学校代码：10284
分 类 号：TP181
密　　级：公开
U D C：004.8
学　　号：MG21370035

南京大学

硕士学位论文

论文题目　基于转换方法的脉冲神经网络训练和剪枝研究

作者姓名　王翔宇

专业名称　计算机科学与技术

研究方向　脉冲神经网络

导师姓名　申富饶 教授

2024 年 5 月 16 日

答辩委员会主席 戴新宇 教授

评 阅 人 戴新宇 教授

徐明华 教授

论文答辩日期 2024 年 5 月 16 日

研究生签名：王翔宇

导师签名：

Research on Training and Pruning of Spiking Neural Networks Based on Conversion Method

by

Wang Xiang-yu

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 Science and Technology

School of Artificial Intelligence

Nanjing University

May 16, 2024

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：基于转换方法的脉冲神经网络训练和剪枝研究

计算机科学与技术 专业 2021 级硕士生姓名：王翔宇

指导教师（姓名、职称）：申富饶 教授

摘　要

脉冲神经网络因其生物仿生性和计算低功耗的潜力在深度学习、类脑计算等领域都广受关注。在深度学习领域，由于日益发展的技术和应用对计算资源的需求逐渐增加，对如何实现深度脉冲神经网络的高性能与低能耗的探索成为热点研究方向。

由于脉冲神经网络计算不可导，传统人工神经网络已有的训练技术无法直接使用，脉冲神经网络训练方法的研究依然是热点与难点。常用的方法有转换训练方法、代理梯度方法，本文聚焦通过转换训练脉冲神经网络的方法，通过形式化转换误差，分析影响转换误差的主要因素，并基于此设计了基于神经元阈值优化的脉冲神经网络转换训练方法，与已有的训练方法进行实验对比，实现了更高的准确率。

为实现脉冲神经网络的低能耗，网络剪枝是一种降低计算能耗的常用方式。此类技术通过减少神经网络参数量来降低模型复杂度和计算成本，在传统人工神经网络中取得显著成效。但由于脉冲神经网络的计算有时间维度且不可导，所以剪枝问题更复杂，传统人工神经网络剪枝方法往往需要适配，且受到时空反向传播和代理梯度的限制。本文基于转换训练方法，尝试通过转换剪枝人工神经网络得到剪枝脉冲神经网络的方式。将人工神经网络结构化剪枝建模成关于转换误差和准确率的二目标优化问题，并通过对转换误差的分析将问题规模按层分解，然后使用演化算法高效求解。与已有方法进行实验对比，在更高剪枝率的同时实现了更高的准确率。

关键词：脉冲神经网络，转换训练方法，结构化剪枝

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Research on Training and Pruning of Spiking Neural Networks Based on Conversion Method

SPECIALIZATION: Computer Science and Technology

POSTGRADUATE: Wang Xiang-yu

MENTOR: Professor Shen Fu-rao

ABSTRACT

Spiking neural networks have attracted widespread attention in deep learning, brain-inspired computing and other fields due to their biological bionics and potential for low power consumption. In the field of deep learning, due to the increasing demand for computing resources from developing technologies and applications, the research direction of how to achieve high performance and low power consumption of deep spiking neural network is a Hot topic.

For the training problem of spiking neural networks, because their calculation is not differentiable, the existing training techniques of traditional artificial neural network can not be directly used, and the research of training methods of spiking neural networks are still difficult problems. The commonly used methods are conversion training method and surrogate gradient method. This paper focuses on the methods of training spiking neural networks through conversion. By formalizing the conversion error, the main factors affecting the conversion error are analyzed. Based on this, we propose the spiking neural networks conversion training method based on neuron threshold optimization. Compared with the existing training methods, the experimental results show our method achieves higher accuracy.

For the low power consumption problem, network pruning is a common way to reduce computing energy consumption. Such technologies reduce the model complexity and computing cost by reducing the number of neural network parameters, have achieved remarkable results in traditional artificial neural networks. However, due to the calculation of spiking neural networks have time dimension and is not derivative,

the pruning problems are more complex. The traditional artificial neural networks pruning methods often need adaptation, and are restricted by spatio-temporal back propagation and surrogate gradient. Based on the conversion training method, we attempt to obtain the pruned spiking neural network by transforming the pruned artificial neural network. The structured pruning problem of artificial neural networks is formulated as a two-objective optimization problem about the conversion error and accuracy rate. The problem is decomposed hierarchically by analyzing the conversion error, and then the evolutionary algorithm is used to solve it efficiently. Compared with the existing methods, this method achieves higher accuracy rate with higher pruning rate.

KEYWORDS: Spiking Neural Networks; Conversion Methods; Structured Pruning

目　　录

插图目录

表格目录

第一章　绪论

1.1　研究背景

近年来，随着人工智能[1]的迅速发展，人工神经网络[2-4]（Artificial Neural Networks, ANNs）成为深度学习[5-7]领域中不可或缺的技术之一。这种神经网络的设计灵感来源于生物神经系统，通过模拟生物神经元的结构和功能，构建出能够进行模式识别和数据处理的网络模型。人工神经网络在计算机视觉、自然语言处理、语音识别、机器翻译等领域得到了广泛应用。例如，在计算机视觉领域，人工神经网络可以对图像进行特征提取和分类，实现高效准确的图像识别；在自然语言处理领域，人工神经网络可以用于文本分类、情感分析、机器翻译等任务。随着深度学习技术在自动驾驶[8]、人机交互和推荐系统[9]等领域的实际应用，对计算资源和能源消耗的需求也日益增加。因此，近年来对低功耗和低计算资源模型的探索已成为研究领域中备受关注的焦点。

随着神经科学[2,10]、硬件[11]和软件[12-13]的发展，越来越多的研究关注对于神经元的模拟。1952 年，Alan Hodgkin 和 Andrew Huxle[10]通过对乌贼巨大神经元的电活动过程的观察，揭示了神经元膜电位变化的离子通道机制，提出了 HH（Hodgkin-Huxley）模型，首次完成了对生物神经元电信号处理过程的数学建模。HH 模型基于乌贼神经元的离子通道动力学，描述了神经元膜上的离子流动和膜电位变化，模型包括四个主要的离子通道如图1-1所示，从左到右分别是：电容、钠离子通道、钾离子通道和泄漏电流，并用一组微分方程来描述离子通道的动态行为。这些微分方程基于实验测量得到的电流-电压关系和离子通道特性，描述了离子通道的开放状态随时间的变化，并通过离子流动来计算膜电位的变化。2003 年，Izhikevich 和 Eugene M 在论文[14]提出了一种简化的脉冲神经元模型——LIF（Leaky-Integrate-and-Fire）模型，该模型模拟生物神经元的兴奋和抑制行为，模拟了生物神经元膜电位的整合、放电和泄露过程。只考虑整合和放电

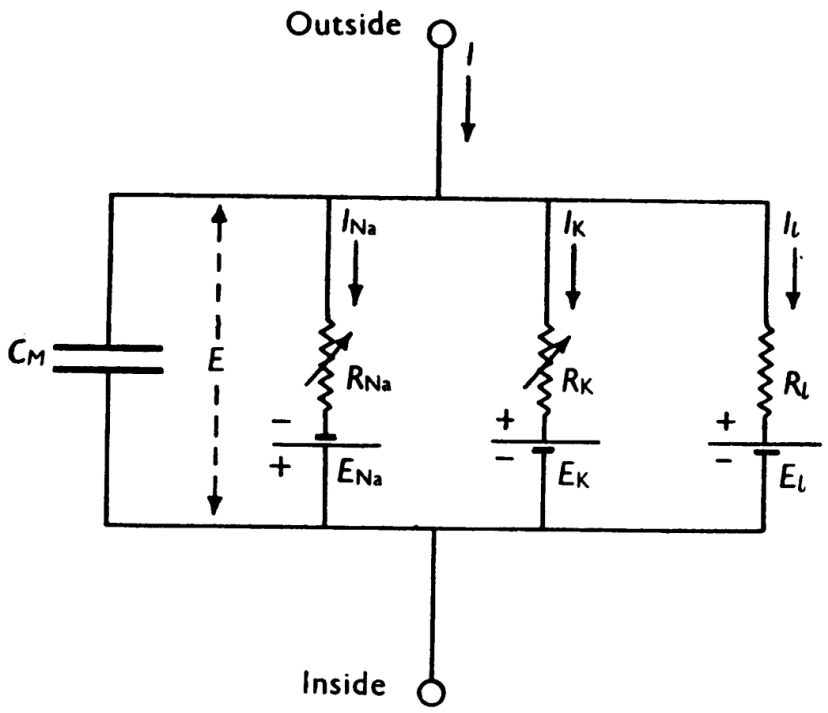

图 1-1 HH 模型[10]

不考虑泄露过程的神经元被称为 IF（Integrate-and-Fire）神经元模型。由脉冲神经元连接构建的脉冲神经网络（Spiking Neural Networks, SNNs）因其更强的生物仿生性和计算稀疏性所带来的硬件计算低功耗潜力在脑科学、人工智能等领域受到大量关注。近年来，随着人工智能领域下深度学习的发展，将人工神经网络积累的成功经验发展、应用到脉冲神经网络领域当中，训练高性能的脉冲神经网络称为一个研究热点。同时，为减少对计算资源和能量消耗的依赖，实现脉冲神经网络计算低功耗的潜力，对脉冲神经网络进行模型压缩近几年也受到广泛关注。本文专注于通过研究训练和剪枝算法实现高性能、低功耗的脉冲神经网络。

关于实现脉冲神经网络高性能的训练方法，由于脉冲神经网络的计算具有离散、不可导的特点，人工神经网络中常用的梯度下降（Gradient Decent）更新参数使网络拟合数据的方法无法直接应用。目前已有的脉冲神经网络训练方法主要有代理梯度训练方法（Surrogate Gradient）、脉冲时序依赖可塑性方法（Spike-Timing-Dependent Plasticity）和人工神经网络转脉冲神经网络方法（ANN-to-SNN）。代理梯度方法属于有监督学习，通过近似计算神经元梯度信息，使得梯度下降可以直接用于脉冲神经网络训练。在前向传播过程中，脉冲信号在网络中进行权重计算和脉冲激活，在反向传播中，根据网络输出与目标值之间的差值，使用与脉冲激活函数近似的可导函数的导数更新网络中的参数。脉冲时序

依赖可塑性方法[15]属于无监督学习，是一种受生物神经元突触可塑性的启发在脉冲神经网络中用于调整突触连接权重的学习规则，这种方法基于突触前后神经元之间的脉冲时序关系，即不同神经元的相对发放时间顺序调整权重，如果突触前神经元在突触后神经元发放脉冲之前发放脉冲，突触权重会增大，反之，突触权重会减小。这种可塑性机制使得脉冲神经网络能够自适应地学习和调整连接权重，以优化网络的功能和性能。人工神经网络转脉冲神经网络是一种间接的监督学习方法，将一个训练好的人工神经网络的参数迁移到脉冲神经网络当中，并进一步通过数据来微调或训练脉冲神经网络中的参数。这三类方法目前都在探索研究中并取得了一定效果。本文关注改进基于转换的脉冲神经网络训练方法，通过分析、降低脉冲神经网络与人工神经网络之间的转换误差实现来实现高性能的深度脉冲神经网络。

关于脉冲神经网络的低能耗问题，与人工神经网络类似，深度脉冲神经网络的训练和推理同样需要大量的计算资源和能源消耗。因此，如何在保持脉冲神经网络高性能的同时尽可能减小计算开销成为近些年脉冲神经网络研究的热点之一。常见的方法有权重量化、低秩近似、网络蒸馏、网络剪枝。权重量化使用参数量化技术，例如使用低精度表示，可以减小模型的存储和计算需求。低秩近似通过将权重矩阵近似为低秩矩阵，同样可以减小模型模型的存储需求和计算量，常见的方法包括奇异值分解和张量分解等。网络蒸馏通过使用一个大型的、精确的网络作为教师网络，来指导训练一个轻量级的网络，通过这种方式使得轻量级网络实现大型网络的性能。网络剪枝通过删除冗余的神经元和连接，可以减小网络的规模和复杂度，是人工神经网络中常用的模型压缩方法。但是由于脉冲神经网络的计算不可导，人工神经网络剪枝方法不能简单适。而且剪枝中常常涉及到对稀疏模型的训练，由于直接训练脉冲神经网络需要使用时空反向传播和代理梯度，稀疏模型尤其是深度模型在训练中易受到梯度爆炸或梯度消失的影响，从而训练不稳定，难以在保证高性能的前提下实现高剪枝率。本文基于转换训练方法，尝试通过转换剪枝人工神经网络得到剪枝脉冲神经网络的方式。将人工神经网络结构化剪枝建模成关于转换误差和准确率的二目标优化问题，并通过对转换误差的分析将问题规模按层分解，然后使用演化算法高效求解，实现了高性能、低功耗的深度脉冲神经网络。

1.2 研究问题

1.2.1 ANN-to-SNN 研究

ANN-to-SNN，通过将训练好的人工神经网络转换成脉冲神经网络，以在脉冲信号计算离散、不可导的脉冲神经网络上实现数据拟合效果。一般地，ANN-to-SNN 包含以下流程：1. 给定一个已经训练好的、特定结构可以用于转换的人工神经网络；2. 初始化一个相同结构的脉冲神经网络，将人工神经网络的权重复制到该网络中；3. 通过转换算法确定脉冲神经元的激活阈值。其示意图如图1-2所示。对于 ANN-to-SNN 算法的效果评价，常在通用数据集上进行对比，有两个常用的评价指标，即脉冲神经网络的准确率和转换造成的准确率下降。

图 1-2 基于转换的脉冲神经网络训练方法的一般性示意图

此类方法基本思路是使用脉冲神经网络的平均激活值拟合人工神经网络的激活值，实现与人工神经网络结构和功能类似的脉冲神经网络。深度学习中常用的卷积神经网络[16]（Convolutional Neural Network，CNN）和残差神经网络[17]（Residual Neural Network，ResNet）在图像分类[6]、目标检测[18]等任务中取得了显著的成果，基于此类结构的脉冲转换成为当前研究热点，一般性示意图如图1-3所示，左侧为 CNN 结构转换示意图，右侧为 ResNet 转换示意图。

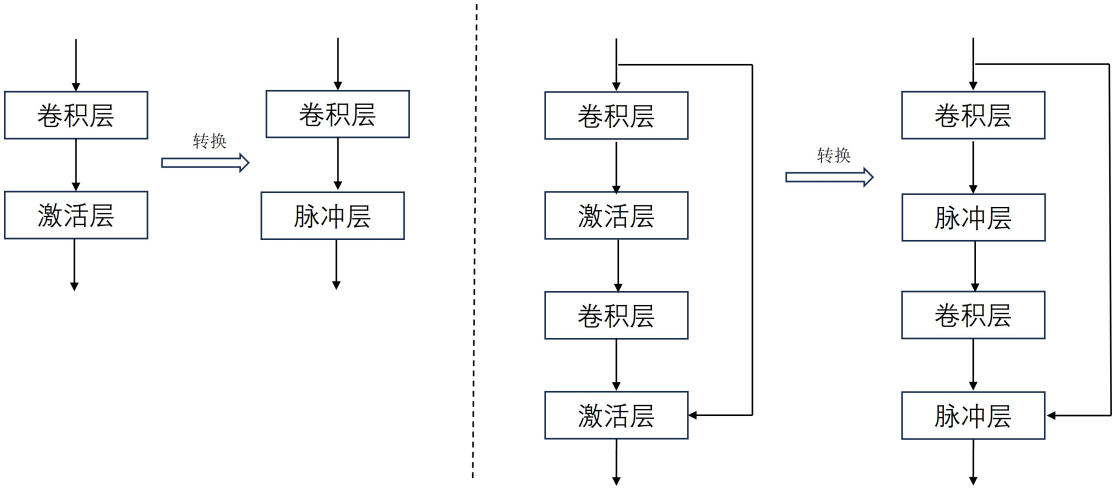

图 1-3 脉冲神经网络转换示意图

根据 ANN-to-SNN 方法是否涉及理论，现有的 ANN-to-SNN 方法可以被分为启发式方法和基于理论的方法。ANN-to-SNN 最早的研究是启发式方法，通过符合直觉的人为设计的规则式，设计特殊结构的人工神经网络或使用特殊的计算层构建人工神经网络，然后在训练完该网络后将权重参数复制迁移给对应结构的脉冲神经网络，脉冲神经网络中的激活阈值也往往凭经验设定，例如通过统计人工神经网络中神经元在不同数据样本上的激活值然后根据划定特定的百分位点来确定激活阈值[19]。这种方式虽然没有理论保证，但在研究初期取得了一定的效果，并且有些设计启发了后来的方法。在此基础上，发展出了转换理论，通过形式化分析人工神经网络和转换所得脉冲神经网络之间的误差，确定影响转换误差的参数，并通过对转换误差的分解设计相应的优化方法确定激活阈值或权重以尽可能消除转换误差。根据确定激活阈值的方法，此类方法分为两类。第一类，在参数迁移复制后再通过优化方式确定阈值或使用梯度微调权重参数；第二类，通过设计使用激活阈值作为网络参数的人工神经网络将激活阈值显式加入到人工神经网络的训练中，使用梯度下降的方式学得恰当的激活阈值和其他网络参数，然后直接将激活阈值和网络参数复制迁移个对应结构的脉冲神经网络。这两类方法都已经在深度卷积神经网络上取得了与传统人工神经网络相近的性能。

1.2.2 脉冲神经网络剪枝研究

脉冲神经网络剪枝，与人工神经网络剪枝类似，即从脉冲神经网络中删除冗余的连接或神经元，在尽可能地保持现有精度前提下，降低模型对存储和计算资源的需求。现有的脉冲神经网络剪枝方法一般地包含以下流程：

1. 训练好一个脉冲神经网络；

2. 确定权重或神经元的重要程度，将不重要的去除；

3. 微调或训练恢复脉冲神经网络的性能。

对于使用卷积结构构建的脉冲神经网络，根据剪枝粒度的不同，脉冲神经网络剪枝方法可以分为非结构化剪枝方法和结构化剪枝方法（Structured pruning）。非结构化剪枝方法对连接权重或神经元进行裁剪，结果往往不规整，难以真正实现硬件加速[20]。结构化剪枝指的是以滤波器、卷积层或结构块为最小单位进行剪枝以实现整体结构的稀疏性，并且在实际应用中真实节约硬件资源、便于

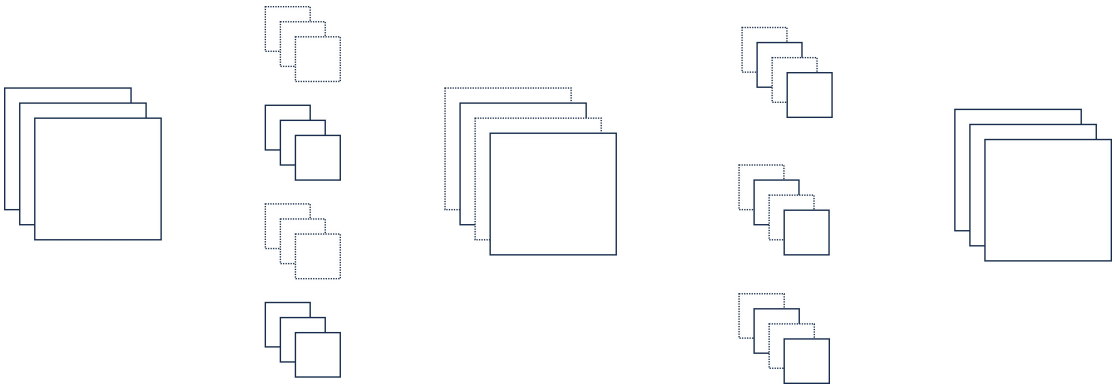

图 1-4 卷积神经网络结构化剪枝示意图

硬件加速。以滤波器为基本单位进行剪枝是当前神经网络结构化剪枝领域内的最主要研究对象，本文关注于以滤波器为基本单位进行脉冲神经网络剪枝。

以一个卷积层为例，其包含的权重参数可以用一个维度为 I×O×H×W 的张量表示，I 表示滤波器的通道数，O 表示滤波器的个数，H 表示滤波器的高，W 表示滤波器的宽。以滤波器为单位进行结构化剪枝就是从 O 所在的维度进行剪枝，如图1-4所示，该图表示卷积神经网络中三层特征图和两层卷积层，图较大的正方形表示特征图的一个通道，较小的正方形表示滤波器的一个通道，该图描述的是对第一层卷积层进行结构化剪枝：在第一层中去掉某些滤波器，输出特征图和下一层卷积层对应的部分也剪去。通过去除冗余的滤波器实现参数量和计算量的下降，剪枝后的滤波器个数为 O'，那么该卷积层权重参数可以用 $I \times O' \times H \times W$ 的张量表示。并且该层的下一层也要做相应变化，如果下一层的权重参数矩阵维度为 $I_1 \times O_1 \times H_1 \times W_1$，其中 $I_1 = O$，那么剪枝后其权重矩阵维度为 $O' \times O_1 \times H_1 \times W_1$。以滤波器为基本单位的脉冲神经网络结构化剪枝，通过剪枝算法确定每一层中要保留的滤波器，并将其余滤波器去除，然后在基于保留的滤波器构建的新的脉冲神经网络上进行微调训练恢复脉冲神经网络性能。衡量脉冲神经网络剪枝效果常用的指标是在数据集上的准确率和模型剪枝率，以单个卷积层为例其剪枝率计算方式：

$$\text{PruningRate} = 1 - \frac{I' \times O' \times H \times W}{I \times O \times H \times W}, \tag{1-1}$$

其中，I' 和 O' 分别为剪枝后的输入和输出通道数。

根据脉冲神经网络剪枝方法是否使用梯度信息，目前已有的脉冲神经网络

剪枝方法可以分为基于规则的方法和基于梯度的方法。基于规则的方法通常基于领域内专家经验或生物启发式地设计规则衡量神经元或连接的重要程度，此类方法依赖经验且需要不断尝试反馈修改，优点是清晰明了，可解释性强。基于梯度的方法通过增加学习参数，在训练的过程中，通过梯度的方式自动学得神经元或连接得重要程度。此类方法优化稀疏网络时训练容易波动，尤其是在脉冲神经网络中，基于梯度训练的方法受到时空计算图和代理梯度的影响，梯度较人工神经网络更加不稳定，易受到梯度爆炸或梯度消失的影响。

然而，目前很少有工作将 ANN-to-SNN 与脉冲神经网络剪枝相结合，该方法通过在人工神经网络上进行剪枝，然后将剪枝后的人工神经网络通过转换算法转换成剪枝的脉冲神经网络，从而规避上述问题。

1.3 本文工作

本文围绕如何实现高性能、低功耗的脉冲神经网络，递进式地设计了脉冲神经网络训练算法和脉冲神经网络剪枝算法。首先基于对转换误差的分析，提出了基于神经元阈值优化的 ANN-to-SNN 算法，在此基础上基于转换中的转换误差设计了脉冲神经网络剪枝算法。

1.3.1 基于神经元阈值优化的 ANN-to-SNN 算法

ANN-to-SNN 算法通过转换一个训练好的人工神经网络来获得脉冲神经网络，获得的脉冲神经网络的性能主要受到两个因素影响：1. 人工神经网络的性能；2. 因转换误差造成的准确度损失。在训练好的人工神经网络性能固定的情况下，降低转换误差起到关键作用。转换误差指的是人工神经网络和转换得到的脉冲神经网络对于相同的数据在计算上的差异性，包括网络内部数值计算的差异和网络输出结果上的差异，并且由于神经网络前向传播的特点，如果网络中某一层的数值计算没有对齐，那么该层的下一层计算结果会出现更大误差，所以网络中浅层的计算误差会随着前向传播被方法，使得网络输出结果出现极大偏离。所以，我们使用逐层对齐的方式，通过优化人工神经网络与转换后的脉冲神经网络的对应层的误差来对齐整个网络。

对于每一层内的对齐方式，我们通过逐一神经元对齐的来最小化该层的转

换误差。例如，对于一个卷积层，其输出特征图维度为 O×H×W，O 为通道数，H 为特征图的高，W 为特征图的宽，那么神经元个数也是 O×H×W，对人工神经网络和相同结构的脉冲神经网络对应位置的神经元的前向传播数值进行逐一对齐，通过这种方式学习脉冲神经网络中的参数。

1.3.2 基于转换误差的脉冲神经网络结构化剪枝算法

通过 ANN-to-SNN 获得的脉冲神经网络，由于使用脉冲信号作为信息传输载体，与使用浮点数计算的人工神经网络之间理论上存在不可完全消除的转换误差，而且由于网络内部参数分布造成的计算差异性，网络不同位置的转换误差也不同，这启发我们利用转换误差的位置差异来寻找与人工神经网络转换误差最小的脉冲神经网络。

本文采用最常用的结构化剪枝中的以滤波器作为剪枝最小粒度的剪枝方式，通过在数据集上统计不同位置滤波器前向推理时人工神经网络与脉冲神经网络的输出差异，将输出差异也就是转换误差较大的那些滤波器去除得到剪枝脉冲神经网络的结构。同时考虑到经转换得到的脉冲神经网络的性能表现还受到人工神经网络影响，所以在选择剪枝滤波器的过程中同时考虑保留对人工神经网络性能有关键作用的滤波器。所以，我们将脉冲神经网络剪枝建模成了一个有关滤波器转换误差和滤波器对性能影响程度的子集选择问题，即将神经网络中的所有滤波器看作一个集合，通过综合考虑这两个因素选择出最优的一个子集，这个子集构成的脉冲神经网络就是剪枝的脉冲神经网络。然后使用演化算法来求解这个子集选择问题来获得高性能、低消耗的脉冲神经网络。

1.3.3 文章结构

本文首先对脉冲神经网络的训练和剪枝做了介绍和分析。以实现高性能、低消耗的脉冲神经网络为目标，专注于 ANN-to-SNN 的训练方法，通过对该方法转换误差和性能的分析，提出了一种基于神经元阈值优化的 ANN-to-SNN 方法，然后在此基础上通过综合考虑转换误差和人工神经元性能设计了以 ANN-to-SNN 为基础的脉冲神经网络剪枝方法。

本文后续章节内容和组织方式如下：

第二章，对本文将用到的脉冲神经网络相关必要的背景知识做介绍，包括脉冲神经元模型和数据编码方式、脉冲神经网络训练方法相关研究和脉冲神经网络相关方法研究。并通过介绍现有方法，分析实现高性能、低消耗脉冲神经网络的当前困难。

第三章，通过对 ANN-to-SNN 方法中转换误差的分析提出了基于神经元阈值优化的 ANN-to-SNN 训练方法，介绍了算法的流程与求解细节，通过实验分析该方法的有效性。

第四章，介绍在转换方法基础上设计的脉冲神经网络剪枝方法，详细介绍算法流程与各个模块求解细节，通过实验分析、与其他方法对比验证该剪枝方式的可行性。

第五章，对本文工作进行总结，展望未来可能的研究方向。

第二章　相关工作

2.1　预备知识

2.1.1　LIF/IF 神经元模型

LIF/IF 神经元是两种常见的简化神经元模型，用于描述神经元的电活动和脉冲放电行为，他们在神经科学和计算神经科学领域得到广泛应用。由生物神经元启发和简化而设计，保留了生物神经元的基本生物学特征。LIF 神经元是一种基于积分和阈值触发机制的神经元模型：1. 整合，当收到外界的电流（current input）输入，神经元膜电位（membrane potential）根据输入电流进行积分；2. 激活，如果累加后的膜电位超过激活阈值（threshold），神经元会激活，向外释放一个脉冲（spike）；3. 重置，释放电压后，脉冲神经元的膜电位会重置（reset potential）。LIF 模型的特点是在膜电位积分过程中引入了漏电机制（leakage mechanism），即膜电位会随时间以一个固定的速率（leakage rate）向重置电位靠近。这种漏电机制反映了神经元膜的电导特性。IF 模型是 LIF 模型的一种特例，它忽略了漏电机制，即没有膜电位的漏电。在 IF 模型中，神经元膜电位只在达到阈值时发放脉冲，然后被重置为重置电位。LIF 和 IF 神经元模型的简化使得它们在计算和分析上更加可行，同时仍能捕捉到一些重要的神经元特性。这些模型被广泛应用于神经科学研究中，例如神经元网络的建模和分析、神经编码的研究以及脉冲信号处理等方面。在实际研究中，LIF/IF 模型常常被扩展和改进，以更好地逼近真实神经元的特性和行为。

在计算机科学领域，LIF/IF 神经元模型被广泛应用于神经网络和脉冲编码的研究，被用作构建计算模型的基础。在深度学习领域，为了实现脉冲神经元的规模模拟，通过离散时间步模拟的方式进行具体实现：时间被离散为固定的模拟时间步长 T，LIF/IF 神经元的电活动模拟通过在模拟时间步长中每个时刻 t

($t \in \{1, 2, ..., T\}$) 更新神经元的状态来进行。具体而言，以下是在深度学习中使用 LIF/IF 神经元模型的一般步骤：

初始化，对于每个神经元初始化器模电位为基础值（例如 0），并设置初始阈值等参数；

模拟时间步长，将模拟时间分为离散的模拟时间步长 T，在每个时刻 t 内，模拟神经元的电活动；

整合，根据输入信号、神经元的权重和膜电位衰减，更新神经元的膜电位：

$$\boldsymbol{I}^l(t) = \lambda \boldsymbol{V}^l(t-1) + \boldsymbol{W}_s^l \boldsymbol{s}^{l-1}(t), \tag{2-1}$$

其中，$\boldsymbol{W}_s^l$ 表示神经元的连接权重，$\boldsymbol{I}^l(t)$ 表示整合上一时刻保留的膜电位与当前时刻上一层的输入的结果，$\lambda \in (0, 1)$ 为泄露系数，λ 小于 1 表示膜电位会随时间衰减，λ 为 1 表示没有衰减，使用前一机制的为 LIF 神经元，使用后一机制的为 IF 神经元。

激活，检查神经元的膜电位是否超过设定的阈值 V_{th}^l。如果膜电位超过阈值，则神经元发放一个脉冲，如果小于阈值则不释放：

$$s^l(t) = \mathbb{I}_{I^l(t) \geqslant V_{th}^l} V_{th}^l, \tag{2-2}$$

其中，$\mathbb{I}$ 为指示函数。

重置，在激活之后，脉冲神经元的膜电位进行重置，如果激活过程中没有释放电压，那么重置过程中膜电位为该时刻的整合电压，如果激活过程中释放了电压，那么重置过程中膜电位在此基础上减去释放的电压：

$$V^l(t) = I^l(t) - s^l(t). \tag{2-3}$$

重复整合、激活、重置，直到模拟时间结束。

通过以上步骤，可以进行深度学习网络中的脉冲神经元模拟。这种模拟方法使得 LIF/IF 神经元能够与传统的深度学习框架兼容，同时能够捕捉脉冲编码的特性，例如时间编码和事件相关性等。

2.1.2 脉冲数据编码方法

脉冲编码是一种将信息转换为脉冲信号的方法，它在神经科学和信息处理领域中被广泛使用。脉冲编码方式可以用于传输和表示信息，其中脉冲的时间、频率或幅度编码了原始数据的特征。以下是几种常见的脉冲数据编码方式：

时间编码，在时间编码中，信息被转换为脉冲的发放时间。具体而言，信号的强度或数值大小被映射到脉冲的发放时间。较大的数值通常对应于更早的发放时间，而较小的数值对应于较晚的发放时间。时间编码可以提供高精度和高分辨率的表示，尤其适用于事件相关任务和时间敏感任务。

幅度编码：在幅度编码中，信息被转换为脉冲的幅度或振幅。信号的强度或数值大小被映射到脉冲的幅度，通常是通过改变脉冲的振幅来表示信息的大小。幅度编码可以提供一种连续的表示方式，适用于模拟信号的传输和表示。

频率编码，在频率编码中，信息被转换为脉冲的发放频率。较大的数值通常对应于更高的发放频率，而较小的数值对应于较低的发放频率。频率编码可以提供一种累积信息的表示方式，其中脉冲的数量对应于信息的强度或数值大小。频率编码可以用于传输和表示连续变量。

脉冲编码方式的选择取决于具体的应用和任务需求。不同的编码方式具有不同的特性和适用性。例如，时间编码适用于事件相关任务和时间敏感任务，而频率编码适用于累积信息的表示，幅度编码适用于连续变量的传输和表示。在具体任务中，需要根据具体的情况选择最适合的编码方式。

本文使用频率编码中的泊松编码方法，它基于泊松过程的概念，利用脉冲的间隔时间来编码信息。在泊松编码中，脉冲的发放时间间隔服从泊松分布，该分布具有随机性和统计独立性。泊松编码的基本原理是将信息转换为脉冲的平均发放率，即单位时间内脉冲发放的平均数量。较高的发放率对应于信息的强度或数值较大，而较低的发放率对应于信息的强度或数值较小。泊松编码产生的脉冲信号，将输入数据编码为发放次数分布符合泊松过程的脉冲序列，在互补相交的时间区间里出现脉冲的个数是相互独立的，且在任意一个区间中，出现脉冲的个数与区间的起点无关，只与区间长度有关。泊松编码具有以下特点和优势：1. 稀疏性，由于泊松过程的随机性，脉冲的发放时间间隔通常是不规则和稀疏的，这种稀疏性使得泊松编码在处理稀疏输入和事件相关任务时具有优势；

2. 高效性，泊松编码可以用较少的脉冲来表示信息，从而节省了传输和处理的资源，这对于神经系统的能耗和计算效率是有益的；3. 鲁棒性，泊松编码对于噪声和干扰具有一定的鲁棒性，由于脉冲的发放时间是随机分布的，泊松编码可以一定程度上抵抗噪声的影响；4. 可逆性，泊松编码是可逆的，即可以从脉冲序列中重构原始信息，这使得泊松编码在信息传输和存储中具有重要的应用价值。

具体而言，对于输入数据先归一化到 $(0,1)$ 的范围内，以单个数值 x 的编码为例，如果需要将其编码为长度为 T 的脉冲序列，那么在每一时刻，x 产生脉冲的概率与 x 的值成正比：

$$P(\text{spike}=1)=x, x\sim \text{U}(0,1), \tag{2-4}$$

其中，$U(a,b)$ 表示 (a,b) 上的均匀分布。不产生脉冲的概率与 1-x 的值成正比：

$$P(\text{spike}=0)=1-x, x\sim \text{U}(0,1), \tag{2-5}$$

通过这种方式就得到了长度为 T 的脉冲序列。例如，模拟时间步 T 为 10，x 为 0.3，得到的脉冲序列可能是 `0010100100`. 对于图片的泊松编码，以一张维度为 3×H×W 的图片为例，每个像素值都通过上述方式编码，就得到了维度为 T×3×H×W 的时空脉冲数据。

2.1.3 卷积脉冲神经网络

卷积脉冲神经网络（Convolutional Spiking Neural Network，CSNN）是一种基于脉冲编码的神经网络模型，主要应用于处理时空数据，如图像、视频等。卷积脉冲神经网络结合了卷积神经网络和脉冲神经网络的特点和优势。

与普通的卷积神经网络类似，卷积脉冲神经网络由脉冲神经元按照层级结构连接而成，每一层内由代表神经元突触的卷积核和神经元本体的激活层构成。其中，卷积核与普通卷积神经网络的卷积核一样，激活层使用 LIF/IF 神经元电信号处理机制。如示意图2-1所示。

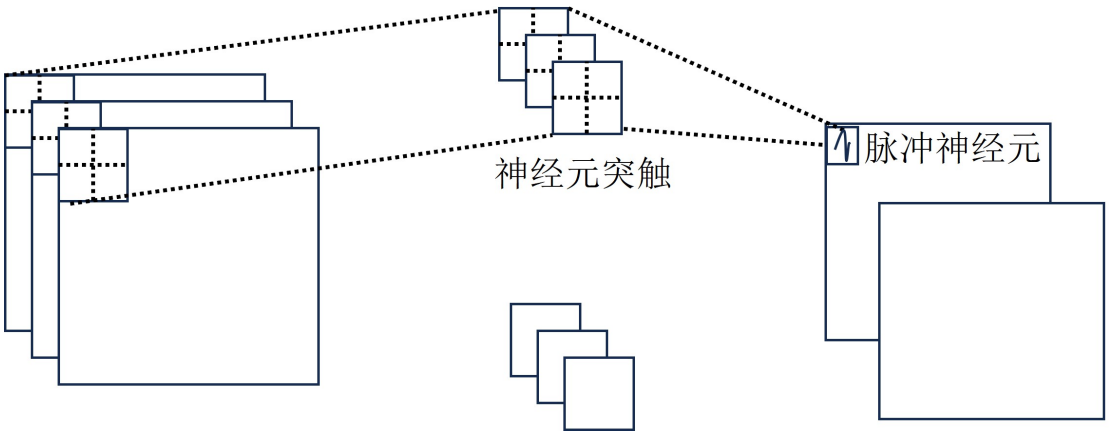

图 2-1 卷积脉冲神经网络单层示意图

2.2 脉冲神经网络训练方法

近年来，脉冲神经网络作为一种模拟生物神经系统的神经网络模型，在神经计算和人工智能领域都引起了广泛的研究兴趣。与传统的人工神经网络不同，脉冲神经网络通过模拟神经元的脉冲放电行为，以时间编码的方式处理和传递信息。然而，由于其离散和非线性的特性，脉冲神经网络的训练方法相较于人工神经网络更具挑战性。研究人员一直致力于开发有效的训练方法来训练脉冲神经网络。脉冲神经网络的训练方法旨在通过调整神经元之间的连接权重和神经元参数，使网络能够学习和适应特定的任务。然而，由于脉冲信号的离散性和不可导性，传统的基于梯度的训练算法，如反向传播算法（BackPropagation），无法直接应用于脉冲神经网络的训练。为了解决这一挑战，研究人员提出了许多方法来训练脉冲神经网络。

2.2.1 时空反向传播算法

时空反向传播算法（Spatio-temporal BackProagation）是一种训练脉冲神经网络的监督学习方法。传统的反向传播算法在人工神经网络中被广泛应用，通过计算网络输出与期望输出之间的误差，然后反向传播误差信号来调整网络中的权重。然而，由于脉冲神经网络的离散和时间敏感的特性，传统的反向传播算法无法直接应用于脉冲神经网路的训练。时空反向传播算法被提出来克服这一挑战，它通过考虑时间维度和空间维度的误差传播和权重调整，使得脉冲神经网络能够学习和适应时空信息。时空反向传播算法的核心思想是将时间视为额外

的维度，并在网络的每个时间步长上进行误差反向传播和权重更新。时空反向传播算法中的误差传播过程类似于传统的反向传播算法，但考虑到了时间因素。首先，根据网络的输出和期望输出之间的差异，计算出误差信号。然后，这个误差信号在时间维度上进行反向传播，从网络的输出层向输入层传播。在每个时间步长上，误差信号被乘以连接权重，并传递到前一时间步的神经元。这样，误差信号会随着时间的推移逐渐传播回网络的初始时间步长。在时空反向传播算法中，权重的调整也是在时空维度上进行的。根据误差信号和输入脉冲的时间信息，计算出权重的梯度，并使用梯度下降等优化算法来更新连接权重。这样，脉冲神经网络可以通过时空反向传播算法在时空维度上进行训练，从而学习和适应输入数据的时序模式。脉冲神经网络的前向过程如图2-2所示，圆形表示算子，箭头表示数据流，其中环表示时间维度上使用上一时刻膜电位计算当前时刻膜电位的过程，类似于循环神经网络中的自连接，该过程的前向传播计算图如图2-3所示，反向传播计算图如图2-4所示。由于其反向传播在时间和空间维度上进行，因此需要存储脉冲神经网络在前向传播期间的中间状态，在实际训练中，时空时空反向传播对现存的需求与模拟时间步的长度呈线性增长关系，这限制了时空反向传播算法的规模和效率。为了解决这个问题，研究人员提出了一些优化和近似方法，例如[21]只存储激活神经元的中间变量来降低整体的现存需求和提高计算速度。

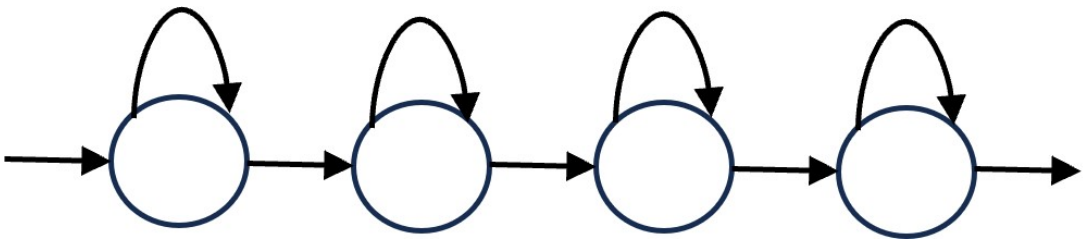

图 2-2 脉冲神经网络前向传播示意图

在时空反向传播算法中，为了计算梯度并进行权重更新，需要将误差信号在时间维度和空间维度上反向传播。然而，由于脉冲神经元的离散性和非线性，在离散时间点上没有可用梯度信息。因此，为了解决这个问题，许多研究引入了代理梯度[22-24]。代理梯度是一种近似的梯度计算方法。具体来说，代理梯度通过对脉冲神经元的脉冲信号进行平滑处理，将离散的脉冲信号转换为连续的可计算梯度的信号。这样，在代理梯度的引导下，误差信号可以在时间维度上进行

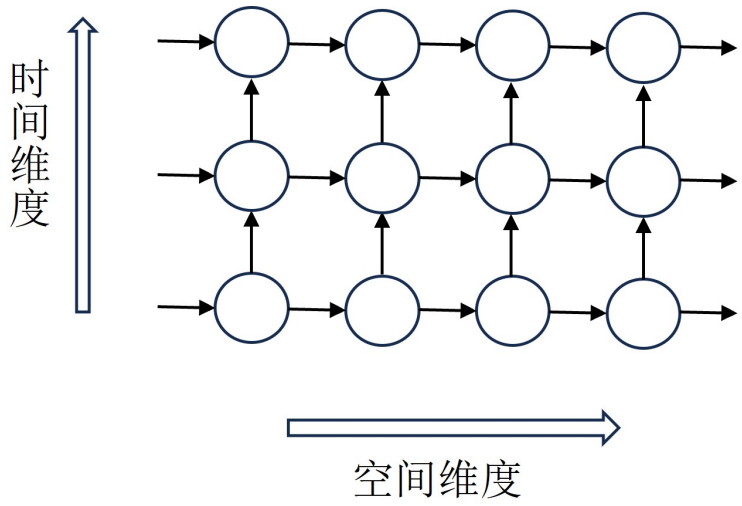

图 2-3 时空反向传播算法前向传播示意图

反向传播，从而实现对脉冲神经网络的训练。然而，代理梯度仅是对真实梯度的近似，并且可能引入了一定的误差。这是因为代理梯度是通过平滑脉冲信号得到的，而平滑过程可能丢失了一些时序信息和非线性特性。因此，代理梯度可能导致训练过程中的信息丢失和模型性能下降。

时空反向传播算法在处理时空信息方面具有一定的优势，能够利用脉冲编码和时间编码的特性进行有效的训练。然而，时空反向传播算法仍然面临一些挑战，如梯度消失和计算复杂性等问题。研究人员仍在不断改进和发展时空反向传播算法，以提高脉冲神经网络的训练性能和应用能力。

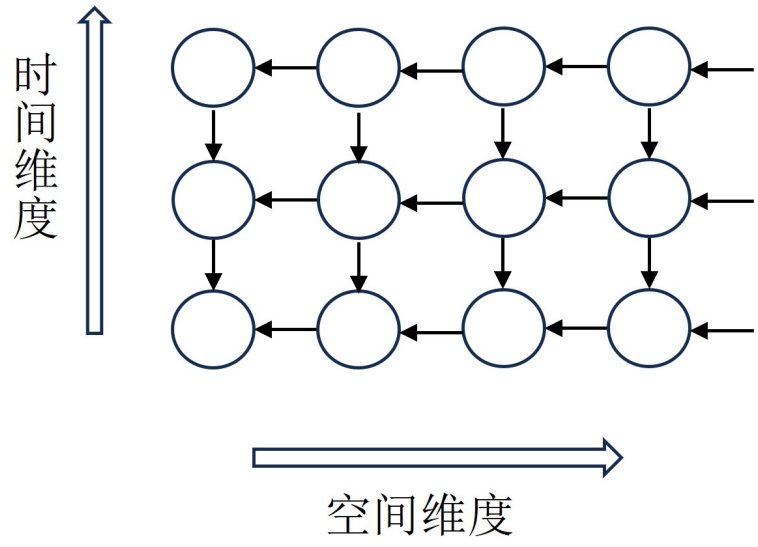

图 2-4 时空反向传播算法反向传播示意图

2.2.2 突触时序可塑性算法

突触时序可塑性[25-27]（Spike-Timing-Dependent Plasticity）是一种突触可塑性规则，最早在论文[25]中提出。该论文描述了时序依赖的突触可塑性现象，基于对海马神经元的实验观察，探索了突触强度和突触时序对突触可塑性的影响。如图2-5所示，横轴表示突触前神经元与突触后神经元释放脉冲相对时间差，竖轴表示兴奋性突触后电流（Excitatory Postsynaptic Current，EPSC）变化幅度（EPSC的测量和分析是神经科学研究中常用实验技术，其可以用于衡量突触连接强度）。具体来讲，当突触后脉冲在突触前脉冲之后到达时，突触权重增加；反之，当突

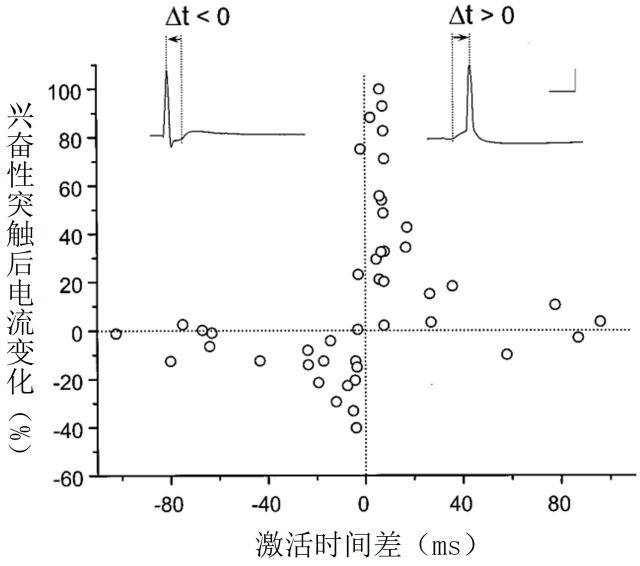

图 2-5 突触时序可塑性示意图[25]

触后脉冲在突触前脉冲之前到达时，突触权重减小。一般性的突触时序可塑性规则可以形式化表示为：

$$\Delta w = \begin{cases} A_{pre} \cdot e^{-\frac{\Delta t}{\tau_{pre}}}, \text{如果} \Delta t > 0 \\ -A_{post} \cdot e^{-\frac{\Delta t}{\tau_{post}}}, \text{其他} \end{cases}, \tag{2-6}$$

其中，Δw 表述突触权重的变化，Δt 表示突触后脉冲与突触前脉冲之间的时间差异，A_{pre} 和 A_{post} 分别表示突触前后脉冲引起的突触权重变化幅度，τ_{pre} 和 τ_{post} 表示突触前后脉冲引起的突触权重变化的时间常数。突触时序可塑性机制使得突触权重可以根据突触前后脉冲的时间顺序进行调整，从而实现对时间相关的

输入模式的学习。在深度学习当中作为一种无监督学习方法已得到初步验证，论文中提出的基于突触时序可塑性的方法[28]在 MNIST 数据集[16]上使用三层卷积网络结构取得了 98.4% 的准确率。

2.2.3 ANN-to-SNN

ANN-to-SNN 是目前常用的一种间接训练脉冲神经网络的有效方法，由于其可以利用人工神经网络训练结果所以在深度神经网络中取得了较好的结果[29-36]，其通过将一个训练好的人工神经网络经过特殊处理和参数迁移转换成对应的脉冲神经网络，通过使用脉冲神经网络的激活频率拟合经典人工神经网络的输出使得脉冲神经网络可以取得与人工神经网络一致的数据拟合能力，其示意图如图2-6所示。ANN-to-SNN 的主要转换步骤包括：

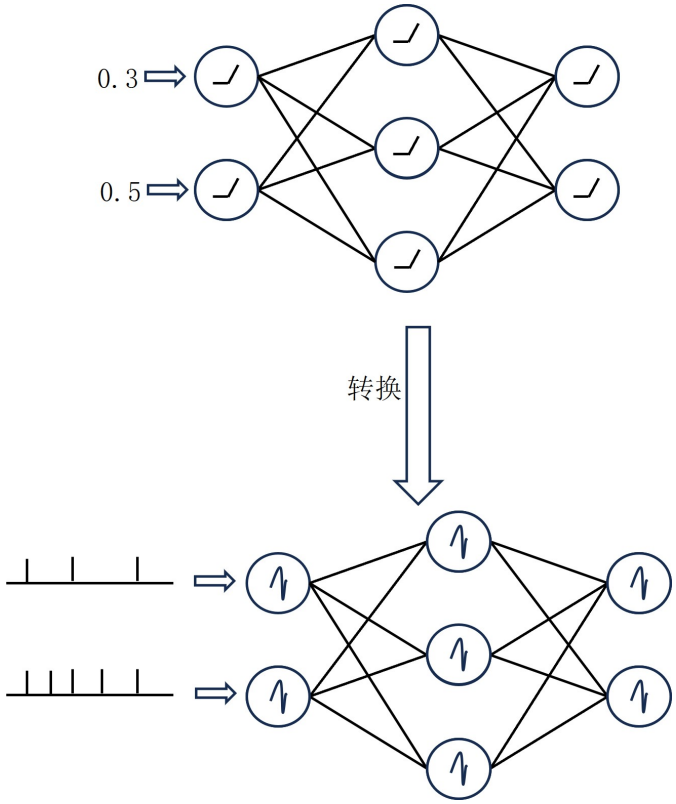

图 2-6 ANN-to-SNN 示意图

1. 时间信息编码，脉冲神经网络基于脉冲的时序操作，其中脉冲的发放时间和频率编码了脉冲信息，在转换过程中，将人工神经网络的发放频率或激活模式，转换为脉冲神经网络的脉冲序列和脉冲激活。

2. 权重映射，将人工神经网络中学习道德权重映射到脉冲神经网络的突触

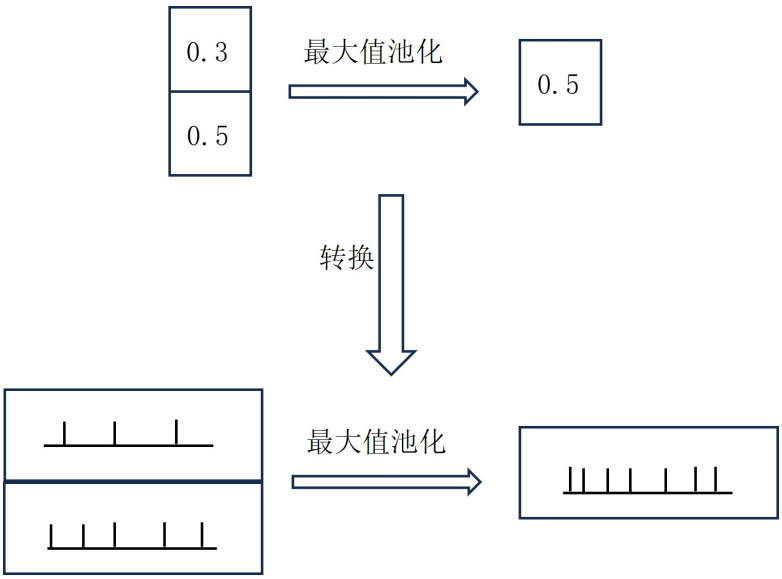

图 2-7 脉冲最大值池化层示意图

连接上，通过这个映射确保脉冲神经网络复制预训练的人工神经网络的功能行为。可以使用比例缩放或基于脉冲的权重归一化[32]方法来实行这个映射。

3. 脉冲神经网络参数求解，在权重映射后，通过对转换误差和脉冲神经网络性能的优化求解或微调网络中的参数，使脉冲神经网络高度拟合训练数据。

对于使用卷积模块的人工神经网络的转换与该流程基本一致，在时间信息编码阶段，常使用泊松编码方法编码图像或其他信息；在权重映射阶段，对卷积神经网络中的各个模块分别进行权重映射，当前卷积神经网络通常由卷积功能模块连接而成，一个卷积模块一般来讲包含卷积层、批归一化层、激活层和池化层，转换算法要对包含权重参数的卷积层和批归一化层进行参数映射，以确保脉冲神经网络与人工神经网络输出一致，使脉冲神经网络继承人工神经网络的特征提取能力和拟合数据的准确性。这类方法根据是否由理论支撑，分为启发式类方法和形式化类方法。

在研究初期，人们通过启发式的方法将经典人工神经网络的模块转换成脉冲神经网络的模块，通过对经典人工神经网络中的模块进行特定处理来使得脉冲神经网络在计算结果上尽量与人工神经网络接近。这类方法在[29]中被首次提出，论文中提出了一种将深度卷积神经网络转换为脉冲神经网络的方法，通过调整人工神经网络架构以适应脉冲神经网络的要求，然后训练调整后的人工神经网络，并将学习到的网络权重应用于由人工神经网络派生出的脉冲神经网络架构。该方法中使用的人工神经网络一个功能模块由三部分组成：卷积层、Tanh

激活层、最大值池化层。对于卷积层，直接将人工神经网络的参数迁移到脉冲神经网络对应层中；对于 Tanh 激活层，由于脉冲神经网络激活频率大于零，但激活函数 tanh(·) 的输出范围为 (-1,1)，所以使用 ReLU[37]激活层替换 Tanh 激活层来保证激活值非负，这种方式在近些年的 ANN-to-SNN 方法中被普遍采用并理论证明了该操作的合理性；对于最大值池化层，如果脉冲神经网络的池化层在每个时间步都进行最大池化的操作，那么在完成 T 个时间步后得到的脉冲序列脉冲频率往往很高并不能拟合经典人工神经网络中对应最大池化层的输出，如图2-7所示，在人工神经网络中，对于 0.3 和 0.5 的最大池化结果是 0.5，但是在脉冲神经网络中，对输入频率为 0.3 和 0.5 的两个脉冲序列在每个时刻都进行最大池化最终得到的结果脉冲频率是 0.7。所以，该论文在人工神经网络中使用平均池化层而不是最大池化层，使得空间线性降采样更容易地转换到脉冲域地空间降采样。该方法在 CIFAR-10[38]数据集上，使用三卷积层加一层全连接的结构，在模拟时间步长 T 设置为 100 的情况下，实现了 77.43% 的准确率，虽然准确率与最近的工作相比没有竞争力，但初步验证了 ANN-to-SNN 思路的可行性，为后续的工作提供了一定的基础和方向。

近几年，对 ANN-to-SNN 问题的理论表述在[34-35,39]被提出，这些工作形式化表述了基于 ReLU 激活的人工神经网络和基于 IF 神经元激活的脉冲神经网络之间的转换误差，并通过分析转换误差设计了合理的准换方法，极大的降低了人工神经网络与脉冲神经网络之间的能力间隔，相对于之前方法极大减小了模拟时间步长，并在常用的卷积神经网络结构和图像分类数据集上取得了与人工神经网络媲美的性能。例如，论文[36]中分析了由于脉冲释放不均匀造成的转换误差，提出了一种具有记忆功能的有符号神经元，通过实现脉冲的均匀发放来降低转换误差。论文[35]通过形式化转换误差与脉冲神经元激活阈值的关系，然后将脉冲神经元做为人工神经网络的训练参数，通过使用反向传播训练人工神经网络获得脉冲神经网络的权重和激活阈值。

2.3 脉冲神经网络剪枝方法

脉冲神经网络剪枝对减小脉冲神经网络规模、实现低功耗的脉冲神经网至关重要。神经网络剪枝是一种常用的模型压缩技术，旨在减少神经网络中冗余

的连接或神经元，从而实现模型的精简和加速。剪枝可以显著减少神经网络的计算和存储开销，同时保持甚至提高模型的性能。

现有大量关于人工神经网络剪枝的研究，根据剪枝过程中是否有一定的结构约束例如卷积神经网路的通道、滤波器或层，剪枝算法可以分为结构化剪枝[40-43]和非结构化剪枝[44-46]。前者在剪枝过程中保持结构约束，以此结构约束为最小剪枝单位；后者在剪枝过程中没有结构约束，以一个神经元或一个连接作为最下剪枝单位。根据剪枝算法在剪枝过程中是否进行多次迭代可以将剪枝方法划分为迭代剪枝[45-46]与非迭代剪枝[40,47-49]。非迭代剪枝在剪枝过程中只进行一次权重重要性评估和一次剪枝操作，这种方法相对快速，但可能会导致剪枝结果不够精确，性能和剪枝比例不容易达到要求；迭代剪枝通过多次剪枝和重训练的迭代循环来逐步优化剪枝结果，在每次迭代中，可以重新评估权重的重要性，并进行剪枝操作，剪枝后的模型可以进行重训练以恢复性能或进一步优化，通过迭代方式可以逐步精细调整剪枝比例和保留的连接或参数以达到更好的性能和剪枝效果。

同经典人工神经网络一样，随着模型不断增大，深度脉冲神经网络的参数量和计算量也逐渐增加，其中也常包含大量参数冗余。通过恰当地剪枝算法在尽量保持模型性能的同时减小计算量和对硬件资源的要求是自然而然的。尤其是对于脉冲神经网络，作为一种具有低功耗潜力的计算模型，对其剪枝可以在模型规模、计算量、所需计算资源、计算能源消耗多方面受益。

对脉冲神经网络的剪枝许多方面可以参考已有的人工神经网络剪枝思路，但由于脉冲神经网络计算的离散、不可导特点，脉冲神经网络的剪枝存在特殊性，许多人工神经网络剪枝的成功经验难以直接简单复制。针对脉冲神经网络自身时空计算、脉冲激活等特点，现有的脉冲神经网络剪枝方法在网络结构规模[20,50-51]、模拟时间步长[52-53]、脉冲激活数量和脉冲激活频率等方面进行针对性的剪枝。根据剪枝算法使用规则或梯度判断权重的重要程度，目前已有的脉冲神经网络剪枝方法可以分为基于规则的方法和基于梯度的方法。

2.3.1 基于规则的脉冲神经网络剪枝方法

基于规则的方法较为直接，基于专家知识、脉冲神经网络计算特点或数学工具制定显式的权重判断准则，然后根据重要性进行剪枝。例如，论文[54]提出了

一种名为 ESL-SNNs 的高效进化结构学习框架，用于从头开始训练具有稀疏结构的脉冲神经网络。这个框架受到人脑中神经网络重连过程的启发，其中神经突触连接在大脑发育期间保持相对稀疏[55-56]。ESL-SNNs 通过动态地剪枝和再生突触连接来实现结构的稀疏性，同时探索所有可能的参数以寻找最优的稀疏连接性。论文[57]首先使用主成分分析（Principal Component Analysis）方法来确定每一层内的重要通道，如图2-8所示，图片右侧灰橙绿的网格表输出特征图各个通道拉直后，每个输出在时间维度上的均值，通过对之使用主成分分析确定重要的特征图和对应的滤波器。该方法还通过在训练时逐渐减小模拟时间步的方式进一步在时间维度上剪枝，以此实现空间、时间上的同时缩减。论文[58]中，

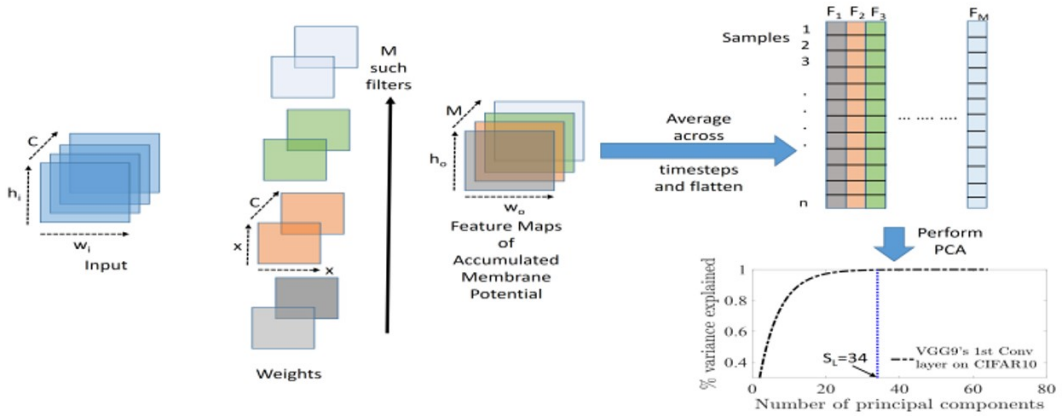

图 2-8 基于主成分分析的脉冲神经网络剪枝方法[57]

作者通过探索脉冲神经网络的神经动态活动和脉冲释放强度，在训练过程中随时间和神经元状态调整剪枝阈值实现网络动态剪枝。这些基于规则的方法为脉冲神经网络的剪枝提供了一些实用的技术和框架。通过结合专家知识、脉冲神经网络的特性和数学工具，这些方法可以指导剪枝操作并优化网络结构，从而提高脉冲神经网络的性能和效率。另一方面，对专家知识的要求提高了算法设计难度，实际应用中往往需要大量实验与试错以适应不同的模型结构或任务。

2.3.2 基于梯度的脉冲神经网络剪枝方法

基于梯度的剪枝方法在人工神经网络剪枝中是一种常用的方法，近年来也被用于脉冲神经网络剪枝。这种方法通过分析梯度信息来评估神经元或连接的重要性，并根据重要性进行剪枝操作。在脉冲神经网络当中，由于脉冲神经元

计算不可导，应用此类剪枝方法通常与基于代理梯度的时空反向传播算法一同使用。例如，Grad R[59]受到人工神经网络剪枝中的 Deep R[60]方法启发，通过在脉冲神经网络训练过程中利用梯度信息评估脉冲神经网络中神经元和连接的重要性，并通过梯度来剪枝或重连网络中的连接来保持网络的连通性以实现非常稀疏的网路结构。文章[61]将脉冲神经网络剪枝问题建模成一个有关权重量化和神经元连接剪枝的约束优化问题，并结合交替方向乘子法（alternating direction method of multipliers）和时空反向传播算法在训练过程中求解该问题，并进一步通过脉冲活动正则化鼓励脉冲神经元的稀疏活动来达到在网络结构、脉冲数量上的综合压缩。

目前基于梯度信息的脉冲神经网络剪枝方法已经在多层全连接结构和多层卷积结构的脉冲神经网络上取得非常稀疏的剪枝效果同时保证了模型性能，这对于探索稀疏的脉冲神经网络模型和实际应用都有重要意义。但是目前此类方法大多为非结构化剪枝方法，常常在神经元和突触连接级别计算梯度和进行剪枝，受制于硬件批量化、结构化计算的特点，非结构化剪枝获得的脉冲神经网络在实际部署中往往不能取得与极为稀疏的网络结构相对应的模型存储和计算量的减少。另一方面，基于梯度信息进行脉冲神经网络剪枝需要用到代理梯度和时空反向传播算法，根据人工神经网络基于梯度信息剪枝的经验，稀疏、动态的网络结构往往极易受到梯度变化的影响而使训练不稳定或大幅降低网络性能，对于使用代理梯度进行反向传播的稀疏脉冲神经网络结构更是如此，目前这类方法尚未在复杂数据集和深度网络结构上取得令人满意的性能，这是一个主要原因。

2.4 本章小结

本章第一小节介绍了脉冲神经网络的背景知识，包括脉冲神经元的计算方式、脉冲数据编码方式和脉冲卷积神经网络结构，通过介绍这些内容说明了基于卷积结构的脉冲神经网络的信息传播计算机制，以及其计算离散、不可导的特点。

本章第二节，通过对现有脉冲神经网络训练方法的介绍，包括时空反向传播算法、突触时间可塑性方法、ANN-to-SNN 方法，说明了该问题的研究现状、特

点和难点。通过对 ANN-to-SNN 方法相关理论和网络中各个计算模块脉冲化方法的介绍分析，说明了已有方法对该问题的研究现状和关键点，说明了设计基于神经元进行阈值优化的 ANN-to-SNN 方法的原因。

本章第三小节，介绍了以脉冲神经网络压缩为目的的脉冲神经网络剪枝方法，介绍了现有方法在模型规模、脉冲激活数量、模拟时间步长等维度进行剪枝的方法，考虑到脉冲神经网络计算的离散、不可导特点，以是否使用梯度信息协助剪枝为依据，将现有方法分为基于规则的脉冲神经网络剪枝方法和基于梯度的脉冲神经网络剪枝方法，举例介绍了两类方法的设计思想和具体操作，说明了当前脉冲神经网络剪枝方法的现状和主要限制因素。

脉冲神经网络因其生物仿生性和计算低功耗特点而具有成为下一代人工神经网络的潜力。随着人工智能中边缘计算、大模型的发展，日益增长的计算量和能源消耗备受关注，对新型计算模型的需求和研究也日益增多。脉冲神经网络近年来受到越来越多的关注。受制于脉冲神经网络计算不可导的特点，脉冲神经网络的训练方法和剪枝方法仍是当前的研究热点和难点。本文将以实现高性能、低功耗的脉冲神经网络为目标，基于对 ANN-to-SNN 方法的分析，提出基于神经元阈值优化的 ANN-to-SNN 方法和基于 ANN-to-SNN 转换误差的脉冲神经网络剪枝方法。

第三章　基于神经元阈值优化的脉冲神经网络训练方法

本章通过对 ANN-to-SNN 转换误差的分析，提出了一种基于神经元阈值优化的脉冲神经网络训练方法。将卷积神经网络参数迁移到对应的脉冲神经网络中，然后根据转换误差最小化确定脉冲神经网络中的参数以实现高新能的脉冲神经网络。

3.1　问题分析

3.1.1　神经元的转换误差

此类方法试图通过脉冲神经网络中的平均脉冲激活值拟合人工神经网络中的激活值。脉冲神经元的平均脉冲激活值定义为：

$$\overline{s}(t)=\frac{\sum_{t=1}^{t}s(t)}{T}, \tag{3-1}$$

其中，T 表示脉冲神经网络的模拟时间步长，$s(t)\in\{0,V_{th}\}$ 表示神经元在 t 时刻释放的脉冲值，V_{th} 为该神经元的激活阈值。

在人工神经网络当中，对于单个神经元来讲，如果其位于网络的第 l 层，其前向传播计算过程可以被形式化为：

$$a^{l}=h(z^{l})=h(\boldsymbol{W}_{a}^{l}\boldsymbol{a}^{l-1}), \tag{3-2}$$

其中，a^{l} 表示该层的激活值，$\boldsymbol{W}_{a}^{l}$ 表示突触的权重向量，$\boldsymbol{a}^{l-1}$ 表示上一层神经元的输出向量，z^{l} 表示输入的加权和，$h(\cdot)$ 表示 ReLU 激活函数。

相应的，对于使用 IF 神经元激活的脉冲神经网络，模拟时间步长设为 T，与人工神经元对应位置的脉冲神经元初始膜电位设为 0，激活阈值为 V_{th}^{l}，输出为 $s^{l}(t) \in \{0, V_{th}^{1}\}$。那么在任意时刻 $t \in \{1, 2, ..., T\}$，其神经元整合该时刻输入后的瞬时膜电位为：

$$\boldsymbol{I}^{l}(t) = \boldsymbol{V}^{l}(t-1) + \boldsymbol{W}_{s}^{l}\boldsymbol{s}^{l-1}(t), \tag{3-3}$$

其中，$\boldsymbol{V}^{l}(t-1)$ 表示 t-1 时刻激活后保留的膜电位，$\boldsymbol{W}_{s}^{l}$ 表示脉冲神经元连接突触的权重，$\boldsymbol{s}^{l-1}(t)$ 表示当前时刻上一层脉冲神经元释放的脉冲信号。当瞬时膜电位大于等于设定好的激活阈值 V_{th}^{l} 时，该神经元会产生一个值为 V_{th}^{l} 的脉冲输出，反之不释放脉冲：

$$s^{l}(t) = \begin{cases} V_{th}^{l}, & \text{如果} \quad \boldsymbol{I}^{l}(t) \geqslant V_{th}^{l} \\ 0, & \text{其他} \end{cases}, \tag{3-4}$$

激活后重置脉冲神经元膜电位，该过程表示为：

$$\boldsymbol{V}^{l}(t) = \boldsymbol{I}^{l}(t) - s^{l}(t). \tag{3-5}$$

为得到脉冲神经元的平均脉冲激活值与人工神经元的激活值之间的关系，首先将公式3-3带入公式3-5：

$$\boldsymbol{V}^{l}(t) = \boldsymbol{V}^{l}(t-1) + \boldsymbol{W}_{s}^{l}\boldsymbol{s}^{l-1}(t) - s^{l}(t), \tag{3-6}$$

然后对该式从时刻 1 到 T 累加脉冲计算过程（初始化膜电位 $\boldsymbol{V}^{l}(0)$ 默认设置为 0）：

$$\boldsymbol{V}^{l}(T) = \boldsymbol{W}_{s}^{l}\sum_{t=1}^{t}\boldsymbol{s}^{l-1}(t) - \sum_{t=1}^{t}s^{l}(t), \tag{3-7}$$

通过对上式除以模拟时间步长 T 得到该脉冲神经元的平均脉冲激活值 $\overline{s}^{l}(t)$ 与前一层释放的平均脉冲激活向量 $\overline{\boldsymbol{s}}^{l-1}(t)$ 之间的关系：

$$\overline{s}^{l}(t) = \boldsymbol{W}_{s}^{l}\overline{\boldsymbol{s}}^{l-1}(t) - \frac{\boldsymbol{V}^{l}(T)}{T}. \tag{3-8}$$

设该神经元释放脉冲的次数为 $m \in \{0, 1, 2, ..., T\}$，那么有：

$$mV_{th}^{l} = \sum_{t=1}^{T} s^{l}(t), \tag{3-9}$$

由公式3-1、公式3-8和公式3-9得到：

$$m = \frac{T}{V_{th}^{l}} \overline{s}^{l}(t) = \frac{T}{V_{th}^{l}} \boldsymbol{W}_{s}^{l} \overline{\boldsymbol{s}}^{l-1}(t) - \frac{V^{l}(T)}{V_{th}^{l}}, \tag{3-10}$$

假设 $V^{l}(T) \in [0, V_{th}^{l})$，那么有：

$$\frac{T}{V_{th}^{l}} \boldsymbol{W}_{s}^{l} \overline{\boldsymbol{s}}^{l-1}(t) - 1 < m \leqslant \frac{T}{V_{th}^{l}} \boldsymbol{W}_{s}^{l} \overline{\boldsymbol{s}}^{l-1}(t), \tag{3-11}$$

可简单表示为：

$$m = clip(\lfloor \frac{T}{V_{th}^{l}} \boldsymbol{W}_{s}^{l} \overline{\boldsymbol{s}}^{l-1}(t) \rfloor, 0, T), \tag{3-12}$$

其中，$clip(x, a, b)$ 表示截取函数，保留 x 在 $[a, b]$ 的部分，$\lfloor \cdot \rfloor$ 表示向下取整。该式代入公式3-10得到平均脉冲激活值在前向传播时的计算方式：

$$\overline{s}^{l}(t) = \frac{V_{th}^{l}}{T} clip(\lfloor \frac{T}{V_{th}^{l}} \boldsymbol{W}_{s}^{l} \overline{\boldsymbol{s}}^{l-1}(t) \rfloor, 0, T), \tag{3-13}$$

由此式可知，如果将人工神经元的权重复制给对应的脉冲神经元 $\boldsymbol{W}_{s}^{l} = \boldsymbol{W}_{a}^{l}$，令脉冲神经元的输入脉冲序列的平均激活值等于人工神经元的输入数值 $\overline{\boldsymbol{s}}^{l-1}(t) = \boldsymbol{a}^{l-1}$，那么该脉冲神经元的平均激活值与人工神经元的激活值有如下关系：

$$\overline{s}^{l}(t) = \frac{V_{th}^{l}}{T} clip(\lfloor \frac{T}{V_{th}^{l}} \boldsymbol{W}_{a}^{l} \boldsymbol{a}^{l-1} \rfloor, 0, T). \tag{3-14}$$

对比人工神经元的激活过程：

$$a^{l} = h(\boldsymbol{W}_{a}^{l} \boldsymbol{a}^{l-1}), \tag{3-15}$$

示意图如图3-1所示，图中直线表示人工神经元对不同加权输入的输出值，折线

表示脉冲神经元对不同加权输入脉冲序列的平均脉冲激活值，折线每一段高度为 $\frac{V_{th}}{T}$，总段数为 T。由于脉冲神经元脉冲信号的离散性，通过参数迁移和输入对齐这种方式进行转换得到的脉冲神经元和人工神经元之间仍存在转换误差。图中直线与折线在竖直方向的差异直接体现了转换误差，该误差分为两部分，阶梯状部分为脉冲信号离散性造成的误差，称为取整误差；右侧开口部分为脉冲激活频率存在最大值与 ReLU 激活的实数值没有上界造成的误差，称为截取误差。容易发现，不同输入对应的转换误差不同，模拟时间步 T 越大，对应阶梯部分段数越多，取整误差越小，反之越大；激活阈值 V_th 越大，截取误差越小但是取整误差越大。由于模拟时间步长 T 越大，实际应用中推理时间越长、计算资源消耗越多，所以在固定大小的 T 下尽量降低转换误差或在尽可能小的 T 下尽量降低转换误差是主要的优化方向。

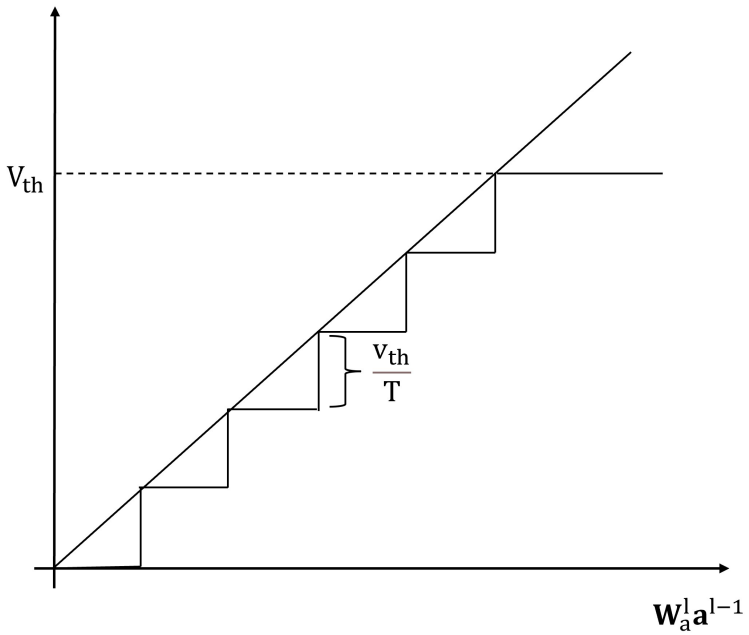

图 3-1 脉冲神经网络平均脉冲激活值与人工神经网络激活值

3.1.2 网络的转换误差

对于深度卷积神经网络来说，其由卷积功能层连接构成，在每一功能层内包含卷积层、批归一化层、激活层和池化层，ANN-to-SNN 需要对这些层进行相应的转换。

卷积层是卷积神经网络的核心，用于提取输入数据的特征，批归一化层包含

权重参数，可直接通过复制的方式将权重参数迁移到脉冲神经网络对应层中。

批归一化层（Batch Normalization）[62]用于加速训练过程和提高模型的稳定性，它通过对每个小批量数据进行标准化处理，使得网络中的每层特征分布相近，从而减少了“内部协变量偏移”问题，减少梯度消失和梯度爆炸的问题，有助于模型在更深的网络中更好的收敛，如今成为深度卷积神经网络的重要组成部分。批归一化层通常通过批归一化层融合[63]的方式先将参数融合进卷积层，然后随卷积层复制到脉冲神经网络中。具体来讲，在一批数量为 N 的小批量数据上，一个卷积层的权重矩阵维度为 I×O×k×k，其中 I 为滤波器通道数，O 为滤波器个数，k 为滤波器宽度，其前向传播时的输出特征维度为 N×O×H×W，其中 H 为输出特征图的高，W 为输出特征图的宽，批归一化计算指的是对于 N 个样本在某一输出通道内的 N×H×W 个值做归一化处理：

$$y = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}} * \gamma + \beta, \tag{3-16}$$

其中，$E[x]$ 表示这些元素的均值，$Var[x]$ 表示这些元素的方差，ϵ 和 β 是可学习的参数，用于对归一化后的数据进行线性变换以适应不同的数据分布，初始化分别为 1 和 0。训练阶段通过移动平均估算整个训练集样本上的均值和方差并通过梯度学习 ϵ 和 β。训练结束后的推理阶段，固定这些参数，使用学到的参数进行前向计算。批归一化层融合技术具体计算方式如下：

$$W \leftarrow W \frac{\gamma}{\sqrt{Var[x] + \epsilon}}, \tag{3-17}$$

$$b \leftarrow \beta + (b - E[x]) \frac{\gamma}{\sqrt{Var[x] + \epsilon}}. \tag{3-18}$$

池化层转换在早期的研究工作中被论述和验证，目前常在人工神经网络中使用平均池化层，脉冲神经网络对应的层也使用平均池化，这样在每个时刻对当前的输入脉冲平均池化，完成模拟时间步长 T 次平均池化后得到的脉冲序列平均激活值与人工神经网络中池化值一致，由于脉冲的离散性，依然存在一定转换误差。

激活层转换使用 IF 激活层替换人工神经网络中中的 ReLU 激活层，由第三

章分析中的公式3-14可知，IF 激活层中的阈值对转换误差有直接影响，许多工作在训练人工神经网络或在参数迁移后设计优化方法学习该参数。

综上，当前卷积神经网络的 ANN-to-SNN 方法对于单个卷积模块转换方式如图3-2所示，每一个模块都使用该方式进行转换就是整个卷积神经网络 ANN-to-SNN 方法的一般流程。基于 ReLU 激活的人工神经网络和基于 IF 激活的脉冲神经网络理论上存在一定的等价关系，同时也有难以完全消除的转换误差。

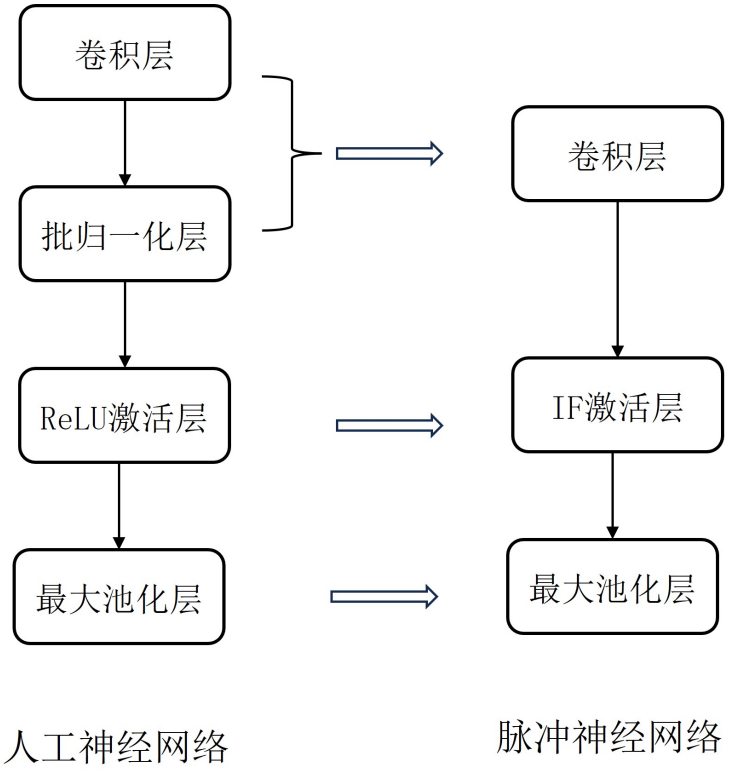

图 3-2 ANN-to-SNN 卷积模块转换示意图

对于脉冲神经网络而言，整体网络上与人工神经网络的转换误差常用输出层（第 n 层）的输出偏移量来衡量：

$$Err_s = \|\boldsymbol{a}^n - \overline{\boldsymbol{s}}^n(T)\|. \tag{3-19}$$

通过对脉冲神经网络结构和前向传播特点的分析容易得知，网络最后一层的输出偏移由转换方法的固有转换误差和最后一层的输入差异造成。逐层反推可知，网络的浅层输出对网络的输出层转换误差有直接影响，浅层中的转换误差和数据脉冲编码造成的信息损失会随着网络的前向传播被逐层放大。所以网络的转换误差与每一个神经元的转换误差有关。

3.1.3 数据分布与转换误差

通过公式3-14和公式3-15对单个神经元的转换误差分析可知，单个神经元的转换误差与该神经元的输入有关。所以说，数据分布会影响转换误差，因为不同数据样本在数值上的差异，会造成人工神经网络内部的激活值的差异。由图3-1可知，人工神经网络的激活值所在坐标位置上直线与折线的竖直举例就是单个样本在单个神经元上的转换误差。所以说，为了实现在整个训练集上转换误差的分析，对某个神经元 n 在训练集 $\mathcal{D}$ 上的转换误差进行统计：

$$\begin{aligned} Err_n(\mathcal{D}) &= \sum_{d\in\mathcal{D}} \|a_d^l - \overline{s}_d^l(T)\| \\ &= \sum_{d\in\mathcal{D}} \|ReLU(\boldsymbol{W}_a^l \boldsymbol{a}_d^{l-1}) - \frac{V_{th}^l}{T} clip(\lfloor \frac{T}{V_{th}^l} \boldsymbol{W}_a^l \boldsymbol{a}_d^{l-1} \rfloor, 0, T)\| \end{aligned} \tag{3-20}$$

可简记为：

$$Err_n(\mathcal{D}) = \sum_{d\in\mathcal{D}} \|a_d^l - ClipFloor(V_{th}^l, T, a_d^l)\|. \tag{3-21}$$

那么对于一个已经训练好的人工神经网络、固定的数据集 $\mathcal{D}$ 和模拟时间步长 T，该神经元的转换误差只与激活阈值 V_{th} 有关，简记为：

$$Err_n(\mathcal{D}) = f(V_{th}). \tag{3-22}$$

那么通过最小化 $Err_n(\mathcal{D})$ 求解 V_{th} 来确定阈值就可实现该神经元 n 在数据集 $\mathcal{D}$ 上转换误差最小。每个神经元都通过这种方式确定阈值，就实现了脉冲神经网络和人工神经网络在整个网络层面的对齐。

3.2 基于神经元阈值优化的脉冲神经网络训练方法

3.2.1 算法流程

通过3.1节对转换误差的分析，我们设计了基于神经元阈值优化的脉冲神经网络训练方法，主要思想是通过在训练集上逐个对齐人工神经网络和脉冲神经网络神经元的输出来实现整个脉冲神经网络输出与人工神经网络的输出偏移最

小。算法流程可以描述为：

1. 使用反向传播算法训练一个人工神经网络；

2. 通过批归一化融合技术和权重迁移操作将人工神经网络中与神经元连接权重相关的参数赋予脉冲神经网络；

3. 通过神经元阈值优化确定脉冲神经元的激活值，完成整个网络的转换。该过程流程图如图3-3所示。

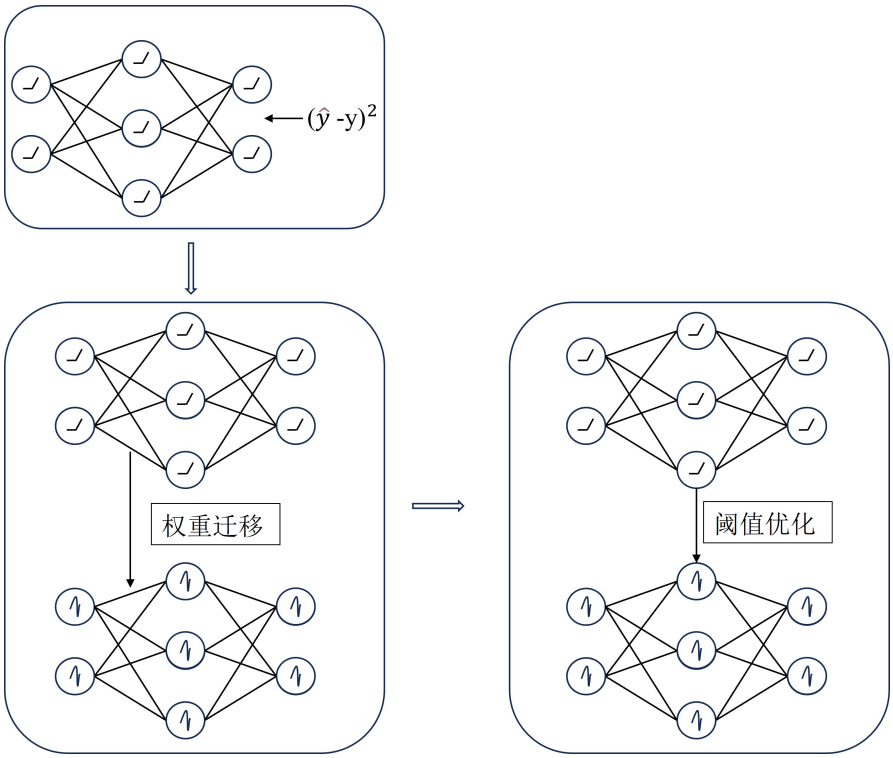

图 3-3 转换算法流程示意图

3.2.2 神经元阈值优化方法

通过3.1节对转换误差的分析可知，单个神经元的阈值优化问题为：

$$\arg\min_{V_{th}} Err_n(D). \tag{3-23}$$

对每一个神经元求解相应的激活阈值就可实现整体网络的转换误差最小。但这个问题是不可导的，本文通过网格搜索（grid search）方法求解。具体而言，将[0,M] 的区间均匀划分为 N 段，每一段右端点为 $n\frac{M}{N}(n=1,2,...,N)$，分别将每个数值带入公式3-22中计算误差大小，然后选择误差最小的数值作为该神经元的激

活阈值。考虑到如果激活阈值大于此人工神经元在该数据集上最大激活值将造成不必要的取整误差和裁剪误差，令 $M = max(\{a_d | d \in \mathcal{D}\})$，$a_d$ 数据样本 d 在网络中前向推理时该神经元的激活值。在实验中发现由于人工神经网络中存在大量（40%-60%）神经元欠激活，神经元最大激活值为 0，会造成通过这种方式搜索得到的脉冲神经元激活阈值为 0，所以实验中使用该神经元周围神经元（神经元所在通道）所有激活值中的最大值确定网格搜索区间。

3.3 实验

本节根据3.2小节中介绍的算法流程和神经元阈值优化方法在图像分类数据集上进行实验以验证提出方法的有效性。我们使用经典的图像分类数据集 CIFAR-10、CIFAR-100[38]和 ImageNet[64]，网络结构使用常用的 VGG[65]和 ResNet[17]。

3.3.1 参数设置

数据集和模型介绍： CIFAR-10 和 CIFAR-100 数据集是 Tiny Images 数据集[66]的一个子集。CIFAR-10 包含 60000 张 32×32 像素的彩色图像，分为 10 个类别，每个类别有 6000 张图像，10 类分别是飞机、汽车、鸟、猫、鹿、狗、青蛙、马、船和卡车，如图3-4所示。每个类别分别划分 5000 张作为训练图像，1000 张作为测试图像。CIFAR-100 数据集包含 60000 张 32×32 像素的彩色图像，分为 100 个类别。这些分类又被分为 20 个超类，我们实验在 100 个类别上进行。每个类别有 500 张训练图像和 100 张测试图像。CIFAR-100 数据集可以更加复杂和详细地进行图像分类任务，它的类别是基于图像中最可能的答案，图像是真实的照片，只包含一个突出的对象实例，对象可能部分遮挡。

ImageNet 是一个大规模的图像数据集，可用于图像分类和目标检测研究，它包含超过 1400 万张手动标注的图像。在图像分类任务中通常只使用 130 万张图片做训练集，因为更多的图像意味着需要更多的计算资源和训练时间，且 130 万张图像可以提供足够的变化和复杂性，包含区分不同类别所需特征。另外有 5 万张图像做测试集，都包含 1000 种不同类别。

VGG 和 ResNet 是计算机视觉任务中流行的网络结构。VGG 模型结构简单并统一，由卷积层、全连接层和池化层线性连接而成，实验中使用的 VGG-16

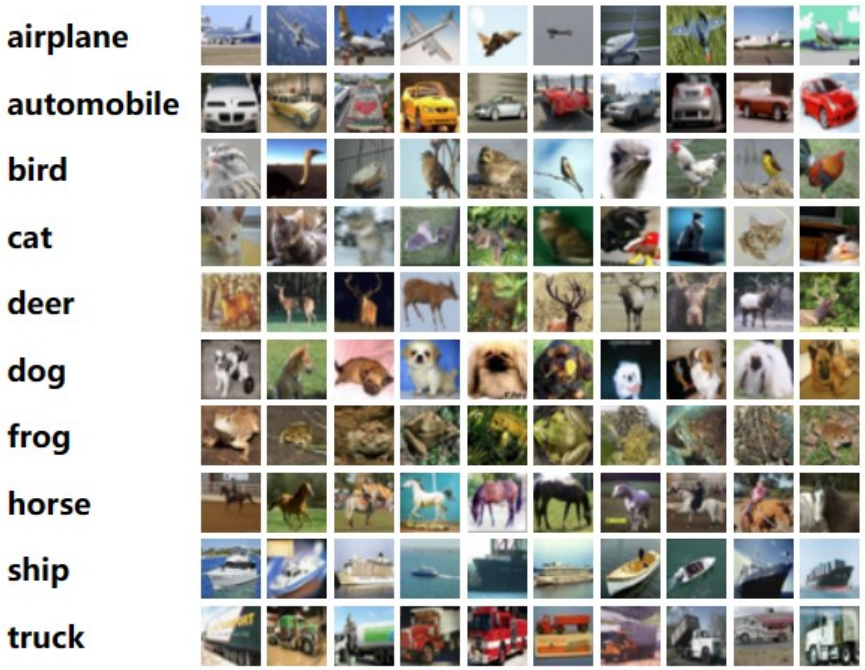

图 3-4 CIFAR-10 数据样本示例[38]

包含 13 个卷积层、3 个全连接层和 5 个池化层，如图3-5所示，图中 conv3-c 表示卷积核大小为 3 通道数为 c，MLP 表示由三层全连接网络构成的多层感知机，全连接层的维度在实验中会根据不同数据集输入维度设置，CIFAR-10 上维度为 512、256、10，CIAFR-100 上为 512、256、100，ImageNet 上为 4096、4096、1000。ResNet 同样由残差块（Residual block）、全连接层和池化层构成，残差块如图3-6所示，其由两个卷积层通过线性连接和跳跃连接构成。用 block-c 表示通道数为 c 的残差块，ResNet-20 结构为 conv3-16、3×block-16、3×block-32、3×block-64、avg-pooling、fc-10/100；ResNet-34 结构为 conv7-64、3×block-64、4×block-128、6×block-256、3×block-512、avg-pooling、fc-1000。

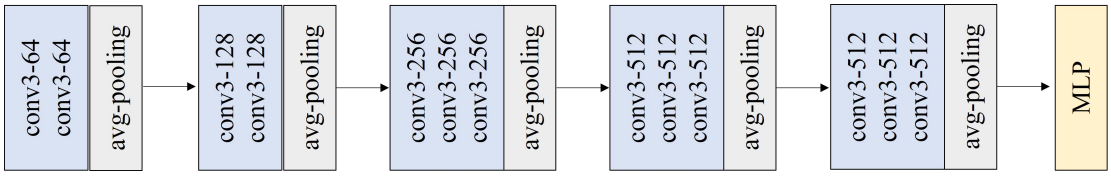

图 3-5 VGG-16 模型结构示意图

评价指标：实验中我们使用模型在测试集上的准确率作为评价指标。具体来讲，在测试集 D 上，使用训练好的脉冲神经网络 S 对数据集中每个样本 d 进行前向推理得到模型对该样本的预测类别 $\hat{y}_d$，如果于该样本的标记真实类别 y_d 一样则分类正确，反之错误，准确率为分类正确样本数量占测试集总量的比例，可形式

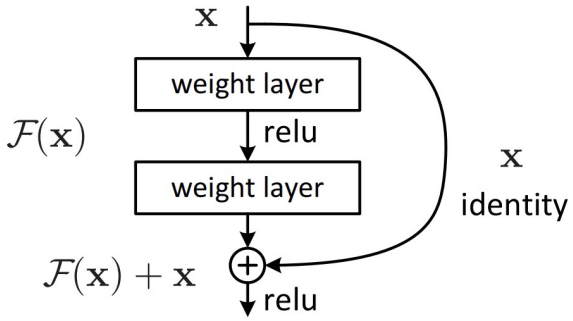

图 3-6 残差块示意图[17]

化表示为：

$$\text{Accuracy} = \frac{1}{|D|} \sum_{d \in D} I(y_d = \hat{y}_d), \tag{3-24}$$

其中 I 为指示函数。

参数设置：在 CIFAR-10 和 CIFAR-100 数据集上，我们使用了 VGG-16 和 ResNet-20 网络结构，训练用于转换的卷积神经网络使用 SGD 优化器[67]，训练 300 个 epoch，数据批量大小为 256；在 ImageNet 数据集上，我们使用了 ResNet-34 网络结构，使用八卡并行训练用于转换的卷积神经网络，训练用于转换的卷积神经网络使用 SGD 优化器，训练 120 个 epoch，数据批量大小为 256。

在进行神经元的阈值求解时，CIFAR-10 和 ICFAR-100 数据集上的实验网格搜索粒度设置为 100，由于 ImageNet 数据集比 CIFAR-10 和 CIFAR-100 复杂得多，神经网络内参数和计算更复杂，所以通过更细粒度的搜索可以获取更准确的激活阈值，ImageNet 上搜索粒度实验设置为 200，每一组实验都是使用 5 个小批量数据用于计算、搜索激活阈值，每批数据量为 256。搜索范围右边界 M 通过人工神经网络在当前批次数据上进行前向推理所得。并使用指数移动平均计算 5 批数据的平均激活阈值。在不同模拟时间步长下进行了多组实验。

3.3.2 实验结果与分析

我们通过对比其他 ANN-to-SNN 方法，对基于神经元阈值优化的 ANN-to-SNN 方法进行分析。其中 Robust Norm[63]、RMP[63]、SNM[36]、TSC[68]属于启发式类方法，Hybrid Train[33]、RNL[69]、OPI[39]、Calibration[34]属于基于转换理论的方法。Robust Norm 通过权重参数归一化降低转换误差；RMP 说明转换过程性能下降由于硬重置，提出了软重置，即激活膜电压不置为零，而是减去释放的电

表 3-1 基于神经元阈值优化的 ANN-to-SNN 算法在 CIFAR-10 数据集上的性能表现

Method	ANN	T = 2	T = 4	T = 6	T = 8	T = 16	T = 32	T=64	T = 128	T = 256
VGG-16										
Robust Norm	92.82	-	-	-	-	-	43.03	81.52	90.80	92.75
Hybrid Train	92.81	-	-	-	-	-	-	-	91.13	-
RMP	93.63	-	-	-	-	-	60.30	90.35	92.41	-
TSC	93.63	-	-	-	-	-	-	92.79	93.27	93.45
SNM+NeuronNorm	94.09	-	-	-	-	-	93.43	94.07	94.07	-
RNL	-	-	-	-	-	57.90	85.40	91.15	92.51	92.95
OPI	94.57	-	-	90.96	93.38	94.20	94.45	94.50	94.49	
Calibration	95.60	-		86.57	-	91.41	93.64	94.81	-	-
Ours	95.71	**72.25**	**91.61**	**93.43**	**94.29**	**95.02**	**95.05**	-	-	-
ResNet-20										
Hybrid Train	93.15	-	-	-	-	-	-	-	-	92.22
RMP	91.47		-	-	-	-	-	-	87.60	89.37
TSC	91.47		-	-	-	-	-	69.38	88.57	90.10
OPI	92.74	-	-	66.24	87.22	91.88	92.57	92.73	92.76	
Calibration	96.72	-	84.70	-	92.98	95.51	96.45	-	-	-
Ours	96.84	**73.21**	**87.97**	**92.29**	**93.93**	**95.86**	96.40	-	-	-

压值；SNM 提出了有符号脉冲神经元来解决脉冲释放不均匀造成的转换损失问题；TSC 通过输入数据动态调整激活阈值来实现无损转换。Hybrid Train 将人工神经网络训好的参数作为脉冲神经网络初始值，然后使用时空反向传播训练脉冲神经网络，使得网络可在几个 epochs 内收敛；RNL 和 OPI 通过训练人工神经网络时优化与转换有关的脉冲神经网络参数实现更高质量转换；Calibration 先将训练好的人工神经网络进行参数迁移，然后在通道上微调激活阈值实现低损失的转换。

在 CIFAR-10 上的实验结果对比如表3-1所示，在 VGG-16 网络结构上，我们的方法在模拟时间步长为 2、4、6、8、16、32 时取得了相较于其他方法更高的准确率，由于在 T=32 时该方法效果已经接近人工神经网络的性能所以我们没有再更大的模拟时间步长上进行实验。我们的方法在低模拟时间步长 T=4 时取得了 91.61% 的准确率，与其他方法相比，由于实现该准确率需要更大的模拟时间步长，意味着我们的方法具有更低的推理时延。与其他基于转换理论的方法相比，我们的方法与人工神经网络的转换性能损失更低。在 ResNet-20 网络结构上，我们的方法在 T 为 2、4、6、8、16 时取得了相较于其他方法更高的准确率，除了 T 为 32 时，与其他方法相比实现了更低模拟时间步长下更高的准确率和更低的转换误差。

在 CIFAR-100 上的实验结果如表3-2所示，在 VGG-16 网络结构上，我们在 T 为 2、4、6、8、16、32 下相较于其他方法都取得了更高的准确率。与基于规则的转换方法相比准确率大幅提升，且实现了极小的时延。与其他基于转换理论的方法相比，在 T 为 4、8、16 时准确率也更高。由于 CIFAR-100 比 CIFAR-10 难度更大，在 T 为 2 和 4 时，转换误差依然较大。在 ResNet-20 网络结构上，由于用于转换的人工神经网络准确率比 VGG-16 的更高，在 T 为 4、6、8 和 16 时取得了与 VGG-16 上相近的准确率，但是转换误差更大。与其他方法相比，可以在更低时延下取得更高准确率。低 T 下转换误差依然很大。

在 ImageNet 数据集上，由于数据集更复杂、模型更大，我们没有在较大的模拟时间步上进行时延，且其他方法准确率不高，我们只与 Calibration 进行对比，相同的 T 下准确率稍高。但与 CIFAR-10 和 CIFAR-100 上的结果相比，所需 T 较大。

值得注意的时，我们的方法在模拟时间步长很小时如 2 和 4，我们的方法准

表 3-2 基于神经元阈值优化的 ANN-to-SNN 算法在 CIFAR-100 数据集上的性能表现

VGG-16										
Method	ANN	T = 2	T = 4	T = 6	T = 8	T = 16	T = 32	T=64	T = 128	T = 256
RMP	71.22	-	-	-	-	-	-	-	63.76	-
TSC	71.22	-	-	-	-	-	-	-	69.86	70.65
SNM+NeuronNorm	74.13	-	-	-	-	-	71.80	73.69	73.95	-
OPI	76.31	-	-	-	60.49	70.72	74.82	75.97	76.25	76.29
Calibration	77.93		55.60		64.13	72.23	75.53	-	-	-
Ours	77.93	**50.30**	**61.80**	**69.47**	**72.20**	**76.08**	**75.71**	-	-	-
ResNet-20										
TSC	68.72		-	-	-	-	-	-	58.42	65.27
OPI	70.43	-	-	-	23.09	52.34	67.18	69.96	70.51	70.59
Calibration	81.51		54.96		71.86	78.13	-	-	-	-
Ours	81.09	**33.80**	**60.57**	**68.03**	**73.26**	**78.96**	**80.11**	-	-	-

	ResNet-34	
Method	ANN	T = 32
Calibration	75.66	64.65
Ours	75.66	**68.90**

表 3-3 基于神经元阈值优化的 ANN-to-SNN 算法在 ImageNet 数据集上的性能表现

确率也不高，这与转换误差分析结论一致，更低的模拟时间步长意味着更大的取整误差，之后可以探索通过尝试动态阈值的方式进一步消除转换误差。

3.3.3 随机噪声对转换误差的影响

根据公式3-14与图3-1可知，由于脉冲信号离散和激活值连续得的原因，脉冲神经元和人工神经元之间存在固有的转化误差，前文中分析过可以分为取整误差与截取误差两部分，本节尝试通过加入噪声平衡取整误差。

由图3-1可知脉冲神经元的平均激活值恒小于人工神经元的激活值，所以尝试通过加入正噪声的方式平衡该部分误差，具体来讲，在每一时刻的膜电位整合计算步骤中加入一个正的随机噪声 $\mathcal{X}$，根据公式3-3有：

$$\boldsymbol{I}^l(t) = V^l(t-1) + \boldsymbol{W}_s^l \boldsymbol{s}^{l-1}(t) + \mathcal{X}, \tag{3-25}$$

那么根据公式3-6该神经元每个时刻计算方式为：

$$V^l(t) = V^l(t-1) + \boldsymbol{W}_s^l \boldsymbol{s}^{l-1}(t) - s^l(t) + \mathcal{X}, \tag{3-26}$$

对该式从时刻 1 到时刻 T 累加脉冲计算过程，然后在时间维度上求均值：

$$\overline{s}^l(t) = \boldsymbol{W}_s^l \overline{\boldsymbol{s}}^{l-1}(t) - \frac{V^l(T)}{T} + \mathcal{X}. \tag{3-27}$$

m 为该神经元激活次数，假设 $v^L(T) \in [0, V_{th}^l)$，根据公式3-1、公式3-10和公式3-27得到：

$$\frac{T}{V_{th}^l}\boldsymbol{W}_s^l \overline{\boldsymbol{s}}^{l-1}(t) - 1 < m - \frac{T}{V_{th}^l}\mathcal{X} \leqslant \frac{T}{V_{th}^l}\boldsymbol{W}_s^l \overline{\boldsymbol{s}}^{l-1}(t), \tag{3-28}$$

可简单表示为：

$$m = clip(\lfloor \frac{T}{V_{th}^{l}} \boldsymbol{W}_{s}^{l} \overline{\boldsymbol{s}}^{l-1}(t) \rfloor, 0, T) + \frac{T}{V_{th}^{l}} \mathcal{X}, \tag{3-29}$$

那么平均脉冲激活值 $\overline{s}^{l}(t)$ 的计算方式为：

$$\overline{s}^{l}(t) = \frac{V_{th}^{l}}{T} clip(\lfloor \frac{T}{V_{th}^{l}} \boldsymbol{W}_{s}^{l} \overline{\boldsymbol{s}}^{l-1}(t) \rfloor, 0, T) + \mathcal{X}, \tag{3-30}$$

如图3-7所示，图中直线表示人工神经元的激活方式，折线表示脉冲神经元的计算方法，两条线的竖直距离代表转化误差。通过这种方式将转换误差调整到同

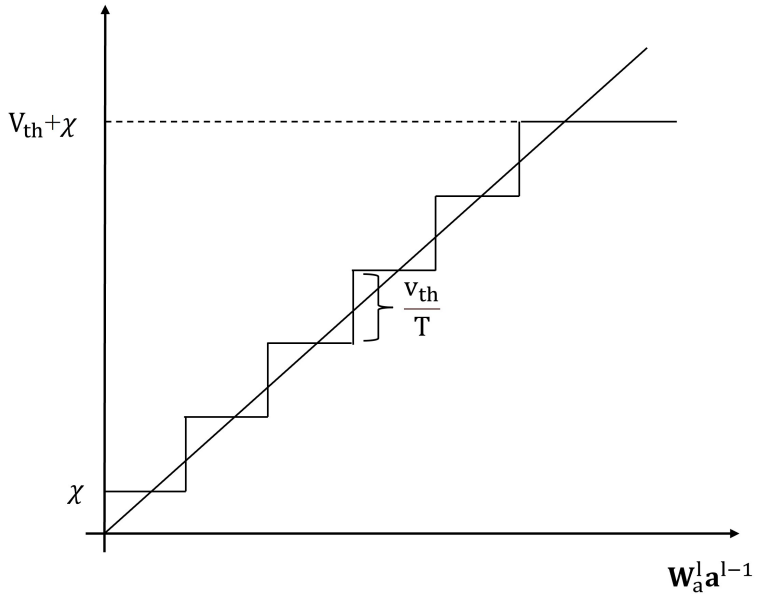

图 3-7 加噪声的脉冲神经网络平均激活值与人工神经网络激活值示意图

时包含正值和负值，避免脉冲神经网络前向传播值整体小于人工神经网络。

然后我们在使用缩放的高斯分布 $\mathcal{X}_1 \sim \frac{V_{th}}{T}\mathcal{N}(0,1)$ 、均匀分布 $\mathcal{X}_2 \sim U(0, \frac{V_{th}}{T})$ 和缩放的绝对值高斯分布 $\mathcal{X}_3 = abs(\mathcal{X}_1)$ 进行实验验证。我们在 CIFAR-10 和 CIFAR-100 上使用 VGG-16 和 ResNet-20 进行实验。

在 CIFAR-10 数据集上的实验结果如表3-4所示。baseline 为没有噪声的组别。模拟时间步长设置为 2、4、6、8、16 和 32。每个实验组用的用于转换的人工神经网络相同，VGG-16 上准确率为 95.71%，ResNet-20 上准确率为 96.84%。在 VGG-16 网络结构上，总体来讲 baseline 在绝大多数 T 下取得了更高的准确率，但是相对于其他组提升幅度不大，例如 T=6 时，准确率较于其他三个噪声提升 0.1% 上下。在 T=2 时，均匀噪声 $\mathcal{X}_2$ 取得了大幅高于其他组的准确率。在 ResNet-20

网络结构上，各个添加噪声的组别相较于 baseline 提升明显，在 T=2 时，均匀噪声比 baseline 准确率高出 3.86%，但是高斯噪声和绝对值高斯噪声比 baseline 有所下降。在 T 大于 2 时，整体上加入噪声可以获得比 baseline 更高的准确率，这也符合对转换误差的理论分析。

在 CIFAR-100 数据集上的实验结果如3-5所示，其中 VGG-16 人工神经网络的准确率为 77.93%，ResNet-20 人工神经网络准确率为 81.09%。整体表现与 CIFAR-10 上类似，在 VGG-16 网络结构上，当 T 为 4、6 和 8 时，加入噪声的组别比 baseline 准确率更高。但是在 T=16 时，加入噪声的组别均低于 baseline 的准确率。在 ResNet-20 网络结构上，在所有 T 下，加入噪声的组别接近或好于 baseline.

综上，通过加入噪声的方式降低固有的取整误差通常可以取得接近或更好的结果，但是实验结果表示加入噪声的效果与数据集、网络结构和模拟时间步长都有关。说明启发式地加入随机噪声由于网络计算地复杂性和噪声的随机性，这种方法也不能完全消除转化误差，之后可以进一步探究转化误差与噪声的关系或通过推理过程中根据输入数据的大小、强度动态调节激活阈值的方式来继续降低转化误差。

表 3-4 探究随机误差对转换的影响实验在 CIFAR-10 数据集上的实验结果

噪声	T=2	T=4	T=6	T=8	T=16	T=32
VGG-16						
baseline	72.25	**91.61**	**93.43**	**94.26**	**95.02**	**95.05**
高斯噪声	67.23	91.32	93.31	94.23	94.55	93.68
均匀噪声	**80.89**	91.31	93.34	94.18	94.60	93.62
绝对值高斯噪声	73.94	91.29	93.34	94.21	94.53	93.60
ResNet-20						
baseline	73.21	87.97	92.29	93.93	95.86	96.40
高斯噪声	64.14	87.88	92.60	94.29	95.84	**96.54**
均匀噪声	**77.07**	**89.11**	**92.83**	94.30	**96.78**	96.49
绝对值高斯噪声	59.70	88.82	92.80	**94.34**	95.86	96.52

表 3-5 探究随机误差对转换的影响实验在 CIFAR-100 数据集上的实验结果

噪声	T=2	T=4	T=6	T=8	T=16	T=32
VGG-16						
baseline	**50.30**	61.80	69.47	72.20	**76.08**	**75.71**
高斯噪声	24.76	62.11	70.30	72.50	71.45	74.36
均匀噪声	49.61	64.49	70.61	**72.91**	71.56	74.38
绝对值高斯噪声	35.54	**65.03**	**70.67**	72.66	71.53	74.38
ResNet-20						
baseline	33.80	60.57	68.03	73.26	**78.96**	80.11
高斯噪声	25.22	60.97	69.13	73.78	78.31	80.73
均匀噪声	**34.10**	63.34	69.62	74.04	78.42	80.70
绝对值高斯噪声	17.95	**63.34**	**70.00**	**74.08**	78.34	**80.80**

3.4 本章小结

本章从单个神经元的转换误差分析入手，解释了单个神经元在 ANN-to-SNN 转换时的转换误差来源，并建模了该误差与激活阈值的关系。然后通过对整个网络在 ANN-to-SNN 转换时的误差分析以及数据分布对转换误差的影响，提出了基于神经元阈值优化的 ANN-to-SNN 方法来降低转换误差，在低模拟时间步长条件下实现高准确率。通过实验与其他方法对比，验证了该方法的有效性。在此基础上我们进一步通过加入随机噪声降低固有的取整误差，实验结果表明多数情况下可以取得比不加入噪声更高的准确率，但是由于噪声的随机性和网络计算的复杂性，该方法存在一定随机性，不能完全消除取整误差。在极低时延下的高性能脉冲神经网络仍需进一步探究，未来可以探索动态阈值等方式进一步降低推理时延。

第四章　基于转换误差的脉冲神经网络剪枝方法

脉冲神经网络剪枝通过去除突触或神经元，实现模型压缩、加速模型推理、降低计算能耗，具有实际应用价值。脉冲神经网络剪枝受其离散的计算方式和时间维度的影响，剪枝算法往往需要精心设计。本文尝试通过转换剪枝人工神经网络的方式实现脉冲神经网络剪枝，基于第三章提出的转换算法，将转换误差引入人工神经网络剪枝中，将该问题建模成关于转换误差和准确率的二目标优化问题，然后结合转换误差和协同演化特点，对剪枝问题进行逐层分解，实现了该问题的高效求解，经实验验证与其他方法相比在更高的剪枝率下实现了更高的准确率。

4.1　问题分析

脉冲神经网络剪枝就是去除网络中冗余的连接或神经元，在尽量不降低模型表达能力的同时减少参数量和计算量。根据剪枝过程中是否有结构约束，网络剪枝方法分为非结构化剪枝方法和结构化剪枝方法。非结构化以神经元或连接突触进行剪枝，往往可以达到很高的剪枝率但不能实现真正硬件加速。结构化剪枝以滤波器、卷积层或结构块为最小单位进行剪枝，剪枝后的模型由于保持结构化，可以实现真正的硬件加速。本文关注于脉冲神经网络的结构化剪枝方法。

剪枝是在训练好的模型上找到重要程度较低的权重，然后去掉这部分权重再重新微调模型以恢复剪枝前的模型性能，或者是在模型训练过程中判断权重的重要程度，边训练边剪枝，剪枝和训练同时完成。在这个过程中，关键的问题有两个：1. 如何判断权重的重要性；2. 如何恢复剪枝模型的性能。

4.1.1 现有方法的限制

当前的脉冲神经网络剪枝方法根据是否使用梯度信息判断权重重要程度，可以分为基于规则的方法和基于梯度的方法。

基于规则的方法通常通过人工设计规则来确定突触或神经元的重要性。例如，论文[57] 使用主成分分析方法来确定每网络一卷积层特征图的重要通道，将非主要的特征图和对应的滤波器剪掉；论文[58] 设计了一个基于神经元活动强度和突触连接强度的自适应剪枝阈值；ESL-SNNs[54] 受到人脑神经网络动态重连过程的启发，设计了突触连接随训练过程可剪枝和再生的训练框架。

基于梯度的方法直接添加权重重要程度的可学习参数，在网络训练过程基于该可学习参数判断连接生成或去除。例如，Grad R[59] 通过在脉冲神经网络训练（使用时空反向传播算法训练）过程中利用梯度信息评估脉冲神经网络中神经元和连接的重要性，并通过梯度来剪枝或重连网络中的连接来保持网络的连通性以实现非常稀疏的网路结构；文章[61] 提出了一种综合的压缩方法，使用时空反向传播算法和交替方向乘子法 (ADMM) 综合求解剪枝和权重量化问题。

当前的脉冲神经网络剪枝方法实现了很好的剪枝率和性能，且结果表明了，与人工神经网络相同，脉冲神经网络中也包含大量冗余参数。

然而，现有方法尚未在深度脉冲神经网络结构上取得好的效果，分析可能的原因：1. 现有脉冲神经网络剪枝方法往往受到人工神经网络剪枝方法的启发而设计，但是由于脉冲神经网络的计算包含时间维度，相较于人工神经网络判断权重的重要性要更困难；2. 现有方法都是用了时空返现传播来训练和微调脉冲神经网络，但是结构处于变化中的稀疏的网络结构训练往往不稳定，加之代理梯度存在偏差，使得网络训练更加波动，在深度脉冲神经网络中还容易受到梯度爆炸或消失的影响。

4.1.2 ANN-to-SNN 用于剪枝的分析

ANN-to-SNN 方法通过参数迁移的方式将一个训练好的人工神经网络转换成脉冲神经网络，由于此类方法不受近似梯度的限制且直接利用已有训练成果，在深度网络结构上取得了很好的性能表现[34,70-71]，而且可以避免时空求导和代理梯度引起的训练不稳定问题。因此，研究将具有转换潜力的剪枝人工神经网络

转换为剪枝脉冲神经网络的范式是有价值的。虽然剪枝人工神经网络相对简单，现有的方法很多[72-74]，但直接将剪枝人工神经网络转换为剪枝脉冲神经网络而不考虑转换误差，使得获得性能良好的剪枝脉冲神经网络具有挑战性。我们通过对结构化剪枝问题和转换误差的综合分析，设计了一种基于 ANN-to-SNN 剪枝脉冲神经网络的方法。

首先，脉冲神经网络的结构化剪枝问题通常可以被建模成一个子集选择问题：将网络中的每个滤波器看作一个集合元素，所有的滤波器构成全集，剪枝就是从集合中选择一个子集（保证可以连成网络）的过程，并通过对不同子集的评估使得选出的最优子集对应的网路结构在我们需要的特性上有好的表现。如图4-1所示，图中方块表示滤波器，蓝色表示保留，白色表示舍弃，最左侧表示整个网络中滤波器的全集，中间表示对不同子集的评估和选择，最右侧表示选出的最优子集对应的网络结构。

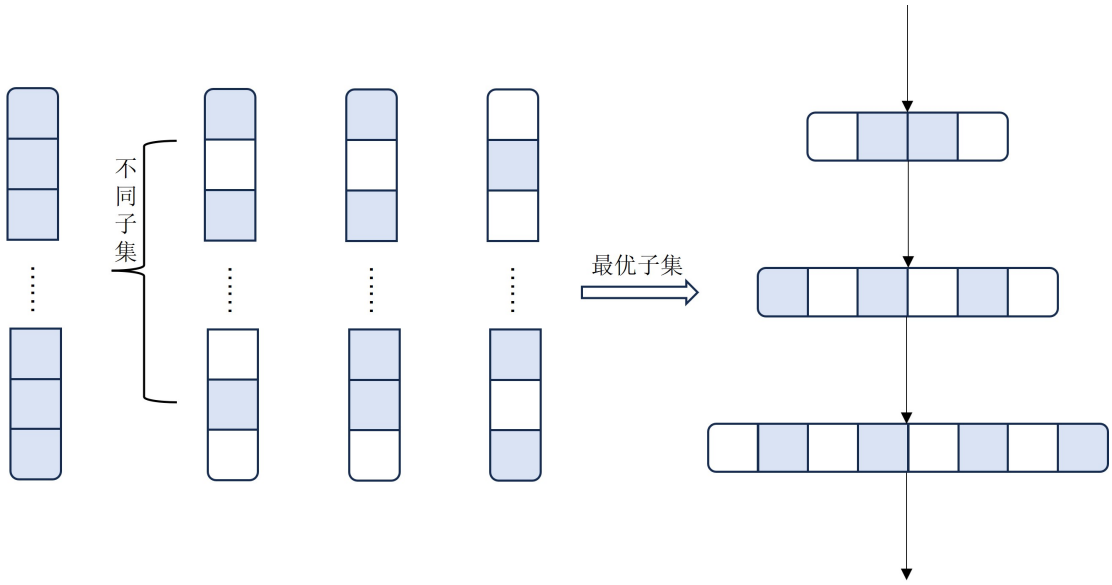

图 4-1 通过子集选择方法建模网络剪枝问题示意图

基于3.1小节对 ANN-to-SNN 转换误差的分析，我们得知，人工神经网络与转换的脉冲神经网络之间有很难完全消除的转换误差，这个转换误差是由于网络中每一个神经元都具有转换误差，也就是特征图一个通道上每一个点都有转换误差。那么对于特征图的一个通道来说，参考公式3-21，我们定义在数据集 D

上通道转换误差为：

$$Err_f(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} \|\boldsymbol{a}_d - \bar{\boldsymbol{s}}_d(T)\|, \tag{4-1}$$

其中，$\boldsymbol{a}_d$ 表示人工神经网络中该特征图的展平后其数值组成的激活向量，$\bar{\boldsymbol{s}}_d(T)$ 表示脉冲神经网络中该特征图展平后其平均脉冲激活值组成的平均激活向量，T 为模拟时间步长。示意图如图4-2所示。图中较大的正方形表示特征图，较小的

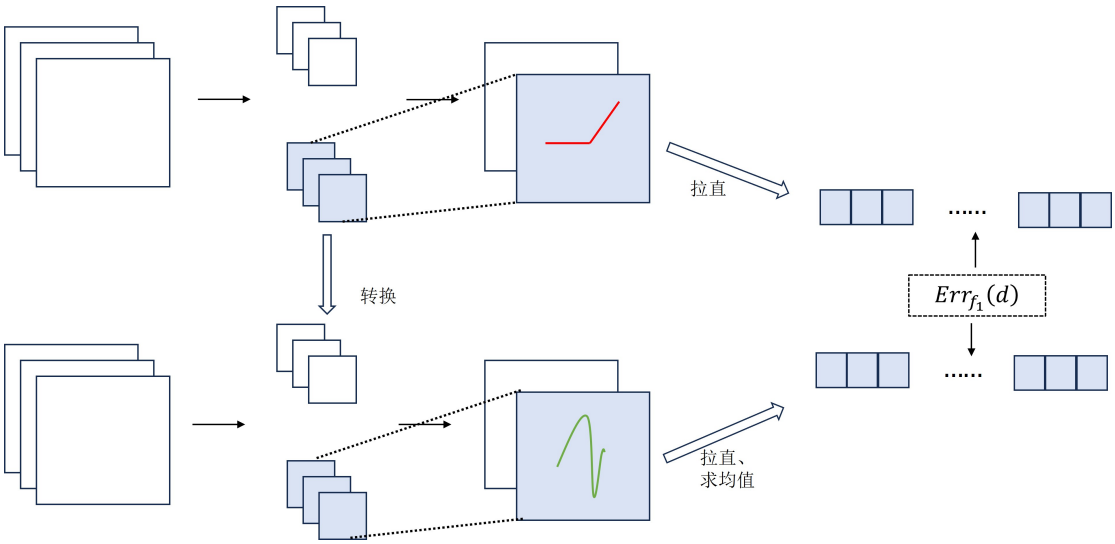

图 4-2 特征图单通道转换误差示意图

正方形为卷积核，$Err_{f_1}(d)$ 表示对数据样本 d 进行前向计算时，人工神经网络与脉冲神经网络在该通道上的转换误差。且特征图该通道对应一个滤波器。

其由于不同特征图输出不同，转换误差不同。那么对脉冲神经网络剪枝这个子集选择问题，基于转换误差最小化选择滤波器的子集是一个自然的想法，通过这种方式可以获得转换误差更小的脉冲神经网络，进而实现脉冲神经网络的高性能。同时考虑到到通过转换获得的脉冲网络性能还受到人工神经网络的性能影响，所以我们综合考虑转换误差与人工神经网络的准确率，将人工神经网络剪枝问题建模成关于这两者的子集选择问题：

$$\underset{\boldsymbol{m}\in\{0,1\}^{|F|}}{\arg\max}\ (-\text{ConversionError}(\mathcal{A}_{\boldsymbol{m}}), \text{Accuracy}(\mathcal{A}_{\boldsymbol{m}})), \tag{4-2}$$

其中，F 表示该网络所有滤波器构成的有序集合，F_i 表示第 i 个滤波器，$\boldsymbol{m} = \{m_i | m_i \in \{0,1\}, i \in \{1,2,...,|F|\}\}$ 为滤波器掩码向量，$\mathcal{A}_{\boldsymbol{m}} = \bigcup_{i=1}^{|F|} F_i m_i$ 表示某一

滤波器子集对应的网络。

4.1.3 问题分解与演化算法求解

公式4-2表示的有关网络剪枝的子集选择问题是不可导的，此类问题常用演化算法进行求解[75-76]。演化算法（Evolutionary algorithms, EAs）是一类基于生物进化原理的优化算法，可用于解决复杂的优化问题。该方法模拟了进化过程中的遗传、变异和选择机制，通过逐代迭代的方式搜索最优解或接近最优解。演化算法的基本思想是通过种群中个体的遗传、变异和环境的自然选择筛选优质个体，一般的演化算法如图4-3所示包含以下流程：

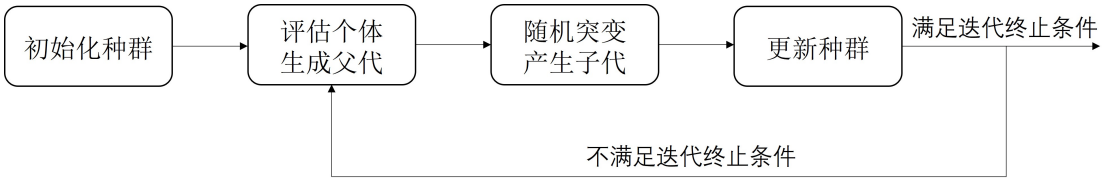

图 4-3 演化算法流程图

1. 初始化种群，随机生成一组初始解作为种群的个体；

2. 评估个体，使用评估函数评估种群每个个体的表现；

3. 选择操作，根据个体评估结果选择一部分个体作为下一种群的父代；

4. 随机突变，通过变异和交叉产生下一代个体；

5. 生成新一代，新的父代与子代形成新的种群；

6. 检查迭代条件，如果满足停止条件，返回最新一代种群，反之重复 2 至 6 直到满足算法停止条件。

通过演化算法求解关于神经网络结构化剪枝的问题，一般是使用每个个体表示一个网络的子集，将优化目标作为评估函数，然后通过上述流程搜索网络的最优子结构。具体而言：

1. 初始化种群，个体使用 $\boldsymbol{m}_i$ 表示，与滤波器掩码一一对应，这样每个个体就表示一个网络子结构，随机初始化 n 个个体作为初始种群；

2. 评估个体，使用公式4-2对应的具体计算方式评估种群中每个个体；

3. 选择操作，选择最优的一部分个体作为新的父代；

4. 随机突变，使用向量逐位随机突变或点位交叉产生新的子代个体；

5. 生成新一代，新的父代与子代形成新的种群；

6. 检查迭代条件，如果满足停止条件，返回最新一代种群，反之重复 2 至 6 直到满足算法停止条件。

演化算法是一种带有随机性的搜索方法，通过不断选择更优的个体使整体种群质量越来越高，从而接近最优解。其搜索空间与神经网络规模呈指数关系，随着神经网络的规模越来越大，直接应用演化算法进行搜索需要巨量的时间，公式4-2所表示的问题直接使用演化算法很难求解。

协同演化通过将问题分解为多个子问题降低搜索规模，并使用独立的进化过程处理这些子问题，从而解决复杂的优化问题[77]，论文[78]指出此类算法能否有效求解问题的关键在于对问题的划分是否采取适当的策略，也就是由求解子问题最优得到的解构成的关于整个问题的解是能否也是最优的。对于公式4-2所表示的问题，其目的在于找到满足优化目标的网络最优子结构。该问题可以通过层划分：将整个网络的搜索空间逐层分解，在每一层中以该层子结构的表示能力和转换误差作为优化目标，通过演化算法求解该层的最优子结构，然后将每一层的最优子结构拼成整个网络的最优子结构。这种划分方式是合适的，因为对于公

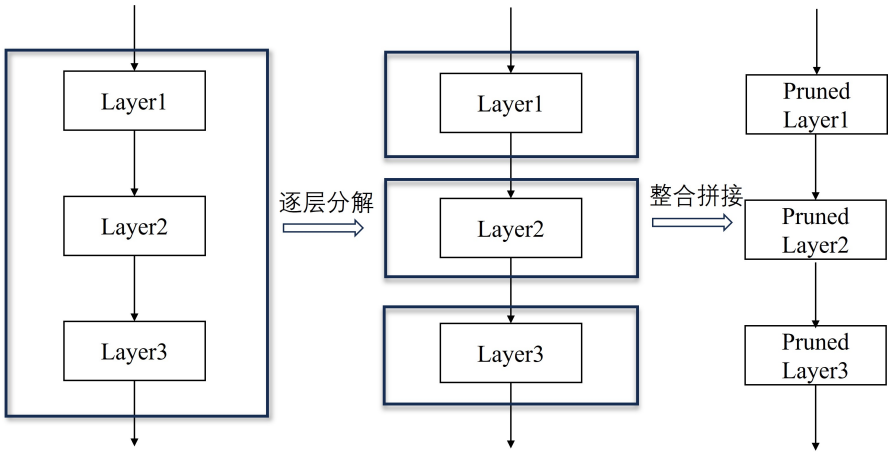

图 4-4 脉冲神经网络剪枝问题求解示意图

式中的准确率优化目标来说，某一层网络的子结构如果表示能力较强，那么其对应的网络结构的准确率也更高[74]；对于公式中的转换误差来说，论文[34]指出整个网络的转换误差以每一层转换误差的线性加权和为上界，所以通过每一层内转换误差最小化搜索的每一层的最优子结构拼成的整体网络转换误差是较小的。基于以上分析，我们将公式4-2表示的子集选择问题的求解逐层分解，先在每一层内通过独立的演化算法得到该层的最优子结构，然后基于每一层的最优子结构拼成整个剪枝网络，这就是求解该问题的主要流程，示意图如图4-4所示。

4.2 基于转换误差的脉冲神经网络剪枝方法

小节4.1中我们通过对 ANN-to-SNN 实现剪枝脉冲神经网络的分析，说明了算法的主要思路，本节中我们详细介绍算法全貌、层内优化问题建模和演化算法求解细节。

4.2.1 算法框架

为先对整体问题有大致的把握，我们先介绍算法框架。基于转换误差的脉冲神经网络剪枝方法是一个迭代式的剪枝方法，算法1给出了算法框架。每一轮开始将一个人工神经网络作为基网络（第一轮以待剪枝的人工神经网络作为基网络），通过转换算法获得对应的脉冲神经网络，然后通过在每一层内搜索在表示能力和转换性能上最优的子结构来获得剪枝的人工神经网络，在训练数据上微调恢复性能后，通过转换获得该轮的剪枝脉冲神经网络，然后该轮中剪枝的人工神经网络作为下一轮的基网络，迭代该过程直到获得的脉冲神经网络剪枝率符合达到要求。算法每一轮都会获得一个剪枝的脉冲神经网络，算法最终返回这些网络。

算法 1 基于转换方法获得剪枝脉冲神经网络的算法框架

输入：一个训练好的人工神经网络 A，剪枝迭代轮数 R，训练集 D_s，微调轮数 E，ANN-to-SNN 方法 C，剪枝脉冲神经网络集合 S
输出：S

```
Let r = 1;
while r ⩽ R do
    使用转换算法 C 转换 A 获得脉冲神经网络 S_tmp;
    通过算法2求解每一层的最优子结构;
    将每一层最优子结构拼接成剪枝人工神经网络 A_p;
    在训练集 D_s 上微调 A_p;
    使用转换算法 C 转换 A_p 获得脉冲神经网络 S_p;
    S = S ∪ S_p;
    A = A_p;
    r = r + 1;
end while
return S
```

4.2.2 层内优化问题

通过4.1小节分析得到，对整体网络的剪枝问题可以按照层划分成多个独立的子问题，通过拼接子问题的解可以获得整体的剪枝网络。且在单层内的最优子结构应当同时具有两个特征：高表达能力和低转换误差，我们通过使用该子结构对应的整体神经网络在训练集上的准确率衡量该子结构的表达能力，并在训练集上的统计转换误差。具体来讲，对于网路的第 i 层，使用 $\boldsymbol{m}_i = \{m_j | m_j \in \{0,1\}, j \in \{1,2,...,l_i\}\}$ 表示滤波器掩码，令 $\mathcal{L}_{i\boldsymbol{m}_i} = \bigcup_{j=1}^{l_i} m_{ij}\mathcal{L}_{ij}$ 表示该层任一子结构，那么 $\mathcal{A}_{\boldsymbol{m}_i} = \mathcal{A} \cap \mathcal{L}_{i\boldsymbol{m}_i}$ 表示该层子结构对应的整个网络，层内优化问题可以表示为：

$$\arg\max_{\boldsymbol{m}_i \in \{0,1\}^{l_i}} (-\text{ConversionError}(\mathcal{A}_{\boldsymbol{m}_i}), \text{Accuracy}(\mathcal{A}_{\boldsymbol{m}_i})), \tag{4-3}$$

其中，准确率直接在训练集上评估，对于该层内子结构的转换误差由公式4-1，该层的子结构在训练集 $\boldsymbol{D}$ 上的转换误差为：

$$\begin{aligned} \text{ConversionError}(\mathcal{A}_{\boldsymbol{m}_i}) &= \frac{1}{|F|}\sum_{f\in F} Err_f(\boldsymbol{D}) \\ &= \frac{1}{|F|}\frac{1}{|\boldsymbol{D}|}\sum_{f\in F}\sum_{d\in \boldsymbol{D}} \|\boldsymbol{a}_d - \bar{\boldsymbol{s}}_d(T)\|, \end{aligned} \tag{4-4}$$

其中，F 为该层子结构所包含的通道。那么公式4-3所表示的单层优化问题就可以在训练集上求解了，而且问题求解难度相较于整个网络大大降低，可以通过演化算法求解。

上述层内优化问题求解如图4-5所示，该图展示了在一层卷积结构中通过演化算法搜索最优子结构的过程。图片下侧示意了演化算法中优化目标的计算过程，其中橙色小方块表示人工神经网络某一子结构的卷积核，绿色小方块表示通过转换得到的脉冲神经网络对应位置的卷积核。通过这两个网络前向推理可得转换误差和准确率。然后通过 average-rank 的方式做演化算法的评估函数，图的上侧是演化算法示意图，下一节将介绍演化算法的细节。

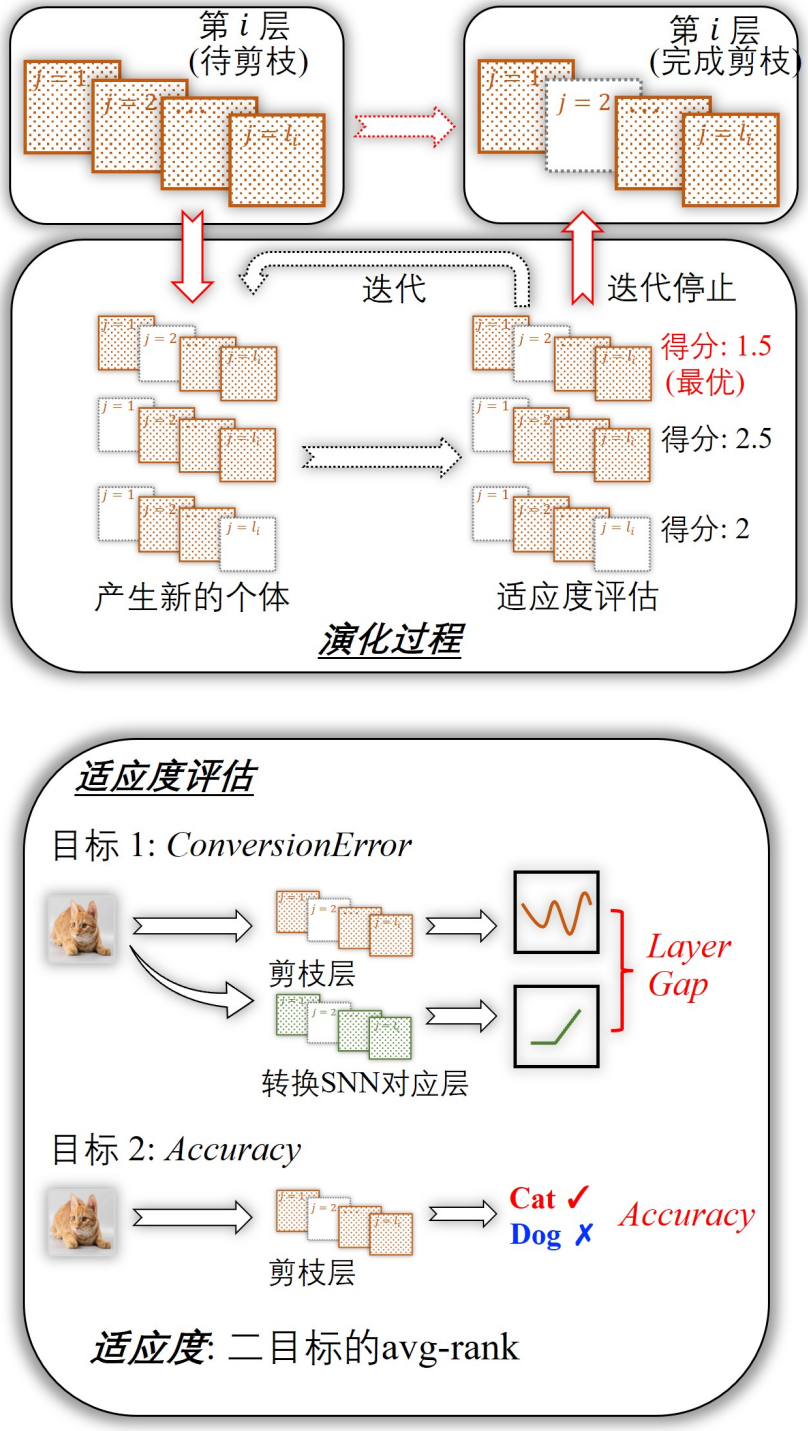

图 4-5 演化算法求解单层最优转换子结构示意图

算法 2 使用演化算法求解单层最优子结构

输入: 数据集 D_s，在数据集上训练好的人工神经网络 $\mathcal{A}$，l_i 表示网络的第 i 层有 l_i 个滤波器，全 1 向量 $\boldsymbol{m}_i^0, |\boldsymbol{m}_i^0| = l_i$，演化迭代轮数 R_e，剪枝率限制 p_0，初始剪枝率限制 p_1，位-突变概率 p_2，演化种群大小 n，训练数据批量大小 b，ANN-to-SNN 方法 C

输出: 该层最优子结构对应的滤波器掩码 $\boldsymbol{m}_i^*$

以 $\boldsymbol{m}_i^0$ 为父本，基于 p_1 和 p_0 通过位-突变生成初始化种群；
Let $r = 1$;
while $r \leqslant R_e$ **do**
  从种群中随机选择 n 个个体，基于 p_2 和 p_0 通过位-突变产生 n 个子代个体；
  for 每个子代个体 $\boldsymbol{m}_i$ **do**
    通过转换算法 C 转换 $\mathcal{A}_{\boldsymbol{m}_i}$ 得到对应的脉冲神经网络；
    通过公式4-4计算 ConversionError($\mathcal{A}_{\boldsymbol{m}_i}$)；
    在训练集 D_s 上计算 Accuracy($\mathcal{A}_{\boldsymbol{m}_i}$);
  end for
  使用 average-rank 综合 ConversionError($\mathcal{A}_{\boldsymbol{m}_i}$) 和 Accuracy($\mathcal{A}_{\boldsymbol{m}_i}$) 对当前所有个体进行排序；
  选出 n 个最优个体作为新的种群；
  $r = r + 1$;
end while
返回 从最后的种群中选出最优个体 $\boldsymbol{m}_i^*$ 返回

4.2.3 使用演化算法求解层内的子结构搜索问题

本节介绍演化算法的求解细节，求解的问题是公式4-3所表示的关于转换误差和准确率的二目标优化问题。

演化算法是一个迭代求解算法，其迭代过程如图4-5所示：图上侧为演化算法的搜索过程，左上角为待剪枝的结构，通过下方大框内的演化流程得到其右侧的剪枝结构。下方大框表示演化算法的迭代过程，如该框中左侧所示每一轮初始有一个种群，下侧所示对当前种群中个体进行评估筛选，如果评估筛选后符合设定条件，则在当前种群中选择排名第一的个体作为搜索结构，不然突变产生新的种群并继续迭代过程。个体的评估通过人工神经网络和脉冲神经网络在数据集上推理得到。

该过程的细节如算法2所示，其中 $\boldsymbol{m}_i^0$ 表示剪枝层的所有滤波器。初始种群由 n 个个体组成，我们以剪枝率 p_1 的概率随机翻转 $\boldsymbol{m}_i^0$ 的每一位，得到包含 n 个体的初始种群。设置一个剪枝速率限制 p_0，以防止剪枝过快。如果一个个体的剪枝率低于 p_0，则它的 0 位被随机翻转为 1，直到该个体满足剪枝率。对于每个个

体 $\boldsymbol{m}_i$，Accuracy($\boldsymbol{m}_i$) 通过在数据集 D_s 上进行推理得到，$-$ConversionError($\boldsymbol{m}_i$) 通过对数据集 b 批采样数据分别在人工神经网络和脉冲神经网络中进行前向推理得到。在初始种群建立之后，演化算法将执行 R_e 轮迭代。在每一轮迭代中，从种群中随机选择 n 个个体进行随机位-突变以产生新的个体，每个位以 p_2 的概率反转，以 p_0 为最低剪枝率限制。对于每个突变个体，转换误差和准确度的计算方法与初始种群中的个体相同。在对每个个体进行评估后，从当前种群中选择最优的 n 个体，并且 n 个后代个体组成下一轮迭代的种群。算法直到达到设定的剪枝轮数停止迭代，在最后的种群中选择最优的个体作为该层的最优子结构。由于这是一个多目标优化问题，因此不能像单目标问题那样直接根据个体的适应度进行排序。为了考虑个体同时在两个目标上的表现，我们采用 average-rank 排序方法，其中 index_1 表示个体在 $-$ConversionError($\boldsymbol{m}_i$) 上的排名，index_2 表示个体在 Accuracy($\boldsymbol{m}_i$) 上的排名，$\frac{\text{index}_1+\text{index}_2}{2}$ 表示排序的平均排名。

4.3 实验与分析

在本节中通过实验测试基于转换误差的脉冲神经网络剪枝方法在 CIFAR-10 和 CIFAR-100[38]上的性能，使用常用的模型架构 VGG-16[79]和 ResNet-56[7]进行实验。随后，我们通过消融实验来分析ConversionError目标的作用。

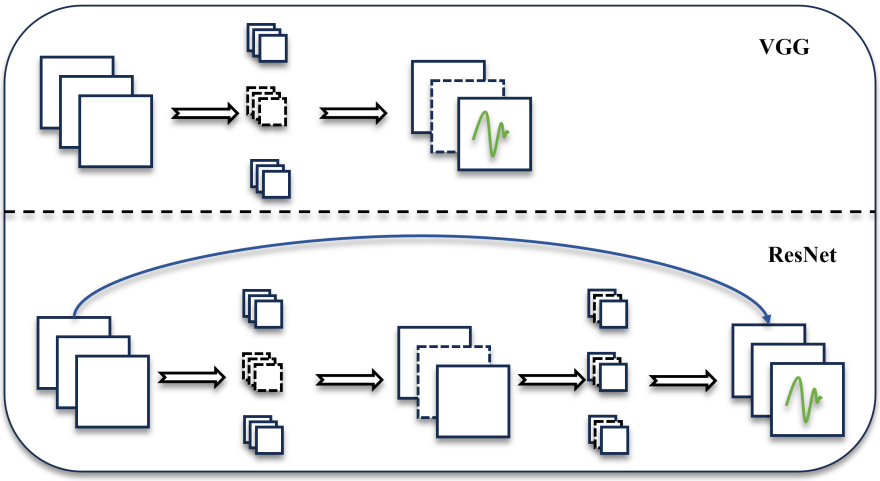

图 4-6 剪枝结构示意图、转换误差计算示意图

4.3.1 实验设置

剪枝结构：我们在滤波器上进行结构化剪枝，这是一种常见的剪枝，与非结构化剪枝方法相比便于硬件加速。与常用的人工神经网络结构化剪枝技术一致，我们在 VGG-16 的每一层和 ResNet-56 的残差块的第一层进行结构化剪枝，如图4-6，以确保剪枝后输出维度一致。

优化目标的计算：对于整个网络的优化问题的层级分级，由于 VGG-16 的卷积层结构与 ResNet-56 的残差块结构不同，我们采用了不同的方法。图4-6给出了直观的描述。具体来说，对于 VGG-16，以单个卷积层划分问题，并利用该层输出来计算ConversionError。另一方面，对于 ResNet-56，以单个残差块（Residual Block）划分问题，并在其第一个卷积层上执行子结构搜索，其中的第二层卷积和其他特征图要做相应的变化，以确保剪枝后层之间的维度匹配。我们将此残差块视为一个整体，以此为整体其计算ConversionError和Accuracy。

参数设置：对于迭代剪枝框架，我们将总的的迭代轮次 R 设置为 20 次，剪枝速率 p_1 设置为 0.04，剪枝速率限制 p_0 设置为 0.04。在实验中，我们发现网络的表达能力对网络中的浅层变化比较敏感，因此，我们对浅层使用更低的剪枝率来增强剪枝的稳定性，对于 VGG-16 中的第 i 层，将剪枝率设置为 $1-(1-p_1)*0.99^i$，并相应地调整 p_0。类似地，对于 ResNet-56，我们按照残差块的深度将其设置为 $1-(1-p_1)*0.995^i$。使用演化算法求解单层的最优子结构时，我们设置 $R_e=10$ 轮迭代，种群大小 n 为 5，并设置突变率 p_2 等于 p_1。在得到每一层的子结构后，我们将它们组合成整个人工神经网络。为了准确计算转换误差，我们使用 warm-up 技术[80]进行微调，初始学习率设置为 1^{-6}，并线性增加 50 次，以达到 CIFAR-10 的最大学习率 1^{-2} 和 CIFAR-100 的最大学习率 5^{-3}，然后余弦减小到 1^{-6}。优化器使用 SGD，数据批次大小设为 300。为了减少实验的所用时间，我们使用 20% 的训练数据来评估准确率，并使用 5 批数据来计算转换误差。

4.3.2 实验效果与对比分析

如表4-1所示，与 CIFAR-10 上的三种非结构化修剪方法相比，我们的方法取得了具有竞争力的结果。其中，ADMM-based[61]方法将脉冲神经网络剪枝建模成有关连接剪枝的优化问题，使用时空方向传播和交替乘子法求解在训练的同

表 4-1 基于转换误差的脉冲神经网络剪枝方法在 CIFAR-10 上的表现

Pruning Method	Architecture	Connectivity (%)	Accuracy (%)	Latency
ADMM-based	7 Conv 2 FC	50.00	89.15	-
Grad R	6 Conv 2 FC	28.41	92.54	8
ESL-SNNs	ResNet-19	50.00	91.09	2
Ours	VGG-16	**44.13**	**92.55**	4
		43.89	**94.67**	8
		43.92	**95.62**	16
	ResNet-56	**49.04**	**94.26**	32

时实现连接剪枝；Grad R[81]方法直接设置了权重重要性的可学习参数，同样通过时空反向传播和代理梯度在训练中进行剪枝；ESL-SNNs[54]受人脑启发设计规则，通过在训练过程中再生和剪枝突触连接探索脉冲神经网络的稀疏结构。非结构化剪枝方法往往可以达到很低的剪枝率，但在实际部署中，结构化剪枝便于实现硬件加速，提高计算效率。

CIFAR-10 上的实验结果如表4-1所示。表中 Connectivity 为连接保留率，与剪枝率计算方式相反，用于衡量剪枝后脉冲神经网络参数占原来网络的比例。表中 Accuracy 为分类准确率。表中 Latency 为脉冲神经网络的模拟时间步长，越小意味着推理次数越少，计算量和能耗越低。我们在 VGG-16 结构上在模拟时间步 4、8 和 16 进行了实验，在 ResNet-34 结构上在模拟时间步长 32 进行了实验。在人工神经网络的剪枝工作中，非结构化剪枝方法剪枝率往往远低于结构化剪枝，我们的结构化剪枝方法与其他非结构化剪枝方法在接近的剪枝率下实现了近似的性能表现。

CIFAR-100 上的实验结果如表4-2所示。由于其他两个方法没有在该数据集上进行实验，所以我们只对比了 ESL-SNNs 方法，该方法用的 ResNet-19 结构为经典的 ResNet-18 加一个卷积层。我们的方法与 ESL-SNNs 性能接近。在 VGG-16 网络上，模拟时间步长为 8 时，在连接保留率为 44.00% 时准确率达到 73.46%，模拟时间步长为 16 时，连接保留率为 43.91% 时准确率达到 75.11%，在保证性能的同时极大的降低了对计算资源的需求。在 ResNet-56 结构中，需要较长的模拟时间步长，这受到转换方法性能的影响，当转换方法在此结构上无法实现高准确率时，剪枝后的模型也难以达到。不过由于我们的方法与转换方法是解耦的，其他转换方法可以即插即用，随着转换方法训练效果越来越好，通过转换进

表 4-2 基于转换误差的脉冲神经网络剪枝方法在 CIFAR-100 上的表现

Pruning Method	Architecture	Connectivity (%)	Accuracy (%)	Latency
ESL-SNNs	ResNet-19	50.00	73.48	4
Ours	VGG-16	**44.00**	**73.46**	8
		43.91	**75.11**	16
	ResNet-56	**49.40**	**69.30**	32

行脉冲神经网络剪枝也会达到更优性能。

4.3.3 消融实验

通过转换误差的脉冲神经网络方法将转换误差加入到人工神经网络的剪枝中，期望实现相同剪枝率的脉冲神经网络更低的转换误差和更好的准确率。为了评估该方法的真实效果，我们通过在剪枝人工神经网络时只用准确率作为演化算法的优化目标而不用转换误差，我们在保持设置和参数一致的情况下进行了一系列消融实验。

CIFAR-10 上的消融实验结果如表4-3所示。表中红色表示性能下降，绿色表示性能上升。在 VGG-16 网络结构上，我们在模拟时间步长 4、8 和 16 下进行了对比，实验结果显示我们的方法对比消融实验在更第的连接保留率下都实现了更搞的准确率，说明通过在人工神经网络剪枝时优化转换误差，可以真正降低该网络结构在转换中的误差，通过这种方式获得剪枝的脉冲神经网络是有效的。

表 4-3 基于转换误差的脉冲神经网络剪枝算法在 CIFAR-10 上的消融实验

Latency	Pruning Method	Connectivity (%)	Accuracy (%)
	VGG-16		
4	**Ours**	**44.13**	**92.55**
	Ablation	46.79 (↑ 2.66)	91.89 (↓ 0.66)
8	**Ours**	**43.89**	**94.67**
	Ablation	46.79 (↑ 2.90)	94.07 (↓ 0.60)
16	**Ours**	**43.92**	**95.62**
	Ablation	46.79 (↑ 2.87)	95.09 (↓ 0.53)
	ResNet-56		
32	**Ours**	**49.04**	**94.26**
	Ablation	51.21 (↑ 2.17)	93.57 (↓ 0.69)

CIFAR-100 上的消融实验结果如表4-3所示，在 VGG-16 网络结构上，我们在模拟时间步长 8、16 下进行实验，在模拟时间步长为 8 时，我们的方法相交于

消融实验在更低的连接保留率下实现了更高的准确率，在时间步为 16 时，连接保留率降低 5.53% 的情况下准确率只降低了 0.10%，说明我们的方法在该组实验中有效优化了转换误差。在 ResNet-56 的结构上，同样实现了更好的性能。

消融实验表明，通过在剪枝人工神经网路中降低转换误差可以起到真实效果，实验验证了我们实验设计的转换误差计算方式、演化算法求解方式的效果。

表 4-4 基于转换误差的脉冲神经网络剪枝算法在 CIFAR-100 上的表现

Pruning Method	Architecture	Connectivity (%)	Accuracy (%)	Latency
Ours	VGG-16	**44.00**	**73.46**	8
		43.91	75.11	16
	ResNet-56	**49.40**	**69.30**	32
Ablation	VGG-16	45.44 (↑ 1.44)	72.22 (↓ 1.24)	8
		49.44 (↑ 5.53)	75.21 (↑ 0.10)	16
	ResNet-56	50.20 (↑ 0.80)	69.08 (↓ 0.22)	32

4.3.4 探究转换误差计算位置对算法性能的影响

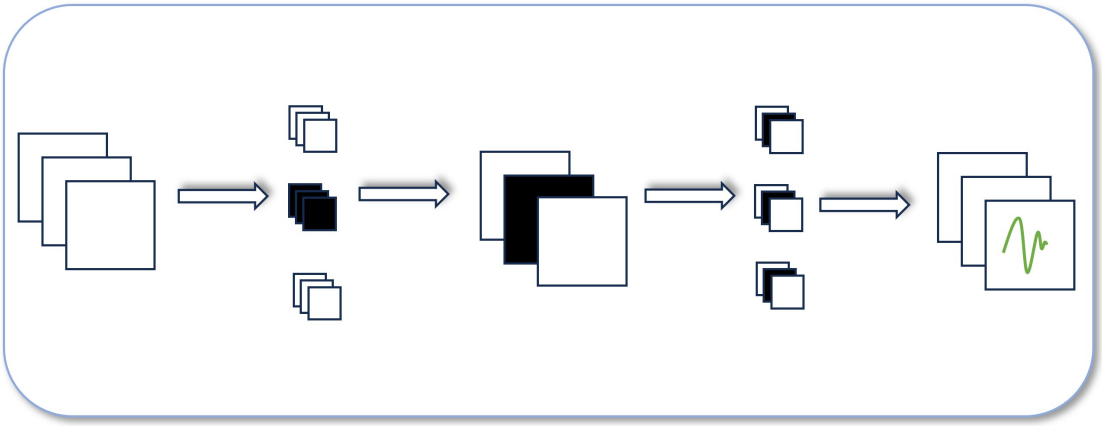

图 4-7 通过剪枝层的下一层计算转换误差示意图

对于图4-6中所示 VGG-16 网络结构上计算转换误差的方式，受到残差块计算转换误差的启发，我们实验探究了通过剪枝层的下一层计算转换误差对算法性能的影响，示意图如图4-7所示。该图表示三层特征成于两层卷积层，我们在第一个卷积层中进行结构化剪枝，黑色表示剪枝，白色表示保留，其后的特征层和卷积层对应变化如下，与图4-6不同的时，这种方式不在剪枝层计算转换误差，而是后一层的输出层计算转换误差。对于网络的最后一个卷积层，保留在本层计算转换误差的方式。我们在 CIFAR-10 数据集上使用 VGG-16 网络结构进行了

表 4-5 转换误差计算方式对算法性能的影响

计算方式	Connectivity (%)	Accuracy (%)	Latency
本层	44.13	92.55	4
	43.89	94.67	8
	43.92	95.62	16
下一层	41.83	92.59	4
	42.20	94.99	8
	42.76	95.60	16

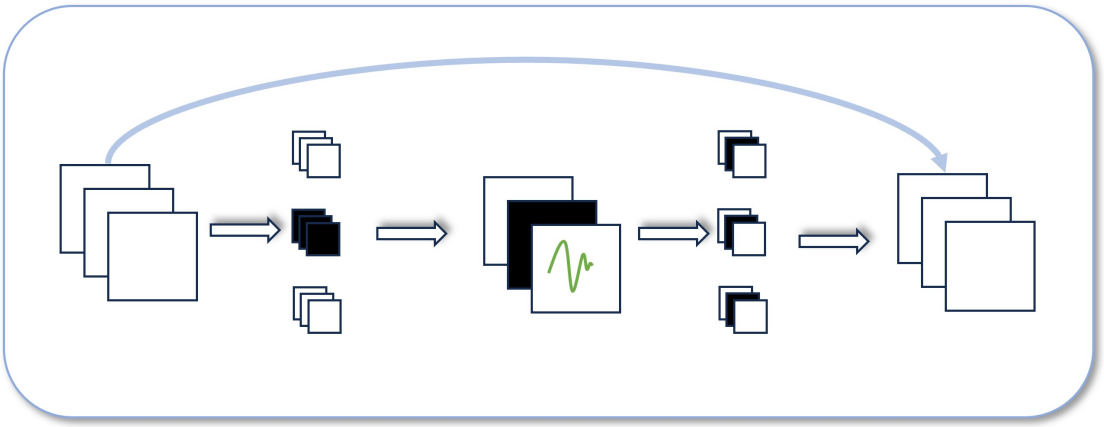

图 4-8 残差块中使用第一层输出计算转换误差的示意图

实验，实验结果如表4-5所示，通过下一层计算转换误差的方式要比本层准确率更高。

与之相反，我们还在残差块上使用第一卷积层对应的输出计算转换误差，如图4-8所示，实验用 ResNet-56 在 CIFAR-10 上进行，结果如表4-6所示。

对这两组实验结果分析可以发现，VGG-16 和 ResNet-56 的表现相反。对于 VGG-16，其使用下一层计算转换误差的实现效果好于本层计算的方式，猜测因为越接近输出层计算网络的转换误差，得到的结果越能反应对准确率的影响。但是在 ResNet-56 网络中，对于残差块，其更像是一个整体，仅通过第一层的对齐不能保证整个残差块的对齐。

表 4-6 残差块中转换误差计算方式对算法性能的影响

计算方式	Connectivity (%)	Accuracy (%)	Latency
本层	51.22	93.90	32
下一层	49.04	94.26	32

4.4 本章小结

脉冲神经网络的计算低功耗特点使其具有广泛的应用前景，特别是在资源受限的边缘设备中。然而，深度脉冲神经网络中大量的参数限制了它们的部署，使得脉冲神经网络剪枝成为一个重要的课题。现有方法主要关注脉冲神经网络的直接剪枝，而忽略了通过 ANN-to-SNN 获得剪枝脉冲神经网络的潜在方法。在本文中，我们提出了一种新的剪枝框架，将转换误差纳入到人工神经网络的剪枝过程中，将其建模成有关转换误差和准确率的优化问题，然后通过对协同演化和转换误差的分析逐层分解该优化问题，实现高效求解。最终，通过转换具有高性能和低转换误差的人工神经网络来实现脉冲神经网络剪枝。我们通过对比和消融实验验证了此种方式的有效性。该类方法稳定易用，但是受转换算法性能的影响，未来随着转换算法的优化，这种方式可实现更好的剪枝效果。

第五章　总结与展望

脉冲神经网络由于其生物仿生性和计算的低功耗潜力近些年广受关注，同时受到日益发展的深度学习对计算资源和能源的需求逐渐增大的影响，许多研究开始以脉冲神经元为载体探索类脑计算模型。这也致使对脉冲神经元和脉冲神经网络的研究成为一个多学科交叉领域。在深度学习领域，实现高性能、第能耗的脉冲神经网络是当前研究热点，本文基于脉冲神经网络的转换训练方式，探究了脉冲神经网络的训练和剪枝算法。

脉冲神经网络的训练方法，受制于脉冲神经网络计算离散、不可导的特点，对该问题的研究在探索当中。本文聚焦于训练脉冲神经网络中的 ANN-to-SNN 方法，通过形式化转换误差，分析了与转换误差相关的因素，并基于此设计了基于神经元阈值优化的 ANN-to-SNN 方法。该方法通过在神经元粒度最小化在训练集上的转化误差实现高性能的脉冲神经网络。

脉冲神经网络剪枝是实现脉冲神经网络的低功耗计算的有效方式，当前有许多研究专注这个问题。与人工神经网络的剪枝类似，脉冲神经网络通过剪枝来减小模型规模、计算复杂度以及对计算、存储资源的需求。并针对脉冲神经网络自身的时空计算、脉冲激活的特点，对脉冲神经网络的模拟时间步长、脉冲激活数量进行剪枝也是实现低功耗的方式。脉冲神经网络剪枝方法常受到人工神经网络剪枝方法的启发，通过规则、梯度等方式判读连接或神经元的重要程度。但是，受到脉冲神经网络时空求导和代理梯度的限制，脉冲神经网络的剪枝往往更复杂和不易训练。本文受到 ANN-to-SNN 方法的启发，尝试使用剪枝的人工神经网络经转换得到剪枝的脉冲神经网络。通过对剪枝和转换问题的联合分析，将该问题建模成关于转换误差和人工神经网络准确率的子集选择问题，通过与演化算法相结合，使得该问题易于求解。

综上，本文通过对脉冲神经网络训练问题和剪枝问题的分析，以实现高性能、低功耗的脉冲神经网络为目标，设计实现了基于神经元阈值优化的脉冲神

经网络训练方法和基于转换误差的脉冲神经网路剪枝算法。未来，随着其他训练方式的发展，基于转换的脉冲神经网络训练方法可以与其他训练方法相结合进行混合训练，以在提升模型性能、节约训练资源等方面取得更好的效果。

参考文献

[1] RUSSELL S J, NORVIG P. Artificial intelligence: a modern approach[M]. Pearson, 2016.

[2] MCCULLOCH W S, PITTS W. A logical calculus of the ideas immanent in nervous activity[J]. The bulletin of mathematical biophysics, 1943, 5: 115-133.

[3] ROSENBLATT F. The perceptron: a probabilistic model for information storage and organization in the brain[J]. Psychological review, 1958, 65(6): 386.

[4] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning representations by back-propagating errors[J]. Nature, 1986, 323(6088): 533-536.

[5] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[6] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[7] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016: 770-778.

[8] HU Y, YANG J, CHEN L, et al. Planning-oriented autonomous driving[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 17853-17862.

[9] HE X, LIAO L, ZHANG H, et al. Neural collaborative filtering[C]//Proceedings of the 26th international conference on world wide web. 2017: 173-182.

[10] 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.

[11] YAO P, WU H, GAO B, et al. Fully hardware-implemented memristor convolutional neural network[J]. Nature, 2020, 577(7792): 641-646.

[12] ZENG Y, ZHANG T, XU B. Improving multi-layer spiking neural networks by incorporating brain-inspired rules[J]. Science China. Information Sciences, 2017, 60(5): 052201.

[13] FANG W, CHEN Y, DING J, et al. SpikingJelly: An open-source machine learning infrastructure platform for spike-based intelligence[J]. Science Advances, 2023, 9(40): eadi1480.

[14] IZHIKEVICH E M. Simple model of spiking neurons[J]. IEEE Transactions on neural networks, 2003, 14(6): 1569-1572.

[15] MARKRAM H, LÜBKE J, FROTSCHER M, et al. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs[J]. Science, 1997, 275(5297): 213-215.

[16] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[17] 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.

[18] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[19] 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.

[20] SCHAEFER C J, TAHERI P, HORENI M, et al. The hardware impact of quantization and pruning for weights in spiking neural networks[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2023.

[21] PEREZ-NIEVES N, GOODMAN D. Sparse spiking gradient descent[J]. Advances in Neural Information Processing Systems, 2021, 34: 11795-11808.

[22] 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.

[23] HUH D, SEJNOWSKI T J. Gradient descent for spiking neural networks[J]. Advances in neural information processing systems, 2018, 31.

[24] ESSER S K, APPUSWAMY R, MEROLLA P, et al. Backpropagation for energy-efficient neuromorphic computing[J]. Advances in neural information processing systems, 2015, 28.

[25] BI G Q, POO M M. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type[J]. Journal of neuroscience, 1998, 18(24): 10464-10472.

[26] GERSTNER W, KISTLER W M. Spiking neuron models: Single neurons, populations, plasticity[M]. Cambridge university press, 2002.

[27] FROEMKE R C, DAN Y. Spike-timing-dependent synaptic modification induced by natural spike trains[J]. Nature, 2002, 416(6879): 433-438.

[28] 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.

[29] 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.

[30] WU Y, DENG L, LI G, et al. Direct training for spiking neural networks: Faster, larger, better[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 33: 01. 2019: 1311-1318.

[31] SENGUPTA A, YE Y, WANG R, et al. Going deeper in spiking neural networks: VGG and residual architectures[J]. Frontiers in neuroscience, 2019, 13: 425055.

[32] 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.

[33] RATHI N, SRINIVASAN G, PANDA P, et al. Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation[J]. ArXiv preprint arXiv:2005.01807, 2020.

[34] LI Y, DENG S, DONG X, et al. A free lunch from ANN: Towards efficient, accurate spiking neural networks calibration[C]//International conference on machine learning. 2021: 6316-6325.

[35] BU T, FANG W, DING J, et al. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks[J]., 2023, arXiv:2303.04347.

[36] WANG Y, ZHANG M, CHEN Y, et al. Signed Neuron with Memory: Towards Simple, Accurate and High-Efficient ANN-SNN Conversion.[C]//IJCAI. 2022: 2501-2508.

[37] GLOROT X, BORDES A, BENGIO Y. Deep sparse rectifier neural networks[C]//Proceedings of the fourteenth international conference on artificial intelligence and statistics. 2011: 315-323.

[38] KRIZHEVSKY A, HINTON G, et al. Learning multiple layers of features from tiny images[J]., 2009.

[39] BU T, DING J, YU Z, et al. Optimized potential initialization for low-latency spiking neural networks[C]//Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI): vol. 36: 1. 2022: 11-20.

[40] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural network[J]. Advances in neural information processing systems, 2015, 28.

[41] FRANKLE J, CARBIN M. The lottery ticket hypothesis: Finding sparse, trainable neural networks[J]. ArXiv preprint arXiv:1803.03635, 2018.

[42] LIN M, JI R, WANG Y, et al. Hrank: Filter pruning using high-rank feature map[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1529-1538.

[43] CAI H, GAN C, WANG T, et al. Once-for-all: Train one network and specialize it for efficient deployment[J]. ArXiv preprint arXiv:1908.09791, 2019.

[44] MOLCHANOV P, TYREE S, KARRAS T, et al. Pruning convolutional neural networks for resource efficient inference[J]. ArXiv preprint arXiv:1611.06440, 2016.

[45] LOUIZOS C, WELLING M, KINGMA D P. Learning sparse neural networks through L_0 regularization[J]. ArXiv preprint arXiv:1712.01312, 2017.

[46] GUO Y, YAO A, CHEN Y. Dynamic network surgery for efficient dnns[J]. Advances in neural information processing systems, 2016, 29.

[47] LUO J H, WU J, LIN W. Thinet: A filter level pruning method for deep neural network compression[C]//Proceedings of the IEEE international conference on computer vision. 2017: 5058-5066.

[48] GAO X, ZHAO Y, DUDZIAK Ł, et al. Dynamic channel pruning: Feature boosting and suppression[J]. ArXiv preprint arXiv:1810.05331, 2018.

[49] HUANG Z, WANG N. Data-driven sparse structure selection for deep neural networks[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 304-320.

[50] HAN B, ZHAO F, ZENG Y, et al. Developmental plasticity-inspired adaptive pruning for deep spiking and artificial neural networks[J]. ArXiv preprint arXiv:2211.12714, 2022.

[51] MENG L, QIAO G, ZHANG X, et al. An efficient pruning and fine-tuning method for deep spiking neural network[J]. Applied Intelligence, 2023, 53(23): 28910-28923.

[52] CHOWDHURY S S, RATHI N, ROY K. Towards ultra low latency spiking neural networks for vision and sequential tasks using temporal pruning[C]//European Conference on Computer Vision. 2022: 709-726.

[53] CHOWDHURY S S, GARG I, ROY K. Spatio-temporal pruning and quantization for low-latency spiking neural networks[C]//2021 International Joint Conference on Neural Networks (IJCNN). 2021: 1-9.

[54] SHEN J, XU Q, LIU J K, et al. ESL-SNNs: An evolutionary structure learning strategy for spiking neural networks[J]., 2023, arXiv:2306.03693.

[55] DE VIVO L, BELLESI M, MARSHALL W, et al. Ultrastructural evidence for synaptic scaling across the wake/sleep cycle[J]. Science, 2017, 355(6324): 507-510.

[56] BENNETT S H, KIRBY A J, FINNERTY G T. Rewiring the connectome: evidence and effects[J]. Neuroscience & Biobehavioral Reviews, 2018, 88: 51-62.

[57] CHOWDHURY S S, GARG I, ROY K. Spatio-Temporal Pruning and Quantization for Low-latency Spiking Neural Networks[C]//Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN). 2021: 1-9.

[58] GUO W, FOUDA M E, YANTIR H E, et al. Unsupervised Adaptive Weight Pruning for Energy-Efficient Neuromorphic Systems[J]. Frontiers in Neuroscience, 2020, 14: 598876.

[59] CHEN Y, YU Z, FANG W, et al. Pruning of Deep Spiking Neural Networks through Gradient Rewiring[C]//Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI). 2021: 1713-1721.

[60] BELLEC G, KAPPEL D, MAASS W, et al. Deep rewiring: Training very sparse deep networks[J]. ArXiv preprint arXiv:1711.05136, 2017.

[61] DENG L, WU Y, HU Y, et al. Comprehensive SNN Compression Using ADMM Optimization and Activity Regularization[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34(6): 2791-2805.

[62] IOFFE S, SZEGEDY C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//International conference on machine learning. 2015: 448-456.

[63] 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.

[64] DENG J, DONG W, SOCHER R, et al. Imagenet: A large-scale hierarchical image database[C]//2009 IEEE conference on computer vision and pattern recognition. 2009: 248-255.

[65] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. ArXiv preprint arXiv:1409.1556, 2014.

[66] BIRHANE A, PRABHU V U. Large image datasets: A pyrrhic win for computer vision?[C]//2021 IEEE Winter Conference on Applications of Computer Vision (WACV). 2021: 1536-1546.

[67] SUTSKEVER I, MARTENS J, DAHL G, et al. On the importance of initialization and momentum in deep learning[C]//International conference on machine learning. 2013: 1139-1147.

[68] HAN B, ROY K. Deep spiking neural network: Energy efficiency through time based coding[C]//European Conference on Computer Vision. 2020: 388-404.

[69] 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.

[70] DING J, YU Z, TIAN Y, et al. Optimal ANN-SNN Conversion for Fast and Accurate Inference in Deep Spiking Neural Networks[C]//Proceedings of the 30th International Joint Conference on Artificial Intelligence(IJCAI). 2021: 2328-2336.

[71] WANG B, CAO J, CHEN J, et al. A new ann-snn conversion method with high accuracy, low latency and good robustness[C]//Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI). 2023: 3067-3075.

[72] LI H, KADAV A, DURDANOVIC I, et al. Pruning Filters for Efficient ConvNets[J]., 2016, arXiv:1608.08710.

[73] LUO J H, WU J. AutoPruner: An end-to-end trainable filter pruning method for efficient deep model inference[J]. Pattern Recognition, 2020, 107: 107461.

[74] SHANG H, WU J L, HONG W, et al. Neural Network Pruning by Cooperative Coevolution[C]//Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI). 2022: 4814-4820.

[75] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE transactions on evolutionary computation, 2002, 6(2): 182-197.

[76] ZHOU A, ZHANG Q, ZHANG G. A multiobjective evolutionary algorithm based on decomposition and probability model[C]//2012 IEEE Congress on Evolutionary Computation. 2012: 1-8.

[77] MA X, LI X, ZHANG Q, et al. Cooperative co-evolutionary algorithms: A survey[G]//IEEE Transactions on Evolutionary Computation. on-line publishing.

[78] MA X, LI X, ZHANG Q, et al. A Survey on Cooperative Co-Evolutionary Algorithms[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(3): 421-441.

[79] SIMONYAN K, ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition[C]//The 3rd International Conference on Learning Representations (ICLR). 2015.

[80] LOSHCHILOV I, HUTTER F. Sgdr: Stochastic gradient descent with warm restarts[J]., 2016, arXiv:1608.03983.

[81] DENG S, LI Y, ZHANG S, et al. Temporal efficient training of spiking neural network via gradient re-weighting[J]. ArXiv preprint arXiv:2202.11946, 2022.

致　　谢

研究生阶段即将结束，回忆即将过去的三年，对能够进入南京大学感到十分幸运，在这里享受了优质的教学资源，并受到“诚朴雄伟，励学敦行”氛围的熏陶。在临别之际，对老师和同学的不舍与感激涌上心头。

感谢我的导师申富饶教授，您在科研上对我的指导和生活上的关心此刻历历在目。有幸能遇到您这样的导师，您的培养与耐心教导使我有机会以一个跨专业的背景了解并学习人工智能领域内的知识与技术；每周的面对面个人讨论您都会耐心仔细地与我们交流，帮助我们合理安排科研工作，解决遇到的难题，极大的提升了我的研究能力；您平日里所展现的严谨、开放的治学态度启发我在今后的工作中要认真务实、敢想敢干。

感谢赵健老师，赵老师在每次组会都认真地聆听大家的组会报告，仔细地提出问题，启发大家思考。赵老师曾给予我耐心的指导，使我能够以严谨、求实的态度进行科研。

感谢在科研与生活中陪伴我的同学和朋友，从你们身上学到许多优良的品质，同时平日里的相处也使我这三年的校园生活变得丰富多彩。

感谢我的家人，你们是我永远的后盾与动力。

简历与科研成果

基本信息

王翔宇，男，汉族，1998 年 06 月出生，山东省临沂市人。

教育背景

2021 年 9 月—2024 年 6 月	南京大学人工智能学院	硕士
2016 年 9 月—2020 年 6 月	内蒙古大学生命科学学院	本科

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 2021 年 9 月-2024 年 6 月)，负责神经网络模型相关研究。

攻读硕士学位期间的发明专利

1. 申富饶，**王翔宇**，赵健.《一种基于转换误差的脉冲神经网络剪枝方法》。专利申请号：202410323203.8