中国科学院大学

University of Chinese Academy of Sciences

博士学位论文

深度脉冲神经网络转换学习算法研究

作者姓名：陈睿智

指导教师：王东琳　研究员　中国科学院自动化研究所

学位类别：工学博士

学科专业：计算机应用技术

培养单位：中国科学院自动化研究所

2019 年 6 月

Research of Converted Learning Algorithms for Deep Spiking Neural Networks

A thesis submitted to the
University of Chinese Academy of Sciences
in partial fulfillment of the requirement
for the degree of
Doctor of Engineering
in Technology of Computer Application

By

Chen Ruizhi

Supervisor: Professor Wang Donglin

Institute of Automation, Chinese Academy of Sciences

June, 2019

中国科学院大学

研究生学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。

作者签名：陈睿智

日　　期：2019.6.6

中国科学院大学

学位论文授权使用声明

本人完全了解并同意遵守中国科学院有关保存和使用学位论文的规定，即中国科学院有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：陈睿智　　导师签名：

日　　期：2019.6.6　　日　　期：2019.6.6

摘 要

脉冲神经网络在低功耗、生物可解释性以及脑机交互实时应用等方面具有优越的性能和广泛的应用前景，因此在类脑计算中占据着重要的地位。但是，脉冲序列的不可微性以及网络中复杂的动态特性使得脉冲神经网络的训练十分困难。目前脉冲神经网络的学习算法还无法取得和深度卷积神经网络相似的性能，这会极大地限制脉冲神经网络的应用。针对该问题，一种可行的解决方案是脉冲神经网络模型转换。该方案首先构建一个结构和规模上都与卷积神经网络类似的脉冲神经网络；然后通过卷积神经网络成熟的训练技术获得高性能卷积神经网络模型；最后通过特定的转换算法将卷积神经网络模型参数转换成脉冲神经网络参数，从而获得与卷积神经网络类似性能的脉冲神经网络模型。

现有的脉冲神经网络模型转换算法研究已经取得了一些非常有前景的成果。但由于两种神经网络之间内在机制的差异，脉冲神经网络模型无法与卷积神经网络模型一一对应，转化后的脉冲神经网络在识别精度、收敛时间等方面与实际应用之间还存在差距，同时现有算法只能获得浅层的脉冲神经网络。本论文针对上述问题，通过详细研究讨论不同转换算法与所得脉冲神经网络在识别精度、收敛时间等方面的内在关系，提出三种转换算法，获得脉冲神经网络的深度结构，为构建高效高性能的深度脉冲神经网络提供重要理论基础和参考设计。本文的主要工作与贡献如下：

(1)多强度深度脉冲神经网络模型转换算法

本文针对脉冲饱和问题，提出一种多强度的脉冲神经网络及其动态剪枝算法，通过降低神经元输出脉冲强度的限制，可以获得多强度大规模深度脉冲神经网络，并提高转换网络的收敛速度。具体来说，首先提出一种多强度的脉冲神经元模型，降低对神经元输出脉冲强度的限制；其次提出多强度脉冲神经网络结构，支持具有深度结构的脉冲神经网络转换；最后，针对深度脉冲神经网络中的大量运算冗余，提出3种脉冲神经网络压缩算法，在保持逼近精度不变的条件下，可以移除原始多强度神经网络中85%的运算操作。实验结果表明，本文提出的算法，在MNIST和CIFAR10数据集上，分别获得99.57%和94.01%的识别精度，较同期最好结果分别提升0.13%和3.16%；并且该网络可以在80个时间步内收敛，比同期的模型转换算法加速3.75倍。

(2)低延迟深度脉冲神经网络模型转换算法

本文提出限制输出预训练算法和错误脉冲抑制算法，在获得具有深度结构的转换脉冲神经网络的同时显著提高转换网络的收敛速度。限制输出预训练算法，通过在卷积神经网络训练过程中进行动态参数规范化，解决脉冲神经网络逼近过程中的脉冲饱和问题；错误脉冲抑制算法，将错误脉冲的抑制问题抽象化为一个线性规划问题，大大减少转换网络中的错误脉冲。实验结果表明，使用这两种算法的转换脉冲神经网络可在30个时间步内收敛，在CIFAR10数据集上取得的最佳逼近精度为94%。

(3)基于反向传播的极低延迟深度脉冲神经网络转换学习算法

本文提出基于反向传播的极低延迟深度脉冲神经网络转换学习算法，利用模型转换算法中两种神经网络之间的联系，使用反向传播算法学习转换网络中脉冲序列的时序信息中的有效特征，获得具有深度结构的脉冲神经网络的同时进一步降低转换网络的收敛时间。具体来说，首先分析总结使用反向传播算法训练深度脉冲神经网络所需满足的三个严苛条件；其次论证模型转换参数可以使脉冲神经网络获得在空间域上处理信息的能力，并设计一种参数初始化算法，使反向传播算法支持更深的脉冲神经网络训练，同时减小反向传播算法的训练迭代次数；最后，提出误差最小化算法以及修改的损失函数，进一步提升反向传播算法的性能。实验结果表明，本部分提出的算法，在MNIST和CIFAR10数据集上，分别将网络收敛时间进一步降低到4和10个时间步，比算法(2)分别提高7.5倍和3倍，同时保持较高的识别精度(分别为99.44%和 91.52%)。

关键词： 脉冲神经网络；反向传播算法；卷积神经网络；脉冲神经元模型；脉冲神经网络学习算法

Abstract

Spiking neural networks (SNNs) have excellent performance and broad application prospects in low power hardware, algorithmic interpretability and real-time applications of brain–machine interfaces, so SNNs play an important role in neuromorphic computing. However, the SNN learning algorithms still can not achieve the similar performance as deep convolutional neural networks (CNN) due to the complex dynamics and non-differentiable spike events in these networks. To solve the problem, a feasible solution is converting CNNs into SNNs (CNN-SNN conversion algorithm). In this solution, a SNN similar to the CNN in structure and model size is firstly constructed; then a high performance CNN is obtained by using the mature CNN training algorithms; finally, the CNN weight parameters are converted into the SNN architecture to get a converted SNN with high performance.

Some promising results have been achieved in the researches of the existing CNN-SNN conversion algorithms. However, there is still a gap of the SNN practical applications in the accuracy and the convergence time aspects. Moreover, the existing conversion algorithms can only convert shallow SNNs. In view of these above problems, we study the inherent relationship between the conversion algorithms and the obtained SNNs in the accuracy, the convergence time and other aspects, then propose three conversion algorithms to provide an important theoretical basis and reference design for the construction of high performance SNNs. The main contributions of this dissertation are:

(1) Deep multi-strength SNN conversion algorithm

This dissertation presents a deep multi-strength conversion algorithm with dynamic pruning, through relaxing the restriction of the spike strength to obtain deep multi-strength SNN (M-SNN) and to decrease the convergence time of the converted SNNs. Specifically, a multi-strength spiking neuron model is firstly proposed. Then, a M-SNN structure is introduced to support large scale deep SNNs. Finally, three aggressive dynamic pruning techniques are applied to reduce the computational operations by 85% while maintaining the same accuracy. Experiments show that our algorithm achieves 99.57% and 94.01% accuracy on the MNIST and the CIFAR10 dataset respectively,

outperforming the best results for the same period with 0.13% and 3.16% accuracy. Meanwhile, the convergence time of the M-SNN is 80 time steps, with 3.7× convergence speedup.

(2) Low latency deep SNN conversion algorithm

This dissertation presents a restricted output training method and a false spike inhibition method, observably decreasing the convergence time. The restricted output training method normalizes the converted weights dynamically in the CNN training phase, solving the firing rate saturation problem. The false spike inhibition method reduces the false spikes through transforming the inhibition problem to a linear programming problem. Experiments show that the converted SNN can converge within 30 time steps, and the accuracy is 94% on the CIFAR10 dataset.

(3) Very low latency deep SNN conversion learning algorithm with the back-propagation process

This dissertation proposes a very low latency deep SNN conversion learning algorithm based on back-propagation. This algorithm brings in the back-propagation algorithm to learn the valid features in the spike trains in the converted SNNs through the connection between CNNs and SNNs, and further reduces the convergence time of the converted SNNs. More concretely, three severe conditions in training deep SNNs with back-propagation are analyzed; then the converted model parameters are proved to be capable of handling the spatial information in the converted SNNs and the weight initialization algorithm is introduced to support the deeper SNN training with back-propagation and decrease the train epochs of back-propagation; finally, an error minimization method and a modified loss function are presented to further improve the training performance. In experiments, these three algorithms achieve the accuracy of 99.44% and 91.52% on the MNIST dataset and the CIFAR10 dataset respectively, and the convergence time steps are 4 and 10.

Keywords: Spiking Neural Networks; Back-Propagation; Convolutional Neural Networks; SNN Learning Algorithms

目 录

图形列表

表格列表

符号列表

缩写

ADALINE	ADAptive LInear NEuron
ANN	Artificial Neural Networks
ART	Adaptive Resonance Theory
BNN	Binary Neural Networks
BPTT	BackPropagation Through Time
BST	Baseline Training
CNN	Convolutional Neural Networks
CD	Contrastive Divergence
DBN	Deep Belief Networks
DoG	Difference of Gaussian filter
DVS	Dynamic-Vision-Sensor
FSI	False Spike Inhibition
HH	Hodgkin-Huxley Model
IF	Integrate-and-Fire Models
IM	Izhikevich Model
LIF	Leaky Integrate-and-Fire Models
PSD	Precise-Spike-Driven
RBM	Restricted Boltzmann Machines
RNN	Recurrent Neural Networks
ROT	Restricted Output Training
SNN	Spiking Neural Networks
SOM	Self-Organizing Map
SPAN	Spike Pattern Association Neuron
SRM	Spike Response Models
STBP	Spatio-Temporal BackPropagation
STDP	Spike Timing-Dependent Plasticity

SVM	Support Vector Machines
TSP	Traveling Salesman Problem

第1章 引言

近年来，随着脑与神经科学领域的新技术不断涌现以及计算机硬件技术的不断发展，各国纷纷推出脑计划来推动人类对大脑的理解以及实现类脑智能系统，类脑计算也越来越受到人们的关注。尽管深度卷积神经网络在某些人工智能任务上取得了类人的效果，但是在算法的训练功耗、泛化性能以及可扩展性等方面，卷积神经网络算法中还存在着一定的局限性。同时，脉冲神经网络在生物可解释性、低功耗以及脑机交互等领域拥有巨大的可能性，这些可能性使得脉冲神经网络研究在类脑计算中占据着重要的地位。但是，脉冲神经网络的学习算法还无法获得性能优越的深度脉冲神经网络，这制约着脉冲神经网络在类脑计算中的应用。通过总结归纳脉冲神经网络学习算法中存在的各种问题，发现脉冲神经网络模型转换算法可以有效地缩减卷积神经网络与脉冲神经网络之间的巨大性能鸿沟。因此，研究脉冲神经网络模型转换算法，从而优化其他脉冲神经网络学习算法，具有很深刻的理论和现实意义。

本章首先简要介绍计算神经科学的发展历程，包括人工神经网络、脉冲神经网络以及类脑计算及相关脑计划；然后介绍了卷积神经网络与脉冲神经网络的研究现状及其特点；接着介绍了各种典型的脉冲神经网络学习算法，指出目前仅能获得浅层脉冲神经网络的瓶颈所在，提出需要解决的问题；最后阐述了本文所研究的问题及内容组织。

1.1 计算神经科学的发展历程

人类的大脑是一个包含上千亿个各种不同类型的脉冲神经元构成的复杂的动态系统，理解大脑的结构及其行为机制是二十一世纪最具有挑战性的科学问题。神经科学(Neuroscience)在分子尺度、细胞尺度、神经网络尺度乃至系统尺度上，研究了大脑的物质、能量以及信息的基本活动规律[1]。该学科的主要目标是阐明大脑的神经结构如何实现认知功能，理解大脑的信息如何进行高效处理，探索大脑各区域如何协同工作等等，其宗旨是揭示大脑的运行机制。与此同时，由于计算机技术的发展，神经计算(Neuromorphic Computing)中使用人工神经网络的技术模拟大脑运行机制成为一个研究热点[2, 3]。该学科侧重于利用现有的数理基础，借助现有的神经科学中的相关证据，构建统一的类脑信息处理系统，完成相关的科学和工程中的人工智能任务。所以，神经科学的目标偏

重于通过生理学实验理解大脑，而神经计算的目标更偏重于使用来自神经科学的启示来构建大脑。这两个学科是相辅相成的，神经计算受神经科学启示，同时神经计算构建的系统又能帮助神经科学进一步理解大脑的运行机制。

下面将从人工神经网络的发展历程、脉冲神经网络的发展历程以及类脑计算及相关脑计划等三个方面来简要介绍计算神经科学的发展历程。其中，人工神经网络的发展历程以神经网络的性能发展为脉络，简要介绍了引起三次人工神经网络研究热潮的相关典型算法，可以发现人工神经网络随着网络层数的不断加深，其处理人工智能任务的能力也不断地增强。卷积神经网络是第二代人工神经网络中的典型，而脉冲神经网络被称之为第三代人工神经网络[4]。由于脉冲神经元更贴近大脑使用脉冲的信息传递方式，脉冲神经网络在未来的类脑计算研究中占据了重要的地位。脉冲神经网络发展历程简要介绍了脉冲神经网络发展过程中最重要的相关典型算法。由于脑科学中新技术的不断涌现以及人工神经网络研究的不断发展，类脑计算将这两方面技术相融合，意图推导出实现强人工智能的相关理论，逐渐成为未来的重要研究方向。类脑计算及相关脑计划简要介绍了各国脑计划，指出了脉冲神经网络在类脑计算技术体系中的重要地位。

1.1.1 人工神经网络的发展历程

人工神经网络的发展历程如图 1.1所示。计算神经科学最早的一个数学模型可追溯到1943年McCulloch 和 Pitts [5]提出的McCulloch-Pitts神经元模型。该模型揭示了一个深刻的科学思想：神经元的空间整合特性以及阈值特性的有限的逻辑组合，即可构成大脑感知外部世界的基本要素。该模型的提出标志着神经计算科学的诞生，同时开启了第一代人工神经网络的萌芽期。

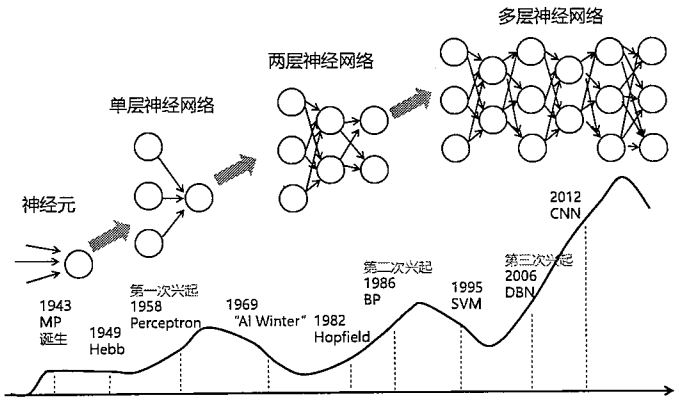

图 1.1 人工神经网络发展历史。

Figure 1.1 The development history of artificial neural networks.

1958年，Rosenblatt [6]提出多层感知器模型(Multi-Layer Perceptron，简称为MLP)，这是第一个具有学习能力的人工神经网络(Artificial Neural Networks，简称为ANN)模型，开启了第一波人工神经网络的研究热潮。1960年，Widrow 和 Hoff [7]提出ADALINE网络模型(ADAptive LInear NEuron)以及Widrow-Hoff学习法则，获得了一个连续取值的自适应线性神经网络模型。在随后的数十年内，很多研究人员认为只要使用足够数目的神经元，任何智能任务都能被解决。但是，1969年，人工智能的创始人之一Minsky 和 Papert [8]指出感知器模型只能解决线性问题，无法解决非线性问题，例如异或问题，从而在一定程度上减慢了感知器模型的发展。在这段低潮期，Grossberg [9]提出了自适应共振理论(Adaptive Resonance Theory)，Fukushima [10]提出一种新认知机(Neocognitron)，Kohonen [11]提出自组织映射网络(Self-Organizing Map)。

直到二十世纪八十年代中叶，Hopfield [12]提出Hopfield网络，并使用李雅普诺夫方法分析网络的稳定性，建立了动力学与人工神经网络的关系，阐明了人工神经网络的稳定性判定依据。随后的1984年，Hopfield [13]使用Hopfield网络找到了著名的旅行商问题(Traveling Salesman Problem，简称TSP问题)的最佳解的近似解，引起了广泛的关注。1985年，Ackley 等 [14]提出玻尔兹曼机(Boltzmann Machines)理论，该理论中使用模拟退火算法，有效地解决了Hopfield网络容易掉到能量局部最小值位置的问题。1986年，反向传播算法被提出，克服了长期以来人工神经网络没有有效的突触权值调整算法的困难。到1988年，该项工作才受到了足够的重视[15]。随后，Broomhead 和 Lowe [16]提出径向基神经网络，是神经网络实用化的一个重要标志。同时，一系列对反向传播算法的分析文章[17–19]指出只要引入足够数目的隐层神经元，神经网络可以逼近任意复杂的函数，重新打开了Minsky 和 Papert [8]关上的神经网络的研究大门。以上这几项工作引起了第二波人工神经网络的研究热潮。

到了上世纪九十年代中期，统计学习方法中的支持向量机(Support Vector Machines，简称SVM)[20]被提出，该算法表现出优于已有方法的性能。同时，人工神经网络理论方面缺乏实质性的进展，由于反向传播算法的过拟合以及参数训练速度慢等问题，关注神经网络的研究视线被转移到了统计学习理论的研究上了。直到2006年，Hinton 和 Salakhutdinov [21]指出多隐层的神经网络可以刻画数据的本质属性，借助无监督的逐层初始化方法可以克服深度神经网络训练困难的问题。同年，Hinton 等 [22]基于受限玻尔兹曼机构建了深度信念网络(Deep Belief Networks，简称DBN网络)，开启了第三波人工神经网络的研究热潮。

到了2012年，随着计算机硬件技术的成熟，Krizhevsky 等 [23]使用一个深度卷积神经网络AlexNet网络取得了ImageNet2012比赛的冠军，其17.0%的Top-5错误率，远超第二名的26.2%的错误率。模型的规模是其取得成功的关键，更大的模型规模意味着模型拥有一个更大的状态空间。人工神经网络的大的状态空间是处理复杂人工智能任务的关键。本质上来说，AlexNet的模型规模足够其获得处理复杂人工智能任务的状态空间，巨大的ImageNet数据集[24]又可以保证模型被充分训练。由此，进一步加大了第三波人工神经网络的研究热潮，卷积神经网络(Convolutional Neural Networks，简称为CNN)已经成为各种人工智能任务研究的主要工具之一[25–33]。

1.1.2 脉冲神经网络的发展历程

文献[3]中给出了近30年来，不同人工神经网络模型在神经计算中硬件实现的逐年发表的代表性论文的数目，如图 1.2所示。在此图中，spiking表示脉冲神经网络，feed-forward表示常见的卷积神经网络，recurrent表示循环神经网络，stochastic表示随机神经网络，unsuepervised表示使用无监督学习法则搭建的神经网络，visual表示受视觉系统启发的人工神经网络。可以看出，卷积神经网络(Convolutional Neural Networks，简称为CNN)和脉冲神经网络(Spiking Neural Networks，简称为SNN)是近十五年来人工神经网络的研究热点(本文中卷积神经网络包括多层全连接神经网络)。脉冲神经网络借鉴大脑中使用脉冲传递信息的方式，使用脉冲神经元模型搭建神经网络，被称之为第三代人工神经网络[4]。第一代人工神经网络是感知器网络，其基本组成计算单元是McCulloch-Pitts神经元模型[5]。第二代人工神经网络包含比较广泛，其中典型的有多层感知器、卷积神经网络以及递归神经网络[34]等等，主要使用的计算单元是使用单调递增的非线性激活函数的McCulloch-Pitts神经元模型。根据基本计算单元可知，前两代人工神经网络中的基本运算单元使用激发脉冲的平均速率的编码方式传递信息。这种编码方式会使数据中的很多信息丢失，同时神经科学实验表明很多生物的神经系统中使用脉冲中的时序信息进行编码[35–39]。后续研究利用了这些实验中的证据，开始探索以脉冲神经元模型为基本运算单元的第三代人工神经网络。

1952年，Hodgkin 和 Huxley [40]模拟乌贼神经元的离子通道、膜电势变化、膜电势阈值以及发送脉冲后的不应期等特性，提出了第一个脉冲神经元模型。该模型是对生物神经元逼近最精确的神经元模型，后续工作致力于简化该模型以适应大规模脉冲神经网络的搭建，典型模型包括Morris-Lecar神经元模

型(1981年)[41]、FitzHugh-Nagumo神经元模型(1961年)[42]、Hindmarsh-Rose神经元模型(1984年)[43]以及Izhikevich神经元模型(2003年)[44]。在脉冲神经网络中应用最广的神经元模型是IF神经元模型(Integrate-and-Fire Model)以及脉冲响应模型(Spike Response Models，简称SRM)[45]。这两个神经元模型都是阈值发送模型，运算相对于上述模型更加简单，但是足够用于搭建脉冲神经网络并实现模式识别等相关功能。1997年，Maass证明了脉冲神经元具有很强的非线性处理能力，用其搭建的脉冲神经网络可以用来逼近任意的连续函数。同时，以脉冲神经元为基本计算单元搭建的脉冲神经网络功耗性能方面强于以Sigmoid神经元搭建的人工神经网络[46–48]。

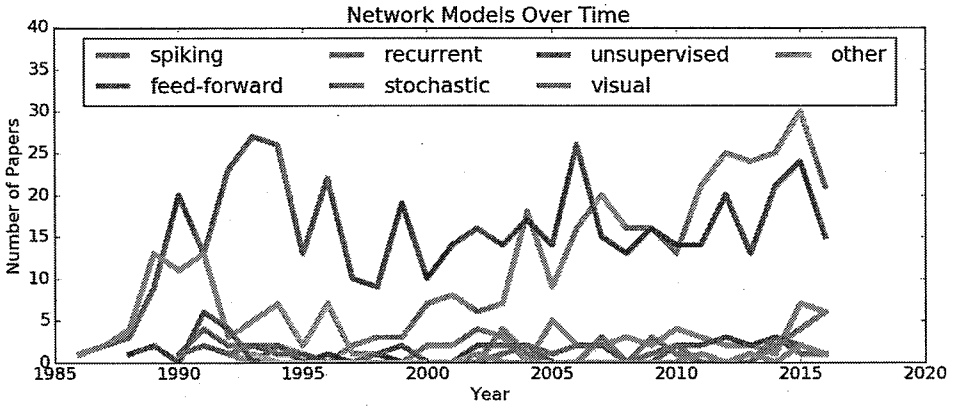

图 1.2 近30年来，不同人工神经网络模型在神经计算中硬件实现的逐年论文发表数目。其中，该图截取自文献[3]中图8。

Figure 1.2 An overview of how models for neuromorphic implementations have changed over time, in terms of the number of papers published per year. This figure is from Figure 8 in the work[3].

人工神经网络中的神经元通过突触相互连接，学习过程就是调整突触权值大小使得神经网络能够获得提取数据中有效特征的能力。传统人工神经网络中采用反向传播算法获得神经元突触的最优调整值。但是脉冲神经元使用脉冲进行信息编码，其中包含很强的不可微性，无法直接将传统人工神经网络的反向传播算法直接应用到脉冲神经网络的训练之中。因此，很多学者开始着力于探索脉冲神经网络的学习算法。

1949年，Hebb [49]通过生理学实验中的证据归纳总结出了第一个生物神经元的突触可塑性的基本原则Hebbian法则。1997年，Ruf 和 Schmitt [50]最早提出了一种基于Hebbian法则的脉冲神经网络监督学习算法。同年，Markram 等 [51]结合生物学实验提出了Hebbian法则的扩展版本STDP法则(Spike Timing-

Dependent Plasticity，脉冲时间依赖可塑性)。很多后续研究使用STDP法则训练脉冲神经网络，其中典型的学习算法包括监督学习算法(2005年-2013年)[52–61]以及非监督学习算法(2002年-2018年)[62–69]。监督学习算法中被引用最多的是ReSuMe算法(2010年)[55]，该算法将STDP法则和anti-STDP法则相结合，可以对脉冲序列的复杂时空模式进行远程学习。这类算法由于其学习性能良好，得到了广泛地应用，但是只能获得单脉冲神经元以及单层脉冲神经网络。在非监督学习算法中，由于STDP法则只能提取局部的数据特征，因此还未取得很好的效果。直到近两三年，Diehl 和 Cook [66]及其后续研究[68, 69]提出的LM-SNN中使用侧向抑制机制提取全局特征，获得了一个三层脉冲神经网络；Kheradpisheh 等 [67]首先使用差分高斯滤波器获取图像边缘信息，然后使用Rank-order编码脉冲输入序列，使用简化STDP法则获得了一个5层的卷积脉冲神经网络。此外，在硬件领域，很多研究使用忆阻器实现STDP法则[70–75]，极大地加速了脉冲神经网络的学习过程。

2002年，Bohte 等 [76]提出了SpikeProp算法，这是第一个基于梯度下降规则的脉冲神经网络监督学习算法。该算法使用SRM神经元模型搭建网络，为了克服神经元内部状态变量由于脉冲发送而导致的不连续性，限制了网络中所有层的神经元只能发送一个脉冲，从而使用梯度下降法推导出一个适用于多层脉冲神经网络的学习算法。随后，该算法的相关变式[77–82](2004年-2017年)在不同方面增强了该算法的性能，但是仍需限制脉冲神经网络中的脉冲发送数目。因此，这些脉冲神经网络中只有一个隐藏层，同时只在异或问题[76]以及Fisher Iris数据集[83]等小数据集上进行测试。2006年，Gütig 和 Sompolinsky [84]提出Tempotron算法，将膜电势阈值之下的后突触膜电势视为所有脉冲神经元输入的脉冲加权和，使用梯度下降法学习脉冲的时空序列模式。该算法仅仅能够获得单层的脉冲神经网络。近年来，一些基于梯度下降规则的脉冲神经网络监督学习算法取得了有竞争力的效果，比如Lee(2016年)[85]、STBP算法(2018年)[86, 87]、SLAYER算法(2018年)[88]、HM2-BP算法(2018年)[89]。这些算法都训练获得了三层以上的脉冲神经网络，同时在MNIST数据集以及CIFAR10数据集上取得了不错的效果。

大部分脉冲神经网络的学习算法都是基于STDP法则以及反向传播算法的，还有一些基于其他方法的典型的脉冲神经网络学习算法研究。其中，SPAN算法(2012年)[90]和PSD算法(2013年)[91]使用卷积核将离散的脉冲事件转换为连续函数，克服了脉冲神经网络中的不可微性。文献[92–95](2013年-2015年)使用受

限玻尔兹曼机的网络结构，提出事件驱动的对比散度算法训练脉冲神经网络。为了减小脉冲神经网络与卷积神经网络的性能之间的差距，文献[96–99](2013年-2017年)中将卷积神经网络模型转换为与之匹配的脉冲神经网络，获得了识别精度与卷积神经网络类似的脉冲神经网络。

1.1.3 类脑计算及相关脑计划

近年来，随着脑与神经科学领域新技术的不断涌现，以及人工智能所依赖的深度学习算法、芯片的计算能力和数据集规模的发展，可以在更深刻的层面支持人工智能研究者对智能的本质进行探究。类脑计算逐渐引起了学术界的广泛注意，其核心目标就是通过借鉴脑神经的结构以及信息处理机制，实现机制类脑、行为类人的下一代人工智能系统。根据各国脑计划的主要内容，采用自底向上的方法可以将类脑计算分为如图 1.3所示的技术体系。

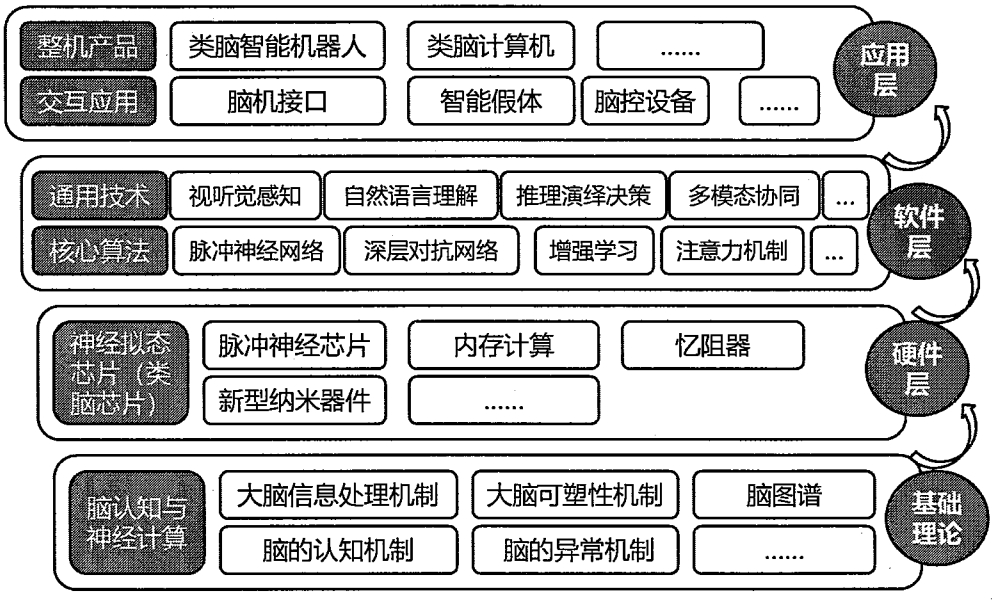

图 1.3 类脑计算的技术体系。

Figure 1.3 The technical system of neuromorphic computing.

类脑计算中影响力最大的工作是近年来各国的脑计划。2013年4月，美国总统奥巴马宣布启动美国创新性神经技术大脑研究计划(Brain Research through Advancing Innovative Neurotechnologies (BRAIN) Initiative)[100, 101]。该计划通过绘制脑部的动态图像，研究大脑功能和行为之间的联系，意在阐述大脑对信息的处理、记忆以及检索过程，改变人们对大脑的认知。同年，欧盟15国推出了欧盟人类脑计划(Human Brain Project)[102]。该计划侧重于通过超级计算机技术来模拟大脑的运行过程，以此来实现强人工智能。2014年，日本推出了日本

大脑研究计划(Brain Mapping by Integrated Neurotechnologies for Disease Studies, Brain/MINDS)[103]。该计划的目标是融合各类灵长类动物中的相关神经科学证据，构建统一的哺乳动物脑发育及疾病发生模型。2016年，澳大利亚推出了澳大利亚脑计划(Australian Brain Initiative)[104]。该计划包含健康、教育以及新产业三个方面，主要目标包括理解以及修复大脑的异常处理机制，发展神经界面记录脑活动数据，推进对大脑的认知以构建新的类脑计算模型。同年，中国经过多年的酝酿，推出了为期15年的中国脑计划(China Brain Project)[105]。该计划拟形成以脑认知功能的解析和技术平台为一体，以认知障碍等相关重大脑疾病诊治研究和类脑计算与脑机智能技术为两翼的‘一体两翼’的研究格局。

依托神经科学的方法研究大脑的结构及行为方法，最终指导实现类脑智能是类脑计算的终极目标之一(也是各国大力推动脑计划的目标之一)。根据图1.3可知，神经拟态芯片(Neuromorphic Chips，类脑芯片)是搭建类脑计算平台的硬件基础。但是，由于大脑中使用脉冲传递信息，现有的硬件结构无法高效地实现类脑计算平台。2009年，Furber及其团队推出SpiNNaker(Spiking Neural Network Architecture)芯片计划[106, 107]，拟通过100万个ARM微处理器搭建一个异步通信的大脑电子模型，作为欧盟脑计划的类脑计算平台。2018年，目前世界上最大的神经形态超级计算机SpiNNaker正式启用，每秒可执行200万亿次操作。基于脉冲的事件驱动特性，已有一些硬件研究致力于使用异步电路搭建极低功耗的类脑芯片。2011年，IBM公司率先推出一款用异步电路搭建的类脑芯片，名为TrueNorth芯片。2014年，第二代TrueNorth芯片被推出[108–111]。该芯片简化了人类大脑神经结构，每个内核包含256个神经元、256个轴突以及6400个突触。其4096个内核可以模拟100万个神经元以及2.56亿个突触。相对于传统神经网络芯片，该芯片的功耗仅为1/10,000。2018年，英特尔公司推出Loihi芯片[112–114]。该芯片采用异步电路搭建，可以像大脑一样通过脉冲传递信息，自主地调节突触权值强度。芯片内部包含128个计算核，每个核可以模拟1024个神经元以及1.3亿个突触。根据英特尔给出的数据表明Loihi芯片的学习效率比其他智能芯片高100万倍，而在完成同一个任务所消耗的能源可节省近1000倍。

由各国脑计划的研究目标以及类脑芯片的研究可知，以脉冲方式在人工神经网络中传递信息并且搭建高性能的脉冲神经网络是未来类脑计算发展的一个必经环节。根据各国脑计划的相关内容，可以作出类脑计算的技术体系图，如图 1.3所示。从图 1.3中硬件层可知，基于异步电路的脉冲神经芯片可以用来低功耗地运行脉冲神经网络，忆阻器可以用来加速脉冲神经网络的STDP学习过

程[70–75]，内存计算可以减少脉冲神经网络中的数据传递。因此，相对于传统人工神经网络，脉冲神经网络更容易在类脑芯片上部署。从图 1.3中基础理论层可知，脉冲神经网络可以加深神经科学家对大脑可塑性以及信息处理机制的理解。从图 1.3中应用层可知，由于脑机接口以及智能假体均需采集人类神经元活动的脉冲信号，脉冲神经网络是该类任务最天然的人工神经网络处理工具。综上所述，脉冲神经网络与类脑计算有着千丝万缕的联系，是类脑计算的一个重要研究方向。

1.2 两种典型的人工神经网络

由上一节可知，卷积神经网络是现阶段处理复杂人工智能任务性能最好的一类人工神经网络，而脉冲神经网络是未来类脑计算研究中一个不可缺少的部分。脉冲神经网络模型转换算法与这两种人工神经网络息息相关。本节简述了人工神经网络中的常见的卷积神经网络以及脉冲神经网络等两种神经网络的特点，指出了脉冲神经网络在未来类脑计算中存在的巨大可能性。

1.2.1 卷积神经网络

卷积神经网络(Convolutional Neural Networks，简称CNN)指的是一类包含卷积运算操作且具有深度结构的前馈神经网络(Feedforward Neural Networks)，是深度学习最具代表性的算法之一。卷积神经网络解决了传统方法当中的多种弊端，包括梯度爆炸，梯度消失、参数过多等等，采用多层网络结构层次化地处理数据信息，大幅提升了人工神经网络在图像识别、语音识别、目标检测等等任务上的效果。因此，该算法的诸多变式成为了各领域的人工智能任务中的一种主要的新工具[25–33]。

同人类的大脑高效处理信息的能力相比，深度卷积神经网络算法还存在着一些局限性：

1. 深度卷积神经网络的网络规模很大，如GoogLeNet v1有22层[115]，需要很高的计算资源以及功耗开销。比如，训练围棋人工智能AlphaGo需要使用1200多块CPU和176块GPU[116]，需要消耗1000多美元的电费，但是大脑在思考的时候仅需消耗20瓦左右的功耗。

2. 深度卷积神经网络算法的自适应能力较弱。传统的深度学习网络需要在训练之前固定网络结构，同时在训练的过程中网络的结构无法被修改。这种方法有别于大脑的可塑性，削弱了模型的性能，甚至限制了深度神经网络算法的

发展。

3. 深度卷积神经网络算法的泛化性能较差。目前的神经网络模型基本上是数据驱动的统计模式识别算法，学习到的参数均与训练数据密切相关，通常难以处理训练数据中未出现过的类别和特征。当需要实时处理多变的动态数据时，比如脑机交互中的数据、光照变化的公路中的车辆数据等等，卷积神经网络并不适用。

4. 深度卷积神经网络的学习过程与大脑不同。深度卷积神经网络使用反向传播算法及其变式进行训练，网络每接收到一个训练样本数据，每个网络中的神经元都会接收到一个全局的误差信号对其突触权值进行调整。神经科学中发现大脑中神经元突触权值使用STDP法则(Spike-Time Dependent Plasticity)进行局部的、无监督的学习调整[117]，并不用反向传播算法进行学习调整。

5. 深度卷积神经网络传递信息方式与大脑不同。大脑中使用脉冲传递信息，这也就意味着二值信号已经足够用于实现相关人工智能。但是，卷积神经网络中使用连续的实数值传递信息。

深度卷积神经网络的上述特点，降低了单独使用卷积神经网络实现强人工智能的可能性。

1.2.2 脉冲神经网络

卷积神经网络成为研究热点的主要原因是其在各类人工智能任务上远超其他方法的优越性能。相反地，脉冲神经网络成为研究热点的主要原因是其巨大的可能性。不同于卷积神经网络中的计算方式，脉冲神经网络使用强度为0或1的脉冲传递信息。当输入脉冲到达脉冲神经元时，该神经元积累膜电势，当膜电势超过某阈值时，该神经元往外发送一个脉冲。基于以上更类脑的处理信息的方式，脉冲神经网络被称之为第三代人工神经网络[4]。

脉冲神经网络研究中的最大的可能性是实现低功耗的人工神经网络。由文献[3]可知，低功耗在近年来已经逐渐成为神经计算硬件实现中的一个主要话题。相对于卷积神经网络，脉冲神经网络的脉冲事件驱动特性使其非常容易被使用异步电路实现。由于异步电路在没有数据变化的时候不工作，因此如果我们获得一个像大脑那样使用稀疏脉冲传递信息的脉冲神经网络，会极大地降低人工神经网络的功耗。于此同时，由于脉冲脑神经网络中使用强度为0或1的脉冲传递信息，可以将卷积神经网络中使用实数值输入与突触权值进行乘累加的过程转换为脉冲神经网络中强度为0或1的脉冲与突触权值的累加过程。在硬件上，使用累加器替换乘累加器不仅可以大幅减少功耗开销，同时可以减小芯片

面积。

相对于卷积神经网络，脉冲神经网络更加有利于科研人员对大脑信息处理机制的理解。相对于脉冲神经网络，卷积神经网络相当于使用激发脉冲的平均频率来传递信息。这种信息传递方式会丢失脉冲序列中的很多时序信息，也许这些时序信息中包含有大脑高效信息处理的关键证据。例如，文献[35–39]等中指出生物神经系统中使用脉冲的时序信息进行编码。如果能够找到脉冲神经网络的学习算法可以获得某些人工智能任务上类人的效果，那么该脉冲神经网络中存在的脉冲序列传递方式，相对于卷积神经网络，更可能是大脑的实际处理信息的方式。

脉冲神经网络在某些交互产品中的应用场景效果可能更好。例如，脑机接口应用中采集的脑电信号以及智能假体中采集到的肌电信号。这些数据是天然适合脉冲神经网络处理的数据。同时，由于交互过程，如果使用卷积神经网络，最后还需要将网络的输出转换为脉冲序列输出。因此，脉冲神经网络更适应于这些应用的端到端部署。

最后，脉冲神经网络更容易在类脑芯片中部署。SpiNNaker芯片[106, 107]、TrueNorth芯片[108–111]以及Loihi芯片[112–114]等类脑芯片均支持脉冲神经网络的直接部署。同时，忆阻器等新型器件的研究，可以加速脉冲神经网络中的STDP学习过程[70–75]。

综上所述，脉冲神经网络在未来的研究中具有巨大的前景，但脉冲神经网络学习算法尚不成熟。

1.3 脉冲神经网络学习算法

脉冲神经元模型较卷积神经网络中的McCulloch-Pitts神经元模型[5]运算更加复杂，同时在脉冲发送附近的时刻还具有不连续性，因此脉冲神经网络的学习算法研究更为困难。脉冲神经网络训练算法主要包括直接训练算法和模型转换算法。

1.3.1 脉冲神经网络直接训练算法

根据脉冲神经网络直接训练算法的影响力将其分为基于反向传播算法的学习算法、基于STDP法则的学习算法以及其他的直接训练算法等三类。

脉冲神经网络的反向传播算法的基本原理与卷积神经网络的反向传播算法类似，但是由于脉冲神经元较卷积神经网络的神经元运算更复杂，同时在脉冲

发送时刻网络的不可微性，因此需对反向传播算法进行相关的改进。SpikeProp算法[76]及其扩展算法[77–82]是提出最早关注最多的一种脉冲神经网络反向传播学习算法。该算法具备求解非线性模式分类问题的能力，但是由于需要限制各层中脉冲数目，使得该算法获得的网络规模不大。随着深度卷积神经网络研究浪潮的到来，神经网络的深度结构被认为是神经网络完成复杂人工智能任务的关键因素之一。脉冲神经网络的反向传播算法也致力于发展深度结构，以获得性能更好的脉冲神经网络。一些算法已经取得了有竞争力的效果，其中典型算法包括[85–89]。总的来说，这类算法可以获得网络规模较大的脉冲神经网络，可以在某些数据集上取得和卷积神经网络类似的效果。但是，类似卷积神经网络的反向传播算法，这类算法只能进行离线训练，仅适用于静态数据的处理。同时，现实世界中获取的时空数据，比如脑机接口数据、功能性核磁共振数据、智能假体肌电数据，这些数据都是连续并且实时的。这类算法使脉冲神经网络失去了处理这类以脉冲序列为基本组成部分的数据的优势。

基于STDP法则的学习算法使用STDP法则作为突触权值调整的基本法则。该法则来源于神经生理学实验的结论，是突触权值可塑性机制Hebbian法则的重要扩展，更贴近于生物系统中学习过程。因此，从STDP法则出发构建脉冲神经网络学习算法，可以帮助神经科学家理解大脑的学习机制。这类算法包括监督学习算法[52–61]以及非监督学习算法[62–69]。其中，监督学习算法中最典型的算法是2010年Ponulak 和 Kasiński [55]提出的ReSuMe算法，该算法可以有效地提取复杂脉冲序列中的重要信息，但是仅适用于单个脉冲神经元的学习。文献[60, 61]提出的基于STDP法则的监督学习算法，获得了带有一个隐含层的脉冲神经网络。由于STDP法则只是提取脉冲序列中的局部特征，非监督学习算法中通常辅之以一些其他机制，使算法获得提取数据全局特征的能力。因此，基于STDP法则的非监督学习算法通常只获得规模很小的网络。直到近年来，该类算法才获得了多层脉冲神经网络[66–69]。总的来说，基于STDP法则的脉冲神经网络学习算法具有生物可解释性强的特点，同时支持在线学习，可以用来处理反向传播算法无法处理的实时动态时空数据。因此，这类算法在未来的研究中有很大的潜力及应用场景。在现阶段，这类算法的主要困难是无法在大的数据集上取得显著的效果，获得具有深度结构的脉冲神经网络。

其他的直接训练算法包括基于脉冲序列卷积的监督学习算法以及基于对比散度算法的监督学习算法。脉冲序列卷积算法使用特定的核函数将离散的脉冲序列转换为连续函数，然后设计算法进行学习过程，典型算法包括[90, 91]。这

类算法仅适用于单神经元以及单层神经网络。基于对比散度的监督学习算法使用受限玻尔兹曼机作为基本的网络结构，通过改进的对比散度算法训练脉冲神经网络，其典型算法包括[92–96]。由于受限玻尔兹曼机结构复杂，使该类算法很难扩展为深度网络，故后续研究较少。

综上所述，脉冲神经网络的直接训练算法中存在的主要问题是无法获得深度脉冲神经网络，以获得和深度卷积神经网络类似的效果。

1.3.2 脉冲神经网络模型转换算法

脉冲神经网络的巨大可能性与脉冲神经网络在实际中的应用之间还存在着巨大的鸿沟。究其原因是因为脉冲神经网络无法训练获得深度结构，以获得与深度卷积神经网络在人工智能任务上类似的效果。为了弥补脉冲神经网络与卷积神经网络在性能上的差距，同时获得脉冲神经网络的深度结构，一个自然的想法是将卷积神经网络模型转换为脉冲神经网络模型。2015年，Cao 等 [97]将卷积神经网络进行裁剪，以适应脉冲神经网络的特性，获得了一个5层的脉冲神经网络。由于脉冲神经元无法表示卷积神经网络中对应神经元大于1的输出值，该算法中存在着脉冲饱和的问题。文献中[99, 118]提出参数规范化算法来解决该问题，但是这些算法又带来了低效脉冲的问题[99]，从而延长了脉冲神经网络的收敛时间。总的来说，模型转换算法可以帮助脉冲神经网络在空间域上获得与卷积神经网络类似的信息处理能力。但是网络的收敛延迟问题是阻碍该算法在实际中应用的主要因素。

脉冲神经网络模型转换算法的研究表明，以现有的脉冲神经元模型，辅之以卷积神经网络中的卷积结构，深度脉冲神经网络就可以获得处理数据信息的认知能力。换就话说，不需要增加新型的脉冲神经元模型以及构造新型的脉冲神经网络结构，就可以获得性能显著的深度脉冲神经网络，从而使研究人员关注于脉冲神经网络学习算法的探索。更进一步地，从模型转换算法中的推导可知，卷积神经网络中的权值参数可以帮助脉冲神经网络在空间域上获得信息处理能力。当使用脉冲神经网络的模型转换算法中的权值参数初始化脉冲神经网络的直接训练算法，可以降低直接训练算法中学习过程的难度，使直接训练算法能够收敛到更好的局部极小值点。因此，本文针对模型转换算法中的脉冲饱和问题和低效脉冲问题，以及转换算法与直接训练算法的结合做了大量的研究与探索。

1.4 本文研究问题及行文组织

随着各国脑计划的不断开始部署以及类脑计算的兴起，脉冲神经网络的研究也越来越受到更多的关注。脉冲神经网络更类脑的事件驱动信息传递的方式，使其在类脑计算中具有广泛的前景。由于脉冲神经网络学习算法还无法获得与深度卷积神经网络类似的性能，脉冲神经网络还很难被部署到实际系统中。因此，开展对脉冲神经网络学习算法的研究十分有意义。脉冲神经网络的直接训练算法更贴近于生物神经系统中的学习过程，但是现有算法还无法在较大的数据集上取得良好的效果。因此，本文基于脉冲神经网络的模型转换算法，通过探索相关技术解决了脉冲网络模型转换算法中的脉冲饱和及低效脉冲问题，获得了识别精度与深度卷积神经网络类似的深度转换脉冲神经网络。

随后，在脉冲神经网络的模型转换算法的基础上，本文研究了脉冲神经网络与卷积神经网络的关系，发现卷积神经网络的权值参数可以帮助脉冲神经网络获得在空间域上处理信息的能力。在此基础上，针对脉冲神经网络直接训练算法难以获得深度脉冲神经网络的问题，本文从脉冲神经网络模型转换算法出发，简化了直接训练算法任务的难度，获得了一种极低延迟的深度脉冲神经网络转换学习算法。本文的具体章节组织结构如下：

第二章介绍脉冲神经网络的基本模型以及典型的学习算法。首先按照仿生性以及运算复杂度将脉冲神经元模型分成四类进行介绍，总结各种神经元模型的特性以及在脉冲神经网络学习算法中的使用情况。然后简要介绍了脉冲神经网络中的6种常见的脉冲编码方式。最后，分析了脉冲神经网络中的典型学习算法，将其分为监督学习算法、非监督学习算法以及模型转换算法进行讨论，总结相关算法的优势以及其中存在的问题，为后续模型转换算法研究提供了理论基础。

第三章针对脉冲神经网络模型转换算法中的脉冲饱和问题，研究了多强度深度脉冲神经网络模型转换算法，用于提升脉冲神经网络模型转换算法的模型识别精度。首先，提出一种多强度脉冲神经元模型，用于解决模型转换算法中脉冲神经元无法表达卷积神经网络中对应神经元大于1的输出值的问题。然后，推导了多强度脉冲神经网络模型转换算法模型精度等价性理论，并总结了多强度脉冲神经网络的优势。接着，使用脉冲神经元搭建了一个19层的多强度脉冲神经网络，并在CIFAR10数据集上将模型转换算法的识别精度从90.85%提升到94.01%。最后，针对该深度脉冲神经网络中存在的大量计算冗余问题，提出三种脉冲神经网络动态剪枝技术，减少转换脉冲神经网络中85%的冗余计算。

第四章提出了限制网络输出预训练算法以及错误脉冲抑制算法，获得了一个低延迟深度脉冲神经网络。为了降低多强度脉冲神经元模型在脉冲神经芯片中直接部署的难度，本章从模型转换算法本身出发，研究能同时解决脉冲饱和问题以及低效脉冲问题的有效算法。首先，提出一种限制网络输出预训练算法，可以在解决脉冲饱和问题的同时将低效脉冲问题的影响限制到最小。基本思想是将之前的参数规范化算法中的参数放缩过程迁移到卷积神经网络的训练过程中，这样可以在卷积神经网络的训练过程中将其神经元的输出值限定在指定值域，动态地完成之前算法中的参数规范化过程。然后，通过研究转换脉冲神经网络中各层脉冲序列的特征，发现了错误脉冲现象。接着，通过将错误脉冲问题抽象化为一个线性优化问题，提出了一种错误脉冲抑制算法。更进一步地，提出一种时序最大值池化算法，可以无损地将卷积神经网络中的最大值池化操作移植到转换脉冲神经网络中。最后，通过实验证明了本章提出的算法可以在保证识别精度的条件下，大大降低转换脉冲神经网络的网络收敛时间。

第五章提出一种基于反向传播的极低延迟深度脉冲神经网络转换学习算法，在模型转化算法中引入了基于反向传播算法的权值参数的学习过程，以进一步减少转换脉冲神经网络的网络收敛时间。首先，归纳了脉冲神经网络学习算法中常用的神经元模型，通过比较各种神经元模型的特点，确定本章算法中使用的LIF神经元模型。然后，基于第二章中关于脉冲神经网络学习算法的分析，选定脉冲神经网络的反向传播算法来进一步降低转换脉冲神经网络的识别延迟。接着，总结了训练深度脉冲神经网络的反向传播算法所需满足的三个严苛条件，提出了使用卷积神经网络权值参数满足上述条件的参数初始化算法。更进一步地，提出误差最小化算法以及修改的损失函数优化参数初始化算法的效果。最后，通过实验表明，该算法可以获得一个极低延迟的深度脉冲神经网络。

第六章是对全文内容的总结。

第2章 脉冲神经网络基本模型与学习算法综述

脉冲神经网络由于更贴近生物神经元的处理信息的方式，在类脑计算中占据着十分重要的地位，同时在低功耗、算法可解释性以及脑机交互应用等方面有着天然优势。脉冲神经元模型较传统的人工神经元模型运算更为复杂，在脉冲发送时刻附近还具有很强的不可微性。因此，设计脉冲神经网络的学习算法更加的困难，现阶段可以获取深度脉冲神经网络的学习算法还十分缺乏。本章描述了现有的常用脉冲神经元模型，分析脉冲神经网络各种学习算法的优势以及其中存在的问题，为后续的研究提供了理论基础。

本章首先将典型的脉冲神经网络中的典型算法的相关特征在表 2.1中进行了总结。其次，介绍脉冲神经网络中常用的基本模型，包括常用的脉冲神经元模型以及脉冲的编码方式。脉冲神经元模型按照其仿生物性以及运算复杂度可以分为4类：类生物学的神经元模型、生物启示的神经元模型、IF神经元模型以及SRM神经元模型。然后，简述了脉冲神经网络中常用的6种脉冲编码方式。接着，本章选取了脉冲神经网络中典型的训练学习算法，将其分为脉冲神经网络监督学习算法、脉冲神经网络非监督学习算法以及脉冲神经网络模型转换算法等3类，并分析了各类算法的主要优势与其中存在的问题。最后，本章对上述学习算法进行简要总结，引出论文中后续主要内容。

2.1 引言

尽管卷积神经网络在很多人工智能任务上取得了突出的效果，但是大脑皮质使用不同的方式来处理信息。相对于卷积神经网络，脉冲神经网络使用脉冲传递信息以及事件驱动的特性更加贴近大脑皮质处理信息的方式。同时，文献[4]指出脉冲神经网络相对于卷积神经网络消耗更少的功耗，将其称之为第三代人工神经网络。基于以上原因，脉冲神经网络还可以推动硬件新的体系结构的探索[108, 112, 119]。举例说明，TrueNorth芯片[108]实现了一个非冯诺依曼结构(non-von Neumann Architecture)，该芯片成功模拟1,000,000个可编程的脉冲神经元的动态运行过程，同时其256,000,000个神经元突触是可配置的。该芯片的功耗仅为传统卷积神经网络处理器的1/10,000。更进一步地，关于脉冲神经网络的研究可以推动对人类大脑运行机制的理解，极有可能导出强人工智能。除此之外，一些任务天然包含脉冲类型数据，比如脑机接口数据，这种任务更适

合脉冲神经网络处理。但是，迄今为止，在大数据集上，脉冲神经网络的训练算法还无法取得深度卷积神经网络相似的识别精度。这一点使得脉冲神经网络的诱人前景黯然失色。

表 2.1 典型的脉冲神经网络学习算法总结表。

Table 2.1 Spiking Neural Networks Learning Algorithms.

Algorithm	Training Type	Learning Rule	Neuron Model	Model Size	Datasets
ReSuMe[55]	Supervised	STDP-based	LIF/HH/IM*	Single neuron	-
SPAN[90]	Supervised	Sequence Convolution	LIF	Single neuron	-
Tempotron[84]	Supervised	BackPropagation	-	Single neuron	-
STDP-network[60]	Supervised	STDP-based	IF/LIF	One hidden layer	MNIST
SWAT[56]	Supervised	STDP-based	IF/LIF	One hidden layer	-
STDP-network[61]	Supervised	STDP-based	IF/LIF	One hidden layer	MNIST
Spike RBM[120]	Supervised	Contrastive Divergence	IF	One hidden layer	MNIST
Spike DBN[121]	Supervised	Contrastive Divergence	IF	One hidden layer	MNIST
Event-driven CD[92]	Supervised	Contrastive Divergence	IF	One hidden layer	MNIST
Spike RDBN[93, 94]	Supervised	Contrastive Divergence	IF	One hidden layer	MNIST
SpikeProp[76]	Supervised	BackPropagation	SRM	One hidden layer	XOR, Fisher Iris
RProp[78]	Supervised	BackPropagation	SRM	One hidden layer	XOR, Fisher Iris
SpikePropAd[79]	Supervised	BackPropagation	SRM	One hidden layer	XOR, Fisher Iris
EvSpikePropRT[80, 81]	Supervised	BackPropagation	SRM	One hidden layer	XOR, Fisher Iris
Multi-SpikeProp[82]	Supervised	BackPropagation	SRM	One hidden layer	XOR, Fisher Iris
SuperSpike[122]	Supervised	BackPropagation	LIF	One hidden layer	-
Lee[85]	Supervised	BackPropagation	LIF	Two hidden layer	MNIST, N-MNIST
STBP[86, 87]	Supervised	BackPropagation	LIF	5Conv+2Linear	MNIST, CIFAR10
SLAYER[88]	Supervised	BackPropagation	SRM	2Conv+1Linear	MNIST, N-MNIST
HM2-BP[89]	Supervised	BackPropagation	LIF	1Conv+1Linear	MNIST, N-MNIST
Panda SpikeCNN[123]	Unsupervised	Auto-Encoder	LIF	3Conv+1Linear	MNIST, CIFAR10
SDNN[67]	Unsupervised	STDP-based	LIF	3Conv+2Linear	MNIST
Diehl[66]	Unsupervised	STDP-based	LIF	1Excitory+1Inhibitory	MNIST
LM-SNN[68, 69]	Unsupervised	STDP-based	LIF	1Excitory+1Inhibitory	MNIST
CNN-SNN[97]	CNN-SNN	Converted	LIF	3Conv+2Linear	MNIST, CIFAR10
CNN-SNN[118]	CNN-SNN	Converted	LIF	3Conv+2Linear	MNIST, CIFAR10
CNN-SNN[124]	CNN-SNN	Converted	LIF	2Conv+2Linear	MNIST, CIFAR10
CNN-SNN[99]	CNN-SNN	Converted	LIF	5-16 layers	MNIST, CIFAR10
CNN-SNN[125]	CNN-SNN	Converted	LIF	3Conv+2Linear	MNIST, CIFAR10

* HH indicates the Hodgkin-Huxley model[126] and IM indicates Izhikevich model[127].

迄今为止，有大量的论文研究脉冲神经网络的训练算法，本章将其中一些典型算法列在了表 2.1中。这些算法可以被分为三类：脉冲神经网络监督学习算法(Supervised Learning)、脉冲神经网络非监督学习算法(Unsupervised Learning)以及脉冲神经网络模型转换算法(CNN-SNN Conversion)。从表 2.1中可以看出，脉冲神经网络监督学习算法是脉冲神经网络学习算法中研究最为广泛的一类学习算法，按照不同的学习规则其又可以被分为四类：基于突触权值可塑性

的监督学习算法(Hebbian-based or STDP-based Learning Algorithms)、基于对比散度算法的监督学习算法(Contrastive Divergence Learning Algorithms)、基于脉冲序列卷积的监督学习算法(Spike Sequence Convolution Learning Algorithms)以及基于梯度下降规则的监督学习算法(BackPropagation-based Learning Algorithms)。这类算法的早期研究中训练得到的脉冲神经网络的规模均较小(仅包含一个隐层)，同时只在较小的数据集上验证，例如XOR数据集以及Fisher Iris数据集。近五年来，逐渐出现一些较为有影响力的工作[85, 86, 88, 89]，获得脉冲神经网络的层数逐步加深，适用的数据集也在逐步加大。从表 2.1中可以看出，脉冲神经网络非监督学习算法还处在起步阶段，效果较好的工作还比较少。相较于其他两种脉冲神经网络学习算法，脉冲神经网络模型转换算法获得的脉冲神经网络的模型规模最大，最多可到16层。该类算法可以在MNIST数据集以及CIFAR10数据集上取得和卷积神经网络类似的效果。

2.2 脉冲神经元模型概述

常见的脉冲神经元模型主要包括四类：类生物学的神经元模型、生物启示的神经元模型、IF神经元模型以及SRM神经元模型。

2.2.1 类生物学的神经元模型

最具有代表性的类生物学的神经元模型是Hodgkin-Huxley神经元模型和Morris-Lecar神经元模型。

1952年，Hodgkin 和 Huxley [40]利用Cole发明的电压钳位技术获得了乌贼轴突的电生理活动的大量实验数据，并在这些数据的基础上推导出一个采用一阶四维非线性微分方程组进行系统描述的数学模型，称为Hodgkin-Huxley神经元模型。该模型主要描述了3种不同类型的离子电流，包括钠离子(Na^+)通道、钾离子(K^+)通道和主要包含Cl^-离子的漏电流通道，其电路等价模型如下图 2.1所示。细胞半透膜控制着细胞内外的离子浓度差，等价于一个电容器。当输入电流$I(t)$输入到细胞中，电容器会积累电荷，同时在细胞膜之间产生漏电流，如下式 (2.1)所示：

$$I(t) = C\frac{du}{dt} + \sum_k I_k(t) \tag{2.1}$$

其中u表示膜电势，等价电容器通过电流(capacity currents)为$I_c(t) = C\frac{du}{dt}$，电容C表示乌贼轴突神经元的脂质双层(lipid bilayer)构成的电容器。三个不同离子

通道电流(ionic currents)由下式 (2.2)决定：

$$\sum_k I_k(t) = g_{N_a} m^3 h(u - E_{N_a}) + g_K n^4 (u - E_K) + g_L(u - E_L) \tag{2.2}$$

其中E_{N_a}、E_K、E_L为三个通道的逆转电位(reversal potentials)，g_{N_a}、g_K、g_L为三个通道的电导率(conductance)，m、n、h为门变量(gating variables)。

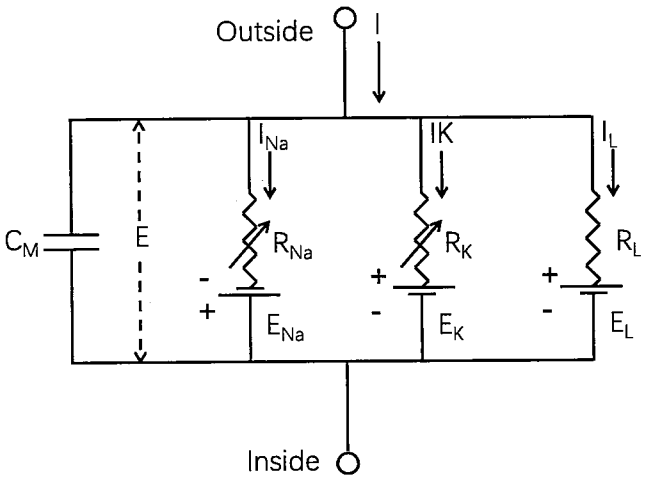

图 2.1 Hodgkin-Huxley神经元模型的等价电路图。

Figure 2.1 The equivalent circuit diagram of the Hodgkin-Huxley model.

假设每个离子通道的电导率(conductance)取决于细胞膜内外的浓度差，那么离子的分布就应该遵循热力学中的玻尔兹曼原理(Boltzmann's principle)。基于以上假设，得到门变量模拟离子通道开闭的动态过程，如下式 (2.3)- (2.5)所示：

$$\frac{dm}{dt} = \alpha_m(u)(1 - m) - \beta_m(u)m \tag{2.3}$$

$$\frac{dn}{dt} = \alpha_m(u)(1 - n) - \beta_n(u)n \tag{2.4}$$

$$\frac{dh}{dt} = \alpha_h(u)(1 - h) - \beta_h(u)h \tag{2.5}$$

上式 (2.3)- (2.5)中，系数$\alpha_m(u)$、$\alpha_m(u)$、$\alpha_h(u)$、$\beta_m(u)$、$\beta_n(u)$、$\beta_h(u)$计算如下：

$$\alpha_m(u) = \frac{0.01(10 - u)}{exp(\frac{10-u}{10}) - 1} \tag{2.6}$$

$$\alpha_m(u) = \frac{0.1(25 - u)}{exp(\frac{25-u}{10}) - 1} \tag{2.7}$$

$$\alpha_h(u) = 0.07exp(-\frac{u}{20}) \tag{2.8}$$

$$\beta_m(u) = 0.125exp(-\frac{u}{80}) \tag{2.9}$$

$$\beta_n(u) = 4exp(-\frac{u}{18}) \tag{2.10}$$

$$\beta_h(u) = \frac{1}{exp(\frac{30-u}{10}) + 1} \tag{2.11}$$

Hodgkin-Huxley神经元模型很重要，不仅在于它的参数具有生物学意义和可测量性，而且为研究人员探究突触整合、树突滤波、树突形态影响、离子流之间的相互作用等相关问题提供了模型基础。该模型消耗的运算量相当大，模拟该模型1ms需要消耗1200次浮点运算(假设指数运算仅消耗10次浮点运算)[127]。因此，使用该模型仅能模拟规模很小的神经网络。

1981年，Morris 和 Lecar [41]使用一个更简单的一阶二维非线性微分方程组简化Hodgkin-Huxley神经元模型，称之为Morris-Lecar神经元模型。该模型描述了巨型藤壶的肌肉纤维中Ca^{++}和K^+离子通道的各种震荡特性。该神经元模型如公式 (2.12)及公式 (2.13)所示：

$$C\frac{du}{dt} = I - g_L(u - E_L) - g_K n(u - E_K) - g_{Ca} m_\infty(u) \times (u - E_{Ca}) \tag{2.12}$$

$$\frac{dn}{dt} = \lambda(u)(n_\infty(u) - n) \tag{2.13}$$

其中u表示膜电势，I表示外加电流，C表示膜电容，g_L、g_{Ca}、g_K、为漏电通道、Ca^{++}、K^+等三个离子通道的电导率(conductance)，E_L、E_{N_a}、E_K为三个通道的逆转电位(reversal potentials)。

门变量$m_\infty(u)$、$n_\infty(u)$、$\lambda(u)$满足下式

$$m_\infty(u) = \frac{1}{2}(1 + tanh[\frac{u - u_1}{u_2}]) \tag{2.14}$$

$$n_\infty(u) = \frac{1}{2}(1 + tanh[\frac{u - u_3}{u_4}]) \tag{2.15}$$

$$\lambda(u) = \hat{\lambda} cosh[\frac{u - u_3}{2u_4}] \tag{2.16}$$

其中u_1、u_2、u_3、u_4表示稳态和时间常数的调整参数，$\hat{\lambda}$为参考频率。这组参数的一组常见的取值[127]为$C = 20\mu F/cm^2$，$g_L = 2mmho/cm^2$，$E_L = -50mV$，$g_{Ca} = 4mmho/cm^2$，$E_{Ca} = 100mV$，$g_K = 8mmho/cm^2$，$E_K = -70mV$，$u1 = 0mV$，$u2 = 15mV$，$u_3 = 10mV$，$u_4 = 10mV$，$\hat{\lambda} = 0.1s^{-1}$。该模型消耗600次浮点操作来模拟模型运行1$ms$[127]。

Hodgkin-Huxley神经元模型和Morris-Lecar神经元模型都是使用非线性微分方程组来精确地模拟生物神经元中电流变化的物理过程，但是这两个模型的运算复杂度都非常高。

2.2.2 生物启示的神经元模型

实现和大脑类似的认知功能并不完全需要精确地进行离子尺度的模拟，因此一部分神经元模型着眼于模拟生物神经元细胞的行为模式，从而减少模型的运算量。代表性神经元模型包括，FitzHugh-Nagumo神经元模型[42]、Hindmarsh-Rose神经元模型[43]和Izhikevich神经元模型[44]。

FitzHugh-Nagumo神经元模型(1961年)是Hodgkin-Huxley神经元模型的二维方程组简化版本。由于Hodgkin-Huxley神经元模型计算量巨大，同时是一个拥有四个变量的非线性微分方程组，进行数学分析十分困难。1950年左右，FitzHugh致力于寻找一种模型能够尽可能多的模拟Hodgkin-Huxley神经元模型的特性，同时大幅简化模型。他观察到Hodgkin-Huxley神经元模型中Na^+离子通道的激活门变量m的变化明显快于K^+离子通道的激活门变量n以及Na^+离子通道抑制门变量h的变化，通过近似地取m为平衡状态的值，且n和h之和为0.8，将Hodgkin-Huxley神经元模型的四维方程组简化为二维方程组。之后，Nagumo用二极管搭建电路，成功模拟了该方程组描述的神经元脉冲发放的动态特性，因此该模型称之为FitzHugh-Nagumo神经元模型。其数学模型如下公式 (2.17)以及公式(2.18)所示：

$$\frac{du}{dt} = a + bu + cu^2 + du^3 - v \tag{2.17}$$

$$\frac{dv}{dt} = \epsilon(eu - v) \tag{2.18}$$

其中，u表示神经元的膜电势的快变量，恢复变量v是一个慢变量，a、b、c、d、e、ϵ为控制方程组动态行为的常数值。该模型可以表现神经元兴奋、震荡以及双稳态等不同的动力学行为。该模型模拟$1ms$神经元运行，需要消耗72次浮点运算操作[127]。

Hindmarsh-Rose神经元模型(1984年)是另一种常见的简化神经元模型，主要用来描述自兴奋型神经元系统中的簇放电模式(the spiking-bursting behavior)，其描述方程组如下公式 (2.19)- (2.21)所示：

$$\frac{du}{dt} = v - F(u) + I - w \tag{2.19}$$

$$\frac{dv}{dt} = G(u) - v \tag{2.20}$$

$$\frac{dw}{dt} = \frac{H(u) - w}{\tau} \tag{2.21}$$

其中，u表示神经元的膜电势，v表示恢复变量，w表示慢变适应电流，I表示外部输入电流，τ为时间常数，F、G、H是一些特定的函数，通常选用如下：

$$F(u) = -au^3 + bu^2 \tag{2.22}$$

$$G(u) = c - du^2 \tag{2.23}$$

通常，给定$a = 1$、$b = 3$、$c = 1$、$d = 5$。函数H可选择的范围较多，可以用于描述神经元的一些混沌行为。该模型模拟1ms神经元运行，需消耗120次浮点操作[127]。

Izhikevich神经元模型(2003年)也是对Hodgkin-Huxley神经元模型的一种降维模拟。在尽可能多的保持Hodgkin-Huxley神经元模型在动力学方面表现出来的生物真实性的条件下，该模型大幅减少了模拟神经元所需的浮点运算次数(模拟1ms神经元运行，仅需要13次浮点运算操作[127])。该神经元模型的数学描述如下公式 (2.24)- (2.26)所示：

$$\frac{du}{dt} = 0.04u^2 + 5u + 140 - v + I \tag{2.24}$$

$$\frac{dv}{dt} = a(bu - v) \tag{2.25}$$

其中，u表示神经元膜电势，I表示外部输入电流，v表示膜电势的恢复变量。当膜电势u超过阈值时，神经元往外发送一个脉冲信号，同时产生如下变化：

$$if \quad u \geq +30mV, \quad then \begin{cases} u \leftarrow c \\ v \leftarrow v + d \end{cases} \tag{2.26}$$

其中，a、b、c、d均为控制参数。公式 (2.26)反映神经元发送脉冲之后的一段不应期。常用的一组参数为$(a, b, c, d) = (0.02, 0.25, -65, 8)$。Izhikevich神经元模型综合了Integrate-and-Fire神经元模型和Hodgkin-Huxley神经元模型的优点，既比较接近真实生物神经元的放电特性，能够复现尖峰放电和簇放电等已知的皮层神经元的放电类型，又便于进行大规模脉冲神经网络的仿真。

2.2.3 IF神经元模型

Integrate-and-Fire神经元模型(以下简称为IF神经元模型)是一系列简化脉冲神经元模型集合，对其他类型的脉冲神经元模型进行了极大的简化。

这类模型中应用最广泛的是著名的Leaky Integrate-and-Fire神经元模型(以下简称为LIF神经元模型)。不同于Hodgkin-Huxley神经元模型使用1个电容器以

及3个电阻器(对应3个离子通道)来精细地模拟神经元的动态特性，LIF神经元模型仅使用包含一个电阻器和一个电容器的简单电路来模拟神经元的变化，如下图 2.2所示。因此，该神经元模型只保留两个特性：1.神经元内部积累膜电势；2.膜电势超过阈值，往外发送脉冲。

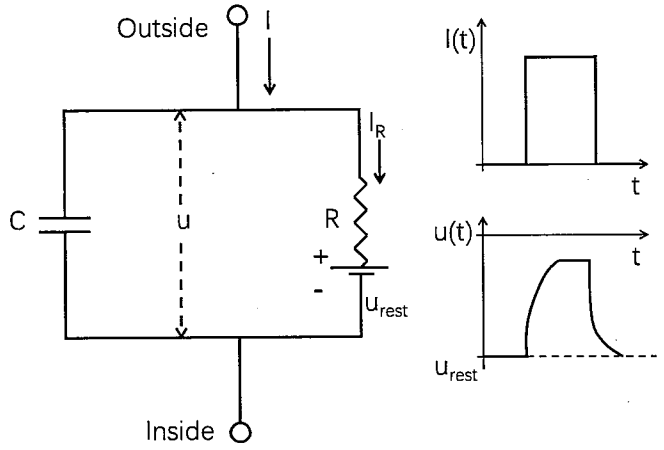

图 2.2 LIF神经元模型的等价电路图。

Figure 2.2 The equivalent circuit diagram of the Leaky Integrate-and-Fire neuron model.

LIF神经元模型可以用数学方式进行描述。首先，输入电流$I(t)$由公式 (2.27) 所示两部分电流组成：

$$I(t) = I_R + I_C \tag{2.27}$$

其中，第一项I_R为通过电阻器R的电流，通过欧姆定律可知

$$I_R = \frac{u_R}{R} = \frac{u - u_{rest}}{R} \tag{2.28}$$

其中，u_{rest}表示复位电势。第二项I_C表示电容器C上积累的电荷，根据定义满足下式：

$$I_C = \frac{dq}{dt} = C\frac{du}{dt} \tag{2.29}$$

其中，q表示电容器积累的电荷量。因此，根据上述公式 (2.27)- (2.29)可以推出

$$I(t) = \frac{u - u_{rest}}{R} + C\frac{du}{dt} \tag{2.30}$$

将公式两边同时乘上R，得到时间常数$\tau_m = RC$(通常称之为漏积分器，leaky integrator)，从而推出LIF神经元模型的基本表示：

$$\tau_m\frac{du}{dt} = -[u - u_{rest}] + RI(t) \tag{2.31}$$

由于只有一个变量，LIF神经元模型进行数值模拟比较简单，适合数学分析。但是因为简单，其无法反映相位刺激、爆发式放电和震荡反应等行为。该模型模拟1ms神经元运行，仅消耗5次浮点运算操作[127]。

LIF神经元模型认为只有发送脉冲的时间有意义，因此无法模拟脉冲的形状以及相关特性。一系列的Integrate-and-Fire神经元模型着眼于解决这个问题，其中最为常见的脉冲神经元模型主要包括Integrate-and-Fire with Adaptation神经元模型[127]、Integrate-and-Fire-or-Burst神经元模型[128]、Resonate-and-Fire神经元模型[129]以及Quadratic Integrate-and-Fire神经元模型[130]。

Integrate-and-Fire with Adaptation神经元模型通过增加一个线性微分方程来描述激活过程，具体公式如下：

$$\frac{du}{dt} = I + a - bu + g(d-u) \tag{2.32}$$

$$\frac{dg}{dt} = \frac{e\delta(t) - g}{\tau} \tag{2.33}$$

公式 (2.33)描述模型自适应地调节脉冲频率的过程，$\delta(\cdot)$表示狄拉克函数，I表示输入电流，τ是时间常数，a、b、d、e是控制参数。该模型模拟1ms神经元运行，消耗10次浮点运算操作[127]。

文献[128]提出了Integrate-and-Fire-or-Burst神经元模型来模拟下丘脑神经元模型，具体公式如下所示：

$$\frac{du}{dt} = I + a - bu + gH(u-u_h)h(u_T - u) \tag{2.34}$$

$$if \quad u = u_{thresh}, \quad then \quad u \leftarrow c \tag{2.35}$$

$$\frac{dh}{dt} = \begin{cases} -\frac{h}{\tau^-}, & if \quad u > u_h \\ \frac{1-h}{\tau^+}, & if \quad u < u_h \end{cases} \tag{2.36}$$

其中，I表示输入电流，u_{thresh}是电势阈值，a、b、c、g、u_u、u_T、τ^+、τ^-是相关控制参数，$H(\cdot)$表示单位阶跃函数。该模型模拟1ms神经元运行，消耗9-13次浮点运算操作[127]。

Resonate-and-Fire神经元模型是一个二维的Integrate-and-Fire神经元模型，具体公式如下：

$$\frac{dz}{dt} = I + (b + iw)z \tag{2.37}$$

$$if \quad Imz = a_{thresh}, \quad then \quad z \leftarrow z_0(z) \tag{2.38}$$

其中，z是复变量，其实值部分表示膜电势，I表示输入电流，a_{thresh}是电势阈值，b、w为控制参数，$z_0(z)$是任意描述神经元复位后时间相关性的函数。该模型模拟1ms神经元运行，消耗10次浮点运算操作[127]。

文献[131]为了探索低脉冲频率的神经网络首次提出Quadratic Integrate-and-Fire神经元模型。该模型的数学描述如下式所示：

$$\frac{du}{dt} = I + a(u - u_{rest})(u - u_{thresh}) \tag{2.39}$$

其中，u表示神经元膜电势，I表示输入电流，u_{rest}是复位电势，u_{thresh}是膜电势阈值，还需保证$a > 0$以及$u_{thresh} > u_{rest}$。该模型模拟1ms神经元运行，消耗7次浮点运算操作[127]。

2.2.4 SRM神经元模型

前面三类脉冲神经元模型使用微分方程组进行描述，而SRM神经元模型(Spike Response Model，简称SRM)[45]使用解析表达式来描述神经元的动态行为。该神经元模型可以看作是IF神经元模型的泛化形式。该神经元模型使用膜电势u作为显示的状态变量，使用不同的核函数η、ϵ以及κ用来描述神经元的输入脉冲以及外部刺激对神经元动态行为的影响。假设，第i个神经元在时刻t的膜电势$u(t)$的变化过程用下面公式 (2.40)描述

$$\begin{aligned} u_i(t) = \eta(t - \hat{t}_i) + \sum_j w_{ij} \sum_f \epsilon_{ij}(t - \hat{t}_i, t - t_j^{(f)}) \\ + \int_0^{\infty} \kappa(t - \hat{t}_i, s) I^{ext}(t - s) ds \end{aligned} \tag{2.40}$$

其中，$\hat{t}_i$表示该神经元i的上一次发送脉冲的时间，$t_j^{(f)}$表示前序神经元j的脉冲输入时间，I^{ext}表示外部输入电流，w_{ij}表示神经元i与神经元j之间的突触权值。

不同于IF神经元模型中固定的膜电势阈值V_{th}，SRM神经元模型的膜电势阈值取决于$t - \hat{t}_i$,如下式所示

$$V_{th} \quad \rightarrow \quad V_{th}(t - \hat{t}_i) \tag{2.41}$$

当神经元处在发送脉冲之后的不应期Δ^{abs}内，膜电势阈值V_{th}被设置成一个很大的值；当时刻$t > \hat{t}_i + \Delta^{abs}$时，膜电势阈值$V_{th}$逐渐衰减到神经元平衡状态的膜电势阈值。

当选取合适的核函数η、ϵ以及κ可以将SRM神经元模型映射成LIF神经元模型，如下式所示

$$\eta(s) = u_r exp(-\frac{s}{\tau_m}) \tag{2.42}$$

$$\epsilon(s,t) = \frac{1}{C}\int_0^s exp(-\frac{t'}{\tau_m})\alpha(t-t')dt' \tag{2.43}$$

$$\kappa(s,t) = \frac{1}{C}exp(-\frac{t}{\tau_m})\Theta(s-t)\Theta(t) \tag{2.44}$$

其中，$\Theta(\cdot)$为Heaviside函数，$\alpha(t-t')$表示前序神经元脉冲对神经元i的影响，u_r为初始的膜电势，τ_m是LIF神经元模型中的时间常数，C是电容器的值。

2.2.5 神经元模型总结

本节简单介绍了4类常见的脉冲神经元模型。神经元模型的性能，通常从神经元模型的仿生性与运算复杂度两个方面进行衡量。在文献[127]中，将生物神经元的最主要的重要特性总结为20种，包括Tonic Spiking，Phasic Spiking等等。文献[127]中根据各种神经元模型是否实现上述特性来确定神经元模型的仿生性能。同时，为了更有效地模拟更大规模的脉冲神经网络，神经元模型的运算复杂度在实际应用中也十分重要。根据各个神经元模型的数学方程组，可以统计出各个神经元模型所需要的运算操作数。因此，定性地画出各类神经元模型之间的仿生性与运算复杂度的示意图，如下图 2.3所示。

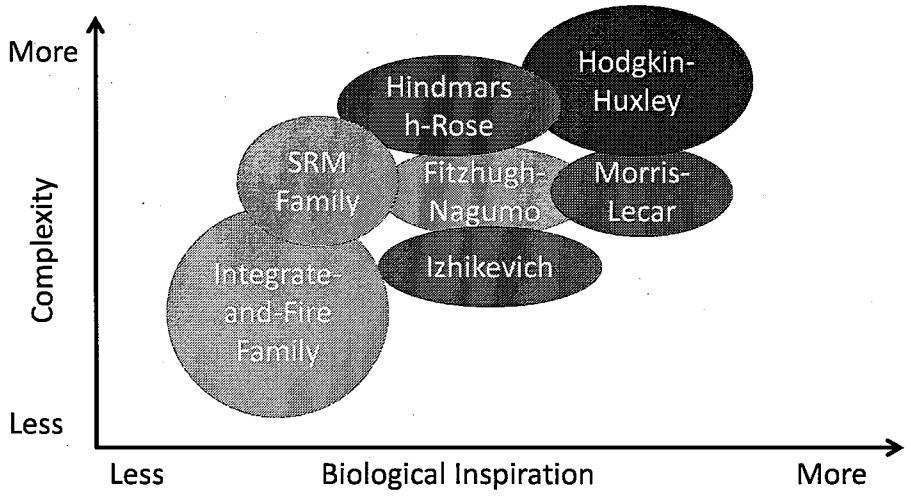

图 2.3 各类神经元模型之间在仿生性与运算复杂度的定性比较示意图。本图根据文献[127]中图2改绘。

Figure 2.3 A qualitative comparison of neuron models in terms of biological inspiration and complexity of the neuron model. This figure is from Figure 2 in the work[127].

由卷积神经网络在各种人工智能任务上取得的显著效果，可知简单的神经元模型已经可以对很多重要的智能模式进行模拟。相较于各种脉冲神经元模型而言，卷积神经网络中的神经元模型的仿生性远远弱于LIF神经元模型。由此说

明，神经元模型对大脑神经元动态特性模拟的完善性并不与最终实现的人工神经网络的性能有直接的关系。从脉冲神经网络模型转换算法的研究[97, 99]中，可以看出使用LIF神经元模型搭建的脉冲神经网络可以实现类似卷积神经网络的性能。由此，本文认为神经元模型中的膜电势累积过程以及脉冲发送规则足以模拟大部分大脑的运行机制。在现阶段，关于脉冲神经网络的研究应该集中于提升神经网络的规模以及性能，简单的神经元模型更有利于实现以上两个目标。因此，绝大部分脉冲神经网络学习算法中选取LIF神经元模型以及SRM神经元模型作为训练脉冲神经网络的神经元模型，如表 2.1中所示。

2.3 脉冲编码方式

大脑每一毫秒会接收到感觉神经元成千上万的脉冲信号，经过处理后决定给出何种对感觉神经元信号的适合反应。有时候，这个处理过程仅仅在数十毫秒内完成。大脑是如何在如此短的时间内完成上述处理的？到目前为止，这还是一个尚未解决的问题。与之相关的两个问题是：信息是如何编码成神经元信号的？进行精确运算所需的信号在时间域上的精度如何？这些问题都与神经元间的脉冲编码方式有关。

早在1926年，Adrian就证明了当给青蛙的皮肤上更大的外部压力，青蛙的皮肤感受神经元会向大脑发送更多的脉冲信号。该发现给脉冲的频率编码(rate code)提供了一些证据。频率编码方式在脉冲神经网络中进行实现以及理论推导十分简单，因此在神经生理学以及人工神经网络的研究中，这种频率编码方式都占据着十分重要的地位。比如，脉冲神经网络模型转换算法大部分都是基于频率编码方式的。但是该种编码需要一段时间收敛到合适的脉冲频率，因此相对于其他编码方式，该编码方式的效率太低。对于该种编码方式批评主要集中在两点：1.频率编码传递信息所需的时间消耗远大于大脑的很多行为反应所需时间；2.频率编码无法提取脉冲序列当中包含的时序信息。比如，Christopher deCharms [132]的研究表明在不改变脉冲频率的条件下，初级听觉皮层中的神经元可以通过神经元动作电位之间的相对时间来进行时序编码(temporal code)。

相对于频率编码，时序编码具有低功耗和速度快的特点。常见的时序编码方式有如下6种，如图 2.4所示。Time-to-first-spike编码表示每个神经元按照该神经元在该时间窗口中首次发送脉冲的时间来确定其强度，如图 2.4(a)所示，主要工作包括生理实验上证据[133, 134]以及神经网络中实现[135, 136]。Rank-order coding编码使用脉冲在神经网络中的先后顺序对其进行编码，如图 2.4(b)所示，

主要工作包括生理实验上证据[137–139]以及神经网络中实现[140, 141]。Latency code使用神经元之间脉冲的时间间隔进行编码，如下图 2.4(c)所示，主要工作包括生理学上的证据[51, 142]、编码本身的研究[143–145]以及神经网络实现[55, 146]。Resonant burst model[147]表示使用一系列爆发式脉冲决定后序神经元是否往外发送脉冲，如图 2.4(d)所示。Coding by synchrony表示当神经网络接收到同一物体的信息时，有相同的一群神经元往外发送脉冲，如图 2.4(e)所示，主要工作包括一些生理学上的证据[148–150]。Phase coding使用一个背景的共振函数的幅值来确定每个细胞的脉冲的强度值，如图 2.4(f)所示，主要工作包括一些生理学上的证据[151, 152]以及神经网络实现[136, 153]。

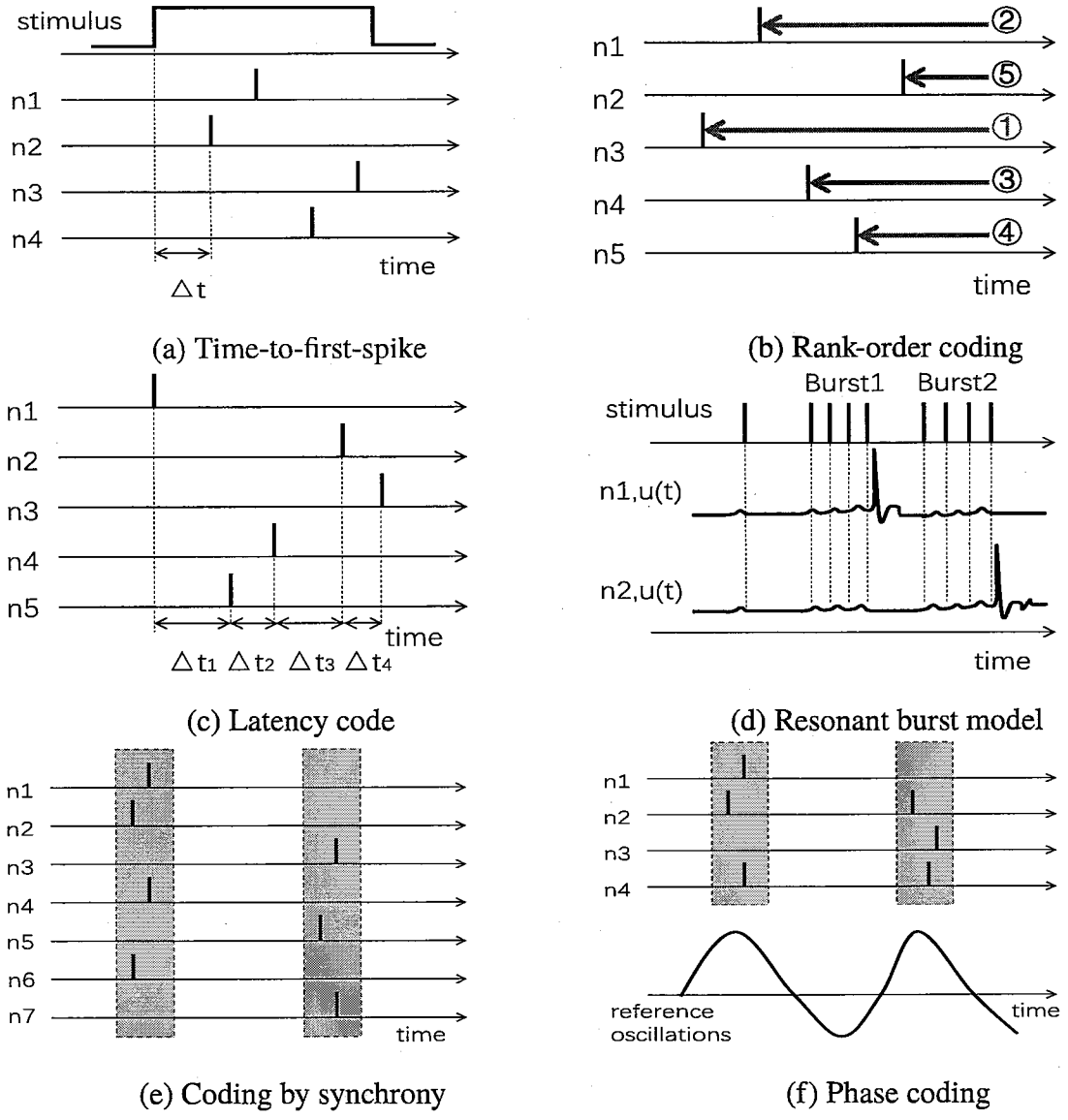

图 2.4 6种常见的基于时序编码的脉冲编码方式。

Figure 2.4 The 6 common spike coding methods based on the temporal code.

尽管时序编码方式在低功耗的实现以及信息传递速度上性能优于频率编码，但是单一的使用上述6种编码方式之一搭建脉冲神经网络还未取得良好的效果。已有一些高性能的脉冲神经网络学习算法可以提取脉冲序列间的时序信息，但是这些算法没有单独使用上述某一种时序编码方式，反而同时混含着频率编码和几种时序编码。如果能从这些神经网络中提取出上述的时序编码方式，结合生理学实验上的一些证据，可以推进神经科学家对大脑信息编码方式的更深刻的理解。

2.4 脉冲神经网络学习算法概述

脉冲神经网络的学习算法还处在研究的开始阶段，可以有效地训练深度脉冲神经网络的学习算法在现阶段还十分缺乏。本节通过脉冲神经网络监督学习算法、脉冲神经网络非监督学习算法以及脉冲神经网络模型转换算法等三个方面描述了现有的脉冲神经网络学习算法之间的关系，分析了各种学习算法之间的优势以及其中存在的问题，为后续研究提供了理论基础。

2.4.1 脉冲神经网络监督学习算法

卷积神经网络的监督学习是指将训练样本的数据输入神经网络，同时将得到的神经网络输出与数据的真实值比较得到误差信号，然后对该误差信号使用梯度下降法将神经网络的突触权值收敛到一个误差局部极小点(理想情况下希望为全局极小点)。当样本情况发生改变时，需要再次使用监督学习算法对突触权值进行修改，使其适应新的数据。生物学实验表明，生物的神经系统特别是感觉运动系统中存在着监督学习，但是生物神经元网络通过怎样的方式实现这一过程还没有明确的结论。对于脉冲神经网络来说，信息是以脉冲的形式表示的，神经元内部状态变量以及误差函数不再满足连续可微的性质，传统人工神经网络的监督学习算法(基于梯度下降法的反向传播算法，BackPropagation算法)已经不能直接使用。因此，脉冲神经网络的监督学习算法成为一个新的研究热点，根据突触权值的学习规则的不同，本小节将现有的脉冲神经网络监督学习算法分为基于突触权值可塑性的监督学习算法、基于脉冲序列卷积的监督学习算法、基于对比散度算法的监督学习算法以及基于梯度下降规则的监督学习算法。

2.4.1.1 基于突触权值可塑性的监督学习算法

1949年，Hebb [154]最先提出一个影响深刻的突触可塑性假说Hebbian法则，该假说十分简单但是反映出生物神经突触变化的本质。Hebbian法则如下：如

果两个神经元同时兴奋，则它们之间的突触增强，反之减弱。1997年，Ruf 和 Schmitt [50]最早提出了一种基于脉冲发放时间的监督Hebbian学习算法。在每个学习周期中，学习的过程由三个脉冲决定，突触权值的学习法则如下式所示：

$$\Delta w_{u,v} = \eta(t_v - t_o) \tag{2.45}$$

其中，u、v表示两个神经元，$\eta > 0$表示学习率。时间差$t_v - t_o$表示前序神经元u在t_0时刻的脉冲和后序神经元v在t_v时刻的脉冲时间上的差值，可以看作是学习过程中的误差。

生物神经网络中脉冲到来的时间顺序是传递信息的重要依据之一，Markram 等 [51]结合生物学实验提出了STDP法则(Spike Timing-Dependent Plasticity，脉冲时间依赖可塑性)。该法则是Hebbian法则的扩展，根据神经元脉冲的先后顺序，调整神经元之间连接的强弱。图 2.5为STDP法则的基本示意图，图中左侧子图表示突触后脉冲先于突触前脉冲到达，引起长时程抑制(long-term depression, LTD)；右侧子图表示突触前脉冲先于突触后脉冲达到，引起长时程增强(long-term potentiation, LTP)；蓝点表示[155]中神经元的实验数据。从此图中可以看出，如果一个神经元B的激活在另一个神经元A的激活之后很快就发生，时间差小于5ms时，A到B的连接权重就会增加约70%，而相反A到B的连接权重就会衰减20%。

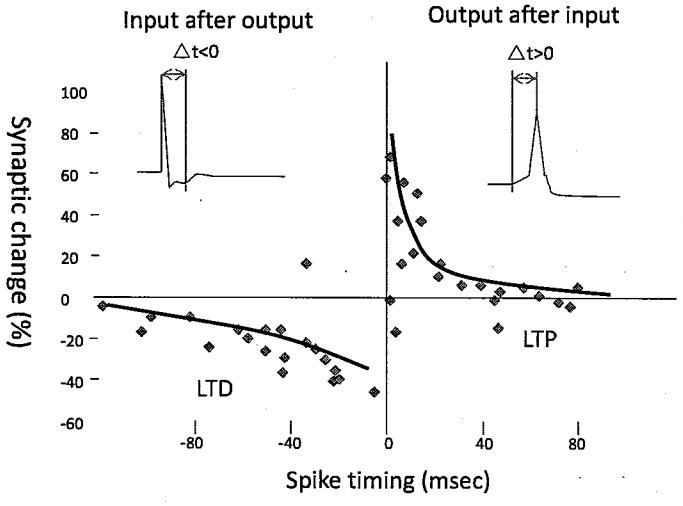

图 2.5 STDP法则示意图。根据[156]中图1重绘。

Figure 2.5 The diagram of the STDP rule. This figure is from Figure 1 in the work据[156].

2010年，Ponulak 和 Kasiński [55]通过将STDP法则和anti-STDP法则两种学习法则结合，提出了一种可对脉冲序列的复杂时空模式进行学习的远程监督学习

算法(Remote Supervised Method，ReSuMe算法)。ReSuMe算法是基于突触权值可塑性的监督学习算法中的一种典型算法，其突触权值的调整规则如下式所示：

$$\frac{d}{dt}w_{oi}(t) = [S_d(t) - S_o(t)][a_d + \int_0^{\infty} a_{di}(s)S_i(t-s)ds] \tag{2.46}$$

其中，$S_i(t)$和$S_o(t)$分别代表突触前的输入脉冲序列和突触后的输出脉冲序列，$S_d(t)$表示期望的输出脉冲序列，a_d是为了加速模型收敛的常量参数，积分核函数$a_{di}(s)$定义了脉冲时间顺序相关性所决定的突触可塑性。对于兴奋型突触，参数a_d为正值，积分核函数$a_{di}(s)$表示STDP法则；对于抑制型突触，参数a_d取负值，$a_{di}(s)$表示anti-STDP法则。其中积分核函数满足下式：

$$a_{di}(s) = +A_{di}e^{-\frac{s}{\tau_{di}}} \tag{2.47}$$

其中，A_{di}是常值参数，τ_{di}表示学习过程中的时间常数。该算法的具体过程如下图2.6所示，图中对于任意神经元i到神经元o的突触连接，$S_i(t)$、$S_o(t)$以及$S_d(t)$分别表示输入、输出以及期望脉冲序列，$W_{oi}(t)$表示突触权值的变化过程。

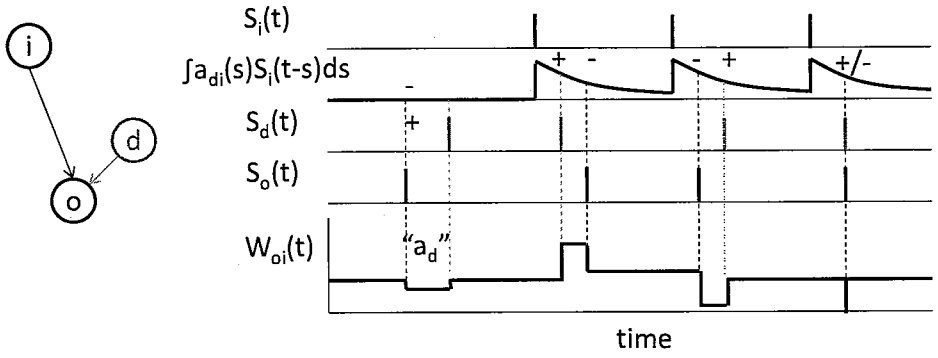

图 2.6 ReSuMe算法示意图。根据文献[55]中图1重新绘制。

Figure 2.6 Illustration of the ReSuMe learning rule. This figure is from Figure 1 in the work[55].

2.4.1.2 基于脉冲序列卷积的监督学习算法

由于脉冲序列是脉冲发放时间构成的离散事件的集合，可以使用特定的核函数$\kappa(t)$对这些离散事件进行卷积，从而将脉冲序列转换为连续函数，如下式所示：

$$\hat{S}(t) = S(t) * \kappa(t) = \sum_{t^f \in F} \kappa(t - t^f) \tag{2.48}$$

其中，$S(t)$和$\hat{S}(t)$分别代表卷积前后的脉冲序列的表示，t^f表示脉冲发放时刻，F表示脉冲发放时间构成的离散事件的时刻的集合。

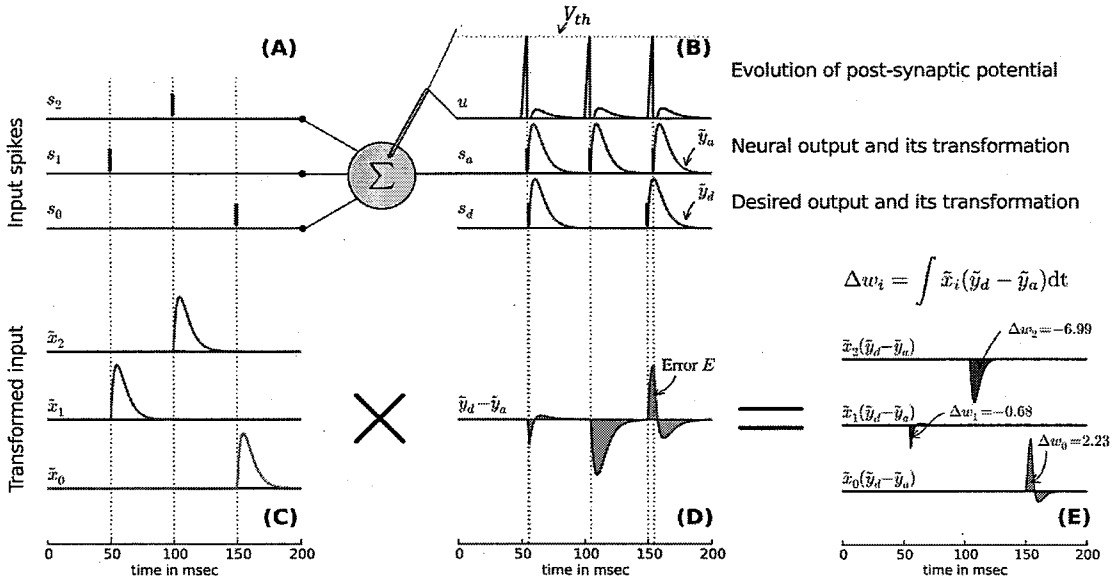

图 2.7 SPAN算法示意图。根据[90]中图1进行重绘。

Figure 2.7 Illustration of the SPAN learning algorithm. This figure is from Figure 1 in the work[90].

2012年，Mohemmed 等 [90]基于上述思想提出了SPAN算法(Spike Pattern Association Neuron)。该算法使用Widrow-Hoff学习法则调整突触权值，如下式所示：

$$\Delta w_i = \eta \int_0^{\infty} \hat{S}_i(t)[\hat{S}_d(t) - \hat{S}_a(t)]dt \tag{2.49}$$

其中，$\hat{S}_i(t)$、$\hat{S}_a(t)$、$\hat{S}_d(t)$分别表示突触前输入脉冲序列、突触后输出脉冲序列、期望的输出脉冲序列。卷积后脉冲序列通过下式计算：

$$\hat{S}(t) = \sum_{t^f \in F} \kappa(t - t^f) = \sum_{t^f \in F} \frac{e}{\tau}(t - t^f)e^{-\frac{t-t^f}{\tau}} H(t - t^f) \tag{2.50}$$

其中，$H(\cdot)$为Heaviside函数。使用Leaky Integrate-and-Fire神经元模型，代入上面公式得到突触权值调节公式如下：

$$\begin{aligned} \Delta w_i &= \lambda \int_0^{\infty} \Delta w_i(t)dt \\ &= \lambda(\frac{e}{2})^2[\sum_g \sum_f (|t_i^f - t_d^g| + \tau)e^{-\frac{|t_i^f - t_d^g|}{\tau}} - \sum_h \sum_f (|t_i^f - t_a^h| + \tau)e^{-\frac{|t_i^f - t_a^h|}{\tau}}] \end{aligned} \tag{2.51}$$

其中，t_i^f、t_a^h、t_d^g分别代表输入、输出、期望脉冲序列的脉冲发放时间。该算法的执行过程如图 2.7所示。图中一个输出神经元与三个输入神经元相连，子图(A)中三个输入脉冲序列s_i被卷积核转换为子图(C)中的连续函数(为了简化示意图，每个脉冲序列均只包含一个脉冲)，子图(B)中S_a和S_d分别代表实际输入以

及期望输出脉冲序列，在脉冲发送时刻t_a^0、t_a^1和t_a^2，经过卷积后得到$\tilde{y}_a$以及$\tilde{y}_d$。子图(C)中的序列值与子图(D)中的序列值相乘后积分，得到最终的突触权值变化，如子图(E)所示。该过程对应公式 (2.51)。

2013年，Yu 等 [91]受到SPAN算法的启发，提出了PSD算法(Precise-Spike-Driven算法)。不同于SPAN算法，在PSD算法只是将输入脉冲序列进行卷积，可以将其表示为突触电流$I_{PSC}^i(t)$，如下式所示：

$$I_{PSC}^i(t) = \sum_{t^f \in F} \kappa(t - t^f) H(t - t^f) \tag{2.52}$$

其中，t^f表示脉冲发放时刻，F表示脉冲发放时间构成的离散事件的集合。$H(\cdot)$为Heaviside函数，$\kappa(\cdot)$表示卷积核函数。推导出突触权值调整公式如下：

$$\begin{aligned} \Delta w_i &= \eta \int_0^{\infty} [S_d(t) - S_o(t)] I_{PSC}^i(t) dt \\ &= \eta [\sum_g \sum_f \kappa(t_d^g - t_i^f) - \sum_h \sum_f \kappa(t_o^h - t_i^f)] \end{aligned} \tag{2.53}$$

其中，$S_i(t)$、$S_o(t)$、$S_d(t)$分别表示突触前输入脉冲序列、突触后输出脉冲序列、期望的输出脉冲序列，t_i^f、t_o^h、t_d^g分别代表输入、输出、期望脉冲序列的脉冲发放时间。

2.4.1.3 基于对比散度算法的监督学习算法

对比散度算法(Contrastive Divergence)是受限玻尔兹曼机(Restricted Boltzmann Machine，简称RBM)的常用学习算法。受限玻尔兹曼机是Smolensky [157]提出的一种随机神经网络生成模型，包含可见变量(visible variable)与隐藏变量(hidden variable)，如下图 2.8中所示。图中v_d表示输入MNIST数据集数据的784个感觉神经元，v_c表示40个类别的标签神经元，W_h和W_c表示层间的突触权值矩阵。

2014年，Neftci 等 [92]根据原始的对比散度算法，为了适应脉冲神经网络的事件驱动的特性，将STDP法则与对比散度算法相结合，提出了一种事件驱动型的对比散度算法。在时间段$(0, 2T)$内，突触权值的平均变化如下式所示：

$$\langle \frac{d}{dt} w_{ij} \rangle_{(0,2T)} = \eta (\bar{v}_i^{+} \bar{h}_j^{+} - \bar{v}_i^{-} \bar{h}_j^{-}) \tag{2.54}$$

其中，$\bar{v}_i^{+}\bar{h}_j^{+}$表示采样的创建过程(construction phase)，$\bar{v}_i^{-}\bar{h}_j^{-}$表示采样的重创建过程，$\eta$是学习率，$\langle \cdot \rangle_{(a,b)} = \frac{1}{b-a} \int_a^b dt \cdot$。

类似的基于对比散度算法的监督学习算法研究还包括[93–96]。由于受限玻尔兹曼机本身的模型结构较为复杂，因此其很难被扩展到深度网络结构中，故这方面的研究成果较少。

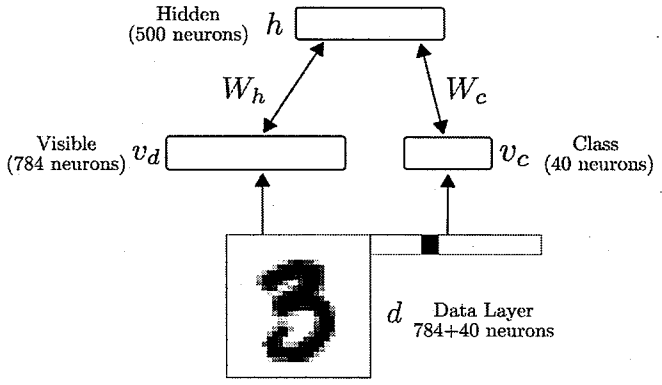

图 2.8 受限玻尔兹曼机网络结构示意图。根据[92]中图5进行重绘。

Figure 2.8 The RBM network consists of a visible and a hidden layer. This figure is from Figure 5 in the work[92].

2.4.1.4 基于梯度下降规则的监督学习算法

这一类算法借鉴人工神经网络中的基于梯度下降的反向传播算法(BP算法，BackPropagation)，采用类似的方式获得脉冲神经网络的突触权值变化量。

2002年，Bohte 等 [76]等人首次提出了适用多层前馈脉冲神经网络的误差反向传播算法，称之为SpikeProp算法。该算法使用了SRM神经元模型，并且为了克服神经元内部状态变量由于脉冲发送而导致的不连续性，限制网络中所有层的神经元只能发送一个脉冲。定义状态变量$x_j(t)$满足下式

$$x_j(t) = \sum_{t_i \in \Gamma_j} w_{ij} \epsilon(t - t_i) \tag{2.55}$$

其中，$\epsilon(t)$是脉冲反应核函数，w_{ij}表示神经元i与神经元j之间的突触权值，Γ_j表示前序神经元脉冲发送时刻t_i的集合。

假设在神经元j发送脉冲时刻t_j^a附近足够小的一段区域，t_j与神经元的状态变量$x_j(t)$可以用线性函数近似，如下图 2.9所示。令其满足$\delta t_j(x_j) = -\delta x_j(t_j^a)/\alpha$，则$\alpha$等于状态变量$x_j(t)$的局部导数：$\alpha = \frac{\partial x_j(t)}{\partial t}(t_j^a)$。按照卷积神经网络中的反向传播算法的推导过程，可以推出SpikeProp算法的链式法则如下

$$\Delta w_{hi}^k = -\eta y_h^k(t_i^a)\delta_i \tag{2.56}$$

其中，w_{hi}^k表示神经元h到神经元i的第k条突触权值，η表示学习率，$y_h^k(t_i^a)$表示脉冲反应函数，δ_i满足下式

$$\delta_i = \frac{\partial t_i^a}{\partial x_i(t_i^a)} \sum_{j \in \Gamma^i} \delta_j \frac{\partial x_j(t_j^a)}{\partial t_i^a} \tag{2.57}$$

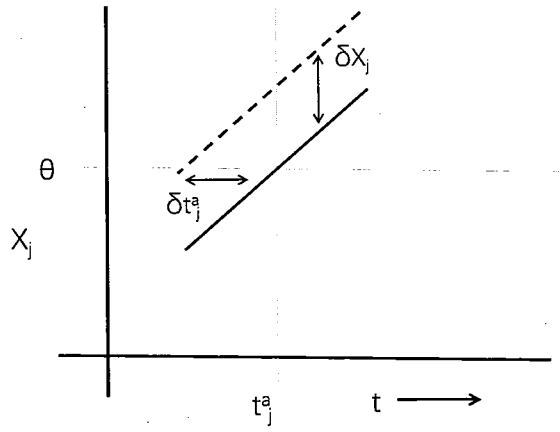

图 2.9 在时刻t_j附近很小区域内，δx_j与δt_j之间的关系示意图。根据[76]中图3进行重绘。

Figure 2.9 Relationship between δx_j and δt_j for an ϵ space around t_j.

其中，Γ^i是神经元i的后序神经元的集合。

SpikeProp算法是基于梯度下降规则的监督学习算法中研究最多的一类算法，很多研究基于该算法进行扩展。在提高网络收敛速度方面，通过对突触延迟、神经元脉冲发送膜电势阈值以及时间常量等等参量的优化提出了各种学习算法，包括RProp[78]、SpikePropAd[79]、EvSpikeRT[80, 81]。在限制脉冲发放个数方面，Ghosh-Dastidar 和 Adeli [82]提出的算法仅需要限制输出层发放脉冲数的个数为一。上述SpikeProp算法及其扩展受限于难以确定脉冲产生与消除的时刻，其脉冲个数被限制在固定的个数以内。由表 2.1中可知，这一系列算法获得的脉冲神经网络仅包含一个隐含层，故这一系列算法难以被扩展到深度脉冲神经网络的训练中。

现阶段已有一些脉冲神经网络的反向传播算法在网络规模方面取得了一定的效果，例如表 2.1中的算法[85, 86, 88, 89]。2016年，Lee 等 [85]将脉冲神经元的膜电势当作可微变量，同时将发送脉冲时刻的不连续性看作是脉冲神经网络中的噪声。这样就可以将误差反向传播机制应用到脉冲神经网络的训练中。在网络的输出层采用赢者通吃电路(Winner-Take-All Circuit)，也即是说，最早发送脉冲的神经元会侧向抑制同层其他脉冲神经元继续往外发送脉冲的行为。2018年，Wu 等 [86]将脉冲神经网络看作是类似循环神经网络的结构(Recurrent Neural Networks，简称RNN)[158]，只是将循环神经网络中的神经元的激活函数Sigmoid函数替换成脉冲发送机制。脉冲发送机制通常为一个类似Heaviside函数的阶跃函数，即是膜电势超过阈值，其值为1，否则为0。因此可以采用一个类似狄拉克函数的函数去逼近脉冲发送机制的函数的导数，从而采用类似于循环神经网络中随时间的反向传播算法(BackPropagation Through Time，简称BPTT)[159]来推

导脉冲神经网络的反向传播算法。

2018年，Jin 等 [89]通过脉冲发送频率上的反向传播获取频率编码误差，通过脉冲序列间的反向传播获取时序编码误差，同时将两种误差结合起来解决了脉冲神经网络反向传播算法中基于频率编码的损失函数与计算出来带有时序信息的梯度之间的不匹配问题。从而得到反向传播梯度如下式所示：

$$\Delta w_{ij} = \delta_i^k e_{i|j}^k (1 + \frac{1}{v}\sum_{l=1}^{r^{k-1}} w_{il} \frac{\partial e_{i|l}^k}{\partial o_i^k}) \tag{2.58}$$

$$\delta_i^k = \begin{cases} \frac{o_i^m - y_i^m}{v} & for \quad output \quad layer, \\ \frac{1}{v}\sum_{l=1}^{r^{k+1}} \delta_l^{k+1} w_{li} \frac{\partial e_{l|i}^{k+1}}{\partial o_i^k}, & for \quad hidden \quad layers. \end{cases} \tag{2.59}$$

其中，Δw_{ij}表示神经元i与神经元j之间的突触权值的变化，$e_{i|j}^k$表示第k层的神经元i与神经元j之间的脉冲之间的误差，o_i^k表示第k层的第i个神经元的脉冲输出，y_i^m表示输出层m的第i个神经元的标记输出，v表示神经元的膜电势。

2018年，Shrestha 和 Orchard [88]指出脉冲神经网络的反向传播算法中只有在脉冲神经元往外发送脉冲时，反向梯度才不为零。也就是说，只有往外发送脉冲的神经元在训练过程中学习突触权值的变化。通过提高脉冲神经元在初始阶段发送脉冲的频率，Shrestha 和 Orchard [88]提出的SLAYER算法有效地缓解了该问题。从而得到反向传播梯度如下式所示：

$$e^{(l)}(t) = \begin{cases} \frac{\partial L(t)}{\partial a^{n_l}}, & if \quad l = n_l \\ (W^{(l)})^T \delta^{(l+1)}(t), & otherwise \end{cases} \tag{2.60}$$

$$\delta^{(l)}(t) = \rho^{(l)}(t) \cdot (\epsilon_d \odot e^{(l)})(t) \tag{2.61}$$

$$\nabla_{W^{(l)}} E = \int_0^T \delta^{(l+1)}(t)(a^{(l)}(t))^T dt \tag{2.62}$$

$$\nabla_{d^{(l)}} E = -\int_0^T \dot{a}^{(l)} \cdot e^{(l)}(t) dt \tag{2.63}$$

其中，l表示脉冲神经网络的层数，$\rho(\cdot)$表示脉冲神经元发送脉冲不可微部分的导数的逼近函数，$a(\cdot)$表示卷积核$\epsilon_d(\cdot)$和神经元输入脉冲序列的卷积函数，$\odot$表示卷积操作。

总的来说，虽然较其他脉冲神经网络直接训练算法，该类算法可以获得网络规模更大的脉冲神经网络。但是，目前这类算法还无法获得网络规模超过10层的脉冲神经网络。

2.4.2 脉冲神经网络非监督学习算法

脉冲神经网络的非监督学习算法通常使用图 2.5中的STDP法则学习神经元之间的突触权值变化。由于STDP法则只能提取网络层间的局部特征，无法类似卷积神经网络的反向传播算法一样全局地提取网络层间的特征。因此，脉冲神经网络的非监督学习算法还未取得良好的效果，同时网络的规模也不是很大。脉冲神经网络的非监督学习算法的研究很多，其中效果比较好的有[66–69]。

2015年，Diehl 和 Cook [66]使用三层网络结构，网络输入层使用泊松分布将像素值转换成脉冲输入序列，兴奋神经元层以及抑制神经元层联合起来提取特征。当兴奋神经元层中的神经元往外发送脉冲时，对应位置的抑制神经元往外发送脉冲同时抑制兴奋神经元层中的其他神经元。采用如下STDP法则对脉冲神经元的突触权值进行调整：

$$\Delta w = \eta(x_{pre} - x_{tar})(w_{max} - w)^{\mu} \tag{2.64}$$

其中，Δw表示突触权值的变化值，η代表学习率，x_{pre}代表神经元输入突触的脉冲的迹(trace)，x_{tar}代表神经元输出突触的脉冲的迹，w_{max}代表突触权值的最大值，μ由之前的突触权值的更新过程决定。兴奋神经元可以通过STDP法则学习输入层的相关特征。当某个神经元学会某个数据的特征时，该神经元会很快往外发送脉冲，同时通过抑制神经元层的侧向抑制作用抑制兴奋神经元层其他的神经元学习该特征，从而不同的脉冲神经元可以学到不同的特征。最后，通过兴奋神经元层的神经元多数投票获得数据的最终分类。该算法主要过程如下图2.10所示，其中Input表示网络输入层，根据MNIST数据集的像素值使用泊松分布产生对应位置的脉冲输入序列；Excitatory Neurons表示网络的兴奋神经元层；Inhibitory Neurons表示抑制神经元层；Lateral Inhibition表示当兴奋型神经元往外发送脉冲时，激活其对应位置的抑制神经元，抑制型神经元同时抑制兴奋神经元层中的神经元往外发送脉冲。后续工作[68, 69](2018年)，简化了该脉冲神经网络的网络结构，同时支持卷积操作。最后，该工作可以获得一个无监督的自组织映射的多层脉冲神经网络。

2018年，Kheradpisheh 等 [67]提出一种基于STDP法则的多层卷积脉冲神经网络，其网络结构如图2.11所示。首先使用ON-和OFF-center的DoG滤波器(Difference of Gaussian filter)提取输入图像的边缘特征，然后根据图 2.4中的Rank-order coding编码方式获得脉冲神经网络的输入脉冲序列。最后卷积层使用如下简化STDP法

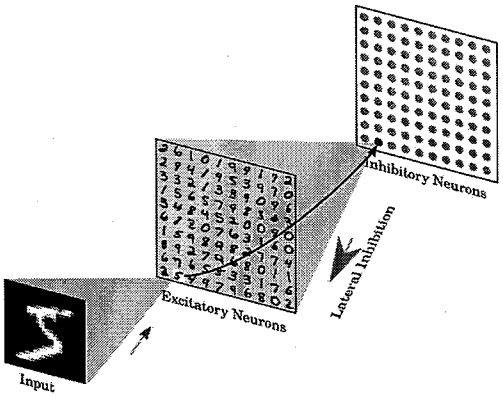

图 2.10 脉冲神经网络无监督学习算法[66]中的网络结构示意图。

Figure 2.10 Network architecture of the learning algorithm in the work[66].

则[64]学习神经元突触权值的变化Δw_{ij}：

$$\Delta w_{ij} = \begin{cases} a^{+} w_{ij}(1 - w_{ij}), & if \quad t_j - t_i \le 0, \\ a^{-} w_{ij}(1 - w_{ij}), & if \quad t_j - t_i > 0. \end{cases} \tag{2.65}$$

其中，i和j分别代表突触前神经元与突触后神经元，t_i和t_j是其对应的脉冲发送时间，a^{+}和a^{-}代表对应的学习率。接着定义变量C_l测量第l个卷积层的网络收敛情况，

$$C_l = \sum_{f} \sum_{i} w_{f,i}(1 - w_{f,i}) / n_w \tag{2.66}$$

其中，$w_{f,i}$代表第f个特征的第i个突触权值，n_w代表第l个卷积层的总的突触数目。当C_l小于0.01时，结束该卷积层的学习过程。最后训练一个线性支持向量机完成模式分类。

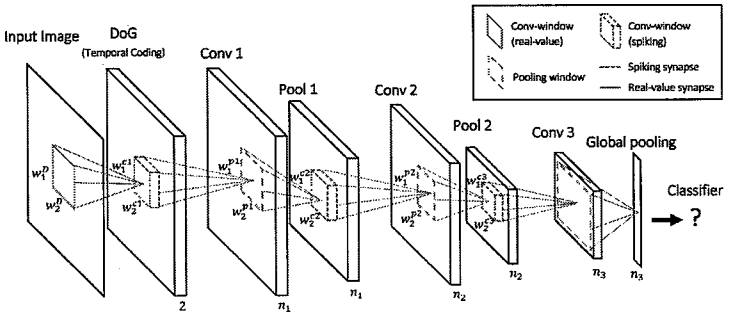

图 2.11 脉冲神经网络无监督学习算法[67]中的网络结构示意图。

Figure 2.11 A sample architecture of the proposed SDNN with three convolutional and three layers.

脉冲神经网络的非监督学习算法的研究还有文献[62–65]。目前，该类算法还未在较大的数据集上取得很有竞争力的网络性能。

2.4.3 脉冲神经网络模型转换算法

脉冲神经网络的监督学习算法以及非监督学习算法在现阶段还不成熟，训练出来的网络的规模较小，无法在实际的智能任务中取得较好的效果。与此同时，在硬件方面，很多脉冲神经网络被实现为低功耗的硬件系统，比如FPGA实现[160, 161]、TrueNorth芯片[162, 163]、SpiNNaker芯片[107]。因此，为了减小脉冲神经网络学习算法与硬件性能之间的间隙，一些研究者提出将传统的人工神经网络转化为脉冲神经网络的方法。这样既利用了成熟的人工神经网络训练算法来得到一个功能强大的深度神经网络，又可以获得脉冲神经网络的诸多优点，同时避免了直接训练脉冲神经网络的困难。

脉冲神经网络的模型转换算法起源于[96, 97]两项工作。在2013年，Pérez-Carrasco 等 [96]使用动态视觉感测器(Dynamic-Vision-Sensor，DVS[164, 165])获得地址事件表达(Address Event Representation)的脉冲序列，然后将卷积神经网络中的神经元转换成一种具有漏电流和不应期的脉冲神经元，从而获得等价的脉冲神经网络。由于动态视觉感测器现在还处在实验室研究阶段，还未在实际任务中大量应用，因此该研究还没有太多的后续研究。

在2015年，Cao 等 [97]使用泊松分布将数据集中的输入图片中的像素值转换为对应的脉冲输入序列。首先，使用反向传播算法训练卷积神经网络，获得相关的突触权值参数。接着，裁剪该卷积神经网络中不适合转换为脉冲神经网络的相关部分，例如替换卷积神经网络中的最大值池化操作为空间线性采样，将卷积神经网络中的偏置设置为零。最后将获得的卷积神经网络转换为对应的脉冲神经网络。该算法使用脉冲神经元的脉冲发送频率逼近卷积神经网络中神经元的输出值，其最大问题是选用的脉冲神经元模型无法表达卷积神经网络中大于1的输出值，被称之为脉冲饱和问题。Diehl 等 [118]提取基于模型的参数正则化算法以及基于数据的参数正则化算法缓解了该问题，提升了转换脉冲神经网络的识别精度，但是这两种算法带来了低效脉冲问题，增加了转换脉冲神经网络的网络收敛时间。Rueckauer 等 [99]提出鲁棒参数正则化算法，可以折中地考虑转换脉冲神经网络中的识别精度以及网络收敛时间，但是没有从本质上解决低效脉冲问题。

总的来说，模型转换算法中的脉冲饱和问题以及低效脉冲问题限制了转换脉冲神经网络获得深度结构，从而影响了转换脉冲神经网络的识别精度以及网络收敛时间。

2.5 本章小结

本章对脉冲神经网络的中的代表性工作进行了回顾和总结，包括脉冲神经网络的基本神经元模型以及相关训练学习算法。脉冲神经元模型按照神经元模型的仿生物性及其运算复杂度可以分为4类：类生物学的神经元模型、生物启示的神经元模型、IF神经元模型以及SRM神经元模型。其中，前两种神经元模型更加类似于生物神经元的行为模式，但是运算复杂度更高。因此，脉冲神经网络学习算法中使用的最多的还是后两种神经元模型。同时，后两种神经元模型的行为模式可以提取出脉冲序列中足够的特征。接着，本章选取了典型的脉冲神经网络学习算法，分析这些算法的特点，将其分为脉冲网络的监督学习算法、脉冲神经网络非监督学习算法以及脉冲神经网络模型转换算法等3类。最后，通过分析这3类算法的特点，发现脉冲神经网络模型转换算法可以获得识别精度与卷积神经网络类似的效果，但是这类算法无法获得深度转换脉冲神经网络以获得更好的网络识别精度，同时转换网络的收敛时间过长。后文将针对该问题，提出三种有效的解决办法。

第3章 多强度深度脉冲神经网络模型转换算法

脉冲神经网络模型转换算法是现阶段最有希望获得接近深度卷积神经网络性能的一类脉冲神经网络学习算法[166]。该类算法通过分析脉冲神经元的脉冲发送机制，匹配卷积神经网络中的神经元的计算特性，使用卷积神经网络的权值参数获得了超过其他学习算法的高性能的脉冲神经网络。由于脉冲神经网络中脉冲神经元的脉冲发送频率小于1，这与卷积神经网络中无限制的神经元输出的特性相违背，造成了转换网络的逼近误差。这类误差会随着转换神经网络的网络规模的增加而逐层累积，因此该算法在转换深度脉冲神经网络时转换网络的精度会急剧减小。

本章针对脉冲神经元无法表达卷积神经网络中神经元输出值大于1的问题，提出了一种多强度的脉冲神经网络结构。首先，分析了脉冲神经网络模型转换算法的相关理论，得到了脉冲神经元无法表达大于1的值的问题的原因。接着，基于上述问题的产生原因，设计了一种多强度的脉冲神经元，并且推导了多强度脉冲神经网络精确逼近卷积神经网络的理论。基于多强度脉冲神经网络的结构，获得了具有深度结构的转换脉冲神经网络，然后提出了三种动态剪枝技术，极大地减少了深度脉冲神经网络中的运算冗余。最后，通过实验测试，证明了本章提出的多强度脉冲神经网络不论是在脉冲神经网络的识别精度，还是收敛速度上，均优于同期的模型转换算法。同时，三种动态剪枝算法在保持多强度脉冲神经网络的识别精度的条件下，移除了原始多强度脉冲神经网络中85%的运算操作。

3.1 脉冲神经网络模型转换算法相关理论

脉冲神经网络模型转换算法的基本原理是用脉冲神经元的脉冲发送频率去匹配卷积神经网络中对应位置的神经元的激活后的输出值。Cao 等 [97]首先提出一种将卷积神经网络的神经元的运算转换为脉冲神经网络的脉冲传递运算的算法。假设每个脉冲神经元每个时间步上更新其膜电势$V(t)$，那么在第t个时间步时，脉冲神经元的膜电势$V(t)$满足如下公式 (3.1)-公式 (3.3)：

$$V(t) = V(t-1) + L + X(t) \tag{3.1}$$

$$\text{如果} \quad V(t) \geq \theta, \quad \text{发送脉冲以及重置膜电势} V(t) = 0 \tag{3.2}$$

$$\text{如果}\quad V(t) < V_{min},\quad \text{重置膜电势} V(t) = V_{min} \tag{3.3}$$

其中，L是常数漏电流参数，$X(t)$表示在第t个时间步时前序神经元对该神经元的输入总和。当膜电势$V(t)$超过阈值θ时，神经元往外发送一个脉冲，并且膜电势$V(t)$复位为零；当膜电势$V(t)$小于V_{min}时，膜电势被设置为V_{min}。

假设该网络中使用7×7卷积核，那么在(i,j)位置的神经元在第t个时间步的输入$X(t)$定义为如下公式：

$$X_{ij}(t) = \sum_{p,q=-3}^{3} A_{p+i,q+j}(t) K_{pq} \tag{3.4}$$

其中，$A_{p+i,q+j}(t)$表示前一层网络的输出脉冲序列，K_{pq}表示7×7卷积核的共享权值参数。

定义脉冲发送频率$r(t)$为脉冲神经元在t个时间步内每个时间步的平均脉冲发送数目，如下公式所示：

$$r(t) = \frac{N_t}{t} \tag{3.5}$$

其中，N_t表示脉冲神经元在t个时间内发送脉冲的总数目。由公式 (3.1)以及公式 (3.5)可知，脉冲发送的总数目N_t不大于t。因此，脉冲神经元的脉冲发送频率$r(t)$不大于1。与此同时，被转换的卷积神经网络中神经元的输出值可能会超过1。换就话说，脉冲神经网络无法表达卷积神经网络中输出大于1的值，这就造成了一定的转换误差。当待转换的神经网络的层数加深时，这类误差会不断累积，于是这类误差就成为阻碍脉冲神经网络的识别精度以及收敛速度提升的主要困难之一。后续工作[98, 99, 118]通过一些参数规范化的办法缓解该问题，但是没有从本质上解决该问题。

3.2 多强度脉冲神经元网络模型转换算法

针对脉冲神经元无法表达大于1的值的问题，本节提出了一种多强度LIF神经元模型，随后基于该神经元模型推导了卷积神经网络与多强度脉冲神经网络的精度等价性理论；最后分析了多强度脉冲神经网络相较于原始的转换脉冲神经网络的优势。

3.2.1 多强度LIF脉冲神经元模型

脉冲神经网络模型转换算法中较常用的模型为Leaky Integrate-and-Fire神经元模型(以下简称为LIF神经元模型)。为了解决LIF神经元无法表示大于1的值的

问题，本章提出一种多强度的LIF脉冲神经元模型。该神经元模型的结构如下图3.1所示。

在第t个时间步时，第l层的第i个多强度LIF神经元模型的膜电势为$v_i^l(t)$。该膜电势受到输入$z_i^l(t)$的影响如下：

$$z_i^l(t) := \lambda I_i^l(t) := \lambda(\sum_{j=1}^{M^l} w_{ij}^l \theta_{j,t}^{l-1} + b_{i,t}^l) \tag{3.6}$$

其中，第l层的第i个神经元在t时刻产生的多强度脉冲$\theta_{i,t}^l$由2部分影响：1.时刻t之前的残余膜电势$v_i^l(t-1)$；2.时刻t的输入。具体过程如下式所示：

$$\theta_{i,t}^l := floor(\frac{max(v_i^l(t-1) + z_i^l(t), 0)}{\tau}) \tag{3.7}$$

多强度LIF神经元不断积累输入$z_i^l(t)$直到膜电势$v_i^l(t)$超过电势阈值V_{th}。为方便后续讨论假设$\tau = V_{th}$。当膜电势$v_i^l(t)$超过τ时，多强度LIF神经元向外发送一个多强度脉冲信号，同时膜电势减小$\tau\theta_{i,t-1}^l$。膜电势$v_i^l(t)$的动态变化过程由下式决定：

$$v_i^l(t) = v_i^l(t-1) + z_i^l(t) - \tau\theta_{i,t-1}^l \tag{3.8}$$

其中，$\theta_{i,t-1}^l$是一个非负整数。

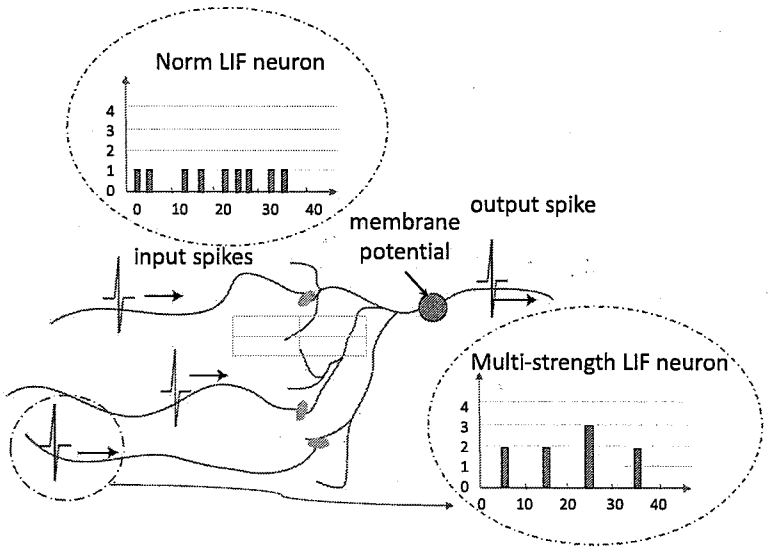

图 3.1 多强度LIF脉冲神经元模型。左上角为常规LIF神经元模型的脉冲产生情况，右下角为多强度LIF神经元模型的脉冲产生情况。

Figure 3.1 Comparison of the multi-strength LIF neuron model and the normal LIF neuron model. Sparer inputs are sent into the multi-strength LIF neuron model.

3.2.2 转换算法模型精度等价性推导

通常卷积神经网络有2个基本的组成部件：卷积核和激活函数。其中全连接层可以看作是全局大小的卷积核。假设使用ReLU函数作为激活函数，那么卷积神经网络的计算过程描述如下：

$$a_i^l := max(0, \sum_{j=1}^{M^{l-1}} w_{ij}^l a_j^{l-1} + b_i^l) \tag{3.9}$$

其中a_i^l表示第l层的第i个卷积神经元的输出，b_i^l是第i个神经元在第l层的偏置。

假设转换后的多强度脉冲神经网络采用与转换之前的卷积神经网络一样的网络结构。那么多强度脉冲神经网络(以下简称为M-SNN)由3个部分组成：输入脉冲序列、多强度LIF神经元以及输出脉冲序列。卷积神经网络的参数被等价地映射到多强度脉冲神经网络中，如图 3.2所示。在此图中，上方子图表示多强度脉冲神经网络中的卷积核结构，输入为前一层的多强度脉冲序列输出，多强度LIF神经元在该输入层上进行卷积操作获得多强度脉冲输出序列。下方子图为多强度脉冲神经网络的基本网络结构。

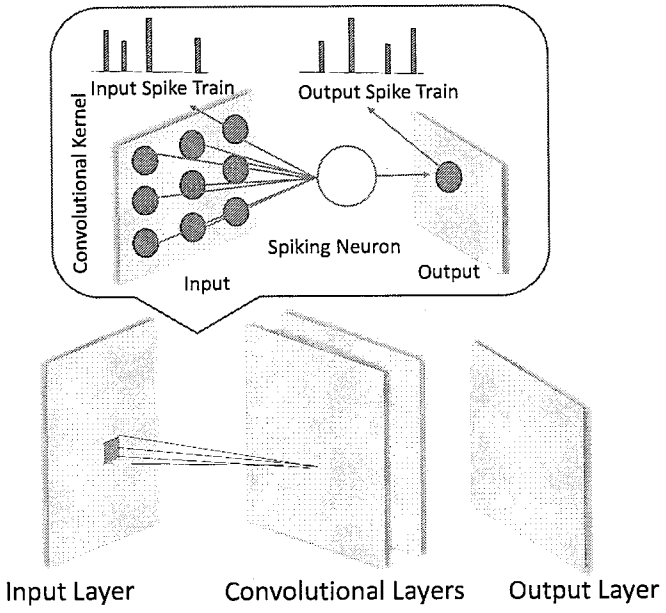

图 3.2 多强度脉冲神经网络(M-SNN)结构示意图。

Figure 3.2 The architecture of M-SNN.

现在来推导M-SNN可以精确地逼近卷积神经网络的理论。定义M-SNN的脉冲发送频率$r_i^l(T)$如下：

$$r_i^l(T) = \frac{N_i^l(T)}{T} = \frac{\sum_{t=0}^{T} \theta_{i,t}^l}{T} \tag{3.10}$$

其中$N_i^l(T)$是第l层的第i个神经元在T时刻内累积的脉冲总强度。

为了证明在T时刻，$r_i^l(T)$可以趋近于卷积神经网络的输出a_i^l。结合公式(3.6)、公式(3.7)、公式(3.8) 以及公式(3.10)，本节可以推出LIF神经元在$T+1$时刻的膜电势如下：

$$v_i^l(T+1)=\sum_{t=1}^{T} z_i^l(t)+z_i^l(T+1)-\tau N_i^l(T) \tag{3.11}$$

将公式 (3.10)代入到公式 (3.11)，同时代入公式 (3.6)，那么本章可以推出(为了简化推导假设$\lambda=\tau$以及$b_{i,t}^l=b_i^l$)：

$$r_i^l(T)=\sum_{j=1}^{M^{l-1}} w_{ij}^l r_j^{l-1}(T)+b_i^l+\Delta E(T) \tag{3.12}$$

根据公式 (3.6)、公式 (3.10)- (3.12)，可以推出逼近误差如下：

$$\Delta E(T)=\frac{z_i^l(T+1)-v_i^l(T+1)}{\tau T} \tag{3.13}$$

由于输入图像像素值以及卷积神经网络的突触权值是有限的，通过分析公式(3.6)以及公式(3.8)可知，存在一个给定的M_1使得$z_i^1(T+1)$和$v_i^1(T+1)$小于$M_1\tau$。假设存在一个给定的M_{l-1}使得$z_i^{l-1}(T+1)$和$v_i^{l-1}(T+1)$小于$M_{l-1}\tau$，那么由公式 (3.7)可知存在一个给定的L使得$\theta_{i,T}^{l-1}$小于$L\tau$。因为$\theta_{i,T}^{l-1}$是有限的，那么从公式 (3.6)中可知$z_i^l(T+1)$是有限的。同时，当膜电势超过阈值的时候，LIF神经元会向外发送脉冲，因此$v_i^l(T+1)$也是有限的。综上所述，当$T\to\infty$时，$z_i^l(T+1)-v_i^l(T+1)$是有限的，因此$\Delta E(T)\to 0$。因此，在有足够长的网络收敛时间T的前提下，$r_i^l(T)$可以准确逼近a_i^l，即M-SNN可以精确地逼近卷积神经网络。

3.2.3 多强度脉冲神经网络的优势

多强度脉冲神经网络相较于原始的转换脉冲神经网络主要有三点优势：更稀疏的输入，更稀疏的层间脉冲，更快的网络收敛速度。

在脉冲神经网络的输入层，公式 (3.7)中的$\theta_{i,t}^0$用于逼近输入图片的第i个像素值p_i。由公式 (3.7)以及公式 (3.11)可知，第一层的输入脉冲频率$r_i^0(T)$需要满足下式：

$$r_i^0(T)=\frac{\sum_{t=1}^{T}\theta_{i,t}^0}{T}\to p_i \tag{3.14}$$

由于可以输入多强度脉冲，因此可以使用更少的脉冲逼近像素值p_i。举例说明，如果像素值$p_i=0.24$，在10个时间步内，原始的转换脉冲神经网络需要发送2个脉冲信号。与之相对地，在多强度脉冲神经网络中可以使用一个强度为2的脉冲

来代替原始的转换脉冲神经网络的脉冲输入。这也就是说，多强度脉冲神经网络的脉冲输入可以更加稀疏。

根据公式 (3.7)可知，当输入层的脉冲的$\theta_{i,t}^{0}$强度更大时，某些时间步上第一层神经元的输入累积电势$z_i^1(t)$更大。结合公式 (3.8)可知，由于某些时间步累积的电势$z_i^1(t)$更大，第一层的层间脉冲$\theta_{i,t}^{1}$更容易在某些时间步上被激发。也就是说，第一层的层间脉冲序列$\theta_{i,t}^{1}$更加稀疏。假设第l层的脉冲序列$\theta_{i,t}^{l}$相较于原始的转换神经网络的脉冲序列更加稀疏，那么类似于上述推导，第$l+1$层的脉冲序列$\theta_{i,t}^{l+1}$更加稀疏。由数学归纳法可知，多强度脉冲神经网络中的层间脉冲序列相较于原始的转换脉冲神经网络的脉冲序列更加稀疏。基于脉冲序列的稀疏性，在大部分时间步上，层间脉冲$\theta_{i,t}^{l}$保持为零。基于以上性质，在硬件设计时，公式 (3.7)以及公式 (3.10)中的运算可以被移除。

由于脉冲神经网络中存在着脉冲神经元无法表达卷积神经网络中大于1的值的问题，一些参数规范化算法[98, 99, 118]被提出缓解该问题，提高转换后脉冲神经网络的识别精度。但是这些参数规范化算法会减少转换网络中的脉冲发送频率，从而加大转换脉冲神经网络的收敛时间。相对于使用参数规范化算法，多强度脉冲神经网络不需要进行权值参数规范化的操作，并且从本质上解决了脉冲神经元无法表示大于1的值的问题。同时，使用多强度脉冲传递信息，多强度脉冲神经元可以更快地积累膜电势，更容易产生脉冲，进而加速脉冲神经元逼近卷积神经网络中对应神经元的输出值的过程。另外，不使用参数规范化算法，转换脉冲神经网络的收敛时间不受这些算法的影响。因此，多强度脉冲神经网络的收敛速度更快。

3.3 深度脉冲神经网络的动态剪枝算法

在深度卷积神经网络中，计算的冗余是造成其功耗较高的主要因素之一。现在已有很多网络压缩算法探索这一领域：Han 等 [167]和Parashar 等 [168]分别删减了全连接层和卷积层中权值为0的突触的乘累加运算；Yu 等 [169]提出一种动态单指令多数据(SMID-aware)的权值剪枝算法用来定制专用卷积神经网络硬件；Qiu 等 [170]探索了一种动态的权值量化算法；Shin 等 [171]提出DNPU硬件结构来实现在线动态定点运算；二值神经网络[172](Binary Neural Networks，简称BNN)、三值神经网络[173](简称Ternary weight networks)以及异或神经网络[174](简称Xnor-net)分别探索了一种不同的极低比特的定点卷积神经网络；Han 等 [175]提出DeepCompression算法，在识别精度接近无损的条件下，VGG16

网络[176]被压缩了49倍。

通常情况下转换脉冲神经网络的层数较少，采用多强度LIF神经元模型可以搭建深度的脉冲神经网络。时序信息没有被考虑到卷积神经网络压缩技术中，因此本章提出了3种压缩转换脉冲神经网络的方法：nNM-SNN、nPM-SNN以及nWM-SNN。

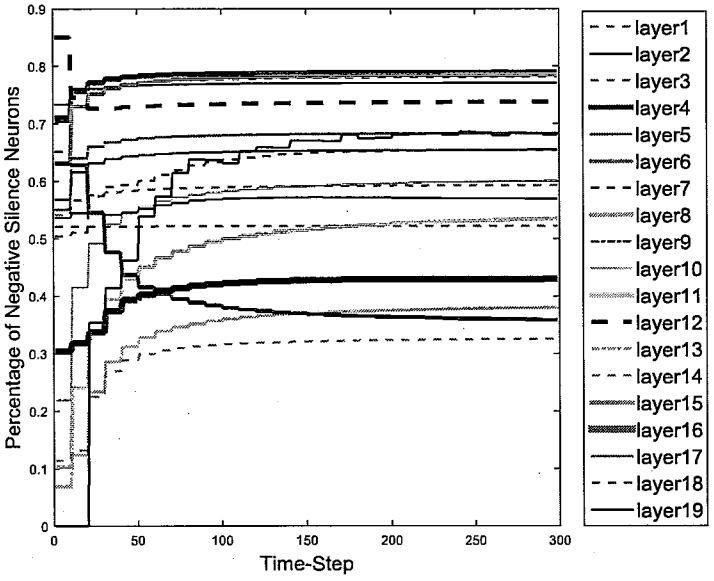

图 3.3 消极的沉默神经元(Negative silence neuron)的分布示意图。

Figure 3.3 Negative silence neuron distribution.

nNM-SNN网络移除转换脉冲神经网络中具有负膜电势的LIF神经元的运算操作。从公式 (3.7)和公式 (3.8)中可以看出，当第l层的第i个LIF神经元的膜电势$v_i^l(t)$在连续几个时刻上均小于0，那么该神经元的输出强度$\theta_{i,t}^l$一直为0。同时，由于$w_{ji}^{l+1}\theta_{i,t}^l$为0，第$l+1$层的输入$z_i^{l+1}(t)$也为0。也就是说，当某个神经元带有负的膜电势的时候，该神经元对后续层的神经元的脉冲产生不贡献任何影响。因此，本章可以对这类神经元相关的一系列操作进行剪枝。定义这类神经元为消极的沉默神经元(Negative silence neuron)，nNM-SNN网络就是M-SNN网络剪枝掉消极的沉默神经元的脉冲神经网络。为了确定需要多少时间步确定LIF神经元为消极的沉默神经元，画出转换脉冲神经网络中每个时间步上带有负的膜电势的神经元分布的示例图，如下图 3.3所示。此图中曲线表示转换脉冲神经网络中各层神经元中带有负膜电势的神经元的百分比，layerN表示第N层神经元。图 3.3中用于转换的卷积神经网络为VGG19网络，可以看出各层中消极的沉默神经元的百分比增长很快，在50个时间步左右趋于稳定。除了第12、15以及19层神经元以外，可以选择一个较小的时间步确定该神经元是否为消极的沉默神经

元。

nPM-SNN网络移除转换脉冲神经网络中具有较小正膜电势的LIF神经元的运算操作。从图 3.4中可以看出，输入脉冲可以分类为2种：强脉冲(strong spike)和弱脉冲(weak spike)，同时神经元突触的权值也可以分为2种：强突触(strong synapses)和弱突触(weak synapses)。如果某神经元的某个弱突触接收到弱脉冲，那么该脉冲对该神经元的输出脉冲产生作用很小。从图 3.5中可以看出，大部分神经元的突触为弱突触同时大部分的脉冲输出也为弱脉冲。因此，本章定义积极的沉默神经元(Positive silence neuron)为在转换脉冲神经网络开始运行的前T个时间步内平均输出强度$\frac{\sum_{t=0}^{T}\theta_i^l(t)}{T}$小于阈值$\xi$的神经元。nPM-SNN网络就是M-SNN网络剪枝掉积极的沉默神经元的脉冲神经网络。

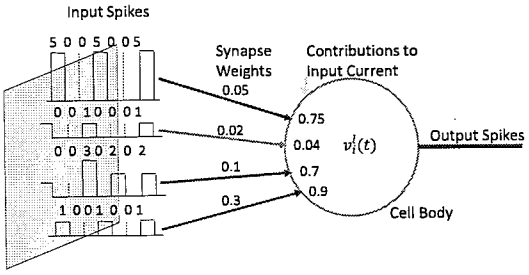

图 3.4 积极的沉默神经元(Positive silence neuron)示意图。该图解释了积极的沉默神经元对后序神经元影响较小的原因。

Figure 3.4 Positive silence neuron. This figure indicates why the positive neuron makes little effect on the neuron of later layers.

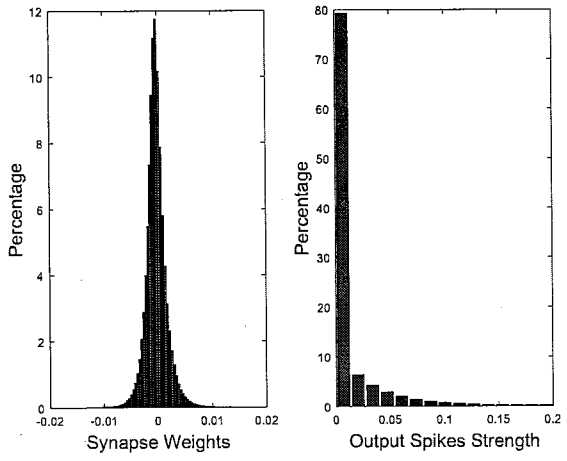

图 3.5 一个VGG19结构的M-SNN中突触权值以及输出脉冲强度的统计分布直方图(任意选取的第10层数据)。

Figure 3.5 The statistical histogram of the synapse weights and the strength of output spikes in the 10th layer of VGG19-structure M-SNN.

nWM-SNN网络移除转换脉冲神经网络中较小权值的突触上的运算操作。当某个突触权值w_{ij}^{l}小于ε时，定义该突触为弱突触(weak synapse)。从图 3.5中可以看出，大部分神经元的输出脉冲强度很弱，也就意味着下一层神经元的输入脉冲强度很弱。如果输入脉冲强度很弱的同时对应的突触强度也很弱，那么其对神经元的脉冲产生贡献很小。因此，可以在不损害识别精度的条件下，剪枝掉转换脉冲神经网络中的弱突触。为了保证剪枝对转换脉冲神经网络的影响较小，对不同的层采用不同的阈值ϵ剪枝。通过实验，卷积层相对于全连接层可以选择更大的剪枝阈值ϵ。本章将剪去弱突触的转换脉冲神经网络称之为nWM-SNN网络。

3.4 实验结果分析

本节首先简要介绍了本章实验所使用的基本实验设置，然后从网络的识别精度、网络的收敛时间、网络的稀疏度、网络的压缩率以及网络的运算复杂度等5个方面分析了多强度脉冲神经网络的性能，最后对实验结果进行了简要的小结。

3.4.1 实验设置

本节主要介绍了验证本章算法所使用的实验数据集以及脉冲神经网络结构。

3.4.1.1 实验数据集

本章采用MNIST数据集和CIFAR10数据集作为主要的测试数据集。MNIST数据集包含70,000张20×28大小的$0-9$的手写数字的灰度图片，其中60,000张作为训练集，10,000张作为测试集。CIFAR10数据集包含60,000张32×32大小的10类彩色自然图片，其中50,000张作为训练集，10,000张作为测试集。

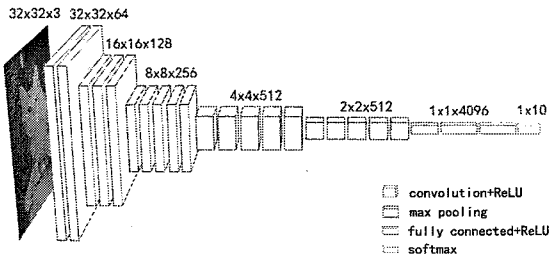

图 3.6 VGG19结构[176]的M-SNN网络示意图。主要用于验证动态剪枝算法的效果。

Figure 3.6 An illustration of the architecture of 19-layer VGG[176] network(VGG19).

3.4.1.2 网络结构设置

本章主要使用5种网络结构进行试验。其中前4种的网络结构如表 3.1中所示，主要用来在MNIST数据集上验证M-SNN的识别精度以及收敛速度的。图3.6中显示的VGG19结构的M-SNN用于验证动态剪枝算法的效果。

表 3.1 网络结构设置(按列显示)。卷积层的参数按照conv(感受野尺寸)-(通道数目)表示。

Table 3.1 Network configurations(shown in columns). The convolutional layer parameters are denoted as conv(receptive field size)-(number of channels).

Network Configuration			
NetworkA	NetworkB	NetworkC	NetworkD
4 weight layers	4 weight layers	4 weight layers	6 weight layers
input($28 \times 28 \times 1$ handwritten digits)			
conv5-32 conv5-64	conv5-32 conv5-64	conv5-32	conv3-32 conv3-32
	max-pool	max-pool	max-pool
		conv5-64	conv3-64 conv3-64
		max-pool	max-pool
FC-1024			
FC-10			
softmax			

3.4.2 网络的识别精度

大部分的脉冲神经网络的直接训练算法以及脉冲神经网络模型转换算法是在MNIST数据集上评估的。表 3.2中比较了10种效果最好的脉冲神经网络学习算法在MNIST数据集上获得的识别精度，其中前7种脉冲神经网络是采用直接训练算法获得的。该表中无监督学习算法STDP-trained network[65]获得最好的识别精度为95.0%；监督学习算法Deep SNN[85]取得的最好识别精度为98.6%；本文之前的脉冲神经网络模型转换算法取得的最好的识别精度为99.44%。本章的M-SNN(NetworkD)相对于已有的脉冲神经网络训练以及模型转换算法取得了最好的识别精度，为99.57%。

本章在CIFAR10数据集上比较了5种先进的的脉冲神经网络直接训练以及模型转换算法。通过将一个二值卷积神经网络(BinaryConnect CNN)转换到脉冲神经网络中，Rueckauer 等 [99]在CIFAR10数据集上取得了本文之前最好的识别精度，为90.85%。在硬件方面，Esser 等 [163]将转换好的脉冲神经网络映射到TrueNorth芯片上去，取得了89.32%的识别精度。在本章中，使用一个更深的VGG19结构的转换脉冲神经网络，在CIFAR10数据集上取得了94.01%的识别

表 3.2 不同训练算法在MNIST数据集上获得的脉冲神经网络的识别精度表。

Table 3.2 Classification accuracy of different SNNs on the MNIST dataset.

Network-type	Preprocessing	Accuracy
Feedward network[60]	Edge-detection	96.5%
Spiking RBM[92]	Thresholding	92.6%
STDP-trained network[65]	None	95.0%
Spiking RBM[121]	Enhanced training set	94.1%
Spiking ConvNet[177]	Scaling,orientation detection	91.3%
Dendritic neurons[178]	Thresholding	90.3%
Deep SNN[85]	N-MNIST dataset	98.6%
Adapting SNN[179]	Analog input	99.1%
Spiking ConvNet[118]	None	99.1%
Spiking ConvNet[99]	None	99.44%
M-SNN(our paper,NetworkD)	None	**99.57%**

精度。

表 3.3 不同训练算法在CIFAR10数据集上获得的脉冲神经网络的识别精度表。

Table 3.3 Classification accuracy of different SNNs on the CIFAR10 dataset.

Network-type	Preprocessing	Accuracy
Spiking ConvNet[97]	Tailor CNNs	77.43%
LIF SNN[180]	Training with noise	83.54%
Q4-SNN[181]	None	84.52%
SNN on TrueNorth[163]	restrict weights	89.32%
BinaryConnect SNN[99]	BinaryConnect Network	90.85%
M-SNN(our paper,VGG19)	None	**94.01%**

3.4.3 网络的收敛时间

通过下式定义脉冲神经网络的收敛时间：

$$C_T = \begin{cases} t, & \text{if } \frac{|a_t - a_{last}|}{a_{last}} < 0.02 \\ T_{last}, & \text{else} \end{cases} \tag{3.15}$$

其中，C_T代表转换脉冲神经网络的收敛时间，a_t表示在第t个时间步时转换脉冲神经网络的收敛的识别精度，a_{last}表示在仿真过程中最后一个时间步T_{last}时转换脉冲神经网络收敛的识别精度。从图 3.7中可以看出，本文之前的模型转换算法获得的转换脉冲神经网络收敛很慢，需要300个时间步左右才能稳定。本文提出的M-SNN仅仅需要80个时间步即可收敛到较佳的识别精度。

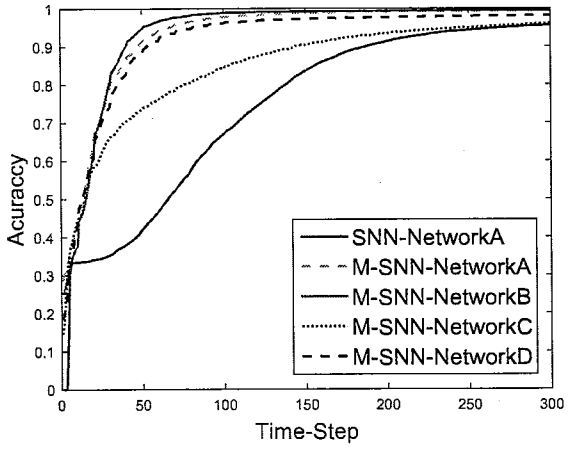

图 3.7 几种不同网络结构的转换脉冲神经网络的收敛时间示意图。其中，NetworkA-D分别对应表 3.1中的4种网络结构。

Figure 3.7 Convergence time for different scenarios. NetworkA-D are related to network configuration from Table 3.1.

3.4.4 网络脉冲稀疏度

定义转换脉冲神经网络中的脉冲稀疏度如下式所示：

$$S_\theta = \frac{\sum_{t=0}^{T} N_\theta(t)}{N_{size} \times T} \tag{3.16}$$

其中，S_θ表示M-SNN网络的脉冲稀疏度，$N_\theta(t)$表示时刻t内M-SNN网络中产生的脉冲的总数目，N_{size}表示M-SNN网络中的神经元的总数目。

从表 3.4中可以看出，通过裁剪积极的沉默神经元，nPM-SNN网络的脉冲稀疏度仅是原始M-SNN网络的1/4。SM-SNN网络表示同时使用动态剪枝算法中的3种技术对M-SNN网络进行裁剪，即是移除M-SNN网络中的消极的沉默神经元、积极的沉默神经元以及弱突触。从表 3.4中发现，SM-SNN网络在脉冲稀疏度的度量上并没有取得最佳的效果。这是由于几种压缩技术带来的识别精度损失会相互累加，造成累积的误差较大，因此当使用3种技术叠加的时候，需要适当减弱3种技术的压缩效果。在不损害识别精度的前提下，本章首先剪枝M-SNN网络的弱突触，然后使用不同的阈值逐层移除消极的沉默神经元与积极的沉默神经元。后续的工作中，将会提出一些算法自动地使用这3种压缩技术进行转换脉冲神经网络剪枝。

3.4.5 网络压缩率

脉冲神经网络动态剪枝算法中主要有3种剪枝技术，其中前2种技术主要是对神经元的运算操作进行裁剪。定义神经元的压缩率C_R为被移除的神经元占转

表 3.4 深度脉冲神经网络动态剪枝算法的实验结果比较(所有的测试网络在CIFAR10都取得和表 3.3相同的识别精度，为94.01%)。S_θ(%)表示公式 (3.16)中定义的脉冲稀疏度，$C_N(GOPS)$表示公式 (3.19)中定义的运算复杂度，C_R(%)表示公式 (3.17)和公式 (3.18)中定义的压缩率。表中上标1表示神经元的压缩率，上标2表示突触的压缩率，上标3中SM-SNN = M-SNN + nPM-SNN + nNM-SNN + nWM-SNN。

Table 3.4 Experimental results with dynamic pruning(all networks with the same accuracy of 94.01%). In this table, S_θ(%) indicates the sparsity of the network in (3.16), $C_N(GOPS)$ indicates the computational complexity of the network in (3.19), C_R(%) indicates the compression ration of the network in (3.17) and (3.18).

Methods	S_θ(%)	$C_N(GOPS)$	C_R(%)	Accuracy
M-SNN	2.3837	1.5199	-	94.01%
M-SNN+nPM-SNN	**0.5739**	0.5789	29.31[1]	94.00%
M-SNN+nNM-SNN	2.3642	0.4865	65.05[1]	94.01%
M-SNN+nWM-SNN	2.3611	0.8390	89.43[2]	94.00%
SM-SNN[3]	1.6186	**0.2165**	**94.07[1]/89.43[2]**	94.00%

换脉冲神经网络总的神经元数目的百分比：

$$C_R = \frac{\sum_{t=1}^{T} N_{silence\_neuron}(t)}{N_{neuron} \times T} \tag{3.17}$$

其中，$N_{silence_neuron}(t)$表示在第t个时间步时沉默神经元的数目，N_{neuron}表示转换脉冲神经网络中总的神经元数目。

第3种剪枝技术主要用来裁剪转换脉冲神经网络中的突触。定义突触的压缩率C_R为被移除的弱突触数目占转换脉冲神经网络中总的突触数目的百分比：

$$C_R = \frac{N_{weak\_synapses}}{N_{synapses}} \tag{3.18}$$

其中，$N_{weak_synapses}$表示被移除的弱突触的数目，$N_{synapses}$表示转换脉冲神经网络中总的突触数目。从表 3.4可以看出，在不降低M-SNN网络识别精度的前提下，SM-SNN网络移除了94%的神经元和89%的弱突触。动态剪枝算法在网络压缩率方面取得了显著的效果。

3.4.6 网络运算复杂度

将乘法、加法以及floor操作均看作一个运算操作，定义网络的运算复杂度如下式所示：

$$C_N = \frac{2 \times OPS_{conv} + 5 \times OPS_{input}}{T} \tag{3.19}$$

其中，$2 \times OPS_{conv}$表示公式 (3.6)中的运算操作数，$5 \times OPS_{input}$表示公式 (3.7)和公式 (3.8)中的运算操作数。

从图 3.8中看出，nNM-SNN网络和nPM-SNN网络相对于nWM-SNN网络在前10层网络中有更低的运算复杂度。但是，nWM-SNN网络在最后3层网络取得更好的性能。从表 3.4中可以看出，在SM-SNN网络中85.7%的运算复杂度被从M-SNN网络中移除了。因为M-SNN网络中的突触权值来源于卷积神经网络，因此时序信息很难被引入到M-SNN网络剪枝技术中。因此，相对于nNM-SNN网络和nPM-SNN网络，nWM-SNN网络的运算复杂度下降的效果没有那么显著。nWM-SNN网络在其网络最后3层取得的效果最好，这也就意味着全连接层中的突触冗余是最大的。

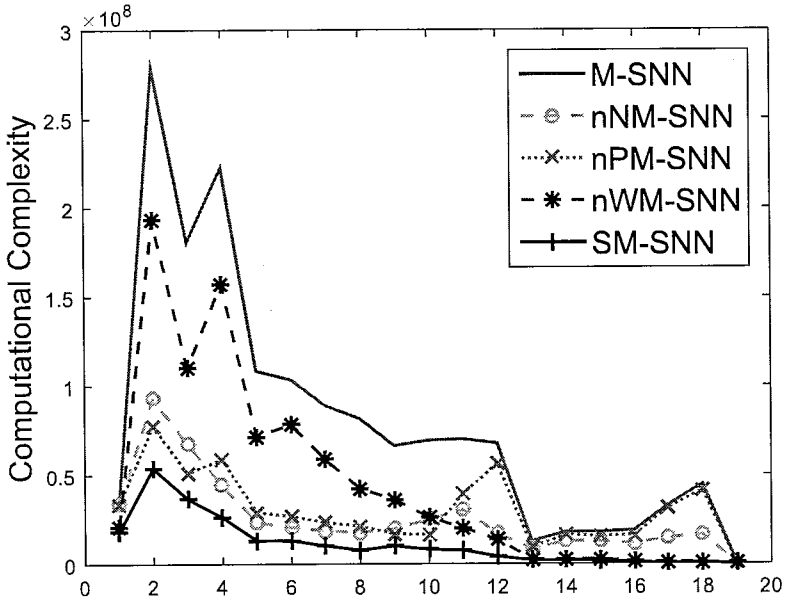

图 3.8 VGG19结构的M-SNN网络中各层的运算复杂度。

Figure 3.8 Computation complexity of each layers in VGG19-structure M-SNN.

3.4.7 实验结果小结

本章从五个方面评估了多强度深度脉冲神经网络模型转换算法的性能。在网络的识别精度方面，多强度脉冲神经网络在MNIST数据集上获得的识别精度为99.57%(相比于同期研究的最佳结果提升0.13%)，在CIFAR10数据集上取得的识别精度为94.01%(相比于同期研究的最佳结果提升3.16%)。在网络的网络收敛时间方面，多强度脉冲神经网络识别精度在MNIST数据集上可以在80个时间步内收敛，而文献[118]中算法需要300个时间步。在减少深度脉冲神经网络中的冗余计算方面，本章提出的三种动态剪枝算法可以移除多强度脉冲神经网络中94%的沉默神经元以及89%的弱突触，减少网络中85%的冗余计算。

总的来说，多强度脉冲神经网络在获得深度结构的同时，提升转换脉冲神

经网络的识别精度。同时，本章提出的三种动态剪枝算法可以大幅裁剪脉冲神经网络中的冗余计算。

3.5 本章小结

针对转换脉冲神经网络中脉冲神经元无法表达卷积神经网络中对应神经元大于1的输出值问题，本章提出了一种基于多强度脉冲神经元的多强度脉冲神经网络算法。首先，定义了多强度脉冲神经元模型的数学表达，用于搭建多强度脉冲神经网络。然后，推导了卷积神经网络与转换后的脉冲神经网络识别精度的等价性理论，并分析了多强度脉冲神经网络相较于原始的转换脉冲神经网络的三个主要优势。使用多强度脉冲神经网络，在CIFAR10数据集上，成功地将VGG19网络转换到了多强度脉冲神经网络中，并获得了94%的识别精度。

获得了极深的转换脉冲神经网络后，本章着手于对脉冲神经网络中的冗余计算进行剪枝。现有的卷积神经网络参数压缩算法中，并没有考虑对脉冲神经网络中时序信息的处理。因此，本章针对脉冲神经网络在时域上传递信息的特点，提出了三种脉冲神经网络的动态剪枝算法。实验表明，使用这三种动态剪枝算法可以移除多强度脉冲神经网络中94%的沉默神经元以及89%的弱突触，减少网络中85%的冗余计算。

第4章 低延迟深度脉冲神经网络模型转换算法

在第三章中，通过提出一种多强度脉冲神经网络，克服了脉冲神经元无法表达卷积神经网络中大于1的输出值的问题。多强度脉冲神经网络在网络的识别精度以及收敛速度上，取得了显著的效果，但是多强度的脉冲神经元与现有的主流类脑芯片中实现的脉冲神经元模型并不相符。这个问题增加了该算法往现有类脑计算芯片上直接部署的难度，同时设计基于异步电路的类脑芯片十分复杂，因此会阻碍转换脉冲神经网络获得极低功耗，从而限制转换脉冲神经网络在实际中的应用。

本章着眼于在使用常规的脉冲神经元模型解决脉冲神经元无法表达卷积神经网络中大于1的输出值的问题的同时，进一步降低转换脉冲神经网络的收敛速度。首先，本章分析了现有的各种参数规范化算法，提出了限制网络输出预训练算法。该预训练算法将参数规范化过程转换到卷积神经网络中的训练过程中，可以动态地进行参数规范化，减少参数规范化算法带来的低效脉冲现象。然后本章发现了转换脉冲神经网络中的错误脉冲现象，将错误脉冲的抑制问题抽象化为一个线性规划问题，大大减少了转换网络中的错误脉冲。接着，本章提出了一种时序最大值池化算法，可以将卷积神经网络中的最大值池化操作无损地迁移到转换脉冲神经网络中。最后，通过实验测试，证明本章的算法可以解决脉冲神经元无法表达卷积神经网络中大于1的输出值的问题，同时获得一个低延迟的深度脉冲神经网络。

4.1 模型描述及相关理论

本章采用脉冲神经网络模型转换算法中的常用脉冲神经元模型[97–99, 118]，其数学模型如下式所示：

$$z_i^l(t) = V_{th}\left(\sum_{j=1}^{M^{l-1}} w_{ij}^l \theta_{t,j}^{l-1} + b_i^l\right) \tag{4.1}$$

$$\theta_{t,i}^l = g(v_i^l(t-1) + z_i^l(t) - V_{th}) \tag{4.2}$$

$$v_i^l(t) = v_i^l(t-1) + z_i^l(t) - V_{th}\theta_{t,i}^l \tag{4.3}$$

其中，$v_i^l(t)$表示第l层第i个脉冲神经元在第t个时间步的膜电势，$z_i^l(t)$表示第l层的第i个神经元在第t个时间步的上一层的输入膜电势总和，$\theta_{t,j}^{l-1}$表示第$l-1$层

的第j个神经元在第t个时间步的输出脉冲，w_{ij}^{l}和b_{i}^{l}分别代表突触权值以及偏置，V_{th}表示膜电势阈值，$g(\cdot)$函数满足下式：

$$g(x)=\begin{cases}1, & if \quad x \geq 0\\ 0, & otherwise\end{cases} \tag{4.4}$$

定义转换脉冲神经网络的第l层的第i个神经元在第t个时间步时的脉冲发送频率$r_{i}^{l}(t)$定义如下：

$$r_{i}^{l}(t)=\frac{N_{i}^{l}(t)}{t}=\frac{\sum_{\hat{t}=1}^{t}\theta_{\hat{t},i}^{l}}{t} \tag{4.5}$$

其中，$N_{i}^{l}(t)$表示在t个时间步内第l层的第i个神经元发送脉冲的总数目。由上式可知，脉冲神经元的无法表达卷积神经网络中对应神经元大于1的输出值。

Diehl 等 [118]提出基于模型的参数规范化(model-based normalization)算法以及基于数据的规范化(data-based normalization)算法等两种参数规范化算法，来解决脉冲神经元的无法表达卷积神经网络中对应神经元大于1的输出值的问题。为了解释方便，给出基于模型的参数规范化算法伪代码如算法 1中所示。由算法1的伪代码可知，该算法只是通过将卷积神经网络训练后的参数进行放缩使得卷积神经网络中的神经元输出值小于1，因此会使参数权值减小，从而降低转换脉冲神经网络的收敛速度。后续工作[98, 99]虽然对该算法进行了改进，但是本质上还是放缩卷积神经网络训练后获得的参数权值，依然会带来降低转换脉冲神经网络收敛速度的问题。

算法 1 基于模型的参数规范化算法(model-based normalization)

```
for layer in layers do
    max_pos_input = 0
    for neuron in layer.neuorns do
        input_sum = 0
        for input_wt in neuron.input_wts do
            input_sum += max(0,input_wt)
        end for
        max_pos_input = max(max_pos_input,input_sum)
    end for
    for neuron in layer.neurons do
        for input_wt in neuron.input_wts do
            input_wt = input_wt/max_pos_input
        end for
    end for
end for
```

4.2 限制网络输出预训练算法研究

大部分脉冲神经网络模型转换算法主要关注参数转换过程中的逼近误差。LIF神经元的脉冲频率的值在区间[0, 1]内是转换过程中产生逼近误差的主要原因之一。如果待转换的卷积神经网络中的神经元的输出特征(Output feature map)值大于1，那么对应的LIF神经元会一直往外输出脉冲。这种现象被称之为脉冲饱和(Firing rate saturation)。从上一小节可知，几种参数规范化(Weight normalization)算法[98, 99, 118]被提出解决这个问题。这些参数规范化的算法总的来说，可以被归纳为下式：

$$\hat{W}^l = \frac{W^l}{\alpha^l} \tag{4.6}$$

其中，W^l表示参数规范化之前的卷积神经网络第l层的权值矩阵，$\hat{W}^l$为规范化之后的第l层权值矩阵，α^l表示参数规范化算法第l层的放缩参数(每层均对应一个不同的放缩参数α^l)。这些参数规范化算法可以使转换后获得的脉冲神经网络获得更好的识别精度，但是由于权值矩阵W^l被缩小，转换后的脉冲神经网络积累膜电势的速度被缩减，造成脉冲产生的频率减小。极低的脉冲频率随着脉冲神经网络由低层向高层传播，会使脉冲神经网络的分类延迟(Classification latency)不断增加，将这种现象称之为低效脉冲现象(Insufficient firing)。Rueckauer等 [99]通过折中考虑脉冲饱和以及低效脉冲的影响，改进了数据驱动的参数规范化算法(data-based normalization)，提出了一种启发式的鲁棒性的参数规范化算法。但是，这些算法(包括之前的参数规范化算法)很难被应用的深度脉冲神经网络中。之前的算法通常在一个6-7层的神经网络上测试。如果使用一个非常深的网络，例如VGG19结构的脉冲神经网络，转换后的脉冲神经网络的网络收敛时间将会难以接受。

基于对之前脉冲神经网络模型转换算法中的参数规范化算法的研究分析，本章将参数规范化过程移到卷积神经网络的训练过程中。通过将参数规范化过程移到卷积神经网络训练过程中，可以使参数规范化过程的适应性更强，同时保证在不降低识别精度的条件下减小网络的收敛时间。图 4.1展示了如何在卷积神经网络训练过程中进行参数规范化的过程，具体算法流程展示在算法 2中。在此图中，子图(a)中间的方框表示限制网络输出的预训练过程。卷积神经网络的神经元的输出裁剪(Clip Output Feature Map)到[0, 1]区间内，卷积神经网络经过几个周期的训练逐渐收敛。算法的输入为训练好的原始卷积神经网络以及权值参数，输出为裁剪好的卷积神经网络以及规范化的权值参数。子图(b)相对于本文之前的参数规范化算法的开环调节过程，该预训练算法可以被看作一个闭环系

统。更好地保证了在减小脉冲饱和的条件下，适应性更强地控制低效脉冲现象。本文之前的参数规范化算法，都是在考虑如何对卷积神经网络的每层设置一个合适的放缩参数α^l使得各层的输出尽量小于1。相较于之前的参数规范化算法，本章算法相当于在卷积神经网络训练的动态过程中对每个神经元动态地寻找合适的放缩参数α，从而使得该算法的适用性更强。本章算法可以在减小脉冲饱和的情况下，有效地避免低效脉冲现象。从图 4.1(b)中看出，相对于之前算法的开环调节过程，本章算法可以被看成一个闭环的调节过程。

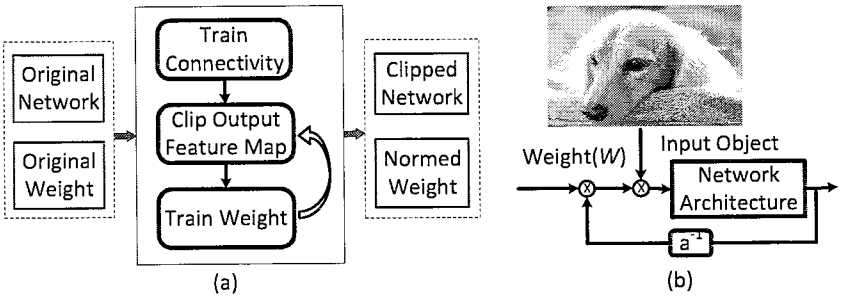

图 4.1 限制网络输出预训练算法示意图。

Figure 4.1 Restricted output training method.

算法 2 限制网络输出预训练算法(Restricted Output Training Algorithm)

1: **输入**：o_network表示原始的卷积神经网络，o_weight表示该卷积神经网络使用正态分布初始化的权值参数；
2: **输出**：clip_network表示裁剪的卷积神经网络，norm_weight表示该算法规范化之后的权值参数；
3: **procedure** RESTRICTEDOUTPUTTRAINING($o\_network, o\_weight$)
4: 使用反向传播算法训练o_network，得到一组权值参数o_weight，识别精度为o_acc；
5: 使用权值参数o_weight，对clip_network进行一次前向传播，得到识别精度clip_acc；
6: norm_weight = o_weight；
7: **while** $\frac{|clip\_acc - o\_acc|}{o\_acc} > 5\%$ **do**
8: 使用反向传播算法训练clip_network，得到权值参数norm_weight；
9: 对clip_network进行一次前向传播，得到识别精度clip_acc；
10: **end while**
11: **return** $clip\_network, norm\_weight$
12: **end procedure**

4.3 错误脉冲抑制算法研究

本节首先阐述了错误脉冲产生的原因，接着度量了错误脉冲在转换脉冲神

经网络各层中的存在情况，然后通过将抑制第一层的错误现象抽象化为一个线性优化问题，最后根据线性优化问题的解设计了一种错误脉冲抑制算法。

4.3.1 错误脉冲产生原因分析

错误脉冲(False spike)指的是转换脉冲神经网络中按照转换之前的卷积神经网络的输出不应该往外发送脉冲，但是实际上转换后的脉冲神经网络中的对应神经元往外发送了脉冲信号。由于输入脉冲序列(Spike train)的随机性，相对于带有正权值的突触，带有负权值的突触可能接收到比期望更少的脉冲。图4.2通过了一个简单的例子解释了错误脉冲现象。如图 4.2左上角所示，卷积神经网络中神经元接收到的2个输入为0.75和0.6，对应的突触强度为-1.0和1.0。经过ReLU激活函数之后，该神经元的输出为0。因此，该神经元对应的转换后的LIF神经元不应该向外发送任何脉冲。但是从图 4.2左下角可知，如果在前4个时间步中，对应输入为0.75的输入脉冲序列为0,1,1,1，对应输入为0.6的输入序列为1,0,1,0，那么在第1个时间步，该LIF神经元将会向外发送一个错误脉冲。因此，脉冲序列的产生机制对LIF神经元膜电势的积累十分重要，不合适的输入脉冲序列可能会使LIF神经元发送意料之外的脉冲。但是当脉冲神经网络模拟的时间步足够多的时候，错误脉冲带来的误差会逐渐收敛，如图 4.2右下角所示。这样也就意味着，错误误差虽然不会带来较大的误差，但是会使深度脉冲神经网络的收敛时间变得难以接受。

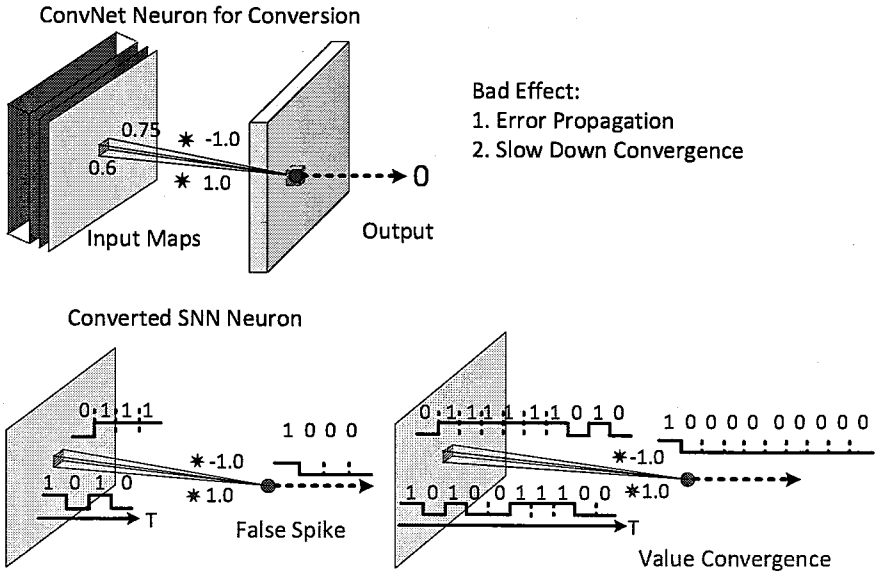

图 4.2 错误脉冲(False spike)产生原因示意图。

Figure 4.2 Illustration of false spike generation.

错误脉冲在本章之前的研究中没有被发现的主要原因是因为本章之前的研究中的脉冲神经网络的网络层数较浅。基于STDP法则的脉冲神经网络无监督学习算法中，抑制型神经元(Inhibitory neurons)接收输入脉冲序列的频率高于兴奋型神经元(Excitatory neurons)，从而解决了错误脉冲(该算法仅训练了一个3层脉冲神经网络)。在之前的脉冲神经网络模型转换算法的研究中通常使用一个6-7层的浅层网络，错误脉冲在该网络上引起的识别延迟不够明显。但是当脉冲神经网络层数加深的时候，错误脉冲将会影响脉冲神经网络的识别精度以及网络收敛时间。

4.3.2 错误脉冲的度量

由上小节可知，错误脉冲问题是引起转换脉冲神经网络过长的网络收敛时间的原因之一。首先本小节对错误脉冲在各层出现的频率进行度量。

假设第l层的第i个LIF神经元期望逼近的卷积神经网络输出特征图(Output feature map)的值a_i^l如下式所示：

$$a_i^l = f(\sum_k (w_k^l * x_i^l + b_k)) \tag{4.7}$$

其中，x_i^l是前序层的特征输出层的值，w_k^l表示通道k的第l个卷积核的权值参数，b_k是通道k对应的偏置权值，符号$*$表示二维卷积操作，函数$f(\cdot)$表示ReLU激活函数。模型转换算法就本质来讲，可以看作是对一系列的$w_i x_i$的和的逼近。假设X_i是一个随机变量，它的期望值$\mu_i = x_i$，方差满足$\sigma_i^2 < \infty$。同时定义一个随机变量$Y = \sum w_i X_i$，用来近似$w_i x_i$之和(其中w_i为卷积神经网络的权值参数)。给定n个时间步的输入，当$n \to \infty$时，随机变量$Z = \frac{Y_1+Y_2+\ldots+Y_n}{n}$的以极大的概率趋近于期望值为$\mu_z = \sum w_i x_i$，方差为$\sigma_z^2 = \sum w_i^2 \sigma_i^2$。当$n$足够大的时候，$Z \in (\mu_z + \alpha\frac{\sigma_z}{\sqrt{n}}, \mu_z + \beta\frac{\sigma_z}{\sqrt{n}}]$ $(\alpha < \beta)$的概率满足下式：

$$\Pr(\mu_z + \alpha\frac{\sigma_z}{\sqrt{n}} < Z \leq \mu_z + \beta\frac{\sigma_z}{\sqrt{n}}) = \Phi(\beta) - \Phi(\alpha) \tag{4.8}$$

其中，Φ为标准正态分布的累积分布函数。由3σ准则可知，参数α和β是常数值(通常$\alpha = -3$以及$\beta = 3$)。

期望值μ_z对应着LIF神经元的膜电势的每个时间步的期望增加量。从公式(4.8)中可知，方差σ_z控制着期望值μ_z的收敛速度。较小的方差σ_z有助于大幅减少转换网络中的错误脉冲。首先，本章通过度量一些典型深度网络的神经元的平均方差来定性地确定转换网络中各层中错误脉冲的数目。假设脉冲神经网络的每层输入符合二项分布，可以得到图 4.3(a)。从图 4.3(a)可以看出，神经元的

平均方差σ_z在脉冲神经网络的第一层中显著地大于其他各层。众所周知，卷积神经网络的输出特征图的值大部分为0。基于二项分布的方差$\sigma_i = np(1-p)$，卷积神经网络的除第一层以外各层的平均方差较小(如果$p \to 0$时，$\sigma_i \to 0$)。与此同时，基于RGB图片的像素值生成的第一层脉冲输入的方差将会大于脉冲神经网络其他各层。以上解释了图 4.3(a)差异化的原因。从图 4.3(b)可以看出，在前几个时间步，脉冲神经网络的第一层20%的输出脉冲为错误脉冲。

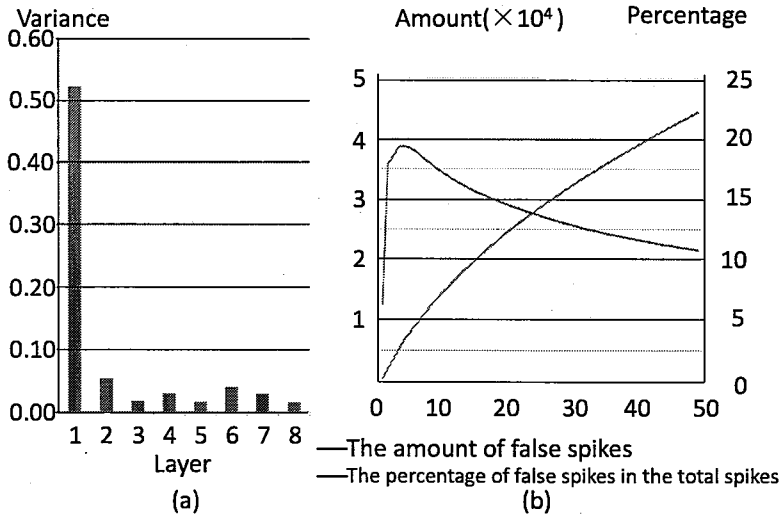

图 4.3 错误脉冲(False spike)的度量。图(a)VGG19结构的脉冲神经网络各层神经元的平均方差(由于第9-19层平均方差接近于0，故在本图中略去)。图(b)转换后脉冲神经网络第一层中错误脉冲的总数目以及百分比。其中，蓝线为总数目，红线为百分比。

Figure 4.3 The measurement of false spikes. (a) The mean variance of SNN neurons in each layer. The mean variance of layers 9-19 is around zeros and these layers are removed for simplicity. (b) Amount and percentage of false spikes in the first layer of the converted SNNs.

4.3.3 错误脉冲抑制机制

由上一小节可知，错误脉冲主要集中在脉冲神经网络的第一层。因此，错误脉冲的控制问题被转化为获取一种可以有效地产生较小σ_z的脉冲生成方式。

为了获取脉冲生成方式的空间更大，本节首先放松对第一层的脉冲强度的限制。假设随机变量$X_i \in \{a_0, a_1, a_2 \ldots a_n\}$，对应的概率为$p_0, p_1 \ldots p_n$。本节将减小

方差σ_i的问题转换为一个线性优化问题，如下所示：

$$\begin{aligned} \min \quad & \sigma_i^2 = \sum_{i=0}^{n} a_m^2 p_m - x_i^2 \\ \text{s.t.} \quad & \sum_{m=1}^{n} a_m p_m = x_i \quad and \quad \sum_{m=0}^{n} p_m = 1 \\ & 0 \le p_m \le 1 \quad \forall m \in \{0, 1, 2...n\} \end{aligned} \tag{4.9}$$

其中，a_m为区间[0, 1]内的实数，同时假设$0 = a_0 \le a_1 \le ... \le a_s \le ... \le a_n = 1$。如果$x_i \in (a_s, a_{s+1}]$以及$a_s = s/n$，该线性优化的解如下所示：

$$p_k = \begin{cases} 1 - (x_i - s/n) \cdot n, & \text{if} \quad \text{k = s} \\ (x_i - s/n) \cdot n, & \text{if} \quad \text{k = s+1} \\ 0, & \text{else.} \end{cases} \tag{4.10}$$

为了简化推导令$n = 2^l$以及$a_s = s/n$。那么方差σ_i随x_i的变化曲线如下图 4.4所示。参数n被称为逼近精度控制参数。从图 4.4中可以看出，当$n \ge 8$时，方差σ_i基本上趋近于0。

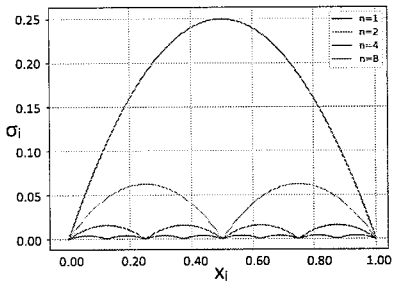

图 4.4 σ_i随x_i的变化曲线示意图。x_i表示卷积神经网络的输入的像素值，σ_i影响着x_i的收敛速度。当σ_i的值越小，转换后的脉冲神经网络的收敛速度越快。

Figure 4.4 The σ_i changes with different x_i solutions. x_i indicates the actual value of the input in CNNs and σ_i measures convergence speed of x_i by using spikes. The lower the value of x_i, the quicker the network converges.

从图 4.4中可以看出，相对于原始的{0, 1}两种脉冲强度，当采用三脉冲强度($n = 2$)时，可以获得更小的方差σ_i。接下来，推导多脉冲强度(multi-state results)可以被转换为原始的{0, 1}两种脉冲强度。举例说明，假设采用三脉冲强度，即是脉冲强度可为{0, 0.5, 1}。如果期望值$x_i = 0.3$，那么在10个时间步内，需要6个脉冲强度为0.5的脉冲和4个脉冲强度为0的脉冲来逼近该期望值(由公式

(4.10)可知)。此时，膜电势的累加之和为$w_i x_i = 6/10 * 0.5 * w_i + 4/10 * 0 * w_i$。如果期望值$x_i = 0.6$，那么在10个时间步内，需要2个脉冲强度为1的脉冲和8个脉冲强度为0.5的脉冲来逼近该值(由公式 (4.10)可知)。此时，膜电势的累加之和为$w_i x_i = 2/10 * 1 * w_i + 8/10 * 0.5 w_i$。令$\hat{w}_i = 0.5 w_i$，当$x_i = 0.6$时有$w_i x_i = 2/10 * 0.5 w_i + 2/10 * 0.5 w_i + 8/10 * 0.5 w_i = 2/10 * \hat{w}_i + 8/10 * 0 * \hat{w}_i + \hat{w}_i$(其中$\hat{w}_i$可以看成是输入脉冲序列的偏置)；当$x_i = 0.3$时有$w_i x_i = 6/10 * \hat{w}_i + 4/10 * 0 * \hat{w}_i$。从而，通过添加上一个偏置，多脉冲强度(multi-state results)被转换为原始的{0, 1}两种脉冲强度。错误脉冲抑制的主要过程在算法 3中展示。

算法 3 错误脉冲抑制算法(False Spike Inhibition Algorithm)

输入：输入RGB图像的像素值x_i(经过规范化)；
输入：需要逼近的值得集合$\{a_0, a_1, ..., a_n\}$；
输出：脉冲神经网络第一层输入脉冲序列I_i；
输出：脉冲偏置Syn_i；
procedure FALSESPIKEINHIBITION(x_i, $\{a_0, a_1, ..., a_n\}$)
    for 输入图片中的像素值x_i **do**
        根据公式 (4.10)计算p_i；
        根据p_i生成输入脉冲序列I_i；
        根据x_i生成对应的脉冲偏置Syn_i；
    end for
    return I_i, Syn_i
end procedure

4.4 时序最大值池化算法研究

在卷积神经网络中，池化操作可以保留网络中任务相关的特征，同时移除一些不相关的细节特征。该操作可以使卷积神经网络的参数更少，同时鲁棒性更强。常见的空间池化操作包括平均值池化(Average pooling)和最大值池化(Max pooling)。平均值池化可以很容易在转化的脉冲神经网络中实现，但是最大值池化在转换的脉冲神经网络中实现更困难一点。从图 4.5中可知，如果使用卷积神经网络中一样的最大值池化操作，脉冲神经网络中的神经元的输出会偏大。

本章之前的研究主要通过设计选取最大脉冲频率的脉冲神经元的方法来实现脉冲神经网络的最大值池化操作的。Cao 等 [97]提出侧向抑制(Lateral inhibition)方法，Orchard 等 [182]提出首脉冲时间编码(time-to-first-spike encoding)，Rueckauer 等 [99]通过一个门函数来预估前序脉冲神经元的脉冲发送频率。但是

这些方法都只是近似地找到最大脉冲发送频率的脉冲神经元，在某些情况下会失效。

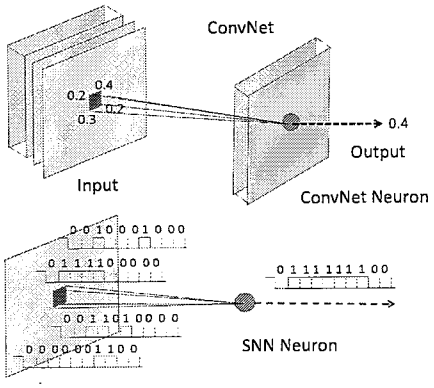

图 4.5 使用卷积神经网络中的最大值池化操作在脉冲神经网络中带来的误差。上图表示卷积神经网络欧中的神经元的输出是0.4，因此对应的脉冲神经网络中的神经元的输出应该为4个脉冲。但是根据当前脉冲输入序列，该脉冲神经网络中的神经元输出了7个脉冲。

Figure 4.5 The errors generated in SNN by converting the max pooling of CNNs with simple mechanism. The output spike train is expected to generate 4 spikes in 10 time steps, whereas 7 spikes are generated.

本章算法研究中不再关注于如何通过输入脉冲序列找到最大脉冲发送频率的脉冲神经元，取而代之，本章设计一个获得输出需要的正确的脉冲个数的技巧。具体来说，本章设计了一种特殊的简单脉冲神经元。该神经元的突触权值均设置为1，且只接收池化操作中的一个脉冲输入序列。特别的，该简单神经元的膜电势在发送脉冲的之后不进行复位操作。在每个时间步上，池化层的输出与具有最大的膜电势累积的简单神经元的输出相同。图 4.6简要解释了该过程。具体过程，见算法 4。

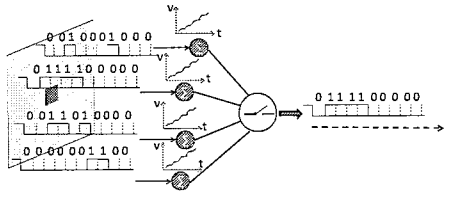

图 4.6 时序最大值池化算法(Temporal max pooling)示意图。

Figure 4.6 Temporal max pooling. The simple neuron, shown in circles, accumulate voltages with the connected input spike trains, which are used to strobe the output channel of the temporal max pooling layer.

算法 4 时序最大值池化算法(Temporal Max Pooling Algorithm)

输入：神经元积累的电势V_t；
输入：当前时间步t，网络模拟过程总时间步数T；
输入：当前时间步输入脉冲S_t；
输入：当前最大脉冲发送频率的神经元索引Id；
输入：当前最大脉冲发送频率的神经元索引的轨迹的集合Id_s；
输出：输出脉冲O_t；
procedure TEMPORALMAXPOOLING(V_t, t, S_t)
    for $t = 0$ **to** T **do**
        $V_t += S_t$;
        $Id_s = max(V_t)$; ▷ 选择具有最大电势积累的简单神经元
        if $Id \notin Id_s$ **then**
            $Id = random\_choose(Id_s)$
        end if
        $O_t = S_t(Id)$ ▷ 输出O_t和被选中的神经元输出保持一样
    end for
    return O_t
end procedure

4.5 实验结果分析

本节首先介绍了本章实验所用的基本设置，然后比较了限制网络输出预训练算法、错误脉冲抑制算法以及时序最大值池化算法对转换脉冲神经网络的识别精度以及网络收敛时间的影响，最后对实验结果进行了简要的小结。

4.5.1 实验设置

本节主要介绍了验证本章算法所使用的实验数据集、脉冲神经网络结构以及卷积神经网络训练中所用的相关技术。

4.5.1.1 实验数据集

本章采用MNIST数据集和CIFAR10数据集作为主要的测试数据集。MNIST数据集包含70,000张28×28大小的$0-9$的手写数字灰度图片，其中60,000张作为训练集，10,000张作为测试集。CIFAR10数据集包含60,000张32×32大小的10类彩色自然图片，其中50,000张作为训练集，10,000张作为测试集。

4.5.1.2 网络结构设置

为了验证本文提出算法的有效性，使用如下表 4.1所示的网络设置。除此之外，本章使用一个VGG19结构的卷积神经网络验证本文算法在深度网络的条件下的有效性。

表 4.1 网络结构设置(按列显示)。卷积层的参数按照conv(感受野尺寸)-(通道数目)表示。

Table 4.1 Network configurations(shown in columns). The convolutional layer parameters denote as conv(receptive field size)-(number of channels).

Network Configuration			
MNIST		CIFAR-10	
NetworkA 4 weight layers	NetworkB 6 weight layers	NetworkC 5 weight layers	NetworkD 7 weight layers
input (28 × 28) gray images		input (32 × 32) RGB images	
conv5-32	conv3-32 conv3-32	conv5-32	conv3-32 conv3-32
max-pool			
conv5-64	conv3-64 conv3-64	conv5-64	conv3-64 conv3-64
max-pool			
FC-1024		FC-2048	
FC-10		FC-2048	
		FC-10	
softmax			

4.5.1.3 卷积神经网络训练技术

本节介绍训练卷积神经网络时，为了获得高性能的卷积神经网络采用的训练技术，主要为Dropout技术以及BatchNormalization技术。Dropout技术是一种能够有效避免卷积神经网络全连接层过拟合的技术。BatchNormalization技术可以使网络各层的输入固定为均值为0方差为1的分布，其主要过程如下式所示：

$$y_i = \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \equiv BN_{\gamma,\beta}(x_i) \tag{4.11}$$

其中，x_i是规范化层(normalization layer)的输入，y_i是该层的输出，μ_B和σ_B^2分别代表x_i的均值以及方差，ϵ是为了防止数值计算不稳定的参数，γ和β是放缩和旋转参数。在转换脉冲神经网络中通过如下公式将卷积层与规范化层合并：

$$\hat{W} = \frac{\gamma}{\sigma_B} W \tag{4.12}$$

$$\hat{b} = \frac{\gamma}{\sigma_B}(b - \mu_B) + \beta \tag{4.13}$$

其中，W、$\hat{W}$、b、$\hat{b}$分别代表合并前后权值矩阵以及偏置向量。

4.5.2 卷积神经网络的识别精度

表 4.2展示了限制网络输出训练算法的性能。在MNIST数据集上，本节中使用表 4.1中的NetworkB；在CIFAR10数据集上，本节中使用表 4.1中的NetworkD。表 4.2的第3行表示使用反向传播算法训练得到的卷积神经网络的识别精度，为限制网络输出预训练算法输入网络。BST列表示使用常规的反向传播算法训练得到的卷积神经网络直接对输入特征图(Output feature map)进行裁剪后，再前向传播获得的识别精度。在MNIST数据集上可以看出，裁剪对精度影响较大，裁剪参数越大，该影响越小；经过限制网络输出预训练算法调整之后，识别精度已经基本可以逼近原始的识别精度99.60%。在CIFAR10数据集上可以看出，经过限制网络输出预训练算法调整之后，卷积神经网络的识别精度甚至有所提升。因此，在数据集规模较小时，限制网络输出预训练算法可以使裁剪后的卷积神经网络获得和算法输入卷积神经网络基本相同的识别精度；当数据集规模增大时，该算法甚至对输入卷积神经网络的识别精度略有提升。

表 4.2 限制网络输出预训练算法的性能。BST是Baseline Training的缩写，表示卷积神经网络的反向传播训练算法；ROT是Restricted Output Training的缩写，表示限制网络输出预训练算法；符号*这行表示使用BST训练得到的卷积神经网络的起始识别精度；clip列表示整个卷积神经网络使用的裁剪参数。

Table 4.2 The performance of restricted output training. BST indicates baseline training, ROT indicates restricted output training.

	MNIST		CIFAR-10	
clip	BST	ROT	BST	ROT
*		**99.60%**		**92.95%**
1.0	91.59%	99.48%	92.89%	92.92%
1.2	96.03%	99.51%	92.92%	92.96%
1.4	97.67%	99.50%	92.92%	**93.04%**
1.6	98.45%	99.52%	92.93%	**93.02%**

4.5.3 脉冲神经网络的识别精度

表 4.3展示了各种不同算法获得的脉冲神经网络的识别精度。在MNIST数据集上，本章提出的算法可以获得与之前的算法研究中类似的识别精度。由于CIFAR10数据集上的识别任务比MNIST数据集要复杂，在CIFAR10数据集上使用更深层的脉冲神经网络效果更好。本章之前的典型算法研究BinaryConnect SNN[99]在CIFAR10数据集上取得最好的识别精度为90.85%，本章提出的算法取得更好的识别精度为94.00%。为了更公平地进行识别精度比较，本章使用与同

期算法[97, 99, 181]相似规模的网络模型NetworkD。本章提出的算法获得的识别精度为92.89%，相对于转换之前的卷积神经网络的识别精度只有0.06%识别精度的损失。

表 4.3 不同训练算法获得的脉冲神经网络的识别精度表。

Table 4.3 Classification accuracy of different SNNs.

Dataset	Network-Type	Accuracy
MNIST	Spiking ConvNet[118]	99.10%
	Spiking ConvNet[99]	99.44%
	Arousal AdSNNs[183]	99.56%
	Our paper (NetworkB)	**99.58%**
CIFAR-10	Spiking ConvNet[97]	77.43%
	LIF SNN[180]	83.54%
	Q4-SNN[181]	84.52%
	Bio-Inspired SNN[184]	86.43%
	SNN on TrueNorth[163]	89.32%
	Arousal AdSNNs[183]	89.88%
	BinaryConnect SNN[108]	90.85%
	Our Paper (NetworkD)	**92.89%**
	Our Paper (VGG19)	**94.00%**

4.5.4 脉冲神经网络的收敛速度

本小节主要通过3个方面考察转换后的脉冲神经网络的收敛速度：1.限制网络输出预训练算法；2.错误脉冲抑制算法；3.时序最大值池化算法。

图 4.7给出了本章提出的算法之间组合的效果比较。相对于本章之前的脉冲神经网络模型转换算法[99, 118]，将限制网络输出预训练算法和错误脉冲抑制算法结合使用可以使转换的脉冲神经网络的收敛速度加快。更进一步，将本章提出的限制网络输出预训练算法以及错误脉冲抑制算法结合使用，可以使转换后获得的脉冲神经网络在30个时间步内收敛。与之相对的，基本算法[99, 118]不仅收敛速度较慢，同时网络的识别精度低于本章算法。仅使用限制网络输出预训练算法时，对基本算法的收敛速度基本没有提升。相对于基本算法，仅使用错误脉冲抑制算法时，可以将网络的收敛时间步数减少到300步左右。限制网络输出预训练算法可以有效地增强错误脉冲抑制算法，将网络的收敛时间步数大幅减小。

- **限制网络输出预训练算法**

从图 4.7中可以看出，单独使用限制网络输出预训练算法时，对脉冲神经网络的收敛速度提升不大。因此，本章在结合错误脉冲抑制算法条件下，调节限

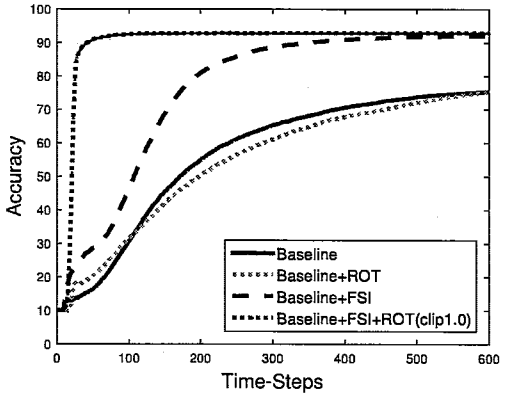

图 4.7 限制网络输出预训练算法以及错误脉冲抑制算法对脉冲神经网络收敛速度影响的示意图。其中，FSI是False Spike Inhibition的缩写，表示限制网络输出预训练算法；ROT是Restricted Output Training的缩写，表示错误脉冲抑制算法；Baseline表示[97]中采用的脉冲神经网络模型转换算法。clip1.0表示限制网络输出预训练算法中裁剪参数为1.0。

Figure 4.7 The performance of restricted output training and false spike inhibition. In this figure, FSI indicates false spike inhibition, ROT indicates restricted output training, Baseline indicates the algorithm in the work[97].

制网络输出预训练算法中裁剪参数，来测试裁剪参数对网络收敛速度的影响。从图 4.8中可知，在25个时间步以内，裁剪参数为1.0时，网络的收敛速度最快。与此同时，裁剪参数为1.4与裁剪参数为1.0的收敛速度基本相同。但是，相对于裁剪参数为1.0时，裁剪参数为1.4时卷积神经网络更容易训练。在更大的数据集和更深的脉冲神经网络上，更多的裁剪参数可以得到相似效果的脉冲神经网络。

- 错误脉冲抑制算法

假设，下一段中FSI表示错误脉冲抑制算法，ROT表示限制网络输出预训练算法，ROT-SNN表示使用ROT技术的模型转换算法获得的转换脉冲神经网络，NOM-SNN表示使用基本算法获得的转换脉冲神经网络，本小节对错误脉冲抑制算法的效果讨论如下。

从图 4.7中Baseline+FSI曲线和Baseline+FSI+ROT(clip1.0)曲线看出，转换后获得的脉冲神经网络的收敛速度被优化。由表 4.4中可知，只有使用错误脉冲抑制算法才可以在360个时间步内达到90%以上的识别精度。使用FSI技术的ROT-SNN在60个时间步之内，相对于原始的ROT-SNN，可以取得4倍的识别精度的提升。在NOM-SNN上使用FSI技术，可以使脉冲神经网络在360个时间步时的识别精度提升22%。更进一步地，结合使用ROT技术和FSI技术，在保证识别精度

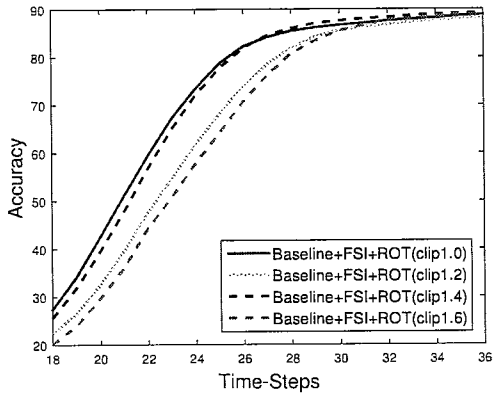

图 4.8 限制网络输出预训练算法对脉冲神经网络收敛速度影响的示意图。其中，FSI是False Spike Inhibition的缩写，表示限制网络输出预训练算法；ROT是Restricted Output Training的缩写，表示错误脉冲抑制算法；Baseline表示[97]中采用的脉冲神经网络模型转换算法。clipF表示限制网络输出预训练算法中裁剪参数为F。

Figure 4.8 The performance of restricted output training. In this figure, FSI indicates false spike inhibition, ROT indicates restricted output training, Baseline indicates the algorithm in the work[97].

至少为91%的条件下，FSI技术将脉冲神经网络收敛时间从360个时间步减小到了60个时间步。在图 4.9中，本章衡量了不同的逼近精度控制参数n对错误脉冲抑制算法效果的影响。从该图中可以看出，当$n \geq 8$时，错误脉冲抑制算法就已经达到了其最佳效果。因此，为了减小错误脉冲抑制算法的复杂度，在本章其他实验中，均使用逼近精度控制参数$n = 8$。

表 4.4 错误脉冲抑制算法效果比较表。NOM-SNN表示使用基本算法[97]获得的转换脉冲神经网络；ROT-SNN表示使用限制网络输出预训练算法获得的转换脉冲神经网络；FSI表示错误脉冲抑制算法。

Table 4.4 Classification accuracy of SNNs with restricted output training and false spike inhibition. NOM-SNN indicates the network in the work[97], ROT-SNN indicates the converted SNN using the restricted output training, FSI indicates false spike inhibition.

time-steps	25	30	60	80	360
ROT-SNN(clip1.0)+FSI	82.18%	87.05%	91.34%	92.12%	92.75%
ROT-SNN(clip1.0)	18.13%	17.74%	22.29%	26.94%	65.66%
NOM-SNN+FSI	22.66%	24.12%	30.26%	37.07%	90.11%
NOM-SNN	13.78%	14.10%	19.01%	24.66%	68.61%

- 时序最大值池化算法

在网络深度不够的时候，时序最大值池化算法对转换后获得的脉冲神经网络提升效果不明显。图 4.10比较了在VGG19结构上转换获得的脉冲神经网络中

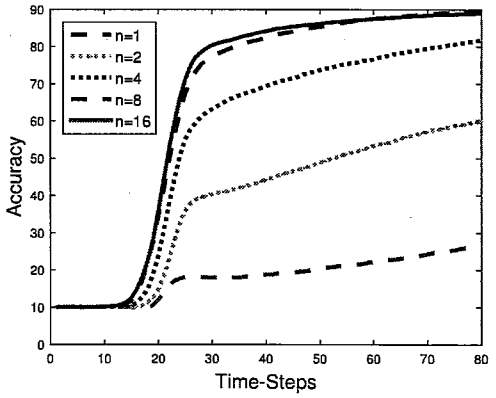

图 4.9 错误脉冲抑制算法对脉冲神经网络收敛速度影响的示意图。该图使用裁剪参数为1.0的限制网络输出预训练算法获得转换之前的卷积神经网络。参数n表示公式(4.10)中的逼近精度控制参数。

Figure 4.9 The performance of false spike inhibition with restricted output training(clip1.0). The parameter n means the approximate precision parameter in (4.10).

使用时序最大值池化算法的效果。通过比较Baseline曲线和Baseline+TP曲线可以看出，在100-300时间步内，时序最大值池化算法对转换后获得的脉冲神经网络在识别精度上有10%的提升。当使用本章提出的所有3种算法，转换后获得的脉冲神经网络在200个时间步内获得92%左右的识别精度。图中前100个时间步，识别精度保持为10%是由于网络层数较大，网络低层中的脉冲需要一定的时间步才可以传输到高层中。

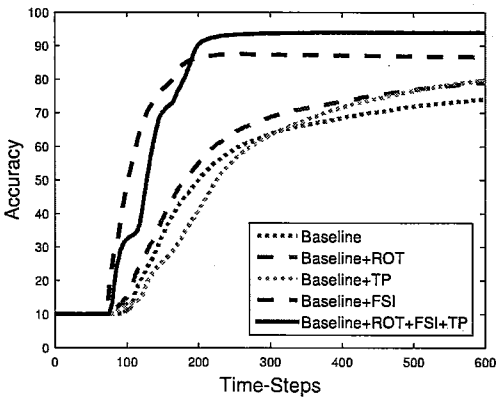

图 4.10 VGG结构的深度脉冲神经网络上的实验效果示意图。ROT是Restricted Output Training的缩写，表示错误脉冲抑制算法；TP是Temporal Max Pooling的缩写，表示时序最大值池化算法；Baseline表示[97]中采用的脉冲神经网络模型转换算法。

Figure 4.10 The experimental results by using VGG19-like CNN to convert SNNs. In this figure, TP means the temporal max pooling and Baseline indicates the algorithm in the work[97].

4.5.5 实验结果小结

为了验证本章所提算法的有效性，本节通过三个方面测试了算法的性能。首先，在卷积神经网络的识别精度方面，验证了限制网络输出的预训练算法可以在保证卷积神经网络的识别精度的前提下，将模型转换算法中的参数规范化过程动态地迁移到卷积神经网络的训练过程中。其次，在转换脉冲神经网络的识别精度方面，本章所提算法取得了较之前算法研究在MNIST数据集以及CIFAR10数据集上更优的实验结果。最后，结合限制网络输出预训练算法以及错误脉冲抑制算法，可以有效地将转换脉冲神经网络的收敛时间限制在30个时间步以内。同时，时序最大值池化算法可以在VGG19结构的深度脉冲神经网络上带来10%的识别精度提升。

4.6 本章小结

脉冲神经网络模型转换算法前面的研究中已经在一些数据集上取得了和卷积神经网络类似的性能，但是转换脉冲神经网络的收敛延迟问题没有得到充分地关注。大部分研究中采用的参数规范化技术增加了转换脉冲神经网络的收敛时间，带来了更多的计算操作，从而损害了脉冲神经网络低功耗的特性。本章首先提出一种限制网络输出的预训练算法，从本质上解决了脉冲神经元无法表达卷积神经网络中对应神经元大于1的输出值的问题，同时不影响转换脉冲神经网络的收敛速度。然后，本章分析了转换脉冲神经网络各层输出的特点，发现了在转换脉冲神经网络中存在的错误脉冲现象，通过将错误脉冲抑制问题抽象化为一个线性规划问题，大大减小了转换脉冲神经网络中的错误脉冲数目。接着，本章提出一种时序最大值池化算法，可以无损地将卷积神经网络中的最大值池化操作移植到转换脉冲神经网络中。最后，实验表明，本章算法可以在保证转换脉冲神经网络识别精度的条件下，得到低延迟的深度脉冲神经网络。

第5章 基于反向传播的极低延迟深度脉冲神经网络转换学习算法

前面两章的工作表明，转换脉冲神经网络已经可以获得和卷积神经网络识别精度相同的深度转换脉冲神经网络，并可以减小转换脉冲神经网络的网络收敛时间。2017年底，Han 等 [125]设计了一种基于同步电路的神经网络硬件加速器用于比较卷积神经网络以及转换脉冲神经网络的功耗，指出在CIFAR10数据集上，转换脉冲神经网络的功耗是相同规模的卷积神经网络的5.4倍。该工作中采用模型转换算法[118]，转换网络的收敛时间为65个时间步。如果可以将转换神经网络的收敛时间步数限制在10以内(上一章所提算法获得的同等规模的转换脉冲神经网络的收敛时间为30个时间步)，初略估计可以使转换脉冲神经网络的功耗低于相同网络规模的卷积神经网络。更进一步地，文献[125]中使用基于频率编码的模型转换算法，如果能够利用脉冲序列中的时序信息，就能够进一步地减小转换脉冲神经网络的收敛时间。因此，进一步地减小转换脉冲神经网络的网络收敛时间十分必要。

本章提出一种基于反向传播算法的脉冲神经网络转换学习算法，通过反向传播算法学习转换网络中脉冲序列的时序信息中的有效特征，进一步地减小了转换脉冲神经网络的网络收敛时间。具体来说，首先，本章通过分析现有脉冲神经网络中使用的神经元模型，选定LIF神经元模型作为搭建深度脉冲神经网络的基本元素。然后，本章总结了现有的脉冲神经网络反向传播算法的优缺点，并且归纳了脉冲神经网络反向传播算法在训练深度脉冲神经网络需要满足的三个苛刻条件。接着，通过分析卷积神经网络与脉冲神经网络之间的联系，本章提出一种了参数初始化算法，可以有效克服训练深度脉冲神经网络时难以满足上述三个苛刻条件的问题。同时，本章提出了一种设计脉冲神经网络超参数的误差最小化算法，修改了损失函数中频率编码的损失项与算法计算出的带时序信息的梯度之间的不匹配关系。本章提出的三种算法支持更深的脉冲神经网络训练，同时显著提升了脉冲神经网络训练算法的收敛速度。最后，实验表明，本章提出的算法可以获得极低延迟的深度脉冲神经网络。

5.1 引言

由于脉冲神经网络事件驱动的特性可以用于设计极低功耗的硬件，以及脉冲这种更类脑的信息传递方式，脉冲神经网络的训练学习算法的研究在深度神

经网络的这波浪潮之中被广泛地关注。按照是否直接使用脉冲神经网络进行训练，可以将这些学习算法分成两类：脉冲神经网络直接训练学习算法与脉冲神经网络模型转换算法。

脉冲神经网络的直接训练学习算法又可以分为脉冲神经网络监督学习算法以及脉冲神经网络非监督学习算法。根据突触权值学习规则的不同，脉冲神经网络监督学习算法又可以分为基于突触权值可塑性的监督学习算法[55, 60, 61]、基于脉冲序列卷积的监督学习算法[90, 91]、基于对比散度算法的监督学习算法[92, 120, 121]以及基于梯度下降规则的监督学习算法[76, 85, 86, 88, 89]等四类。其中，基于梯度下降规则的监督学习算法是目前最有希望获得高性能的深度脉冲神经网络的一类算法。这类算法的反向传播机制类似于卷积神经网络中的反向传播算法，只是由于脉冲神经网络在发送脉冲的时候的不可微性，使其难以获得准确的反向梯度。同时，这类算法只在产生脉冲的附近小的时间域内产生学习作用，因此其训练算法较卷积神经网络的训练算法更加难以收敛。当脉冲神经网络的网络规模逐渐增加的时候，这类算法会慢慢失效。与此同时，脉冲神经网络的非监督学习算法还处在研究的初始阶段[66, 68, 69, 123]，还未取得良好的效果。

由前文可知，脉冲神经网络模型转换算法可以取得与卷积神经网络相似的识别精度[97, 99, 118]。但是，转换脉冲神经网络仅利用了脉冲序列间的空间域信息，无法合理地利用脉冲序列的时间域信息，因此转换脉冲神经网络的网络收敛时间很长。基于以上原因，转换脉冲神经网络很难实现低功耗硬件[125]。

一个直观的主意是将基于梯度下降规则的脉冲神经监督学习算法和脉冲神经网络转换学习算法结合起来。通过脉冲神经网络模型转换算法，本章可以获得一个可以处理空间域信息的转换脉冲神经网络。然后，通过脉冲神经网络的反向传播算法可以使该转换脉冲神经网络通过学习过程逐渐获得处理脉冲序列间的时序信息的能力。这样做既可以减轻脉冲神经网络的反向传播算法的任务难度，使其能够取得更好的效果，又可以减小转换脉冲神经网络的网络收敛时间较长的缺点。最终，本章将获得一个极低延迟的深度转换脉冲神经网络。

5.2 神经元模型描述

本节首先总结了脉冲神经网络中常用的两种脉冲神经元模型，选定LIF神经元模型作为后面实验的神经元模型；然后介绍了离散LIF神经元模型的基本数学模型。

5.2.1 神经元模型选择

计算神经科学中有各种不同的脉冲神经元模型，其中Hodgkin-Huxley神经元模型[40]是最早的一个使用常微分方程组定量化生物神经元的脉冲神经元模型。该模型是迄今为止对生物神经元近似效果最好的模型，但是使用该模型实现深度脉冲神经网络的复杂度过高。从第二章的表 2.1中可知，LIF神经元模型以及SRM神经元模型是脉冲神经网络训练算法中常用的两种基本神经元模型。LIF神经元模型是IF神经元模型的一个简化版本，主要包含两个独立的过程来描述它的动态行为：1. 一个微分方程描述膜电势的累积过程；2. 脉冲产生机制的描述。该模型的主要缺点是无法记录之前的脉冲序列的情况。这个模型通过发送脉冲之前的膜电势变化来传输时序信息。SRM模型使用解析表达式来描述神经元的动态行为，将独立的脉冲发送时间当作状态变量。该模型包含脉冲发送后的不应期(refractoriness)的模拟，但是IF神经元模型一般不能模拟神经元的不应期。所以SRM模型的复杂度比LIF神经元模型的复杂度要高。同时，独立的脉冲时间序列使得学习算法难以决定什么时候应该产生或者删除脉冲。

在这两个神经元模型之间，本章选用LIF神经元模型。基于以下原因：1. LIF神经元模型只是积累膜电势，然后往外发送脉冲。本章中认为这两项基本功能就足以应付脉冲神经网络中大多数的信息处理过程；2. 使用更简单的神经元模型训练深度脉冲神经网络更加有利于获得性能更佳的脉冲神经网络，同时避免训练过程中的一些难以意料的问题。一个简单的神经元模型对于学习算法意味着只需要搜索一个更小的参数空间，学习算法在不需要更多的附加技术的辅助下就可以得到很好的结果；3. 大多数的脉冲神经网络模型转换算法采用LIF神经元模型。如第二章的表 2.1中所示，这些模型转换算法相对于其他学习算法已经可以获得和卷积神经网络性能类似的大规模脉冲神经网络。

5.2.2 离散LIF神经元模型

在本章中，选用著名的Leaky Integrate-and-Fire神经元模型作为脉冲神经元模型(以下简称为LIF神经元模型)。该模型膜电势变化的动态特性可以用以下公式表示：

$$\tau\frac{du(t)}{dt} = -[u(t) - u_{rest}] + RI(t) \tag{5.1}$$

其中，$u(t)$表示膜电势在t时刻的值，$I(t)$表示前序神经元的外部输入电流，R表示输入电阻，u_{rest}表示重置电势，τ表示膜时间常数。当膜电势$u(t)$超过阈值V_{th}时，膜电势$u(t)$被重置为u_{rest}。从上式可以看出，脉冲在神经网络内的传输主

要可以分为2个部分：1.在空间域，脉冲将信息从网络的低层向高层逐层传播；2.在时间域，膜电势的累积过程代表着前序神经元的当前时刻之前发送给该神经元的脉冲带来的影响。使用上述一阶线性微分方程搭建深度脉冲神经网络，进行训练时运算量需求过大。因此本章对该模型进行离散化简化，获得了离散化LIF神经元模型，如下式所示：

$$u_i^l(t+1) = \tau(1-o_i^{t,l})u_i^l(t) + x_i^{t+1,l} \tag{5.2}$$

$$x_i^{t,l} = \sum_{j=1}^{M^{l-1}} w_{ij}^l o_j^{t,l-1} + b_i^l \tag{5.3}$$

$$o_i^{t,l} = g(u_i^l(t)) \tag{5.4}$$

其中，上标l表示神经元所在神经网络的层数，下标i和j表示神经元在该层的位置。公式 (5.2)中，时刻t的膜电势$u_i^l(t+1)$取决于上一时刻t的膜电势$u_i^l(t)$和前序神经元输入$x_i^{t+1,l}$。公式 (5.1)中外部输入$RI(t)$对应于公式 (5.3)中的$w_{ij}^l o^{t,l-1}$之和，$o^{t,l-1}$是上一层神经元的脉冲输出，b_i^l表示神经元的偏置。公式 (5.4)中，w_{ij}^l表示第l层的第i个神经元的第j个突触权值，M^{l-1}表示第$l-1$层神经元的数目。公式 (5.4)中，脉冲产生的规则如下式所示：

$$g(u_i) = \begin{cases} 1, & if \quad u_i \geq V_{th} \\ 0, & otherwise \end{cases} \tag{5.5}$$

其中V_{th}是膜电势的阈值。公式 (5.2)、公式 (5.3)和公式 (5.4)表示的过程如下示意图5.1。

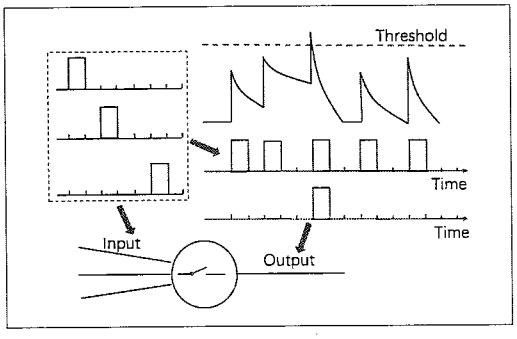

图 5.1 离散化LIF神经元的动态模型。在每个时间步，该神经元收集前序神经元的脉冲输入，积累膜电势；当膜电势超过阈值的时候，该神经元向后序神经元发送一个脉冲。

Figure 5.1 Iterative LIF neuron. At each time step, the neuron collects the input spikes, accumulates the membrane potential and generates the output spikes when the membrane potential exceeds the potential threshold.

5.3 脉冲神经网络反向传播算法分析

本节主要包括4个小节：1. 首先分析了现有的典型的脉冲神经网络反向传播算法，然后发现STBP算法[87]对于训练深度脉冲神经网络具有很大的潜力；2. 为了后续的算法分析，详细介绍了STBP算法的细节；3. 推导了训练深度脉冲神经网络的三个严苛条件；4. 最后，分析了卷积神经网络和脉冲神经网络之间的联系，利用该联系，证明了可以使用该联系满足上一步提到的三个严苛条件。

5.3.1 脉冲神经网络反向传播算法比较

从第二章表 2.1中可以看出，反向传播算法是脉冲神经网络学习算法中研究最广的一种算法，同时这类算法可以被用来训练多层脉冲神经网络。

SpikeProp算法[76]是最早提出的基于反向传播算法的脉冲神经网络监督学习算法之一。在SpikeProp算法中，使用实际输出脉冲时间序列与期望输出脉冲时间序列之间在时间域上的差距来定义误差。通过神经元膜电势在脉冲发送时间附近的极小范围内的线性关系，该算法近似了脉冲之间的不可微性。然后，就可以使用卷积神经网络的反向传播算法类似的过程推导脉冲神经网络的反向传播算法。SpikeProp算法有很多变式，典型算法包括RProp算法[78]、SpikeProp-Ad算法[79]、Extending SpikeProp算法[77]、EvSpikePropRT算法[80, 81]，这些算法加速了SpikeProp算法的收敛，Multi-SpikeProp算法[82]只需要限制输出神经元层的神经元发送一个脉冲。这类算法的主要问题是网络中的每个神经元只能发送事先定义好的脉冲个数。因此，这类算法只在很小的数据集上进行评估，比如异或问题以及Fisher Iris数据集。

近年来，一些基于反向传播算法的脉冲神经网络学习算法已经开始显示出有竞争力的效果了。Lee 等 [85]将脉冲之间的不可微性看作是一种噪声，然后将脉冲神经元的膜电势看作是反向传播算法中的可微分变量，进而推导出一种训练多层脉冲神经网络的监督学习算法。该项工作用隐式的方法获得脉冲之间的时域上的相关信息。Wu 等 [86]使用类似随时间的反向传播算法(BackPropagation Through Time，BPTT算法)的推导得到了STBP算法(Spatio-Temporal BackPropagation)，可以显式地获取脉冲之间的时序关系。该算法通过对脉冲进行平滑来解决脉冲之间的不可微性，从而计算出反向传播梯度。上述两项工作均采用频率编码(rate code)方式的损失函数，与计算出来的时域上的梯度存在着一定的不匹配。Jin 等 [89]提出一种HM2-BP算法来避免上述的一些限制。该算法中实现了两个尺度的反向传播算法，包括在脉冲频率上的方向传播算法(macro-level)和

脉冲序列上的反向传播算法(micro-level)，然后通过两个尺度上的反向传播算法的交互获得反向传播的梯度。但是上述的所有研究中都存在着沉默神经元的问题(dead neuron problem)，即是学习算法只有在神经元往外发送脉冲的时候进行权值调整学习。当所训练的脉冲神经网络层数加深的时候，该问题会变得更加难以解决。Shrestha 和 Orchard [88]提出的SLAYER算法给出了一种解决办法。该算法使用更高的脉冲频率(甚至不应该发送脉冲的神经元也让其往外发送脉冲)来开始神经网络的训练，来缓解沉默神经元的问题。同时，该算法不仅学习权值的变化，而且学习轴突上的延迟的变化。但是，因为SLAYER算法包含两个积分过程，这个算法比上述其他算法消耗的运算量要大得多。

从第二章表 2.1中可以看出，STBP算法比其他的脉冲神经网络学习直接训练算法训练所得的脉冲神经网络的模型大小要大一些。与此同时，脉冲神经网络的模型转换算法训练所得的模型大小是最大的。从Han 等 [125]的工作中可以看出，在CIFAR10数据集上，转换后获得的脉冲神经网络需要65个时间步精度才能收敛，转换后的脉冲神经网络消耗的功耗是原始的卷积神经网络的5.4倍。只需将转换脉冲神经网络的网络收敛时间减小到10个时间步以内，即可获得功耗比卷积神经网络更低的转换脉冲神经网络。由于STBP算法足够简单，可以将STBP算法中的脉冲神经网络和原始的卷积神经网络中找到一定的联系，从而使用STBP算法将转换后的脉冲神经网络的收敛时间减少到10个时间步以内。该联系的另外一个优势是，研究者还可以从因特网上下载卷积神经网络的相关参数来初始化STBP算法中的权值参数。同时，这个联系还可以避免深度神经网络训练中的沉默神经元问题。基于以上的原因，本章选择STBP算法来加速脉冲神经网络的模型转换算法，以获得性能更优的转换脉冲神经网络。

5.3.2 STBP算法背景介绍

本章采用STBP算法[86]来训练脉冲神经网络。该算法可以充分地利用空间域-时间域(spatio-temporal domain)的信息来搭建多层脉冲神经网络。给定训练集的第s个数据的数据标签向量y_s以及脉冲神经网络的输出向量o_s，定义均方误差(mean squared loss)函数如下式所示：

$$Loss = \frac{1}{2S}\sum_{s=1}^{S}||y_s - \frac{1}{T}\sum_{t=1}^{T}o_s^{t,L}||_2^2 \tag{5.6}$$

其中，L表示脉冲神经网络的最后一层，S表示每批(batch)处理的训练数据的数目，T表示一次脉冲神经网络模拟的总的时间步，t表示第t个时间步。

为了简化梯度的计算过程，定义变量$\delta_i^{t,l}$如下所示：

$$\delta_i^{t,l} = \frac{\partial Loss}{\partial o_i^{t,l}} \tag{5.7}$$

其中，$o_i^{t,l}$表示第l层的第i个神经元在第t个时间步的输出信号。当$t = T$以及$l = L$时，可以推出

$$\frac{\partial Loss}{\partial o_i^{T,L}} = -\frac{1}{TS}(y_i - \frac{1}{T}\sum_{t=1}^{T} o_i^{t,L}) \tag{5.8}$$

当$t < T$以及$l < L$时，偏导数$\frac{\partial Loss}{\partial u_i^{t,l}}$既在时间域(temporal domain)上受影响，也在空间域(spatial domain)上受影响。在空间域上，权值误差信号从网络高层往低层传播。在时间域上，传播的权值误差来自于膜电势的动态变化。因此，计算偏导数$\frac{\partial Loss}{\partial u_i^{t,l}}$如下：

$$\begin{aligned}\frac{\partial Loss}{\partial u_i^{t,l}} &= \frac{\partial Loss}{\partial o_i^{t,l}}\frac{\partial o_i^{t,l}}{\partial u_i^{t,l}} + \frac{\partial Loss}{\partial o_i^{t+1,l}}\frac{\partial o_i^{t+1,l}}{\partial u_i^{t,l}} \\ &= \delta_i^{t,l}\frac{\partial g}{\partial u_i^{t,l}} + \tau(1 - o_i^{t,l})\delta_i^{t+1,l}\frac{\partial g}{\partial u_i^{t+1,l}}\end{aligned} \tag{5.9}$$

与此同时，变量$\delta_i^{t,l}$计算如下：

$$\begin{aligned}\delta_i^{t,l} &= \sum_{j=1}^{M^{l+1}} \delta_j^{t,l+1}\frac{\partial o_j^{t,l+1}}{\partial o_i^{t,l}} + \delta_i^{t+1,l}\frac{\partial o_i^{t+1,l}}{\partial o_i^{t,l}} \\ &= \sum_{j=1}^{M^{l+1}} \delta_j^{t,l+1}\frac{\partial g}{\partial u_i^{t,l}}w_{ji}^{l+1} - \tau\delta_i^{t+1,l}\frac{\partial g}{\partial u_i^{t+1,l}}u_i^{t+1,l}\end{aligned} \tag{5.10}$$

从公式 (5.9)中看出，偏导数$\frac{\partial g}{\partial u_i^{t,l}}$中的输出门变量函数$g(\cdot)$在每个神经元发送脉冲的时候不可微。理论上，该偏导数$\frac{\partial g}{\partial u_i^{t,l}}$是狄拉克函数$\delta(u)$。狄拉克函数$\delta(u)$难以在反向传播算法中应用。因此，此处使用如下近似函数$h(\cdot)$对偏导数$\frac{\partial g}{\partial u_i^{t,l}}$进行近似：

$$h(u) = \frac{1}{\Delta}sign(|u - V_{th}| < \frac{\Delta}{2}) \tag{5.11}$$

其中，Δ是一个常值变量。当$\Delta \to 0^+$时，该近似函数$h(\cdot)$可以完美地逼近偏导数$\frac{\partial g}{\partial u_i^{t,l}}$：

$$\lim_{\Delta \to 0^+} h(u) = \frac{dg}{du} \tag{5.12}$$

由此可以推出权值矩阵$\mathbf{W}$以及偏置向量$\mathbf{b}$的梯度如下式所示：

$$\begin{aligned}\frac{\partial Loss}{\partial \mathbf{W}^l} &= \sum_{t=1}^{T}\frac{\partial Loss}{\partial \mathbf{u}^{t,l}}\frac{\partial \mathbf{u}^{t,l}}{\partial \mathbf{x}^{t,l}}\frac{\partial \mathbf{x}^{t,l}}{\partial \mathbf{W}^l} \\ &= \sum_{t=1}^{T}\frac{\partial Loss}{\partial \mathbf{u}^{t,l}}(\mathbf{o}^{t,l-1})^T\end{aligned} \tag{5.13}$$

$$\frac{\partial Loss}{\partial \mathbf{b}^l} = \sum_{t=1}^{T} \frac{\partial Loss}{\partial \mathbf{u}^{t,l}} \frac{\partial \mathbf{u}^{t,l}}{\partial \mathbf{b}^l} = \sum_{t=1}^{T} \frac{\partial Loss}{\partial \mathbf{u}^{t,l}} \tag{5.14}$$

其中，$(\mathbf{o}^{t,l-1})^T$是神经网络层输出向量$\mathbf{o}^{t,l-1}$的转置。在公式 (5.7)- (5.12)中，为了更容易理解变量之间的关系，使用分量对反向传播算法进行推导。公式 (5.13)和(5.14)给出了反向传播梯度的矩阵以及向量表达式。

5.3.3 脉冲神经网络中反向传播算法的难点

本章仔细地研究了反向传播算法的学习过程，发现基于反向传播算法训练深度脉冲神经网络需要满足极其严苛的条件。首先，初始化之后的脉冲神经网络中不能有太多的沉默神经元，否则会产生沉默神经元问题(dead neuron problem)。然而，从表 5.1中可以看出，如果使用常用的初始化方法初始化脉冲神经网络，沉默神经元的数目会随着脉冲神经网络的层数的增加而增加。其次，当且仅当反向传播路径上的突触权值较大时，传播的梯度才能从脉冲神经网络的高层传递到网络的低层。与之类似的，当训练深度卷积神经网络的时候，如果反向传播路径之上有某个突触权值较小，梯度很难从卷积神经网络的输出层传到网络的低层。这是因为梯度传递时，传递项中包含权值的连乘项，当有较小的突触权值的时候，该连乘项很容易为零。在此处，本章将使用STBP算法为例阐述脉冲神经网络中的严苛条件的推导。

在空间域上，公式 (5.10)右边的第一项表示梯度随网络逐层传递的过程。如果偏导数$\frac{\partial g}{\partial u_i^{t,l}}$为零，从公式 (5.10)看出，第$l+1$层变量$\delta_j^{t,l+1}$不会对第$l$层变量$\delta_i^{t,l}$产生任何作用。因此，偏导数$\frac{\partial g}{\partial u_i^{t,l}}$大于等于零，当且仅当

$$u_i^{t,l} \in (V_{th} - \frac{\Delta}{2}, V_{th} + \frac{\Delta}{2}). \tag{5.15}$$

从公式 (5.12)中可以看出，参数Δ需要选一个较小的值。从上式推出，当且仅当第l层的第i个脉冲神经元往外发送脉冲时，变量$\delta_i^{t,l}$才可以收到第$l+1$层网络变量$\delta_j^{t,l+1}$的影响。从公式 (5.9)中，可以推出

$$\frac{\partial Loss}{\partial u_i^{t,l}} > 0 \quad if \quad \delta_i^{t,l} > 0 \quad or \quad \delta_i^{t+1,l} > 0. \tag{5.16}$$

在第t个时间步，从公式 (5.13)中，可以推出

$$\frac{\partial Loss}{\partial w_{ji}} > 0 \quad if \quad \frac{\partial Loss}{\partial \mathbf{u}_i^{t,l}} > 0 \quad and \quad \mathbf{o}_j^{t,l-1} > 0. \tag{5.17}$$

上述公式表明，第$l-1$层的第j个神经元与第l层的第i个神经元之间的突触权值改变，当且仅当第i个神经元接收到第j个神经元的脉冲同时第i个神经元处于接

近于向外发送脉冲的状态。更进一步地，突触权值改变当且仅当该突触处在一条脉冲从网络的输入层传输到输出层的路径上，如图 5.2中所示。因此，本章可以推导出训练深度脉冲神经网络的第一个严苛条件如下：

条件 1: 在脉冲神经网络训练开始的时候，网络中有足够多的脉冲路径。

与此同时，突触权值也应该考虑在内。从第l层脉冲神经网络传输到最后第L输出层，变量$\delta_i^{t,l}$通过公式 (5.10)可以推出

$$\delta_i^{t,l} \propto \delta_{i_L}^{t,L} w_{i_1 i}^{l+1} w_{i_2 i_1}^{l+2} \ldots w_{i_L i_{L-1}}^{L}. \tag{5.18}$$

其中，下标索引i_s表示各层神经元的索引。本文将公式 (5.18)中的突触权值连乘项定义为权值链(weight chain)。这些突触权值的连乘项使得训练深度脉冲神经网络的难度加大。当权值链中有任意一个突触的权值很小时，反向传播的梯度就很难传递到脉冲神经网络的低层。因此，推导出训练深度脉冲神经网络的第二个严苛条件如下：

条件 2: 脉冲路径上的权值链的权值需要保证足够的强度来传输梯度。

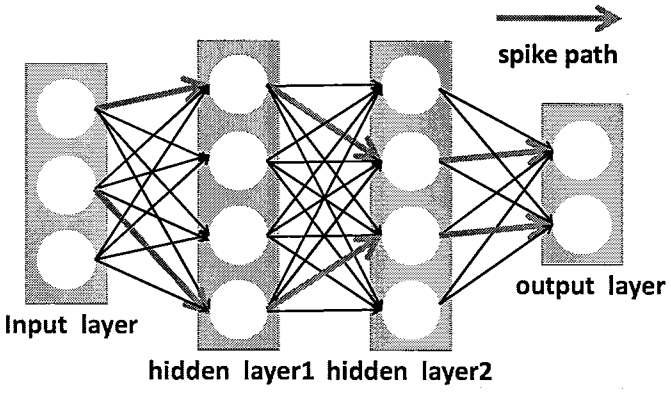

图 5.2 脉冲路径(spike path)示意图。输入某个数据时，脉冲路径指的是脉冲从脉冲神经网络的输入层传递到输出层的路径。图中的蓝线为脉冲路径。

Figure 5.2 The diagram of spike paths. For one input data, the spike path is the path where spikes are transmitted through the input layer to the output layer.

公式 (5.10)右边的第二项表示梯度在时间域上的传递。当第l层的第i个神经元在第t个时间步往外发送脉冲的时候，其膜电势$u_i^{t+1,l}$为零，此时第二项对变量$\delta_i^{t,l}$的贡献为零。从公式 (5.15)中可以推出，偏导数$\frac{\partial g}{\partial u_i^{t,l}}$大于零当且仅当$u_i^{t,l} > V_{th} - \Delta/2$。所以，权值变化只有在$(V_{th} - \Delta/2, V_{th})$范围内积累，同时参数$\Delta$控制着权值变化在多少个时间步内累积。因此，可以总结出训练深度脉冲神经网络的第三个严苛条件如下：

条件 3: 参数Δ只有设置的适合的时候，反向传播学习算法才可以获取脉冲神经网络之间传输的时序信息。

上面这个条件仅为STBP算法的要求，但是那类基于反向传播算法的脉冲神经网络的训练算法也需要设置类似的超参数。所以，第三个条件可以看作是基于反向传播算法的脉冲神经网络训练算法中的参数设置条件，本章中将参数Δ的设置看作一个例子。

当训练的脉冲神经网络的层数逐渐变深的时候，满足以上三个条件的难度变得十分困难。如果像STBP算法一样使用均匀分布来初始化脉冲神经网络中的权值参数，在网络的前几层脉冲路径就会逐渐消失。举例说明，从表 5.1中左边几列看出，使用一个类似VGG11结构的脉冲神经网络进行实验，该网络的第7-11层的发送的脉冲总数目为零。对于第二个条件，假设有一条脉冲路径，同时脉冲路径上的突触权值为 $w_1, w_2, ...w_{11}$。那么这些突触权值均大于0.1($|w_i| > 0.1$)的概率为$0.9^{11} \approx 31\%$。这就意味着，在训练过程中，只有31%的脉冲路径可以学习到权值的变化。实际上，真实的情况更加严重。举例说明，如果满足$w_{10} = 0.12$以及$w_{11} = 0.11$，由于$w_{10}w_{11} \approx 0.01$，该权值链很弱。当前两个条件满足的时候，第三个条件也更加容易满足。因此，本章致力于发现一些权值参数来满足前两个条件。

5.3.4 卷积神经网络与脉冲神经网络的联系

卷积神经网络与脉冲神经网络之间的联系可以提供一些满足上一小节提出的前两个严苛条件的线索。为了阐述该联系，首先假设两种网络之间的权值参数完全相同。常见的脉冲神经网络模型转换算法是本小节推导的一个特殊情况，即参数τ和参数V_{th}满足$\tau = V_{th}$[99]。假设卷积神经网络第l层的第i个神经元的输出a_i^l满足如下公式：

$$a_i^l = f(\sum_{j=1}^{M^{l-1}} W_{ij}^l a_j^{l-1} + b_i^l) \tag{5.19}$$

其中，W_{ij}^l表示第l层的第i个卷积神经元的第j个突触的权值，b_i^l是其对应的偏置。如果采用ReLU函数作为激活函数，那么$f(x) = max(0, x)$。

本章使用脉冲神经元的脉冲发送频率来逼近卷积神经网络中神经元的模拟输出。假设公式 (5.3)和公式 (5.19)使用同一组权值参数，即$w_{ij}^l = W_{ij}^l$。定义第l层第i个脉冲神经元的脉冲发送频率r_i^l如下式所示：

$$r_i^l = \frac{\sum_{t=1}^{T} o_i^{t,l}}{T} \tag{5.20}$$

其中，T为模拟脉冲神经网络前馈过程的时间步数，$o_i^{t,l}$表示第l层的第i个神经元在第t个时间步的输出脉冲信号。假设脉冲神经元的脉冲发放频率r_i^l趋近于卷积神经网络的模拟输出a_i^l，即$r_i^l \to a_i^l$。那么由公式 (5.20)可知，$E(o_i^{l-1}) \to a_i^{l-1}$。从而，可以假设$E(x_i^{t,l}) = a_i^l$。给定$a_i^l$，结合公式 (5.2)和公式 (5.4)，可以推出以下公式：

$$r_i^l = \begin{cases} 1, & if \quad a_i^l \in [V_{th}, \infty) \\ \frac{1}{2}, & if \quad a_i^l \in [\frac{V_{th}}{1+\tau}, V_{th}) \\ \frac{1}{n}, & if \quad a_i^l \in [\frac{V_{th}}{1+\tau+\ldots+\tau^{n-1}}, \frac{V_{th}}{1+\tau+\ldots+\tau^{n-2}}), \quad n > 2 \\ 0, & otherwise \end{cases} \tag{5.21}$$

根据上式，固定$V_{th} = 1$，作出$a_i - r_i$关系图，如下图 5.3所示。由图 5.3中可以看出，通过设置合适的参数τ和参数V_{th}可以满足上一小节的第一个条件。举例说明，固定参数$\tau = 0.8$，设置参数V_{th}可以使得图 5.3中的左子图的蓝色曲线向左移动。然后，脉冲神经网络就可以和卷积神经网络有相似的输出。同时，就像在卷积神经网络中有效信息从网络的输入层沿着某条路径传输到网络的输出层，脉冲路径也在脉冲神经网络对应于卷积神经网络的相似位置被建立起来。更进一步地，从图 5.3中可以看出，脉冲发放频率r_i等于零的区间很小(对应卷积神经网络的输出a_i大于0的区域，因为$a_i = 0$的区间也可以保证$r_i = 0$，不产生误差)。因此，使用卷积神经网络的参数初始化脉冲神经网络，脉冲神经网络中的沉默神经元的数目很小。

利用卷积神经网络与脉冲神经网络之间的联系，上一小节的第二个条件同样可以被满足。从卷积神经网络的反向传播算法的推导中可知，在卷积神经网络的反向传播过程中同样存在着权值链强度的问题。在卷积神经网络的训练算法中，现在已有一些有效的解决办法，比如ReLU函数、BatchNormalization技术[185]、ResNet[186]等等。这些技术可以对抗权值链中突触权值较小的问题。因此，使用卷积神经网络突触权值参数初始化的方法可以使脉冲神经网络获得足够的权值链用于突触权值学习过程。更进一步地，这种方式初始化的脉冲神经网络已经初步具备在空间域上处理信息的能力，后续的学习阶段可以将任务聚焦在提取脉冲时间序列中的时序信息。任务变得更加简单，从而公式 (5.15)中的参数Δ可设置的范围也更广。

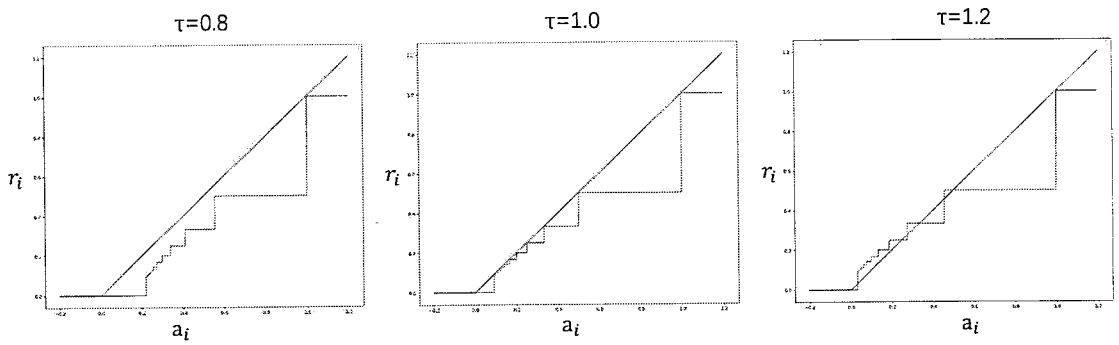

图 5.3 卷积神经网络的输出a_i与脉冲神经网络神经元脉冲发送频率r_i的$a_i - r_i$关系图。固定膜电势阈值$V_{th} = 1.0$，参数$\tau = 0.8$，$\tau = 1.0$以及$\tau = 1.2$分别对应左中右三个子图。其中红线为ReLU函数曲线，蓝线为$a_i - r_i$关系曲线。该图解释了公式 (5.21)。

Figure 5.3 The relation between the CNN neuron output a_i and the firing rate r_i of the spike neuron. Suppose the membrane potential threshold $V_{th} = 1$ and the red line indicates the ReLU function. The parameters τ of the three subfigure are 0.8, 1.0 and 1.2. This figure explains the equation (5.21).

5.4 基于反向传播的模型转换算法优化

本节提出三种不同的技术来克服训练深度脉冲神经网络中的困难。首先，提出参数初始化算法使脉冲神经网络在训练开始的时候可以获得更好的训练起点。然后，提出误差最小化算法来指导脉冲神经网络中的超参数选取的问题。最后，修改了STBP算法中的损失函数，避免了基于频率编码的损失函数和STBP算法算出的带时序信息的梯度之间不匹配的问题。

5.4.1 参数初始化算法

通过第 5.3.4小节的分析可知，训练后获得的卷积神经网络的权值参数可以帮助脉冲神经网络在空间域上获得提取特征的能力。换句话说，这些权值参数可以帮助脉冲神经网络满足第 5.3.3小节的前两个条件。另外，从表 5.1中左边几列可以看出，如果使用原始的STBP算法的参数初始化算法来设置权值参数，脉冲神经网络的高层中的脉冲总数目会逐渐下降到零。本节可以从表 5.1推测出该现象的原因。假设模拟脉冲神经网络的总时间步数$T = 10$，那么脉冲神经元可以往外发送脉冲当且仅当对应的卷积神经网络中的神经元的输出$a_i > 0.1$(为了简化分析过程，假设参数$\tau = 1$以及$V_{th} = 1$)。如果卷积神经网络中的神经元输出$a_i < 0.1$，对应的脉冲神经元的脉冲发送频率$r_i < 0.1$。这就意味着该脉冲神经元在10个时间步内很难往外发送脉冲信号。换句话说，第 5.3.3小节中的条件1等价于在脉冲路径上对应的卷积神经网络的神经元输出a_i大于零。因此，使用常

规的权值参数初始化的算法很难满足第 5.3.3小节中的条件1。

在下图 5.4中，解释了原始STBP算法中的初始化算法以及本节提出的参数初始化算法的区别。图中的红点可以看作是使用本小节提出的参数初始化算法初始化STBP算法的训练起点，蓝点表示原始STBP算法使用的初始化算法获得的训练起点。从图 5.4中看出，从蓝点的分布可以看出，使用常规的初始化算法初始化脉冲神经网络的权值参数，STBP算法更难收敛。有时候甚至使STBP算法难以收敛，比如脉冲神经网络的层数过深的时候。与之相对地，红点的分布意味着本小节提出的参数初始化可以加速STBP算法的收敛，同时使用STBP算法训练深度脉冲神经网络成为可能。

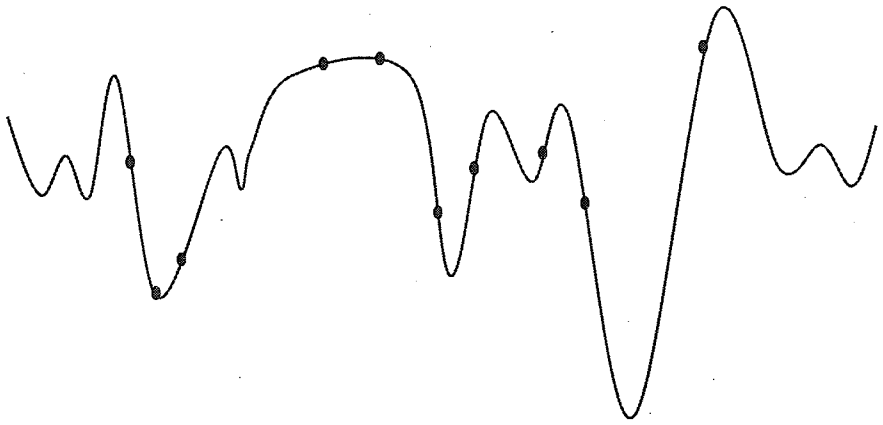

图 5.4 不同的参数初始化算法的影响的示意图。蓝点表示原始STBP算法的参数初始化算法获得的训练起点，红点表示本章中提出的参数初始化算法获得的训练起点。

Figure 5.4 The influence of different weight initialization methods. The blue points denote the starting points with the original STBP algorithm and the red points denote the starting points with the proposed weight initialization algorithm.

参数初始化算法的具体过程如下算法 5所示。首先，训练深度卷积神经网络获得权值参数矩阵，或者从因特网上下载权值参数矩阵。然后，使用脉冲神经网络模型转换算法中类似的方法将卷积神经网络转换为脉冲神经网络，用离散化的LIF脉冲神经元替换模型转换算法中的脉冲神经元。最后，使用第一步中获得的权值参数矩阵初始化脉冲神经网络，使用STBP算法训练该脉冲神经网络。从表 5.1中可以看出，本小节提出的参数初始化算法可以有效地避免深度脉冲神经网络中网络高层中的沉默神经元的问题。该表的最右边的几列表明，参数初始化算法可以找到脉冲神经网络中的脉冲路径。

算法 5 参数初始化算法(Weight Initialization Algorithm)

输入：卷积神经网络训练后获得的参数矩阵W；
输入：输入数据D_i；
输出：STBP算法训练后获得的参数矩阵$\hat{W}$；
输出：输出脉冲序列O_i；
procedure WEIGHTINITIALIZATION(W, D_i)
    使用离散化的LIF神经元模型替换转换学习算法中的脉冲神经元；
    构建和训练好的卷积神经网络的网络结构一样的脉冲神经网络；
    使用权值参数矩阵W初始化带训练脉冲神经网络；
    输入数据D_i，使用STBP算法训练脉冲神经网络；
    return $\hat{W}, O_i$
end procedure

表 5.1 脉冲神经网络中各层神经元的脉冲的总数目。使用的网络结构为表 5.2中的NetworkD。其中，STBP表示使用原始的STBP算法中的初始化方式，Weight Initialization表示使用本章的参数初始化算法进行网络初始化。

Table 5.1 Spike amount of each layer in the NetworkD before training between the original STBP algorithm and the STBP algorithm with weight initialization.

	Algorithm	Original STBP		Weight Initialization	
Layer	Neuron Amount	Spikes	Precent	Spikes	Precent
1	65536	16639.32	25.39%	8917.36	13.61%
2	32768	4848.61	14.80%	2156.82	7.68%
3	16384	1721.53	10.51%	1070.12	6.21%
4	16384	525.88	3.21%	640.47	3.91%
5	8192	202.64	2.47%	227.15	2.77%
6	8192	6.06	0.07%	161.84	1.98%
7	2048	0.00	0.00%	16.65	0.81%
8	2048	0.00	0.00%	49.50	2.42%
9	1024	0.00	0.00%	43.41	4.24%
10	1024	0.00	0.00%	104.72	10.23%
11	10	0.00	0.00%	0.68	6.85%

5.4.2 误差最小化算法

根据第 5.3.4小节，定义卷积神经网络与脉冲神经网络中第l层的每个神经元输出的平均误差e_l如下

$$e_l = \frac{\sum_{i=1}^{N} |a_i^l - r_i^l|}{N} \tag{5.22}$$

其中，N表示第l层神经元的总数目。本节可以通过设置不同的参数τ和参数V_{th}来最小化上式中的平均误差e_l。假设脉冲神经网络中第l层的第i个神经元的脉冲发送频率r_i^l满足函数$g(\cdot)$如下

$$r_i^l = g(a_i^l, \tau, V_{th}). \tag{5.23}$$

函数$g(\cdot)$和公式 (5.21)中函数是等价的。然后，统计卷积神经网络中的第l层的第i个神经元的输出a_i^l以及其值为a_i^l的概率为$p(a_i^l)$。除了卷积神经网络中的输出层，卷积神经网络的其他各层的神经元输出分布满足长尾分布。这就意味着大部分卷积神经元的输出a_i^l接近于零。所以，脉冲神经网络的各层逼近的平均误差e_l可以被改写如下

$$e_l = \sum_i p(a_i^l)|a_i^l - g(a_i^l, \tau, V_{th})| \tag{5.24}$$

由于卷积神经网络的大部分层的输出满足长尾效应，为了减小平均误差e_l，函数$g(\cdot)$主要需要关注对于靠近零附近的a_i^l的值的逼近。因此，各层平均误差有相似的等高线图，如下图 5.5所示。本章绘制了表 5.2中NetworkC-E中各层的神经元平均误差e_l的示意图，并且使用NetowrkD中的第一层为例给出了图 5.5。从图 5.5中可以看出，神经元的平均误差e_l的最小值在参数$\tau = 1.0$以及参数$V_{th} = 0.85$附近。

在实验部分，本章选择参数$\tau = 1.0$以及参数$V_{th} = 0.85$。从第 5.3.4小节可知，脉冲神经元的脉冲发送频率r_i与卷积神经网络的输出a_i之间逼近误差最大的地方有两处。首先，假设膜电势的阈值$V_{th} = 1$，当卷积神经网络的输出$a_i > 1$时，脉冲神经元的脉冲发送频率$r_i = 1$。由于卷积神经网络各层输出的长尾分布特性，只有很少一部分的卷积神经网络的神经元输出满足$a_i > 1$。因此，这部分的逼近误差对最终的平均误差e_l影响不大。其次，由于卷积神经网络的各层输出的长尾分布特性，卷积神经网络的各层输出a_i靠近于零附近的部分的逼近对平均误差e_l的影响很大。假设固定脉冲神经网络模拟的总时间步数$T = 10$，然后脉冲神经网络中的每个神经元的脉冲发送频率r_i^l在集合$\{0.1, 0.2, ...1.0\}$中取值。假设参数$\tau = 1.0$，从公式 (5.21)中可知，当脉冲神经元的脉冲发送频率$r_i = 0.1$时，卷积神经网络的神经元输出a_i落在区间$[\frac{V_{th}}{10}, \frac{V_{th}}{9})$内。所以，可以通过减小膜电势阈

值V_{th}来减小靠近零附近的逼近误差。但是也不能将膜电势阈值V_{th}设置得过小，否则区间[0.1, 1.0]内的逼近误差会大幅增加。当然，本节也可以改变参数τ的大小。当参数τ靠近1附近，该超参数已经能够取得不错的效果，剩下的工作可以交给STBP算法去优化脉冲神经网络。

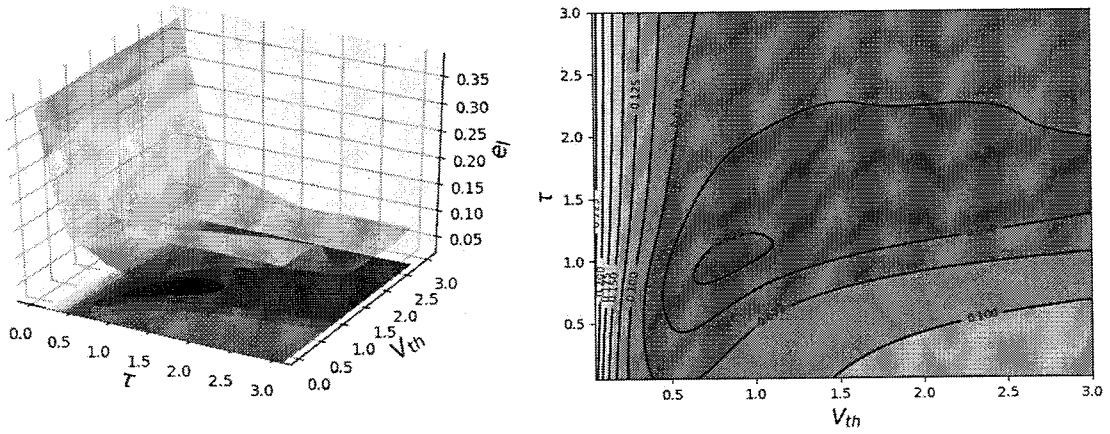

图 5.5 卷积神经网络与脉冲神经网络中神经元之间的平均逼近误差e_l的示意图。左子图为平均误差的曲面图，右子图为平均误差的等高线图。本图根据表 5.2中的NetworkD的第一层的平均误差情况绘制。

Figure 5.5 The average neuron error between CNN and SNN. The left subfigure is the surface of the average error and the right subfigure is the contour map. This figure is form the first layer from NetworkD in Table 5.2.

5.4.3 修改的损失函数

从前文可知，公式 (5.6)中定义的基于脉冲发送频率(rate code)的损失函数与STBP算法中脉冲神经网络各层中计算出的带时序信息的梯度不匹配。根据第5.3.3小节的分析可知，参数Δ可以控制脉冲神经网络各层间脉序列间的时序信息的获取。为了获得更多的时序信息，本节在损失函数中添加一个放缩参数α。举例说明，从公式 (5.26)中可知，当参数$\alpha = 1.05$时，如果输出层脉冲在第一个时间步到来，那么该脉冲对损失函数的贡献是1.05^9，当脉冲在第二个时间步到来，该脉冲对损失函数的贡献是1.05^8，以此类推。因此，到达时间越早的脉冲对脉冲神经网络最终的输出影响越大。

与此同时，由第 5.3.4小节的讨论，由于一些卷积神经网络的输出层神经元在输入某些数据的时候输出极小，那么其对应脉冲神经网络中的神经元不会发送任何脉冲。为了确定这些神经元的输出分类，本节将膜电势的变化添加为损失函数的第二项。

最终，定义基于交叉熵的损失函数如下

$$Loss = -\frac{1}{S}\sum_{s=1}^{S} log\frac{exp(y_s[class])}{\sum_j exp[x_s[j]]} \tag{5.25}$$

其中，$class$表示目标类别的索引，s表示第s个输入数据的索引，每次批处理训练的数据数目为S。上式中$x_s[j]$定义如下

$$x_s[j] = \frac{1}{T}\sum_{t=1}^{T}\{\alpha^{T-t}o_{s,j}^{t,L} + \beta u_{s,j}^{t,L}\}. \tag{5.26}$$

其中，参数α和参数β是放缩参数，j表示脉冲神经网络输入向量的第j个值，t表示第t个时间步。

5.5 实验结果分析

本节首先介绍了本章实验所用的基本设置，然后从网络的识别精度、训练迭代次数、误差最小化算法及修改的损失函数的影响、网络的收敛时间等4个方面比较本章提出的三种算法之间的效果，最后对实验结果进行了简要的小结。

5.5.1 实验设置

本小节主要介绍本章实验所需的相关设置，包括实验用的数据集、使用的脉冲神经网络的网络结构、脉冲神经网络的相关超参数以及实验的硬件环境。

5.5.1.1 实验数据集

本章选用脉冲神经网络中常用的MNIST数据集以及CIFAR10数据集来评估本章提出算法的性能。其中，MNIST数据集包含60,000张大小为28×28的灰度手写数字训练图片，对应的测试图片为10,000张。CIFAR10数据集比MNIST数据集稍复杂，包含50,000张训练图片以及10,000张测试图片，图片均为大小为32×32的自然物体的RGB彩色图片。

5.5.1.2 网络结构设置

本章中使用了如表 5.2中所示的5种网络结构来验证本章提出算法的有效性。该表中的NetworkA以及NetworkB与[86, 97, 99, 187, 188]中的常用的网络结构十分相似。该表中的NetworkC、NetworkD以及NetworkE用来评估本章提出的算法在非常深的脉冲神经网络上的效果。网络的层数逐渐增加是为了测试脉冲神经网络的深度变化对学习算法的影响。在MNIST数据集上，本章仅使用表中的NetworkA以及NetworkB进行相关试验，因为这个任务不需要训练深度脉冲神经网络即可获得很好的效果。

5.5.1.3 脉冲神经网络参数设置

本章实验中使用到的主要的脉冲神经网络的超参数如下表 5.3所示。

5.5.1.4 仿真硬件环境

本章的所有实验均在一个64-bit的计算机上进行，操作系统为ubuntu 16.04 LTS，内存大小为31.4GiB。处理器是Intel®Core™i7-7700CPU@3.60GHz×8，显卡为GeForce GTX 1080Ti/PCIe/SSE2。使用的深度学习框架为PyTorch，版本号为0.4.1。

表 5.2 网络结构设置(按列显示)。卷积层的参数按照conv(感受野尺寸)-(通道数目)表示。

Table 5.2 network configurations(shown in columns). The convolutional layer parameters are denoted as conv(receptive field size)-(number of channels).

<table>
<tr><th colspan="5">Network Configuration</th></tr>
<tr><td>NetworkA
4 weight layers</td><td>NetworkB
6 weight layers</td><td>NetworkC
8 weight layers</td><td>NetworkD
11 weight layers</td><td>NetworkE
19 weight layers</td></tr>
<tr><td>conv3-32</td><td>conv3-32
conv3-32</td><td>conv3-32
conv3-32</td><td>conv3-64</td><td>conv3-64
conv3-64</td></tr>
<tr><td colspan="5">max-pool</td></tr>
<tr><td>conv3-32</td><td>conv3-64
conv3-64</td><td>conv3-64
conv3-64</td><td>conv3-128</td><td>conv3-128
conv3-128</td></tr>
<tr><td colspan="5">max-pool</td></tr>
<tr><td></td><td></td><td>conv3-128
conv3-128</td><td>conv3-256
conv3-256</td><td>conv3-256
conv3-256
conv3-256
conv3-256</td></tr>
<tr><td colspan="2"></td><td colspan="3">max-pool</td></tr>
<tr><td colspan="3"></td><td>conv3-512
conv3-512</td><td>conv3-512
conv3-512
conv3-512
conv3-512</td></tr>
<tr><td colspan="3"></td><td colspan="2">max-pool</td></tr>
<tr><td colspan="3"></td><td>conv3-512
conv3-512</td><td>conv3-512
conv3-512
conv3-512
conv3-512</td></tr>
<tr><td colspan="3"></td><td colspan="2">max-pool</td></tr>
<tr><td colspan="3">FC-512</td><td colspan="2">FC-1024</td></tr>
<tr><td colspan="3">FC-10</td><td colspan="2">FC-1024</td></tr>
<tr><td colspan="3"></td><td colspan="2">FC-10</td></tr>
<tr><td colspan="5">softmax</td></tr>
</table>

5.5.2 网络的识别精度

从表 5.4中可以看出，不同的脉冲神经网络训练算法在MNIST数据集上取得

表 5.3 脉冲神经网络中的超参数设置。

Table 5.3 Parameters set in our experiments.

Network Parameter	Description	Value
T	Time window	10ms
V_{th}	Potential threshold	0.5mV
τ	Decline constant	0.6,1.0
dt	Simulation time step	1ms
a	Derivative approximation parameter	0.5mV

表 5.4 不同脉冲神经网络学习算法的识别精度对比表。

Table 5.4 Classification accuracy of different SNNs.

Dataset	Network-Type	Accuracy	Algorithm
MNIST	LM-SNN[68]	94.07%*	Unsupervised
	DCSNN(STDP)[189]	97.20%	Unsupervised
	SpikeCNN(Panda)[123]	99.05%	Unsupervised
	Spiking CNN(Lee)[85]	99.31%	Supervised
	SYLAYER[88]	99.36%	Supervised
	Spiking CNN (STBP)[86]	99.42%	Supervised
	Spiking ConvNet[99]	99.44%	CNN-SNN
	HM2-BP[89]	99.49%	Supervised
	ROT-FSI SNN[188]	99.58%	CNN-SNN
	Our paper (NetworkA)	99.44%#	Supervised
	Our paper (NetworkB)	**99.58%**#	Supervised
CIFAR10	Spiking CNN (STBP)[86]	50.70%	Supervised
	SpikeCNN(Panda)[123]	75.42%	Unsupervised
	Spiking ConvNet[97]	77.43%	CNN-SNN
	STBP+NeuNorm[87]	90.53%	Supervised
	BinaryConnect SNN[99]	90.85%	CNN-SNN
	ROT-FSI SNN[188]	92.89%	CNN-SNN
	Our Paper (NetworkA)	83.43%##	Supervised
	Our Paper (NetworkB)	89.11%##	Supervised
	Our Paper (NetworkC)	91.53%##	Supervised
	Our Paper (NetworkD)	89.97%##	Supervised
	Our Paper (NetworkE)	85.44%##	Supervised

* The results of related works are extracted from corresponding works.

The results only use the weight initialization algorithm in Section 5.4.

The results use both three algorithms in Section 5.4.

的识别精度差不多。本章使用参数初始化算法在表 5.2中NetworkB上取得了最好的识别精度99.58%。

在CIFAR10数据集上，脉冲神经网络模型转换算法ROT-FSI SNN[188]取得最好的识别精度92.89%，其识别精度在25、30、60个时间步内分别为82.18%、87.05%以及92.34%。全部使用本章提出的三种算法可以在表 5.2中的NetworkB上在10个时间步内取得89.11%的识别精度。其中，NetworkB和ROT-FSI SNN中的网络结构相似。这也就意味着，使用参数初始化算法开始STBP算法的训练过程，可以在保持脉冲神经网络的识别精度的条件下，大幅度地减小网络的运算操作数。本章提出的三种算法在表 5.2中NetworkC上的识别精度为91.53%，相较于STBP算法的改进算法STBP+NeuNorm[87]取得了1%识别精度的提升。Wu等 [87]提出一种辅助神经元来缓解沉默神经元问题，在该增强算法中使用的脉冲神经网络为CIFAR10Net网络 (128C3-256C3-AP2-512C3-AP2-1024C3-512C3-1024FC-512FC-Voting)。该网络结构CIFAR10Net与表 5.2中NetworkC网络结构类似，但是CIFAR10Net网络中的卷积通道数远多于NetworkC。一个原因是使用更多的卷积通道用于避免网络中存在过多的沉默神经元。因此，CIFAR10Net网络比NetworkC消耗更多的运算操作数。

5.5.3 训练迭代次数

本小节从两个方面来衡量参数初始化算法在脉冲神经网络的训练迭代次数(train epochs)上的性能。

首先，为了进行公平的比较，本小节对比了不同的脉冲神经网络训练算法在MNIST数据集上训练所需的迭代次数。基于反向传播的脉冲神经网络监督学习算法HM2-BP[89]在100次训练迭代时，取得了最好的识别精度98.93%。该算法中使用两种方式的反向传播算法，其训练过程比STBP算法复杂，消耗的运算操作数也更多。与此同时，本章提出的参数初始化算法使用5次卷积神经网络训练迭代以及15次脉冲神经网络训练迭代即可取得98.73%的识别精度。相较于训练过程中运算操作数的减少，0.2%的识别精度的减少可以被忽略不计。与STBP算法对比，本小节中将训练迭代次数从STBP算法的200次训练迭代，减少到5次卷积神经网络训练迭代以及15次脉冲神经网络训练迭代。更多地，如果脉冲神经网络总的运行时间步$T = 10$，1次卷积神经网络训练的迭代消耗的运算操作数仅相当于STBP算法中脉冲神经网络训练1次迭代运算操作数的1/10。这也就是说，卷积神经网络中的运算操作次数可以忽略，因此参数初始化算法在MNIST数据集上相对于原始的STBP算法节省了$200/15 \approx 13.3\times$的运算操作

数。

表 5.5 在MNIST数据集上，不同的脉冲神经网络训练算法的训练迭代次数比较表。为了公平比较，本文选择了[89]中提供的数据。

Table 5.5 Comparison of train epochs of different SNN models on MNIST dataset. For fair comparison, the data comes from [89].

Model	Hidden layers	Accuracy	Best	Epochs
Spiking MLP(CNN-SNN)[121]	500-500	94.09%	94.09%	50
Spiking MLP(CNN-SNN)[124]	500-200	98.37%	98.37%	160
Spiking MLP(CNN-SNN)[118]	1200-1200	98.64%	98.64%	50
Spiking MLP(Lee)[85]	800	98.71%	98.71%	200
Spiking MLP(STBP)[86]	800	98.89%	98.89%	200
Spiking MLP(HM2-BP)[89]	800	98.84 ± 0.02%	98.93%	100
Spiking MLP(this work)	800	98.73%	98.73%	5(CNN)+15(SNN)*

* The first item indicates the CNN train epochs and the second item indicates the SNN train epochs. The test accuracy of the CNN is 98.26% at the 5th epoch.

其次，本小节比较了原始的STBP算法和参数初始化算法在训练深度脉冲神经网络时迭代次数上的差距。在本小节中，将两种算法在CIFAR10数据集上的测试精度收敛随算法训练迭代次数的变化关系示意图绘制在图 5.6中。从图中可以看出，参数初始化算法在不同的深度的网络结构上，均可以在10次训练迭代内收敛到一个可接受的识别精度上。相较于原始STBP算法，参数初始化算法大大减少了训练的迭代次数。同时，在NetworkB和NetworkC上，STBP算法需要80次训练迭代识别精度才会收敛。由于参数初始化算法解决了STBP算法中的沉默神经元问题(dead neuron problem)，参数初始化算法所需的训练迭代次数远少于原始STBP算法。更进一步地，在NetworkD上，STBP算法只能取得20%的识别精度，同时在NetworkE上，STBP算法会失效。这也就是说，随着网络规模的不断加大，基于反向传播算法的脉冲神经网络中的前两个苛刻条件越来越难以满足。相对地，参数初始化算法可以帮助脉冲神经网络找到足够的脉冲路径，同时路径上的权值链的强度足够。因此，参数初始化算法在图 5.6中对应的曲线均能在10次训练迭代次数之内收敛。究其原因，通过分析卷积神经网络与脉冲神经网络之间的联系可知，来自卷积神经网络中的权值参数可以帮助脉冲神经网络在空间域上获得提取有效信息的能力。当脉冲神经网络获得在空间域上处理信息的能力时，STBP算法只需要重点关注学习在时间域上处理信息的能力。因此，STBP算法需要学习的任务难度降低了。如图 5.6所示，参数初始化算法在NetworkE上依然有效，同时训练算法训练收敛速度很快。因此，参数初始化算法可以扩大STBP算法的应用范围，同时加速STBP算法的训练收敛速度。

总的来说，参数初始化算法可以减小训练迭代次数，并且在极深度脉冲神经网络训练中取得显著的效果。

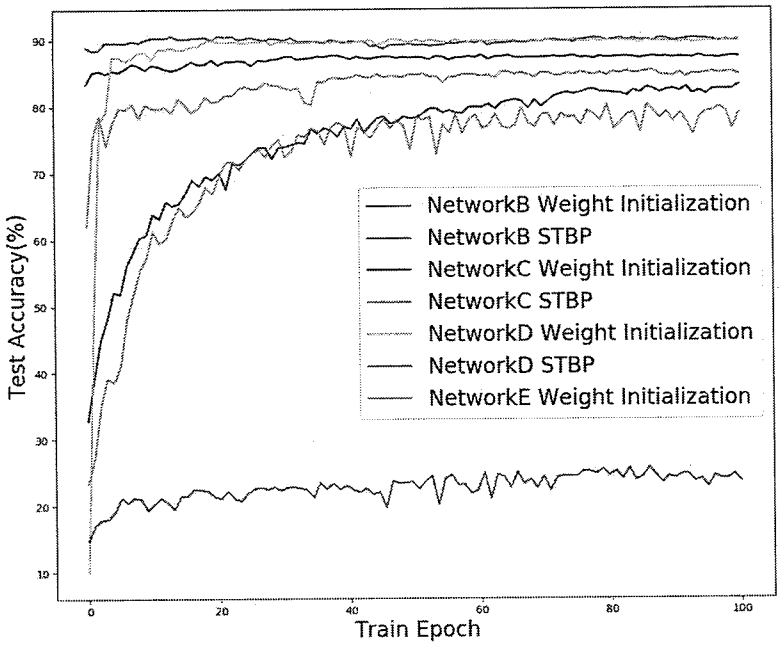

图 5.6 在CIFAR10数据集上，STBP算法与参数初始化算法的训练迭代次数示意图。NetworkB-E来自表 5.2。STBP表示[86]中提出的脉冲神经网络监督学习算法；Weight Initialization表示使用参数初始化算法初始化脉冲神经网络的突触权值参数，然后使用STBP算法训练脉冲神经网络。图中所有的脉冲神经网络运行的总的时间步数$T = 10$。STBP算法在训练NetworkE时，该算法会完全失效，因此没有绘制在图中。

Figure 5.6 Training acceleration on CIFAR10 dataset. NetworkB-E are from Table 5.2. STBP means the training algorithm from [86] and Weight Initialization means the STBP algorithm with the weight initialization. The total time steps T for all these SNNs are 10 time steps. The line of NetworkE STBP is not drew for that the STBP algorithm fails in this situation.

5.5.4 误差最小化算法及修改的损失函数的影响

本小节使用表 5.2中NetworkC以及CIFAR10数据集评估本章提出的误差最小化算法以及修改的损失函数的影响，结果如表 5.6中所示。所有的实验均使用参数初始化算法对脉冲神经网络的突触权值参数进行初始化。在表 5.6中，Original行表示使用常规的网络训练参数($\alpha = 1.0$, $\tau = 0.6$, $V_{th} = 0.5$)，Error Minimization行代表使用本章提出的误差最小化算法以及修改的损失函数(对应的参数为$\alpha = 1.05$, $\tau = 1.0$, $V_{th} = 0.85$)，Reference行代表实验的对照参考组(对应的参数为$\alpha = 1.05$, $\tau = 0.6$, $V_{th} = 0.5$)。

如表5.6所示，对于Original行，识别精度随参数β增加而增大。这也就是说，使用参数β可以缓解转换脉冲神经网络中输出层不产生脉冲的问题。然而，训练迭代次数也随着参数β的增大而增加，识别精度在参数β在区间[0.4, 1.0]内变化不大。

从表5.6中可知，参数α也可以影响识别精度。如果不使用修改的损失函数中的参数β(即$\beta = 0.0$)，在Original行、Reference行以及Error Minimization行上识别精度分别为90.04%、88.01%、88.37%。另外，选择一个合适的参数β(比如$\beta = 0.4$)，Reference行的识别精度相较于Original行($\beta = 0.0$)有1.28%的精度提升。

当参数$\beta = 0.4$时，同时使用参数初始化算法、误差最小化算法以及修改的损失函数时，Error Minimization行可以取得91.53%的识别精度(对应于表5.4中的NetworkC的识别精度)。相对于仅使用参数初始化的STBP算法，Error Minimization行可以取得1.49%的识别精度提升。通过同时使用本章提出的三种算法时，相较于仅使用参数初始化算法，训练迭代次数也从87次训练迭代次数下降到63次训练迭代次数。因此，同时使用误差最小化算法以及修改的损失函数，训练深度脉冲神经网络可以获得更好的识别精度以及更少的训练迭代次数。

表5.6 误差最小化算法以及修改的损失函数的影响。表中所有的实验均基于参数初始化算法，同时在CIFAR10数据集上训练。第一行中的数字代表公式(5.26)中的参数β。

Table 5.6 Effects of error minimization and loss function. All these experiments are based on the weight initialization algorithm and the CIFAR10 dataset. The numbers of the first row indicates the parameter β in (5.26).

Algorithm	β	0.0#	0.2	0.4	0.6	0.8	1.0
Original($\alpha = 1.0, \tau = 0.6, V_{th} = 0.5$)	Test Accuracy	90.04%	90.73%	91.19%	91.20%	91.12%	91.17%
	Train Epochs	87	173	138	159	212	274
Reference($\alpha = 1.05, \tau = 0.6, V_{th} = 0.5$)	Test Accuracy	88.01%	91.28%	91.32%	90.86%	91.22%	86.38%
	Train Epochs	96	84	72	163	151	277
Error Minimization($\alpha = 1.05, \tau = 1.0, V_{th} = 0.85$)	Test Accuracy	88.37%	90.69%	91.53%	91.16%	90.83%	86.45%
	Train Epochs	102	93	63	193	140	289

In this column, the parameter $\beta = 0.0$ in (5.26). It means that the temporal loss is moved from these training epochs.

5.5.5 网络的收敛时间

从前面的研究 [97, 99, 118, 124, 187, 188]中可知，脉冲神经网络模型转换算法可以在深度脉冲网络的识别精度方面取得显著的效果。这类算法的主要问题是转换脉冲神经网络的分类收敛时间太长。在文献[125]中，如果转换脉冲神经网络的分类收敛时间下降5.4×时，使用转换神经网络实现的硬件比使用

卷积神经网络实现相似网络规模的硬件消耗的能量更少。在表 5.7中，本小节比较了脉冲神经网络模型转换算法(CNN-SNN Algorithm)[97]、STBP算法(STBP Algorithm)[86]以及使用参数初始化算法初始化网络权值参数的STBP算法(Weight Initialization)的分类收敛时间和识别精度。

从表 5.7中可知，在MNIST数据集上，在4个时间步以内，参数初始化算法、STBP算法以及脉冲神经网络转换学习算法的识别精度为99.20%、99.18%以及75.38%。同时，转换学习算法获得的脉冲神经网络在100个时间步内的识别精度为99.02%。这就是说，在MNIST数据集上，使用参数初始化算法的STBP算法可以取得和转换脉冲神经网络相似的识别精度的同时，还可以对脉冲神经网络的分类收敛时间进行25×的加速。

在CIFAR10数据集上，脉冲神经网络模型转换算法的网络分类收敛时间最长，是其他两种算法网络分类收敛时间的25×。使用参数初始化算法的STBP算法的识别精度在相同的网络分类收敛时间下高于原始的STBP算法。比如，在4个时间步的网络分类收敛时间内，表 5.2中NetworkB可以取得3.46%的识别精度的提升。更进一步地，从图 5.6中可以看出，如果待训练的脉冲神经网络的网络规模不断增大，使用参数初始化算法的STBP算法相对于原始的STBP算法具有更好的鲁棒性。也就是说，使用参数初始化算法的STBP算法可以取得更好的性能。

综上所述，本章提出的参数初始化算法可以有效地加速脉冲神经网络的模型转换算法。

5.5.6 实验结果小结

本章通过4个方面比较分析了所提出的三种算法的效果。在转换网络的识别精度方面，本章算法在MNIST数据集以及CIFAR10数据集上分别取得99.58%以及91.53%的识别精度。在训练的迭代次数方面，在MNIST数据集上，相较于本章之前的典型算法，本章的参数初始化算法训练迭代次数最少，同时比原始STBP算法取得了13倍的训练加速。在CIFAR10数据集上，参数初始化算法在6-19层脉冲神经网络的训练中，均可在10次训练迭代内收敛到不错的识别精度。在网络的收敛时间方面，本章的参数初始化算法对分类收敛时间取得了25倍的加速。同时使用本章提出的三种算法比仅使用参数初始化算法，可以将训练迭代次数由87次减小到63次。

表 5.7 脉冲神经网络的分类收敛时间以及识别精度比较表。

Table 5.7 Convergence time and test accuracy.

Dataset	Network-Type	Algorithms	Test Accuracy				
Time-Step			4	6	8	10	100
MNIST	NetworkA(99.27%)	CNN-SNN Algorithm*	75.38%	90.79%	96.36%	98.21%	99.02%
		STBP Algorithm**	99.18%	99.20%	99.21%	99.24%	
		Weight Initialization	99.20%	99.18%	99.20%	99.34%	
	NetworkB(99.56%)	CNN-SNN	10.45%	71.22%	67.50%	90.32%	97.27%
		STBP Algorithm	99.25%	99.30%	99.32%	99.40%	
		Weight Initialization	99.44%	99.50%	99.51%	99.58%	
CIFAR10	NetworkA(82.32%)	CNN-SNN Algorithm	13.87%	39.81%	41.77%	52.93%	70.32%
		STBP Algorithm	76.32%	77.61%	78.01%	78.13%	
		Weight Initialization	77.33%	78.21%	78.90%	79.10%	
	NetworkB(90.81%)	CNN-SNN Algorithm	10.45%	12.11%	15.52%	17.43%	77.27%
		STBP Algorithm	82.74%	83.66%	84.36%	85.48%	
		Weight Initialization	86.20%	87.00%	87.58%	87.64%	

* This CNN-SNN algorithm is from [97], it is suitable for small networks.

** This STBP algorithm is from [86] and we realize it through PyTorch.

5.6 本章小结

本章针对脉冲神经网络模型转换算法中转换脉冲神经网络的网络收敛时间过长的问题，提出了基于反向传播算法的极低延迟的深度脉冲神经网络转换学习算法。首先，本章分析了脉冲神经网络反向传播算法的现有研究，总结出了该算法训练深度脉冲神经网络时需要满足的三个严苛的条件。接着，通过分析脉冲神经元与卷积神经网络神经元之间的联系，发现卷积神经网络中的权值参数可以帮助脉冲神经网络满足上述的三个严苛条件。基于以上原理，本章提出一种参数初始化算法。然后，本章提出了一种误差最小化算法以及修改的损失函数，进一步地优化算法以获得性能更好的脉冲神经网络。最后，实验表明，本章算法可以获得极低延迟的深度脉冲神经网络。

第6章 总结

在计算神经科学中，脉冲神经网络是类脑计算的重要研究方向之一。相对于深度卷积神经网络，脉冲神经网络在生物可解释性、低功耗以及脑机交互应用等方面存在着巨大前景。深度脉冲神经网络学习算法的缺失是阻碍脉冲神经网络在类脑计算中广泛应用的主要原因。因此，研究深度脉冲神经网络学习算法，从而提升脉冲神经网络在相关人工智能任务中的性能，具有很强的理论及现实意义。

本文通过总结分析各种典型的脉冲神经网络学习算法，发现脉冲神经网络模型转换算法具有实现深度脉冲神经网络的潜力，从而有效缩减脉冲神经网络与深度卷积神经网络之间的性能差距。因此，本文主要研究脉冲神经网络模型转换算法中存在的相关问题，以获得高性能的深度转换脉冲神经网络。本文取得的主要成果如下：

(1)针对转换脉冲神经网络中脉冲神经元无法表示卷积神经网络中大于1的输出值的问题，提出了一种多强度深度脉冲神经网络模型转换算法。首先，通过放松脉冲神经元的输出强度的限制，提出了一种多强度脉冲神经元。然后，推导了转换模型精度等价性理论，同时在推导过程中总结了多强度脉冲神经网络的相关优势。接着，利用多强度脉冲神经元为基本组成部件，成功搭建了一个19层的深度转换脉冲神经网络。为了减小该深度脉冲神经网络中存在大量冗余计算，针对脉冲神经网络在时间域上的计算冗余特性，提出了三种不同的动态剪枝算法。最终，实验表明，这三种动态剪枝算法可以移除多强度脉冲神经网络中94%的沉默神经元以及89%的弱突触，减少网络中85%的冗余计算。

(2)为了降低多强度脉冲神经元在脉冲神经芯片中直接部署的难度，提出了一种低延迟深度脉冲神经网络模型转换算法，在使用常见的脉冲神经元的同时解决了模型转换算法中的脉冲饱和问题。首先，提出了一种限制网络输出预训练算法，可以动态地进行参数规范化过程。具体来说，通过在卷积神经网络的训练过程中增加限制网络输出的操作，从而保证待转换的卷积神经网络中神经元输出值被限制到固定的值域。这也就相当于在卷积神经网络的模型训练过程中动态地完成了之前模型转换算法中的参数规范化过程，从而可以在消除脉冲饱和问题的同时不带来参数规范化过程中的低效脉冲问题。然后，提出了一种错误脉冲抑制算法。具体来说，先观察到转换脉冲神经网络中存在的错误脉冲

问题，再将抑制错误脉冲问题抽象化为一个线性最优化问题，基于以上抽象提出了一种错误脉冲抑制机制。接着，提出了一种时序最大值池化算法，可以无损地将卷积神经网络中的最大值池化操作移植到转换脉冲神经网络中。最终在保持模型转换算法的识别精度的条件下，大大地缩减了转换脉冲神经网络的网络收敛时间。

(3)为了进一步缩减转换脉冲神经网络的网络收敛时间，提出一种基于反向传播的极低延迟深度脉冲神经网络转换学习算法。首先，总结了各种脉冲神经网络的反向传播算法的优缺点，选定了一种与转换学习最匹配的反向传播算法。然后，归纳总结了使用反向传播算法训练深度脉冲神经网络需要满足的三个严苛条件，阐明了模型转换参数帮助反向传播算法满足上述的三个严苛条件的原因，提出参数初始化算法获得了极低延迟的深度脉冲神经网络。接着，提出了误差最小化算法以及修改的损失函数进一步地优化了参数初始化算法的性能。最后，实验表明，该算法最多可以对转换脉冲神经网络的网络收敛时间进行25×的加速，同时可以对反向传播算法取得至少13×的训练加速。

本文在深入分析脉冲神经网络模型转换算法的基础上，提出了三种改进的深度脉冲神经网络转换学习算法。这三个算法的基本目标是获得高性能的深度脉冲神经网络，同期的脉冲神经网络模型转换算法中的主要问题包括脉冲饱和问题以及低效脉冲问题。随着脉冲神经网络的层数的增加，脉冲饱和问题会使得转换脉冲神经网络的识别精度下降。本论文中的算法(1)提出一种特殊的多强度脉冲神经元模型解决了脉冲饱和问题，获得了一个深度的多强度脉冲神经网络。但是多强度脉冲神经元运算较常规的脉冲神经元更加复杂，算法(2)使用常规的脉冲神经元模型，提出一种限制输出预训练算法在解决脉冲饱和问题的同时减轻了低效脉冲问题的影响。经过评估，使用算法(2)中的转换脉冲神经网络实现类脑计算芯片，不能大幅减少神经网络硬件的功耗。因此，基于模型转换参数可以帮助脉冲神经网络在空间域上获得信息处理能力的理论，算法(3)研究了使用模型转换参数初始化脉冲神经网络以训练深度脉冲神经网络的学习算法的方法。该算法获得了一个极低延迟的深度脉冲神经网络，解决了脉冲饱和问题以及低效脉冲问题。

后续工作中，可以在高效的脉冲神经网络的学习算法、脉冲神经网络的可解释性以及脉冲神经网络的低功耗特性、脉冲神经网络在脑机交互等实时场景中的应用等四方面进行更深度的探索，从而获得高效高性能的脉冲神经网络的理论基础与参考设计。在研究高效的脉冲神经网络学习算法方面，从脉冲神经

网络的模型转换算法出发，主要包括两个方面探索：(1)进一步完善基于反向传播的脉冲神经网络转换学习算法，进一步加强超参数的设定法则，研究超参数在训练过程中的自学习算法，使该算法可以在更大的数据集上取得可以接受的识别精度；(2)探索将模型转换参数应用到突触权值可塑性的脉冲神经网络学习算法，从而获得一种低功耗、生物可解释性更强、更易在脉冲神经芯片中部署的深度脉冲神经网络，推动类脑计算的快速发展。在研究脉冲神经网络的可解释性方面，主要包括两个方面的探索：(1)研究脉冲的编码方式，探索基于时序编码的脉冲神经网络模型转换算法，从而获得脉冲传递信息的方式；(2)结合更多的生物神经科学方面的证据，研究更加类脑的脉冲神经网络的网络结构，搜寻人类智能是如何实现的相关关键证据。在脉冲神经网络的低功耗特性方面，研究深度脉冲神经网络的相关存储计算特性，分析深度脉冲神经网络算法在低功耗方面的优势，进行基于脉冲神经网络的硬件体系结构探索。在脉冲神经网络在脑机交互等实时场景中应用方面，这类应用场景与卷积神经网络的特性不太相符，适合脉冲神经网络对其应用进行相关探索。