3. **Full**: 同时使用多头级别阈值和神经元级别阈值。

表4-4展示了消融实验的结果，列出了 ViT 模型在 CIFAR-10 和 CIFAR-100 数据集使用不同实验设置下的分类准确率。在 CIFAR-10 数据集上，移除神经元级别阈值 (w/o neuron) 或多头级别阈值 (w/o head) 均导致模型性能下降，验证了这两个模块对性能提升的重要性。例如，在 $T=1$ 时，w/o neuron 和 w/o head 的准确率分别为 66.39% 和 67.26%，均明显低于 Full 配置的 71.96%。这种性能差距随着时间步的增加而逐渐缩小，但即使在 $T=32$ 时，Full 配置的准确率 (91.96%) 仍高于 w/o neuron (91.68%) 和 w/o head (91.13%)。这表明神经元级别阈值和多头级别阈值在短时间步下对性能的贡献尤为显著。

在 CIFAR-100 数据集上，性能下降的趋势更加明显。在 $T=1$ 时，移除神经元级别阈值 (w/o neuron) 的准确率仅为 26.81%，而移除多头阈值 (w/o head) 的准确率为 42.46%，均显著低于 Full 配置的 46.48%。即使在 $T=32$ 时，Full 配置的准确率为 71.19%，仍优于 w/o neuron (70.99%) 和 w/o head (69.37%)。这一结果表明，在复杂任务中，多头级别阈值和神经元级别阈值的联合作用对提升模型性能具有更大的影响。

实验结果表明，多头阈值和神经元阈值在提高脉冲 Transformer 模型性能方面具有关键作用。多头级别阈值能够增强多头注意力机制的特征捕获能力，而神经元级别阈值则提升了全连接层的表达能力。两者的协同作用特别在时间步较短时尤为显著。

4.3.5 理论支持实验

为了验证 HN-QCFS 激活函数在 Transformer 模型中训练得到的阈值的合理性，我们设计了理论支持实验。实验旨在探讨 HN-QCFS 激活函数是否能够有效捕获多头注意力机制中不同头之间以及全连接层中不同神经元之间的差异性，并分析训练过程中产生的阈值分布特征。为定量评估这种差异性，我们引入以下指标：

多头注意力机制的差异性指标。对于某个多头注意力层，其内各头的阈值最大值和最小值之间的比例差定义为 $Diff_{head}$ ，其计算公式如下：

$$Diff_{head} = \frac{MaxThre_{head} - MinThre_{head}}{MinThre_{head}} \quad (4-8)$$

其中， $MaxThre_{head}$ 和 $MinThre_{head}$ 分别表示该多头注意力层内最高和最低的阈值。为对模型整体的多头注意力机制差异性进行衡量，定义以下两个全局指标： $MaxDiff_{head}$ 表示所有多头注意力层中 $Diff_{head}$ 的最大值， $AvgDiff_{head}$ 表示所有多头注意力层中 $Diff_{head}$ 的平均值。

全连接层神经元的差异性指标。类似地，对于全连接层中的每个输出神经元，其阈值差异性用 $Diff_{neuron}$ 表示。具体计算公式为：

$$Diff_{neuron} = \frac{MaxThre_{neuron} - MinThre_{neuron}}{MinThre_{neuron}} \quad (4-9)$$

其中， $MaxThre_{neuron}$ 和 $MinThre_{neuron}$ 分别表示全连接层中神经元阈值的最大值和最小值。全局指标包括： $MaxDiff_{neuron}$ 表示所有全连接层中 $Diff_{neuron}$ 的最大值， $AvgDiff_{neuron}$ 表示所有全连接层中 $Diff_{neuron}$ 的平均值。

在实验中，我们分别对 CIFAR-10 和 CIFAR-100 数据集上的 ViT 和 Swin-T 模型计算上述指标，以全面分析 HN-QCFS 激活函数在不同任务中的表现。实验结果如表 4-5 所示。

从多头注意力机制的角度来看， $MaxDiff_{head}$ 和 $AvgDiff_{head}$ 两个指标展现了 HN-QCFS 激活函数在捕获不同头之间差异性方面的显著效果。以 Swin-T 模型为例，在 CIFAR-10 数据集上， $MaxDiff_{head}$ 为 57.9%，而在 CIFAR-100 数据集上提升至 64.1%，同时 $AvgDiff_{head}$ 也从 18.7% 提升到 21.4%。这表明，在类别数量

表 4-5 多头级别阈值和神经元级别阈值验证实验结果

激活函数	数据集	网络架构	$MaxDiff_{head}$	$AvgDiff_{head}$	$MaxDiff_{neuron}$	$AvgDiff_{neuron}$
HN-QCFS	CIFAR-10	Swin-T	57.9%	18.7%	176%	40.2%
		ViT	35.4%	9.8%	70.5%	25.2%
	CIFAR-100	Swin-T	64.1%	21.4%	354.8%	46.3%
		ViT	34.4%	9.9%	93.1%	33.47%

较多且任务复杂度较高的 CIFAR-100 数据集中，注意力头之间的阈值分布差异性更加显著。类似地，ViT 模型在两个数据集上的 $MaxDiff_{head}$ 分别为 35.4% 和 34.4%，虽整体低于 Swin-T 模型，但仍然能够体现出一定程度的差异性。这表明 HN-QCFS 激活函数能够在训练过程中自适应地为每个多头注意力机制中的不同头分配合理的阈值，从而更好地满足各头对不同特征的敏感性需求。

在全连接层的分析中， $MaxDiff_{neuron}$ 和 $AvgDiff_{neuron}$ 指标进一步验证了 HN-QCFS 激活函数在捕获不同神经元之间差异性方面的能力。特别是在 CIFAR-100 数据集上，Swin-T 模型的 $MaxDiff_{neuron}$ 达到 354.8%，显著高于 CIFAR-10 数据集的 176%，而 $AvgDiff_{neuron}$ 也从 40.2% 上升至 46.3%。这一差异性表明，在复杂任务中，不同神经元对特定特征的响应需求更加多样化，HN-QCFS 激活函数通过训练优化为每个神经元分配合理的阈值，从而提升了模型对复杂特征的适应能力。对于 ViT 模型，在 CIFAR-100 数据集上， $MaxDiff_{neuron}$ 为 93.1%，而在 CIFAR-10 数据集上则为 70.5%， $AvgDiff_{neuron}$ 则分别为 33.47% 和 25.2%。这种差异性表明，尽管 ViT 模型的全连接层的阈值差异性较 Swin-T 模型小，但在面对更加复杂的任务时，ViT 模型中的神经元阈值分布也表现出了随任务复杂度变化的多样性。这进一步验证了 HN-QCFS 激活函数能够为每个神经元分配合适的阈值，从而帮助模型更好地适应复杂特征的处理需求。

综上所述，实验结果充分表明，HN-QCFS 激活函数在捕获多头注意力机制中不同头之间的差异性以及全连接层中不同神经元之间的差异性方面表现出色。尤其在复杂任务（如 CIFAR-100）中，模型中的阈值分布差异性更为明显，验证了 HN-QCFS 激活函数在提升模型自适应性和任务复杂性处理能力方面的重要作用。这为基于转换的脉冲 Transformer 训练方法提供了理论依据和实践支持。

4.4 本章小结

在本章中，我们针对 Transformer 模型中的阈值训练问题，提出了基于转换的脉冲 Transformer 训练方法。通过对 Transformer 中关键组件的深入分析，我们发现多头注意力机制和全连接层中存在显著的通道和神经元差异性，传统的 ANN-SNN 转换方法难以有效应对这些差异。因此，提出了两项重要的创新。

首先，针对多头注意力机制中的差异性问题的，我们提出了多头 QCFS 方法。通过为每个注意力头独立训练阈值，确保了模型能够更好地捕捉到每个头的独特信息，从而增强了多头注意力层在脉冲神经网络中的表达能力。其次，为了应对 Transformer 中全连接层的神经元差异性，我们引入了神经元级别的 QCFS 方法。通过为每个输出神经元训练单独的阈值，进一步提升了模型的适应性和性能。

在此基础上，我们提出了 HN-QCFS，该方法结合了多头 QCFS 和神经元级别 QCFS 的优点，为 Transformer 中的每个多头注意力头和全连接层的输出神经元分别训练阈值，从而在人工神经网络到脉冲神经网络的转换过程中更好地适应了 Transformer 的结构特点。这一创新不仅使得脉冲神经网络的性能得到显著提升，还保持了 Transformer 的原始优势，增强了其在脉冲神经网络中的应用潜力。

通过在 CIFAR 数据集上的实验验证，我们的改进方法在多个任务中取得了优异的成绩。实验结果表明，HN-QCFS 能够有效缩小脉冲神经网络与人工神经网络之间的性能差距。此外，消融实验和理论验证实验进一步表明了 HN-QCFS 方法的有效性与优势。

第五章 基于转换的脉冲神经网络训练系统

5.1 系统研发背景

随着脉冲神经网络 (SNN) 研究的不断深入, 其在计算效率和能效方面的潜力逐渐受到学术界和工业界的关注。SNN 作为一种模拟生物神经元活动的计算模型, 其计算方式基于脉冲事件, 具有高效的信息传递机制, 尤其适合于处理时序数据和低功耗计算需求。尽管 SNN 具备显著优势, 但由于其训练和应用的复杂性, 相比于传统的 ANN, SNN 的广泛应用仍受到一定限制。

近年来, ANN-SNN 的转换方法被提出, 并获得了广泛的关注。该方法通过将 ANN 的训练成果转化为 SNN 的可用形式, 从而实现 ANN 和 SNN 之间的知识迁移。尽管已有一些基于转换的脉冲神经网络训练方法取得了初步进展, 但现有的 ANN-SNN 转换方法通常在神经元阈值的设定、训练过程中的稳定性以及模型的实际性能等方面存在一定的局限性。此外, 也如缺乏高效的训练框架和操作界面, 使得非脉冲神经网络领域的专业人员在实际应用中面临一定的挑战。

为了解决这一问题, 本文提出了一种基于转换的脉冲神经网络训练系统。该系统结合了第三章和第四章中提出的基于转换的脉冲 CNN 训练方法和基于转换的脉冲 Transformer 训练方法, 旨在为非脉冲神经网络领域的专业人员提供一种便捷的训练平台。通过提供友好的用户界面 (UI) 和自动化的训练流程, 该系统能够有效降低脉冲神经网络模型训练的门槛, 使得 ANN-SNN 的转换过程更加直观和易于操作。

此外, 该系统将支持对多种网络架构的训练, 涵盖卷积神经网络 (CNN) 和 Transformer 模型, 并根据不同的需求, 提供针对性优化。通过集成第三章和第四章的创新方法, 该系统不仅能够提高训练效率, 还能够确保转换后的 SNN 模型能够更好地保持原有 ANN 模型的性能。

5.2 需求分析

在研发基于转换的脉冲神经网络训练系统时，需求分析是确保系统功能设计合理和满足用户实际需求的重要步骤。该系统需要在多个方面提供完整支持，包括数据隐私保护、用户界面友好性、模型训练与测试的灵活性以及系统性能优化等。以下是系统的主要需求：

1. 数据隐私保护。在深度学习领域，尤其是在涉及个人数据和敏感信息的应用中，保护用户数据的隐私至关重要。为了确保数据的安全性和隐私性，系统必须采取严格的数据保护措施，避免任何未经授权的数据泄露或滥用。

2. 用户友好界面。用户友好的界面是降低使用门槛的关键。系统应设计直观且简洁的操作界面，帮助用户无障碍完成模型训练流程。模块化的界面布局可以将功能划分为预训练、训练和测试等步骤，用户可按需逐步操作。

3. 预训练模块。系统应该包含预训练模块，允许用户上传已有的模型或从头训练新模型，并支持自定义参数设置，包括训练轮次、学习率和优化器选择等。

4. Quantized ANN 训练模块。量化 ANN 训练模块则实现了 ANN 到量化 ANN 的转换，该模块融合了本文第三章和第四章提出的转换算法，通过替换 ReLU 激活函数为 QCFS 激活函数并训练相应参数，提升 ANN 的可转换性。系统需支持不同类型的 QCFS 激活函数选择，包括针对 CNN 模型的 CQCFS、SQCFs 以及 CSQCFS，及针对 Transformer 模型的 HQCFs、NQCFS 和 HN-QCFS。

5. 测试模块。系统还需提供全面的测试功能。测试模块应允许用户评估 ANN 和 SNN 的性能，并支持调整时间步长以测试 SNN 在不同设置下的表现。通过对测试结果的可视化展示和详细对比，用户能够更深入地了解两种模型的差异，为后续优化提供依据。

6. 性能优化与扩展性。为保证系统在复杂任务和大规模数据下的高效运行，性能优化与扩展性设计尤为重要。系统需支持分布式训练和硬件加速技术，如多 GPU 并行计算或 TPU 加速，以提高训练效率。同时，系统应具备模块化设计，便于未来功能的扩展和升级，例如集成新的激活函数或支持更多神经网络结构。这种开放性设计能够确保系统在技术发展的同时保持适应性。

本系统的需求分析全面覆盖了从数据隐私保护到训练和测试流程的各个环

节，并注重系统的易用性和性能优化。通过满足这些需求，本系统将为用户提供一个高效、便捷且安全的脉冲神经网络训练平台，尤其适用于 ANN-SNN 转换的研究和应用。

5.3 系统设计

本系统采用前后端分离的架构，以提高开发效率和系统的可维护性，并实现良好的用户体验和性能优化。前端负责用户界面的呈现和交互，后端负责数据处理、模型训练服务和系统逻辑的实现，整体架构如图5-1所示。

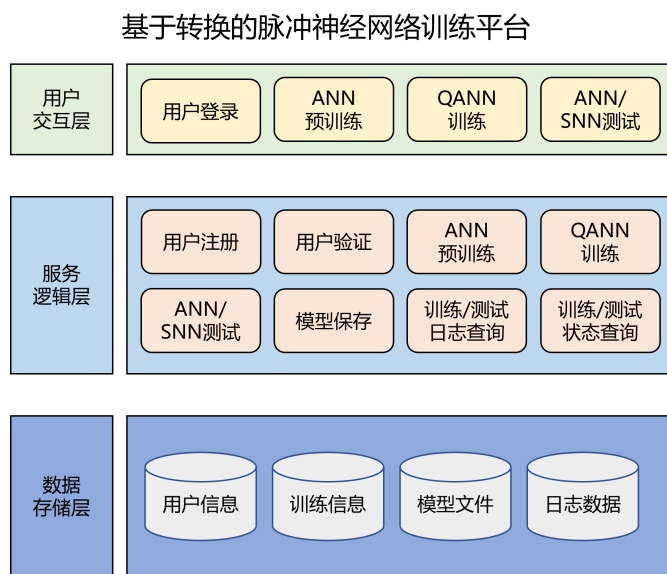


图 5-1 平台整体架构

系统主要分为三层：用户交互层、服务逻辑层和数据存储层。各层相互协作，分别承担不同的职责，以实现基于转换的脉冲神经网络训练系统的核心功能。以下将结合图中架构对系统功能进行详细说明。

用户交互层是系统的前端部分，直接面向用户，提供友好的界面和操作功能。具体包括以下功能模块：用户登录模块实现身份验证；ANN 预训练模块支持用户上传已训练模型或从头训练模型，允许设置训练参数如训练轮次、学习率等；QANN 训练模块基于 ANN 模型进行量化训练，支持选择 QCFS 激活函数类型（如 CNN 的 C/S/CS 或 Transformer 的 H/N/HN）并设置参数；ANN/SNN 测试模块支持测试 ANN 与 SNN 性能，测试 SNN 时用户可选择不同的时间步长来进行测试。

服务逻辑层是系统的后端部分，负责处理用户请求与模型训练任务。功能模块包括用户注册与验证、ANN 预训练、QANN 训练、模型测试、模型保存与管理，以及日志与状态查询。其中，用户注册与验证模块支持新用户注册并通过身份认证机制保护用户数据安全；ANN 预训练模块用于处理用户上传模型或从头训练模型的请求，通过调用相应算法完成模型训练；QANN 训练模块基于用户指定的 QCFS 激活函数和训练参数，对 ANN 模型进行量化训练；ANN/SNN 测试模块提供对模型的测试功能，支持计算性能指标并返回结果供前端展示；模型保存模块统一管理用户上传与生成的模型；日志与状态查询模块记录模型训练状态，支持用户随时查询训练和测试过程的详细日志和任务状态。

数据存储层基于 MongoDB 实现，存储并管理系统运行过程中产生的各种数据，包括用户信息、训练信息、模型文件和日志数据。其中，用户信息用于存储用户的基本信息（如用户名、密码和权限等）；训练信息记录每次模型训练的详细信息、执行状态及中间数据，为用户提供训练查询功能；模型文件存储用户上传的模型、训练后的模型的文件，方便用户随时调用和下载；日志数据则保存模型训练日志，方便用户排查问题。

5.4 系统实现

5.4.1 系统开发环境

本系统基于前后端分离架构实现，前端采用 Vue.js 和 Element UI 技术栈，后端采用 Node.js、Express 框架以及 MongoDB 数据库。具体地，前端采用 Vue 2 作为主框架，结合 Vuex、Vue-Router 和 Element-UI 实现组件化开发和状态管理，同时使用 Webpack 进行模块打包和优化，支持现代化的 ES6/7 语法以及 Less 样式预处理。后端采用 Node.js 与 Express 框架构建，确保服务的轻量化和高效性。通过 RESTful API 实现前后端的数据交互，同时基于 Mongoose 提供对 MongoDB 数据库的操作封装。数据存储层采用 MongoDB，结合 Mongoose 进行数据建模与操作。前后端通过 RESTful API 进行数据交互，接口采用统一规范设计，便于功能扩展与维护。前端发起 HTTP 请求，后端处理逻辑后返回 JSON 格式的数据。

5.4.2 界面展示和使用流程

本节内容旨在展示基于转换的脉冲神经网络训练系统中若干关键功能的操作界面，用户可通过浏览器进行访问和登录以体验这些功能。

登录与首页



图 5-2 登录页面



图 5-3 首页

登录页面和首页分别如图5-2和图5-3所示。注册过的用户可凭账号密码登录，未登录过的新用户将自动注册账号。登录系统后将进入首页，首页对本系统的主要模块进行了介绍，主要包含 ANN 预训练模块、QANN 训练模块和测试模块。此外，首页也展示了基于转换脉冲神经网络训练方法的流程图。

5.4.3 ANN 预训练模块

ANN 预训练页面如图5-4所示，用户可以自行选择网络架构、数据集、优化器、学习率、随机数种子等参数。其中网络架构、数据集和优化器为必选，其他参数不填写时将使用默认值。网络目前支持 ResNet、VGG、ViT、Swin-Transformer 系列的模型，数据集支持 CIFAR10、CIFAR100、Tiny-ImageNet。参数填写完毕后即可点击最下方的开始训练按钮发起训练。

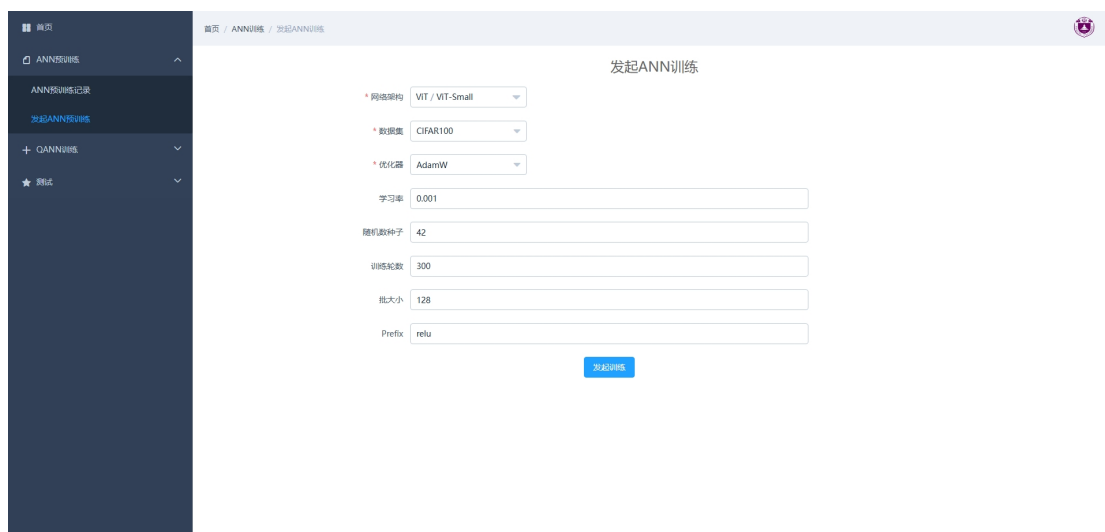


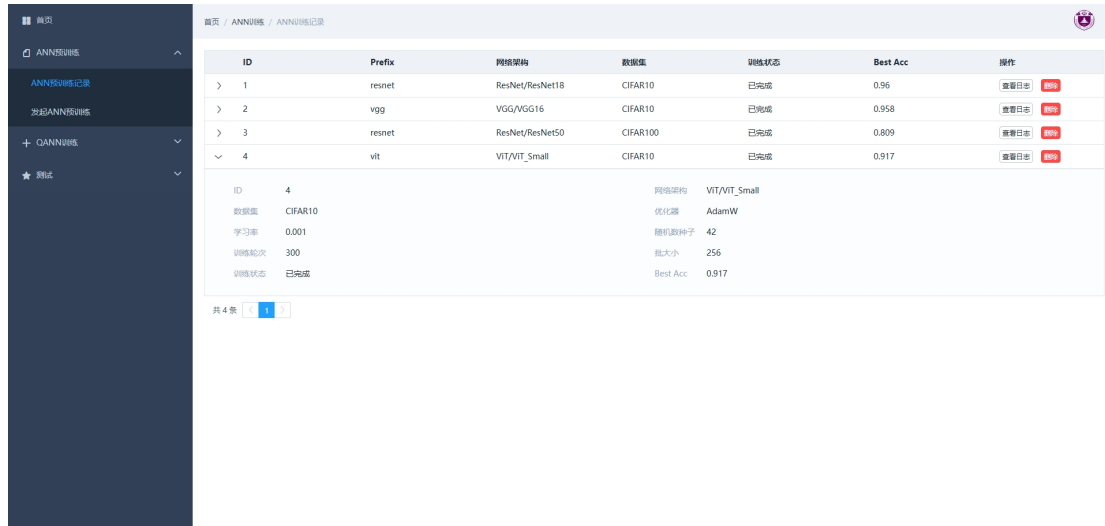
图 5-4 发起 ANN 训练

ANN 预训练记录页面如图5-5所示，用户可以在此页面查看之前发起的 ANN 预训练记录。每条记录展示了一些基本信息，如训练 Prefix、网络架构、数据集、训练状态以及 Best Acc。点击每条记录最左边的箭头可以对训练记录进行展开，可以看到更详细的训练信息。用户也可以点击每条记录最右边的删除按钮对训练记录进行删除或者停止训练中的任务并删除。此外，用户也可以点击查看日志按钮查看详细的训练日志。

训练日志页面如图5-6所示，此页面展示了训练任务的详细训练日志，用户可以在此查看更详细的信息，如训练时间、训练轮次、每个 epoch 的训练损失、Acc1、Acc5、显存占用、测试 Acc1、测试 Acc5 等信息。

5.4.4 QANN 训练模块

QANN 训练界面如图5-7所示，相比 ANN 预训练，QANN 训练将 ReLu 激活函数替换为了 QCFS 激活函数，因此，用户在训练时除了需要设置网络架构、



ID	Prefix	网络架构	数据集	训练状态	Best Acc	操作
> 1	resnet	ResNet/ResNet18	CIFAR10	已完成	0.96	查看详情
> 2	vgg	VGG/VGG16	CIFAR10	已完成	0.958	查看详情
> 3	resnet	ResNet/ResNet50	CIFAR100	已完成	0.809	查看详情
> 4	vit	VIT/VIT_Small	CIFAR10	已完成	0.917	查看详情

ID	网络架构	数据集	训练状态	Best Acc
4	VIT/VIT_Small	CIFAR10	已完成	0.917

共 4 条

图 5-5 ANN 预训练记录



ANN训练日志

```

[2025-01-06 11:55:38 vit] (gain_unm_pp 309): INFO Train: [296/200] [10/48] eta 0:10:27 lr 0.000100 wd 0.0500 time 0.4501 (0.7216) loss 1.7601 (1.1596) Acc@1 91.609 (87.607) Acc@5 100.000
[2025-01-06 11:56:03 vit] (gain_unm_pp 309): INFO Train: [296/200] [10/48] eta 0:10:19 lr 0.000100 wd 0.0500 time 0.4527 (0.8298) loss 1.3200 (1.0804) Acc@1 91.902 (87.529) Acc@5 99.414
[2025-01-06 11:56:37 vit] (gain_unm_pp 309): INFO Train: [296/200] [10/48] eta 0:10:09 lr 0.000100 wd 0.0500 time 0.4597 (0.5535) loss 1.2338 (1.0927) Acc@1 84.570 (87.374) Acc@5 99.414
[2025-01-06 11:56:15 vit] (gain_unm_pp 309): INFO Train: [296/200] [160/63] eta 0:10:04 lr 0.000100 wd 0.0500 time 0.4593 (0.5304) loss 1.4785 (1.1345) Acc@1 92.244 (86.507) Acc@5 99.121
[2025-01-06 11:56:15 vit] (gain_unm_pp 309): INFO EPOCH 296 training takes 0:00:25
[2025-01-06 11:56:15 vit] (gain_unm_pp 311): INFO * Acc@1 95.302 Acc@5 99.499
[2025-01-06 11:56:15 vit] (gain_unm_pp 302): INFO Test: [0/10] Time 2.331 (2.331) Loss 0.4246 (0.4246) Acc@1 92.188 (92.188) Acc@5 99.707 (99.707) Mem 15400MB
[2025-01-06 11:56:19 vit] (gain_unm_pp 369): INFO * Acc@1 91.570 Acc@5 99.550
[2025-01-06 11:56:19 vit] (gain_unm_pp 399): INFO accuracy of the network on the 10000 test images: 91.4%
[2025-01-06 11:56:19 vit] (gain_unm_pp 201): INFO Res accuracy: 91.71%
[2025-01-06 11:56:23 vit] (gain_unm_pp 309): INFO Train: [297/200] [10/48] eta 0:10:23 lr 0.000100 wd 0.0500 time 0.1582 (3.1582) loss 1.0894 (1.0894) Acc@1 98.438 (96.438) Acc@5 100.000
[2025-01-06 11:56:27 vit] (gain_unm_pp 309): INFO Train: [297/200] [10/48] eta 0:10:26 lr 0.000100 wd 0.0500 time 0.4604 (0.7099) loss 1.0796 (1.0652) Acc@1 83.789 (86.874) Acc@5 99.512
[2025-01-06 11:56:32 vit] (gain_unm_pp 309): INFO Train: [297/200] [10/48] eta 0:10:26 lr 0.000100 wd 0.0500 time 0.4608 (0.5912) loss 0.9782 (1.0462) Acc@1 92.571 (90.578) Acc@5 99.962
[2025-01-06 11:56:36 vit] (gain_unm_pp 309): INFO Train: [297/200] [10/48] eta 0:10:09 lr 0.000100 wd 0.0500 time 0.4609 (0.5465) loss 1.4588 (1.0200) Acc@1 78.927 (86.140) Acc@5 98.730
[2025-01-06 11:56:41 vit] (gain_unm_pp 309): INFO Train: [297/200] [60/48] eta 0:10:04 lr 0.000100 wd 0.0500 time 0.4583 (0.5273) loss 1.2114 (1.0374) Acc@1 88.672 (89.370) Acc@5 99.805
[2025-01-06 11:56:45 vit] (gain_unm_pp 309): INFO EPOCH 297 training takes 0:00:25
[2025-01-06 11:56:45 vit] (gain_unm_pp 321): INFO * Acc@1 88.767 Acc@5 99.538
[2025-01-06 11:56:47 vit] (gain_unm_pp 362): INFO Test: [0/10] Time 2.159 (2.159) Loss 0.4230 (0.4230) Acc@1 91.602 (91.602) Acc@5 99.707 (99.707) Mem 15400MB
[2025-01-06 11:56:49 vit] (gain_unm_pp 369): INFO * Acc@1 91.410 Acc@5 99.380
[2025-01-06 11:56:49 vit] (gain_unm_pp 399): INFO accuracy of the network on the 10000 test images: 91.4%
[2025-01-06 11:56:49 vit] (gain_unm_pp 201): INFO Res accuracy: 91.71%
[2025-01-06 11:56:53 vit] (gain_unm_pp 309): INFO Train: [298/200] [10/48] eta 0:10:23 lr 0.000100 wd 0.0500 time 0.2811 (2.2811) loss 1.0188 (1.0188) Acc@1 99.512 (99.512) Acc@5 100.000
[2025-01-06 11:56:56 vit] (gain_unm_pp 309): INFO Train: [298/200] [10/48] eta 0:10:09 lr 0.000100 wd 0.0500 time 0.4589 (0.6598) loss 0.7519 (1.0368) Acc@1 91.676 (86.676) Acc@5 99.902
[2025-01-06 11:57:01 vit] (gain_unm_pp 309): INFO Train: [298/200] [10/48] eta 0:10:18 lr 0.000100 wd 0.0500 time 0.4605 (0.5628) loss 0.5735 (1.0583) Acc@1 72.559 (85.053) Acc@5 98.242
[2025-01-06 11:57:05 vit] (gain_unm_pp 309): INFO Train: [298/200] [10/48] eta 0:10:09 lr 0.000100 wd 0.0500 time 0.4625 (0.5488) loss 1.1878 (1.0553) Acc@1 90.438 (86.581) Acc@5 99.326
[2025-01-06 11:57:10 vit] (gain_unm_pp 309): INFO Train: [298/200] [140/48] eta 0:10:04 lr 0.000100 wd 0.0500 time 0.4587 (0.5234) loss 1.2648 (1.0475) Acc@1 79.297 (88.589) Acc@5 99.121
[2025-01-06 11:57:14 vit] (gain_unm_pp 320): INFO EPOCH 298 training takes 0:00:24
[2025-01-06 11:57:14 vit] (gain_unm_pp 321): INFO * Acc@1 88.064 Acc@5 99.491
[2025-01-06 11:57:16 vit] (gain_unm_pp 362): INFO Test: [0/10] Time 2.103 (2.103) Loss 0.4104 (0.4104) Acc@1 91.895 (91.895) Acc@5 99.609 (99.609) Mem 15400MB
[2025-01-06 11:57:17 vit] (gain_unm_pp 369): INFO * Acc@1 91.569 Acc@5 99.444
[2025-01-06 11:57:17 vit] (gain_unm_pp 399): INFO accuracy of the network on the 10000 test images: 91.4%
[2025-01-06 11:57:21 vit] (gain_unm_pp 201): INFO Res accuracy: 91.71%
[2025-01-06 11:57:21 vit] (gain_unm_pp 309): INFO Train: [299/200] [10/48] eta 0:10:24 lr 0.000100 wd 0.0500 time 0.4177 (3.4177) loss 1.1219 (1.1219) Acc@1 97.559 (97.559) Acc@5 100.000
[2025-01-06 11:57:26 vit] (gain_unm_pp 309): INFO Train: [299/200] [10/48] eta 0:10:27 lr 0.000100 wd 0.0500 time 0.4609 (0.7290) loss 1.3644 (1.0521) Acc@1 91.188 (85.322) Acc@5 99.707
[2025-01-06 11:57:28 vit] (gain_unm_pp 309): INFO Train: [299/200] [10/48] eta 0:10:19 lr 0.000100 wd 0.0500 time 0.4609 (0.6033) loss 0.9288 (1.0453) Acc@1 88.089 (85.723) Acc@5 99.414
[2025-01-06 11:57:33 vit] (gain_unm_pp 309): INFO Train: [299/200] [10/48] eta 0:10:10 lr 0.000100 wd 0.0500 time 0.4606 (0.5588) loss 1.2234 (1.1021) Acc@1 74.805 (88.379) Acc@5 98.730
[2025-01-06 11:57:37 vit] (gain_unm_pp 309): INFO Train: [299/200] [160/63] eta 0:10:04 lr 0.000100 wd 0.0500 time 0.4606 (0.5324) loss 1.3744 (1.0931) Acc@1 90.489 (88.113) Acc@5 99.414
[2025-01-06 11:57:43 vit] (gain_unm_pp 311): INFO * Acc@1 91.813 Acc@5 99.502
[2025-01-06 11:57:45 vit] (gain_unm_pp 302): INFO Test: [0/10] Time 2.048 (2.048) Loss 0.4141 (0.4141) Acc@1 91.895 (91.895) Acc@5 99.609 (99.609) Mem 15400MB
[2025-01-06 11:57:47 vit] (gain_unm_pp 369): INFO * Acc@1 91.600 Acc@5 99.520
[2025-01-06 11:57:47 vit] (gain_unm_pp 399): INFO accuracy of the network on the 10000 test images: 91.7%
[2025-01-06 11:57:47 vit] (gain_unm_pp 201): INFO Res accuracy: 91.71%
[2025-01-06 11:57:47 vit] (gain_unm_pp 203): INFO Training time 2:50:39

```

图 5-6 ANN 训练日志

数据集、优化器、学习率、随机数种子等常规参数外，也需要配置一些 QCFs 激活函数相关的参数，如 QCFs 激活函数的类型，初始阈值、量化步长。对于 CNN 模型，QCFs 激活函数可以选择 QCFs、CQCFs、SQCFs 和 CSQCFs；对于 Transformer，QCFs 激活函数可以选择 QCFs、HQCFs、NQCFS、HN-QCFs。此外，还可以选择是否使用预训练模型，可以选择使用 ANN 预训练阶段的模型作为 QANN 的预训练模型，也可以选择从头开始训练 QANN 模型。参数设置完毕后，点击最下方的开始训练按钮即可进行 QANN 训练。

类似地，用户可以查询 QANN 训练记录，如图 5-8 所示。每条记录展示了 Prefix、网络架构、数据集、QCFs 类型、训练状态等。点击每条记录最左边的箭头可以查看更详细的训练信息。用户同样可以终止或删除 QANN 训练任务，以

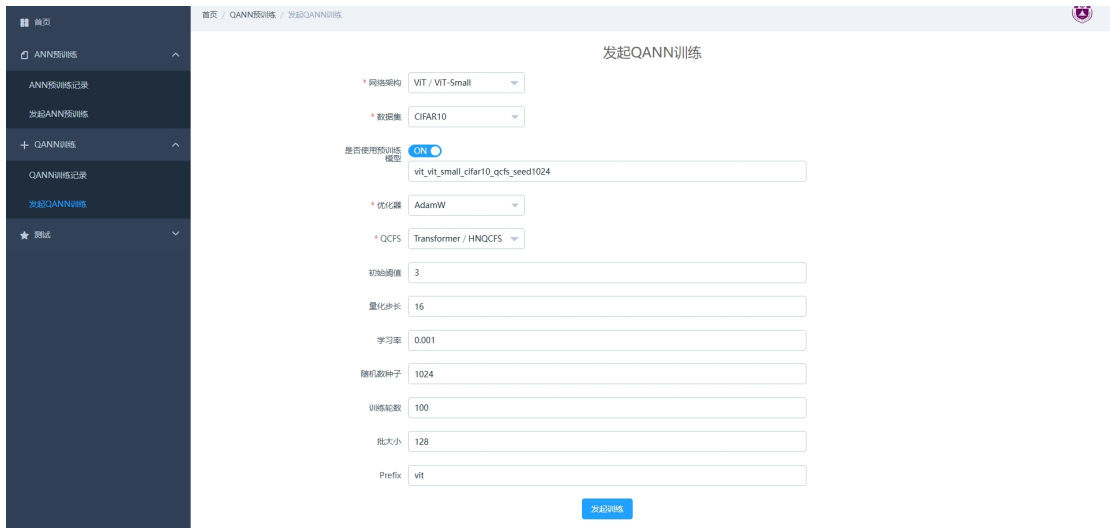


图 5-7 发起 QANN 训练

及查看 QANN 训练日志。

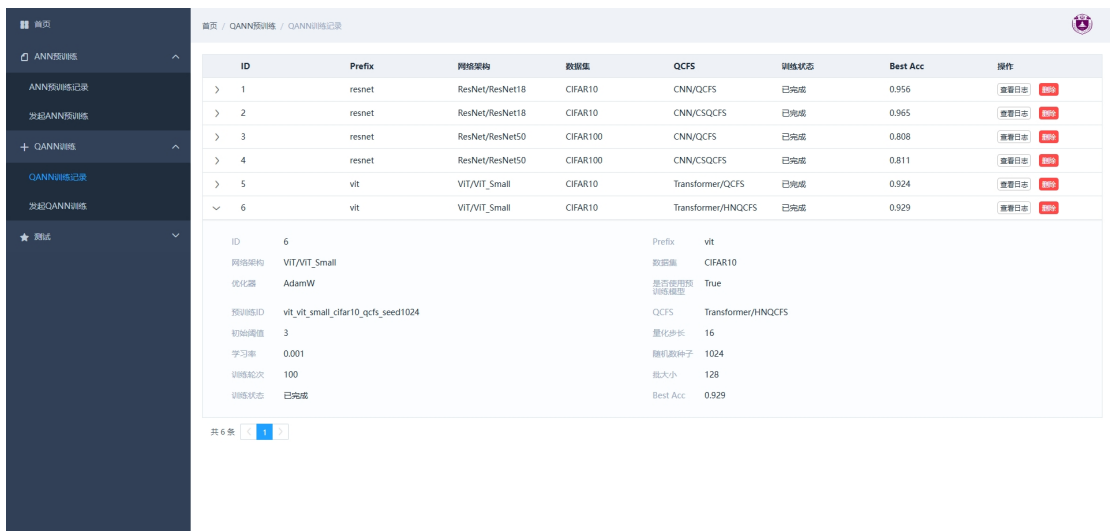


图 5-8 QANN 训练记录

QANN 训练日志页面如图5-9所示，展示了 QANN 训练任务的详细日志。

5.4.5 测试模块

SNN 测试页面如图5-10所示，用户需要选择要测试的网络架构、数据集、模型 ID、批大小、QCFs 类型以及 SNN 模型步长等参数。此外，用户在测试时还可以配置是否使用 $\text{COS}^{[33]}$ 技术。配置完成后，点击发起测试按钮进行测试，在下面会实时显示测试日志，包含测试进度以及在不同模拟步长下的精度。

此外，用户还可以在测试记录页面查看历史测试记录，如图5-11所示。类似

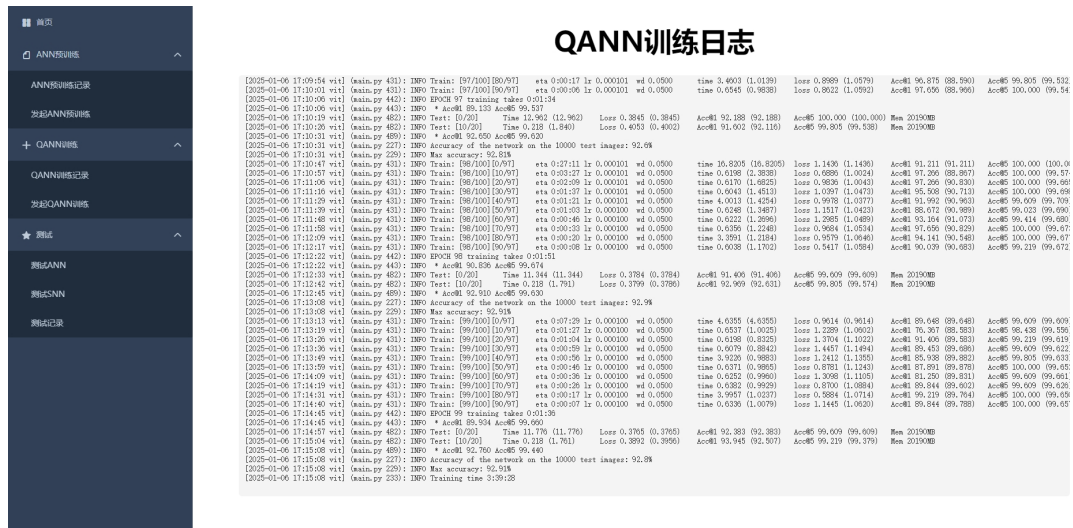


图 5-9 QANN 训练日志

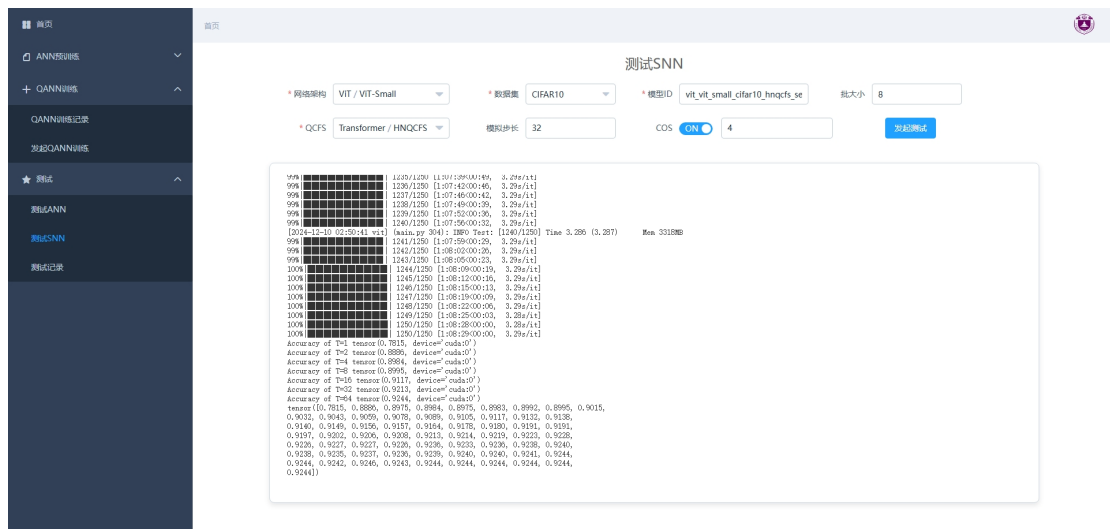


图 5-10 测试 SNN

训练记录页面，测试记录包含了网络类型（ANN/SNN）、网络架构、数据集、模型 ID 等基础信息，点击左边的箭头可以查看更详细的信息。用户也可以通过查看日志按钮和删除按钮来查看详细的测试日志或终止/删除测试任务。

5.5 本章小结

本章主要介绍了基于转换的脉冲神经网络训练系统的设计与实现过程。首先，我们阐述了系统研发的背景，分析了该系统在脉冲神经网络领域中的实际应用需求。基于这些需求，我们进行了详细的需求分析，为系统的功能和设计提供了清晰的指导方向。

ID	网络类型	网络架构	数据集	模型ID	测试状态	操作
> 1	ANN	ResNet/ResNet18	CIFAR10	resnet_resnet18_cifar10_cs_qcfs_seed42	已完成	查看日志 删除
∨ 2	SNN	VIT/VIT-Small	CIFAR10	vit_vit_small_cifar10_hnqcfs_seed1024	已完成	查看日志 删除

ID	2	网络类型	SNN
网络架构	VIT/VIT-Small	数据集	CIFAR10
模型ID	vit_vit_small_cifar10_hnqcfs_seed1024	QCFS	Transformer/HNQCFS
批大小	8	训练状态	已完成
Test Acc	0.9244	模型步长	32
COS	true	COS长度	4

共 2 条 1 2

图 5-11 测试记录

在系统设计部分，我们提出了系统的整体架构，主要分为用户交互层、服务逻辑层以及数据存储层，并详细描述了各个层级的设计思路和功能需求。系统设计的核心目标是构建一个便于非脉冲神经网络领域专业人员使用的训练平台，以支持脉冲卷积神经网络和脉冲 Transformer 的转换训练。

系统实现方面，我们介绍了系统的开发环境，包括所采用的开发工具、平台及其配置。我们还展示了系统的界面设计和使用流程，确保系统操作的简便性和易用性。具体到各模块的实现，ANN 预训练模块、QANN 训练模块及测试模块使用流程均有详细阐述。通过这些模块的紧密配合，系统能够顺利完成 ANN 到 SNN 的转换，并有效进行训练与测试。

第六章 总结与展望

随着脉冲神经网络 (SNN) 技术的不断发展,其在生物启发计算和低功耗神经形态计算中的潜力得到了广泛关注。然而,由于 SNN 自身的稀疏性和动态性特征,与传统人工神经网络 (ANN) 相比,SNN 在训练方法和模型性能方面仍然存在较大差距。本文针对这一现状,从 ANN 到 SNN 转换的角度出发,提出了基于转换的脉冲卷积神经网络和脉冲 Transformer 的训练方法,并开发了相关的训练系统,以支持非 SNN 领域专业人士的应用需求,为 SNN 的实际应用提供了有效的技术支持。

在脉冲卷积神经网络方面,本文针对现有 QCFS 激活函数在卷积神经网络中的局限性(如通道差异性和负阈值问题),提出了 CS-QCFS 函数。该方法通过为每个通道独立训练阈值,确保通道间差异能够被有效捕捉,同时引入 Softplus 变换避免负阈值问题,从而提高了 SNN 与 ANN 之间的性能一致性。实验结果表明,CS-QCFS 函数能够在 CIFAR 数据集上显著提升性能,缩小 SNN 与 ANN 的性能差距。

在脉冲 Transformer 的研究中,本文针对 Transformer 模型中多头注意力机制和全连接层的神经元差异性,提出了多头 QCFS 和神经元 QCFS 函数,并结合两者设计了 HN-QCFS 函数。这一方法通过为每个注意力头和全连接层神经元独立训练阈值,有效增强了 Transformer 模型在 SNN 中的表现力和自适应性。实验验证表明,HN-QCFS 函数能够进一步提升脉冲 Transformer 的性能,缩小 SNN 与 ANN 之间的性能差距,并为脉冲 Transformer 在更多任务中的应用提供了支持。

此外,本文设计并实现了一个基于转换的脉冲神经网络训练系统。该系统集成了本文提出的脉冲卷积神经网络和脉冲 Transformer 的训练方法,为用户提供了包括 ANN 预训练、QANN 训练和测试在内的一站式操作流程。通过模块化设计和用户友好的界面,系统降低了 ANN 到 SNN 转换的使用门槛,为神经形态计算在广泛应用中的落地提供了实际支持。

尽管本研究在基于转换的脉冲神经网络训练方法和系统设计方面取得了一

些进展,但仍存在一些亟待解决的问题,并有许多值得探索的方向。未来的研究可以从以下几个方面进行扩展和深入:

1. 扩展到更多的神经网络架构。目前,本论文的工作主要集中在卷积神经网络和 Transformer 模型上。未来的研究可以将基于转换的脉冲神经网络方法扩展到其他神经网络架构,如图神经网络 (GNN)、循环神经网络 (RNN) 等。通过为这些网络设计适配的转换方法,能够进一步丰富 SNN 在各类网络中的应用潜力。
2. 大规模数据集和更复杂任务的验证。本论文的实验主要集中在 CIFAR 数据集上,未来的研究可以在更大规模的数据集 (如 ImageNet、COCO 等) 上进行验证,以评估所提出方法在处理大规模数据时的表现。此外,针对更加复杂的任务,如目标检测、语义分割、自然语言处理等领域,脉冲神经网络的潜力仍有待进一步挖掘。
3. 进一步优化激活函数。尽管本文提出的 CS-QCFS 和 HN-QCFS 激活函数在一定程度上解决了 SNN 与 ANN 之间的性能差距,但在一些复杂任务中,仍可能面临性能瓶颈。因此,未来的研究可以进一步优化这些激活函数,探索更适合脉冲神经网络的非线性变换函数,以提高 SNN 的表达能力和计算效率。

参考文献

- [1] WINSTON P H. Artificial intelligence[M]. Addison-Wesley Longman Publishing Co., Inc., 1984.
- [2] WINSTON P H. Artificial intelligence[M]. Addison-Wesley Longman Publishing Co., Inc., 1992.
- [3] BALLARD D H, BROWN C M. Computer vision[M]. Prentice Hall Professional Technical Reference, 1982.
- [4] VOULODIMOS A, DOULAMIS N, DOULAMIS A, et al. Deep learning for computer vision: A brief review[J]. Computational intelligence and neuroscience, 2018, 2018(1): 7068349.
- [5] SZELISKI R. Computer vision: algorithms and applications[M]. Springer Nature, 2022.
- [6] NADKARNI P M, OHNO-MACHADO L, CHAPMAN W W. Natural language processing: an introduction[J]. Journal of the American Medical Informatics Association, 2011, 18(5): 544-551.
- [7] HIRSCHBERG J, MANNING C D. Advances in natural language processing[J]. Science, 2015, 349(6245): 261-266.
- [8] CHOWDHARY K, CHOWDHARY K. Natural language processing[J]. Fundamentals of artificial intelligence, 2020: 603-649.
- [9] ANUSUYA M, KATTI S K. Speech recognition by machine, a review[J]. ArXiv preprint arXiv:1001.2267, 2010.
- [10] GAIKWAD S K, GAWALI B W, YANNAWAR P. A review on speech recognition technique[J]. International Journal of Computer Applications, 2010, 10(3): 16-24.

- [11] ACHIAM J, ADLER S, AGARWAL S, et al. Gpt-4 technical report[J]. ArXiv preprint arXiv:2303.08774, 2023.
- [12] DOSOVITSKIY A. An image is worth 16x16 words: Transformers for image recognition at scale[J]. ArXiv preprint arXiv:2010.11929, 2020.
- [13] LI C, SHANG Z, SHI L, et al. IC-SNN: Optimal ANN2SNN conversion at low latency[J]. Mathematics, 2022, 11(1): 58.
- [14] STRUKOV D, INDIVERI G, GROLLIER J, et al. Building brain-inspired computing[J]. Nature Communications, 2019(10): 4838-2019.
- [15] MEHONIC A, KENYON A J. Brain-inspired computing needs a master plan[J]. Nature, 2022, 604(7905): 255-260.
- [16] GHOSH-DASTIDAR S, ADELI H. Spiking neural networks[J]. International journal of neural systems, 2009, 19(04): 295-308.
- [17] NEFTCI E O, MOSTAFA H, ZENKE F. 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.
- [18] CAO Y, CHEN Y, KHOSLA D. Spiking deep convolutional neural networks for energy-efficient object recognition[J]. International Journal of Computer Vision, 2015, 113: 54-66.
- [19] KHERADPISHEH S R, GANJTABESH M, THORPE S J, et al. STDP-based spiking deep convolutional neural networks for object recognition[J]. Neural Networks, 2018, 99: 56-67.
- [20] LI Z, LIU F, YANG W, et al. A survey of convolutional neural networks: analysis, applications, and prospects[J]. IEEE transactions on neural networks and learning systems, 2021, 33(12): 6999-7019.
- [21] VASWANI A. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017.

- [22] WANG Y, ZHANG M, CHEN Y, et al. Signed neuron with memory: Towards simple, accurate and high-efficient ann-snn conversion[C] // International Joint Conference on Artificial Intelligence. 2022.
- [23] RUECKAUER B, LUNGU I A, HU Y, et al. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification[J]. *Frontiers in neuroscience*, 2017, 11: 682.
- [24] DIEHL P U, NEIL D, BINAS J, et al. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing[C] // 2015 International joint conference on neural networks (IJCNN). 2015: 1-8.
- [25] KIM S, PARK S, NA B, et al. Spiking-yolo: spiking neural network for energy-efficient object detection[C] // Proceedings of the AAAI conference on artificial intelligence: vol. 34: 07. 2020: 11270-11277.
- [26] HAN B, SRINIVASAN G, ROY K. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network[C] // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 13558-13567.
- [27] 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.
- [28] HO N D, CHANG I J. TCL: an ANN-to-SNN conversion with trainable clipping layers[C] // 2021 58th ACM/IEEE Design Automation Conference (DAC). 2021: 793-798.
- [29] DENG S, GU S. Optimal conversion of conventional artificial neural networks to spiking neural networks[J]. *ArXiv preprint arXiv:2103.00476*, 2021.
- [30] LI Y, DENG S, DONG X, et al. Converting artificial neural networks to spiking neural networks via parameter calibration[J]. *ArXiv preprint arXiv:2205.10121*, 2022.

-
- [31] BU T, FANG W, DING J, et al. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks[J]. ArXiv preprint arXiv:2303.04347, 2023.
- [32] HAO Z, BU T, DING J, et al. Reducing ANN-SNN Conversion Error through Residual Membrane Potential[J]. ArXiv preprint arXiv:2302.02091, 2023.
- [33] HAO Z, DING J, BU T, et al. Bridging the Gap between ANNs and SNNs by Calibrating Offset Spikes[J]. ArXiv preprint arXiv:2302.10685, 2023.
- [34] SENGUPTA A, YE Y, WANG R, et al. Going deeper in spiking neural networks: VGG and residual architectures[J]. *Frontiers in neuroscience*, 2019, 13: 95.
- [35] ZHOU Z, ZHU Y, HE C, et al. Spikformer: When spiking neural network meets transformer[J]. ArXiv preprint arXiv:2209.15425, 2022.
- [36] ZHOU C, YU L, ZHOU Z, et al. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network[J]. ArXiv preprint arXiv:2304.11954, 2023.
- [37] YAO M, HU J, ZHOU Z, et al. Spike-driven transformer[J]. *Advances in neural information processing systems*, 2024, 36.
- [38] WANG Z, FANG Y, CAO J, et al. Masked spiking transformer[C] // *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023: 1761-1771.
- [39] SHI X, HAO Z, YU Z. SpikingResformer: Bridging ResNet and Vision Transformer in Spiking Neural Networks[C] // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024: 5610-5619.
- [40] ZHOU Z, CHE K, FANG W, et al. Spikformer v2: Join the high accuracy club on imagenet with an snn ticket[J]. ArXiv preprint arXiv:2401.02020, 2024.
- [41] YAO M, HU J, HU T, et al. Spike-driven transformer v2: Meta spiking neural network architecture inspiring the design of next-generation neuromorphic chips[J]. ArXiv preprint arXiv:2404.03663, 2024.

- [42] HODGKIN A L, HUXLEY A F. A quantitative description of membrane current and its application to conduction and excitation in nerve[J]. The Journal of physiology, 1952, 117(4): 500.
- [43] IZHIKEVICH E M. Simple model of spiking neurons[J]. IEEE Transactions on neural networks, 2003, 14(6): 1569-1572.
- [44] TAL D, SCHWARTZ E L. Computing with the leaky integrate-and-fire neuron: logarithmic computation and multiplication[J]. Neural computation, 1997, 9(2): 305-318.
- [45] WEISSTEIN E W. Heaviside step function[J]. <https://mathworld.wolfram.com/>, 2002.
- [46] CUI X, BAI H, ZHAO Y, et al. Combination of Fast Finite Shear Wave Transform and Optimized Deep Convolutional Neural Network: A Better Method for Noise Reduction of Wetland Test Images[J]. Electronics, 2023, 12(17): 3557.
- [47] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.
- [48] LIU F, ZHAO W, CHEN Y, et al. Spikeconverter: An efficient conversion framework zipping the gap between artificial neural networks and spiking neural networks[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 36: 2. 2022: 1692-1701.
- [49] BURKITT A N. A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input[J]. Biological cybernetics, 2006, 95: 1-19.
- [50] IOFFE S. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. ArXiv preprint arXiv:1502.03167, 2015.
- [51] YOU K, XU Z, NIE C, et al. SpikeZIP-TF: Conversion is All You Need for Transformer-based SNN[J]. ArXiv preprint arXiv:2406.03470, 2024.
- [52] ZHANG A, LIPTON Z C, LI M, et al. Dive into deep learning[M]. Cambridge University Press, 2023.

- [53] QIN Z, YU F, LIU C, et al. How convolutional neural network see the world-A survey of convolutional neural network visualization methods[J]. ArXiv preprint arXiv:1804.11191, 2018.
- [54] OLAH C, SATYANARAYAN A, JOHNSON I, et al. The building blocks of interpretability[J]. Distill, 2018, 3(3): e10.
- [55] ZEILER M D, FERGUS R. Visualizing and understanding convolutional networks[C]//Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13. 2014: 818-833.
- [56] ZHENG H, YANG Z, LIU W, et al. Improving deep neural networks using soft-plus units[C]//2015 International joint conference on neural networks (IJCNN). 2015: 1-4.
- [57] BENGIO Y, LÉONARD N, COURVILLE A. Estimating or propagating gradients through stochastic neurons for conditional computation[J]. ArXiv preprint arXiv:1308.3432, 2013.
- [58] KRIZHEVSKY A, HINTON G, et al. Learning multiple layers of features from tiny images[J]., 2009.
- [59] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
- [60] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. ArXiv preprint arXiv:1409.1556, 2014.
- [61] BOTTOU L. Stochastic gradient descent tricks[G]//Neural Networks: Tricks of the Trade: Second Edition. Springer, 2012: 421-436.
- [62] LOSHCHILOV I, HUTTER F. Sgdr: Stochastic gradient descent with warm restarts[J]. ArXiv preprint arXiv:1608.03983, 2016.

-
- [63] YANG S, GUO S, ZHAO J, et al. Investigating the effectiveness of data augmentation from similarity and diversity: An empirical study[J/OL]. *Pattern Recognition*, 2024, 148: 110204. <https://www.sciencedirect.com/science/article/pii/S031320323009019>. DOI: <https://doi.org/10.1016/j.patcog.2023.110204>.
- [64] DEVRIES T, TAYLOR G W. Improved regularization of convolutional neural networks with cutout[J]. *ArXiv preprint arXiv:1708.04552*, 2017.
- [65] CUBUK E D, ZOPH B, MANE D, et al. Autoaugment: Learning augmentation strategies from data[C] // *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019: 113-123.
- [66] MICHEL P, LEVY O, NEUBIG G. Are sixteen heads really better than one?[J]. *Advances in neural information processing systems*, 2019, 32.
- [67] LIU Z, LIN Y, CAO Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C] // *Proceedings of the IEEE/CVF international conference on computer vision*. 2021: 10012-10022.
- [68] LOSHCHILOV I, HUTTER F. Decoupled weight decay regularization[J]. *ArXiv preprint arXiv:1711.05101*, 2017.

致 谢

行文至此，落笔为终。回首研究生三年的点滴时光，无数感动与成长涌上心头。犹记电影《无问西东》里的叩问：“如果提前了解你所要面对的人生，你是否还有勇气前来？”在无数个调试代码的深夜，在一次次实验数据与理论预期偏离的迷惘时刻，在求职季收到一封封感谢信时，我也曾这样自问。而今答案依然清晰——纵然提前知晓这三年的艰辛，我仍会坚定地踏上这条求知之路。此刻，谨以最真挚的笔墨，向一路陪伴、支持与指引我的人们致以最深切的谢意。

首先，感谢我的导师申富饶教授。感谢申老师以渊博的学识与严谨的治学态度为我点亮学术道路的明灯。每周的一对一个人讨论中，您总能以高屋建瓴的视角为我厘清研究思路，又以极大的耐心包容我的试错与探索。更令我感激的是，您始终尊重学生的学术志趣，给予我们自由选择研究方向的空间，更以张弛有度的指导方式让我在科研中保有热情与创造力。在您的关怀下，我的研究生生活不仅充满成长和收获，更因您的言传身教而倍感温暖和力量。

其次，感谢 RINC 研究组的每一位成员，是你们让我的研究生时光绽放出多元的色彩。我们曾为科研难题围坐长谈，也曾在球场上挥洒汗水，亦会在茶余饭后探讨社会热点，在食堂餐桌、街角小馆的烟火气里分享美食。这段并肩前行的岁月，早已成为我成长中不可或缺的精神财富。

最后，感谢我的家人和挚友，你们是最坚实的后盾。你们用鼓励为我注入动力，用陪伴驱散科研路上的孤单。谢谢你们让我知道，前行从不孤单，这份爱与支持，是我永远的底气。

简历和科研成果

基本信息

杨洪朝，男，汉族，2001年2月出生，河南新乡人。

教育背景

2022年9月 - 2025年6月 南京大学计算机学院 硕士
2018年9月 - 2022年6月 武汉大学计算机学院 本科

攻读硕士学位期间完成的学术成果

1. **Hongchao Yang**, Suorong Yang, Lingming Zhang, Hui Dou, Furao Shen, Jian Zhao. CS-QCFS: Bridging the performance gap in ultra-low latency spiking neural networks. *Neural Networks*, 2025.
2. Suorong Yang, **Hongchao Yang**, Suhan Guo, Furao Shen, Jian Zhao. Not All Data Matters: An End-to-End Adaptive Dataset Pruning Framework for Enhancing Model Performance and Efficiency. *arXiv preprint arXiv:2312.05599*, 2023.
3. 杨锁荣, **杨洪朝**, 申富饶, 赵健. 面向深度学习的图像数据增强综述. *软件学报*, 2024.
4. **杨洪朝**, 易梦军, 李培佳, 张瀚文, 申富饶, 赵健, 王刘旺. 同态加密在深度学习中的应用综述. *计算机科学与探索*, 2024.

攻读硕士学位期间参与的科研课题

1. 科技部重大项目“基于神经可塑性的脉冲网络高效学习机制与类脑智能系统”（参与课题年限2022年9月—2025年6月），负责脉冲神经网络模型相关研究。