学校代码：10284
分 类 号：TP183
密　　级：公开
U D C：004.8
学　　号：502022370015

南京大学

硕 士 学 位 论 文

论 文 题 目　具有生物合理性的脉冲神经网络局部非 BP 训练算法研究

作 者 姓 名　胡嘉骏
专 业 名 称　计算机科学与技术
研 究 方 向　脉冲神经网络
导 师 姓 名　申富饶　教授

2025 年 5 月 26 日

答辩委员会主席 武港山 教授

评　　阅　　人 张剡 高工

徐明华 教授

论文答辩日期 2024 年 5 月 16 日

研究生签名：胡嘉骏

导师签名：

Research on Biologically Plausible Local Non-Backpropagation Training Algorithms for Spiking Neural Networks

by

Hu Jiajun

Supervised by

Shen Furao

A dissertation submitted to

the graduate school of Nanjing University

in partial fulfilment of the requirements for the degree of

MASTER

in

Computer Science and Technology

School of Artificial Intelligence

Nanjing University

May 26, 2025

南京大学学位论文原创性声明

本人郑重声明，所提交的学位论文是本人在导师指导下独立进行科学研究工作所取得的成果。除本论文中已经注明引用的内容外，本论文不包含其他个人或集体已经发表或撰写过的研究成果，也不包含为获得南京大学或其他教育机构的学位证书而使用过的材料。对本文的研究做出重要贡献的个人和集体，均已在论文的致谢部分明确标明。本人郑重申明愿承担本声明的法律责任。

研究生签名：胡嘉骏

日期： 2025年5月26日

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：具有生物合理性的脉冲神经网络局部非 BP 训练算法研究

计算机科学与技术 专业 2022 级硕士生姓名：胡嘉骏

指导教师（姓名、职称）：申富饶 教授

摘 要

脉冲神经网络（Spiking Neural Networks）被誉为第三代神经网络，在生物合理性和计算能效方面展现出了潜力，正在得到越来越多的关注。然而由于脉冲激活函数的不可微性，其高效训练算法仍是亟待解决的挑战。主流的几种训练算法中，生物可解释性很高的 STDP 方法在深层网络和复杂任务中性能有限；基于替代梯度的反向传播方法继承了 BP 算法的生物不合理性以及梯度消失爆炸、近似误差累积等问题，计算资源开销也大；人工神经网络 ANN 转换 SNN 的方法需要大量的时间步来弥补转换误差，计算开销大，生物合理性低，且一般是离线转换，难以实现在线推理。

基于每种 SNN 训练算法都有明显缺点的情况，本文尝试在生物合理性、计算效率和模型精度等多个目标间权衡，研究多目标平衡的新型 SNN 训练算法。为此，本文提出了两个符合上述目标的 SNN 非反向传播（BP）式局部学习算法：

第一项工作是脉冲神经网络混合损失局部学习算法（SNN-HLL），通过逐层构建辅助分类器，无需全局梯度计算，仅依赖局部信息生成分层监督信号，并结合预测损失、相似度匹配损失等不同类型的损失函数，形成多角度互补优化目标。实验表明该方法在 MNIST、Fashion-MNIST 和 CIFAR-10 数据集上分别达到 99.35%、93.46% 和 91.44% 的准确率，达到甚至超过了基于全局误差传播的替代梯度下降法的准确率。在实现较高精确度的同时，本算法计算开销和内存更小，生物学合理性更高，可以实现并行化训练，易于部署到神经形态硬件上，兼顾了本文所提出的目标中的多个方面。

第二项工作是脉冲神经网络 HSIC 信息瓶颈赫布学习算法（SNN-HBH），利用希尔伯特-施密特独立性准则（Hilbert-Schmidt independence criterion）压缩输入冗余信息并保留任务相关特征，结合突触前后脉冲活动的局部信息与 HSIC 瓶

颈提供的全局信息约束，形成类似生物的局部突触和全局调制信号协同的学习规则，实验显示其性能接近常规的替代梯度下降法。和第一个工作类似，本工作突破传统 BP 框架，通过 HSIC 瓶颈导出的局部误差信号和分层优化形式，避免了脉冲函数不可微难以求梯度、反向传播的权重对称和更新锁定等问题，一定程度上解决了准确率、能耗和生物合理性之间的取舍矛盾，为 SNN 的训练算法提供了兼顾性能与仿生性的新解决方案。

关键词：脉冲神经网络；非 BP 式算法；分层局部学习；混合损失；HSIC 瓶颈

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Research on Biologically Plausible Local Non-Backpropagation Training Algorithms for Spiking Neural Networks

SPECIALIZATION: Computer Science and Technology

POSTGRADUATE: Hu Jiajun

MENTOR: Shen Furao

ABSTRACT

Spiking Neural Networks (SNNs), known as the third generation of neural networks, have shown great potential in terms of biological plausibility and computational efficiency, and are attracting increasing attention. However, due to the non-differentiability of the spiking activation function, efficient training algorithms for SNNs remain a significant challenge. Among mainstream training methods, Spike-Timing-Dependent Plasticity (STDP), which is highly biologically interpretable, performs poorly on deep networks and complex tasks. Surrogate gradient-based backpropagation inherits the biological implausibility of the BP algorithm and suffers from issues such as vanishing/exploding gradients and accumulated approximation errors, while also being computationally expensive. Conversion-based methods that map Artificial Neural Networks (ANNs) to SNNs require a large number of time steps to mitigate conversion errors, leading to high computational cost, low biological plausibility, and are usually conducted offline, making online inference difficult.

Given the apparent limitations of each SNN training approach, this work seeks to balance multiple objectives—biological plausibility, computational efficiency, and model accuracy—by exploring novel multi-objective training algorithms for SNNs. To this end, we propose two non-backpropagation (non-BP) local learning algorithms for SNNs that align with these goals:

The first is the Hybrid Loss-based Local Learning for SNNs (SNN-HLL). This method constructs auxiliary classifiers layer by layer, eliminating the need for global gradient computation by generating layer-wise supervisory signals based solely on local

information. It integrates multiple complementary loss functions, such as predictive loss and similarity-matching loss, to form a multi-perspective optimization objective. Experiments show that SNN-HLL achieves accuracies of 99.35%, 93.46%, and 91.44% on the MNIST, Fashion-MNIST, and CIFAR-10 datasets, respectively—matching or surpassing those achieved by surrogate gradient-based global error propagation. The method achieves high accuracy while requiring less computation and memory, offering higher biological plausibility, enabling parallel training, and making it easier to deploy on neuromorphic hardware—addressing multiple goals proposed in this work.

The second is the HSIC Bottleneck Hebbian Learning for SNNs (SNN-HBH). This method leverages the Hilbert-Schmidt Independence Criterion (HSIC) to compress redundant input information while preserving task-relevant features. By combining local spike activity between pre- and post-synaptic neurons with global constraints from the HSIC bottleneck, the method forms a biologically inspired learning rule that synergizes local synaptic updates with global modulation signals. Experimental results show that its performance approaches that of conventional surrogate gradient descent methods. Like the first method, this work breaks away from the traditional BP framework. By using locally derived error signals based on the HSIC bottleneck and a layer-wise optimization structure, it avoids issues such as non-differentiable spiking functions, weight symmetry, and update locking in backpropagation. It offers a new training paradigm for SNNs that reconciles accuracy, energy efficiency, and biological plausibility, providing a promising alternative that balances performance and bio-inspiration.

KEYWORDS: Spiking Neural Network; Non-BP Algorithm; Hierarchical Local Learning; Hybrid Loss; HSIC Bottleneck

目　　录

插图目录

表格目录

第一章　绪论

1.1　研究背景

人类的大脑代表着地球上最高水平的生物智慧，其在生物进化过程中逐渐形成了语言表达、逻辑推理、思考决策和复杂问题处理的能力，这些能力在生物界中是绝无仅有的，令人叹为观止。关于人脑智慧的奥秘，一直是受到广泛关注的研究重点，人们希望通过神经科学的最新发现，探寻并揭示智能的本质，进而实现能力接近甚至超越人类智慧的人工智能。类脑智能便是这样的一个研究领域，它是融合了生物学和计算机科学理念的交叉学科，架起了连接神经科学和人工智能的桥梁，极具研究价值。

而在类脑智能的研究范畴里，脉冲神经网络处于绝对的核心位置，它通过对生物神经元的连接结构、信息传递方式的模拟，试图达成以下两个目标：1. 利用计算机仿真技术，模拟和探究大脑皮层的神经元连接方式和脉冲信号处理机制，辅助神经科学的前沿研究；2. 开发神经形态脉冲计算架构，实现突破冯·诺依曼架构能效比的新型低功耗仿生智能，为人工智能提供新的解决方案。总之，脉冲神经网络有较高的研究意义，近几年来的研究热度也逐步攀升。

脉冲神经网络被誉为第三代神经网络，在此之前，神经网络已历经两代技术范式演进，取得了长足的发展，如图1-1所示。第一代神经网络最早可以追溯到二战时期，1943 年心理学家 Warren McCulloch 和数理逻辑学家 Walter Pitts 借鉴生物神经元的运算方式，首次提出了 MP 神经元模型[1]，开启了神经网络的研究时代。1949 年心理学家 Donald Olding Hebb 提出了著名的 Hebb 学习规则[2]。进一步，1957 年神经学家 Frank Rosenblatt 提出可以模拟人类感知能力的机器，称之为“感知机”[3]，并于 1960 年实现了能够识别一些英文字母的基于感知机的神经计算机 Mark 1，第一代神经网络就此诞生。尽管感知机能完成一些简单图形的分类任务，但是其单层的结构严重限制了在稍复杂问题上的能力，比如它无法解决异或（XOR）等线性不可分问题[4]。

第一代神经网络的局限性催生了神经网络技术革新的需求。1974 年，Paul Werbos 在博士论文[5]中首次把误差反向传播（BP）算法用于神经网络的训练。1986 年，David Rumelhart、Geoffrey Hinton 和 Ronald Williams 在 Nature 上发表著名文章[6]，系统清晰地提出了误差反向传播算法，使得多层感知机（MLP）的训练成为可能，标志着第二代神经网络的诞生。第二代神经网络在 BP 算法和激活函数的加持下，通过隐藏层堆叠成功构建了深层非线性映射，利用链式求导实现梯度稳定传播，实现了从单层到多层的突破，网络的表征能力大大提升，理论上可以逼近任何复杂的连续映射。BP 算法的大获成功，引发了人们的研究热潮，一大批新的神经网络模型被提出，如卷积神经网络（CNN）[7]、循环神经网络（RNN）[8]、深度信念网络（DBN）[9]等。2012 年深度神经网络（DNN）[10]兴起，第二代神经网络进入深度学习时代，VGG、ResNet 等一系列更深更复杂的模型结构出现，新型神经网络如变分自编码器（Variational Autoencoder,VAE）[11]、Transformer[12]也相继诞生，2020 年 GPT-3[13]已实现 1750 亿参数的规模训练，神经网络领域的发展和应用达到鼎盛。这一切其实离不开 GPU 等硬件提供的算力的大幅提升，强大的算力让深度神经网络的学习得以实现。

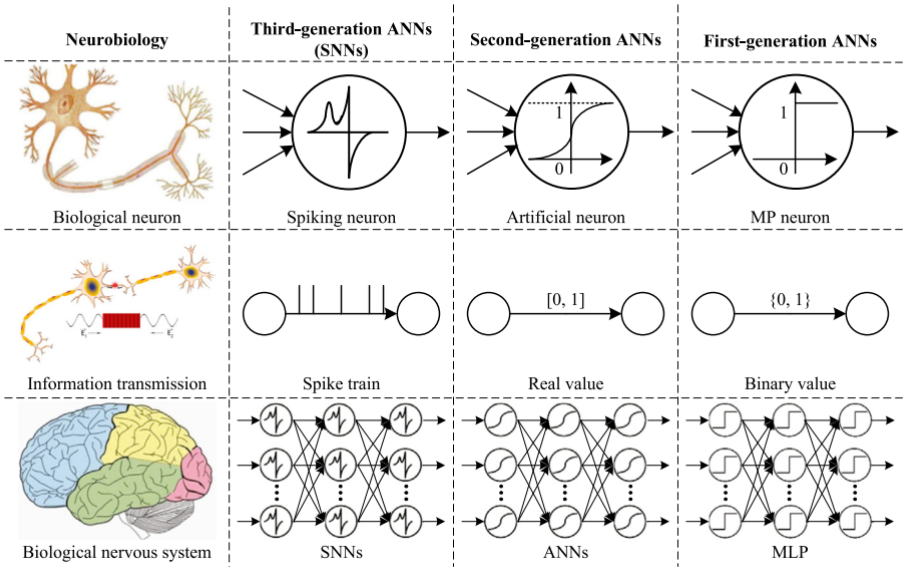

图 1-1　三代神经网络对照图[14]

然而，在深度学习空前繁荣的背后，第二代神经网络的缺陷也愈发显现：1. 资源开销大：深度神经网络的训练需要海量的数据和巨大的 GPU 计算资源，以 GPT3 为例，其单次训练消耗高达 1287 兆瓦时（MWh）电力，相当于 120 个美

国家庭一年的用电量。2. 生物合理性低：基于反向传播的全局更新需要整个网络的梯度的存储和同步更新，这和生物神经网络中局部的脉冲时序依赖可塑性（STDP）存在极大差异，后者的代表———人脑的思考能耗仅为 20W 左右，这和深度神经网络反向传播过程中涉及的大量矩阵乘法和梯度计算形成鲜明对比。而且大脑中的信息传递是单向的，没有观察到反向传播机制[15]，因此 BP 算法被认为是生物学不合理的。3. 时域建模能力弱：传统人工神经网络（ANN）处理的是固定时间步的离散帧数据，而面对动态视觉传感器（DVS）[16]捕捉的动态事件流数据则有些无能为力。在这样的背景下，第三代神经网络——脉冲神经网络（Spiking Neural Networks，SNN）[17]应运而生。脉冲神经网络以独特的脉冲信息传播方式、生物学可解释的学习机制、时空域数据的处理能力和较低的能耗，与传统人工神经网络（Artificial Neural Networks，ANN）区分开来。

1.2 研究现状与挑战

脉冲神经网络（SNN）是一种受脑科学启发的新型神经网络模型，它通过时空动力学模拟神经行为，使用二进制脉冲信号在神经元之间通信。简其核心工作机制在于：脉冲神经网络中传输的是稀疏的脉冲信号，只有当膜电压逐渐积累达到发放阈值时，才会向下一层发放脉冲，如图1-2所示，这和传统 ANN 的连续向量运算不同，极大程度节省了神经元在非工作状态下的能耗，这种由事件驱动的稀疏 0-1 脉冲串信息表示方法正是脉冲神经网络节约能耗的优点的来源。人们研究脉冲神经网络，不光是因为它能逼真地模拟人脑的生物计算过程，还因为它被部署在神经形态芯片上时，有比 ANN 在 GPU 上运行时更好的能量效率，这为突破冯·诺依曼架构存算分离瓶颈提供了新路径。

1.2.1 脉冲神经网络的研究现状

脉冲神经网络在最近几年获得了迅速的发展，目前呈现出多维度突破态势，其核心研究重点可以归结为以下几个方面：脉冲神经元的建模、脉冲编码方式的研究、高效低耗的脉冲神经网络学习算法的优化、脉冲神经网络的硬件协同设计、脉冲神经网络的跨学科应用领域拓展等，下面将分别做出介绍。

首先是脉冲神经元的建模方面，常见的脉冲神经元模型有 HH 模型、LIF 模

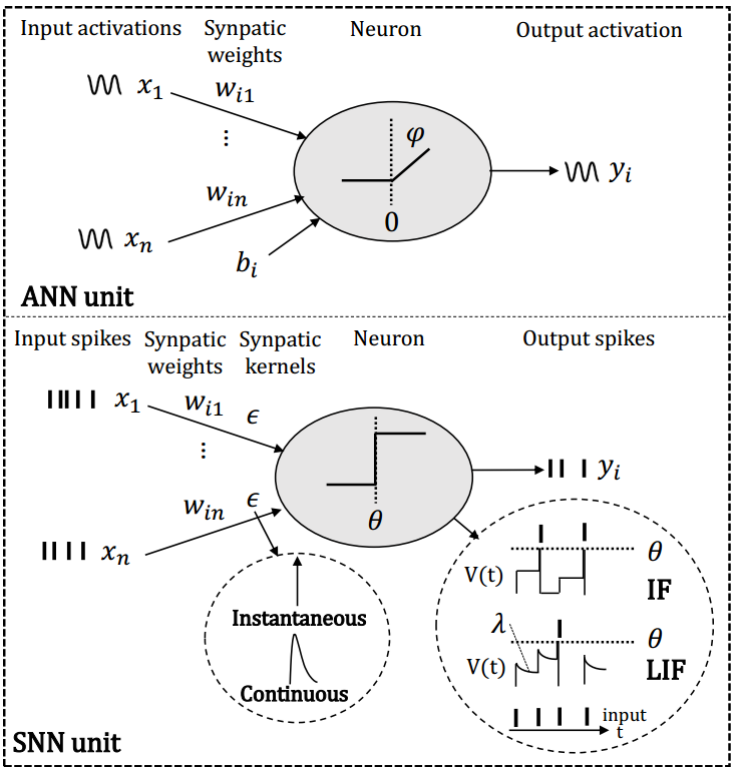

图 1-2　人工神经元和脉冲神经元对比图[18]

型、Izhikevich 模型等。1952 年，Alan Hodgkin 和 Andrew Huxle 通过对乌贼的巨大神经轴突的电活动过程的观察，揭示了神经元膜电位变化的离子通道机制，提出了 Hodgkin-Huxley（HH）模型[19]，该模型通过一组微分方程描述离子通道的动态变化，并通过离子流动计算膜电压状态。虽然 HH 模型对生物学观察结果还原度很高，但是它较为复杂，不适合大规模仿真。Leaky Integrate-and-Fire（LIF）模型[20]克服了以上不足，LIF 模型模拟了膜电位的整合、放电和泄露过程，它足够简约，便于大规模仿真计算，因此使用最为普遍。但 LIF 模型并非完美，因为模型较简单，其信息表达能力偏弱。2003 年，Izhikevich 博士对 HH 模型进行化简，提出了 Izhikevich 模型[21]，它在计算复杂度和生物可解释性之间进行了权衡。除此之外，还有不少神经元模型，比如 Integrate-and-Fire（IF）模型、Spike Response Model（SRM）等，不同的模型各有优缺点。最新研究进展表明，研究者正在构建兼顾生物可解释性和计算可行性的新脉冲神经元模型。

其次是脉冲编码方式的研究，常见的脉冲编码方式有频率编码、时间编码等。频率编码较为常见，它使用概率采样将每个像素强度和发放率相匹配来生成脉冲序列，像素值越高，相应输入的激发率就越高。由于概率采样通常采用泊松分布，因此频率编码也可以称为泊松编码。时间编码顾名思义利用脉冲的精确时

间或时间间隔进行编码，主要又分为以下两种：首个脉冲时间编码和延迟编码。首个脉冲时间编码用第一个脉冲的发放时刻来传达信息，输入值越大，第一个脉冲时间越早；延迟编码用脉冲之间的时间间隔来表达信息。相较于频率编码，时间编码是稀疏的，每一个脉冲都携带重要信息，因此需要较高的时间分辨率。此外，还有一些编码方式如：德尔塔调制、种群编码等等。不同的脉冲编码方式有各自的优势和应用场景，选择合适的编码方式对提升网络性能和效率至关重要。随着研究的深入，未来的工作可能着眼于优化编码信息效率、提高时间分辨率的同时降低计算复杂度。

接着是高效低耗的脉冲神经网络学习算法的优化，脉冲神经网络主流的学习算法分为三类[22]：脉冲时序依赖可塑性（Spike Timing Dependent Plasticity，STDP）、替代梯度下降法（Surrogate Gradient Descend）和人工神经网络转换脉冲神经网络的方法（ANN-to-SNN）。STDP 属于无监督学习算法，是一种仿照生物神经元突触可塑性启发的调整突触连接权重的局部 Hebb 学习规则，它根据突触前和突触后脉冲之间的相对时间差决定了突触权重变化的方向和大小。该方法的优点是生物学合理性高，不需要标签信息，计算开销小；缺点是精度有限，仅限于较简单的网络和任务[23]。替代梯度下降法[24]属于直接训练的监督学习算法，由于脉冲激活函数的不可微特点，ANN 中的误差 BP 算法在 SNN 中会遇到求导困难的问题，因此替代梯度下降法使用连续可微函数的导数代替脉冲激活函数的导数，从而使得反向传播过程得以正常实现。替代梯度下降法的优点是一定程度上克服了脉冲神经网络训练困难的难题；缺点是不但继承了 ANN 中 BP 算法的梯度消失/爆炸、梯度计算开销大等问题，替代梯度本身带来的替代误差也会影响网络的精度。ANN-to-SNN 方法[25]是一种非直接训练的监督学习方法，将一个训练好的 ANN 的参数迁移到 SNN 当中，并进一步进行微调或训练脉冲神经网络中的参数。ANN-to-SNN 方法的优点是精度高，和同结构的 ANN 差距很小；缺点是生物学合理性缺失，且模拟时间步较多带来的计算开销大。除了三大主流学习算法外，还有一些混合学习方法和其他小众的学习算法，这些方法都取得了一定的成效。脉冲神经网络学习算法的优化在各个研究方向中可谓是重中之重，由于没有统一的算法占据主导地位，各种新颖独特的算法创新研究仍在推进当中。

然后是脉冲神经网络的硬件协同设计，神经形态芯片的架构创新是释放 SNN

能效潜力的关键。IBM 公司设计的神经形态芯片 TrueNorth[26]含有约 100 万个神经元和 2.56 亿个突触，可实现基于 SNN 的类脑计算。英特尔推出的的 Loihi[27]神经形态芯片拥有 13 万个神经元和 1.3 亿个突触，可以灵活模拟各种 SNN 模型。浙江大学联合之江实验室研发的类脑计算机，有 1.2 亿个神经元和 720 亿个突触，达到了小型哺乳动物大脑的规模量级，典型运行功耗只需 350-500 瓦，体现了神经形态硬件的低能耗特性。脉冲神经网络和神经形态硬件的协同发展，将充分发挥存算一体、低能耗、并行运算等方面发挥优势，尤其是在能耗方面显著低于传统硬件。神经形态为类脑通用智能奠定了硬件基石，一旦得到普及，将会大力推动人工智能在嵌入式设备、边缘计算和物联网等领域的应用，前景非常广阔。

最后是脉冲神经网络的跨学科应用领域拓展。脉冲神经网络在很多领域都有应用，比如动态视觉传感器、脑机接口[28]、生物信号异常检测[29]、神经形态具身智能、物联网和边缘计算等领域。以动态视觉传感器（DVS）为例，它能够实时捕捉场景中的动态变化，当某个像素的亮度变化超过一定阈值，它会记录并生成“事件”数据，因此 DVS 生成的数据本质上是脉冲信号，非常符合 SNN 处理信息的方式。相比于传统视频帧，DVS 记录的事件数据仅在场景发生动态变化时触发，极大减少了计算冗余，具有低能耗的优点。可以展望，SNN 和 DVS 的结合能够处理动态变化复杂场景，在智能监控、自动驾驶、机器人等应用场景下前景可期。

1.2.2 脉冲神经网络训练算法的研究挑战

尽管脉冲神经网络在理论和应用中取得了一定的进展，但目前仍面临许多挑战，主要的挑战有：神经元模型和编码方式仍有改进空间、训练算法的难题、生物学合理性和计算能效的平衡困境、软硬件架构的不成熟、落地应用场景亟需拓展等。其中训练算法的难题以及生物学合理性和计算能效的平衡困境是当前面临的诸多挑战中的重点问题，也是本工作的着眼点，因此将会重点展开介绍：

一直以来，脉冲神经网络的训练算法难题是困扰研究者的核心问题。由于脉冲激活函数是阶跃的 Heaviside 函数，它在间断处是不可微的，只有广义导数为 Dirac Delta 函数，该函数处处为零，除了在间断处为正无穷，这会导致使用常规 ANN 中的 BP 算法时遇到梯度难以计算的困难。针对这一问题根源，研究者探索出了多种不同的可行解决方案：即主流的三种 SNN 训练算法 STDP、替代梯度

下降法、ANN-to-SNN，还有一些方案例如混合学习方法，总之并没有一种统一的最佳方案。几种主流算法各有亟待解决的缺陷：STDP 生物学可解释性高，但是其无监督的学习方式意味着没有使用标签信息，这限制了它在深层网络和复杂任务上的性能，难以成为深度脉冲神经网络的有效训练算法。替代梯度下降法一定程度上缓解了脉冲神经网络由于脉冲激活函数不可微带来的难以梯度反向传播的问题，但是使用代理函数替代脉冲函数这一过程是有近似误差的，近似误差会在深层网络中逐渐积累，这从根本上决定了 SNN 的精度比不上相同结构的 ANN。ANN-to-SNN 的方法需要借助 ANN 的辅助，将训练好的权重迁移给 SNN，严格意义上不是 SNN 自身的训练算法，这有悖 SNN 仿生性的设计初衷，很难想象一种生物可解释的神经网络却依赖生物学合理性较低的 ANN 才能训练。总体来看，这些方法虽然为脉冲神经网络的训练提供了多样的可能，但是没有一种普适性、能解决深度 SNN 中所有问题的通用方案，要么为了生物学合理性牺牲了精度（STDP），要么为了精度牺牲了生物学合理性和能耗（ANN-to-SNN），要么在精度、能耗、生物学合理性三方面都表现一般（替代梯度下降法）。未来的研究可能在生物学可解释性、计算能耗和精度之间寻找更好的平衡，改进现有方法或者发展新的算法，使得 SNN 能够在更复杂的任务中发挥更大的潜力。

对于脉冲神经网络训练算法的难题，有以下一些已被证实可行或值得尝试的技术路线：一是奖励调制的 STDP 方法，让全局奖励信号调整 STDP 过程，解决了传统 STDP 缺乏全局优化能力的问题[30]。二是混合方法，可以尝试混合两种或多种主流学习算法，例如使用梯度下降的全局误差信号调制 STDP[31]，可以集成不同学习算法的优点；三是基于能量的平衡传播方法[32]，该方法在每个训练阶段收敛到能量最小值来实现稳定平衡态，是一种时空局部的更新规则；四是使用反馈对齐算法替代 BP 算法[33]，使用直接误差信号反馈取代梯度计算，简单的说，BP 是一种经典全局空间信度分配（确定每个权重对总损失的贡献度）的算法，但它不够类生物和节能高效，而反馈对齐是一种全新的空间信度分配方案，克服了 BP 的上述缺点，因此该算法被用于 SNN 的训练是自然的；五是一些分层贪婪局部学习手段，在深层网络中为每一层分配独立的局部损失函数[34]，用训练浅层网络的办法成功训练深层脉冲神经网络，从而避免 SNN 难以梯度反向传播的困境。

1.3 研究内容与贡献

本文本着追求生物学合理性、计算能耗和精度三者平衡的理念，设计了两种兼顾三方面的脉冲神经网络非反向传播（non-BP）式局部学习算法。首先是一种混合损失局部学习算法，为脉冲神经网络每一层构建一个局部分类器，每一层分别计算局部误差并进行权重更新，能够有效地让整个网络得到学习；其次是一种基于 HSIC 信息瓶颈损失 + 三因素赫布学习规则的学习算法，同样独立更新 SNN 的每一层，每一层除了使用突触前后突触后信号这两个因素外，还有 HSIC 瓶颈作为全局调制第三因素，本方法提供了一个新颖的 SNN 学习框架。研究内容要点罗列如下：

1. 本文提出了一种适合脉冲神经网络的混合损失局部学习方法（Spiking Neural Network Hybrid Loss Local Learning Algorithm，SNN-HLL）。使用局部损失是迈向更生物合理的深度脉冲神经网络的一步，因为全局损失不必被逐层回传至隐藏层，这巧妙地避开了脉冲函数不可微的难题和替代梯度带来的累积误差问题。为了使局部损失的训练效果不逊色于 BP 全局损失的训练效果，尝试使用不同损失的组合以加强误差信号特征，主要损失类型有：相似度匹配损失（similarity loss）、预测损失（prediction loss）、重构损失（reconstruction loss）等。实验表明混合局部损失 prediction-similarity loss 的性能完全不输全局反向传播算法，在某些情况下前者可以超越后者。本算法作为脉冲神经网络局部学习算法有较高的生物学合理性，同时避免了反向传播链式法则中的大量矩阵和梯度计算，计算开销较低，且实验精度较高，同时兼顾了三个方面的考量。

2. 本文提出了一种能训练脉冲神经网络的 HSIC 瓶颈三因素赫布学习方法（Spiking Neural Network HSIC Bottleneck Three-Factor Hebbian Learning Algorithm，SNN-HBH）。信息论中的互信息可以很好地刻画两个随机变量的相关性，而基于互信息定义的信息瓶颈原理可以尽量压缩网络学习中的噪声数据时，尽量保留想表达的信息。由于信息瓶颈的分布估计麻烦，因此转而寻求使用信息瓶颈的替代品——希尔伯特-施密特独立性准则（Hilbert-Schmidt Independence Criterion，HSIC），并通过局部可塑性三因素赫布学习规则的形式：神经元前、后突触的脉冲发放率（常规赫布学习的形式）+HSIC 瓶颈作为误差指导信号（第三因素）。在图像分类任务上的实验表明，该方法的性能接近替代梯度下降法的性能。本工

作和常规损失学习聚焦预测的正确与否截然不同，而是关注网络中不同层之间的信息相关性，较为新颖，且遵循可塑性学习规则，生物可解释性高。

1.4 本文组织结构

本文一共分为五个章节，如图1-3所示，。本章为第一章绪论，简要介绍了脉冲神经网络的发展背景和实际意义，随后介绍了脉冲神经网络的研究现状，并探讨了相关的研究挑战，从而引出了本文为应对这些挑战进行的努力，即研究内容与贡献中的两项主要工作。随后的四章节的内容和组织方式如下：

第二章为相关工作，先对脉冲神经网络必备的基础背景知识做了介绍，包括模型、编码和算法，然后介绍了主要工作中涉及的一些局部学习算法的技术原理，这为后续两个章节起到了预热铺垫的作用。

第三章提出了一种基于相似度匹配损失、预测损失和重构损失的混合损失脉冲神经网络局部学习方法，并进行了算法设计、实验和分析的流程。

第四章提出了一种基于 HSIC 瓶颈的三因素赫布学习方法，并进行了算法设计、实验和分析的流程。

第五章为总结和展望。

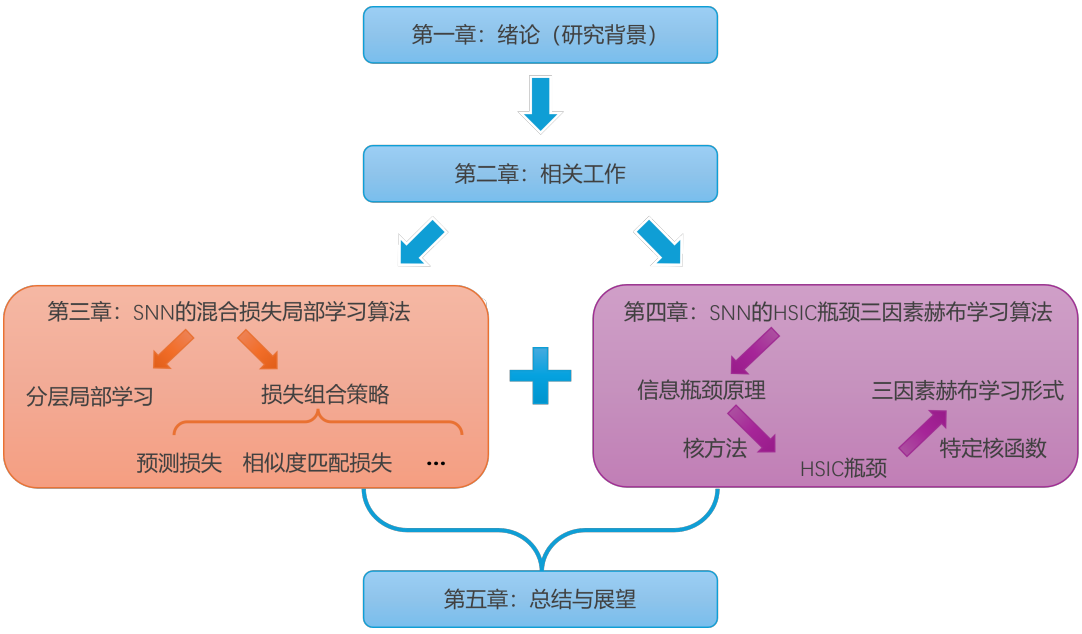

图 1-3 本文章节结构图

第二章　相关工作

本章节主要介绍生物学合理的脉冲神经网络的非 BP 式局部学习训练算法的基础知识和关键技术，并概述了典型相关工作，服务于后续的第三、四章。首先介绍了脉冲神经网络的基础知识，包括神经元模型、编码和算法；然后介绍了后续工作涉及的一些局部学习技术，例如反馈对齐、信息瓶颈等。

2.1　脉冲神经网络基础

脉冲神经网络（SNN）是一类基于生物神经系统的计算模型。与传统的人工神经网络（ANN）不同，SNN 使用脉冲信号（spikes）进行信息传递，而不是连续的激活值。这种设计使得 SNN 能够更真实地模拟大脑中神经元的活动，尤其是在信息的时间编码和事件驱动计算方面。SNN 具有低功耗、高并行度和高效的时序信息处理能力，已成为神经形态计算和类脑智能研究的核心技术之一。在 SNN 的基础知识中，神经元模型、编码方式和训练算法是其三个基本组成部分，分别涉及神经元的动态行为、如何表示输入信息以及如何通过学习来调整神经网络的参数。下面将详细介绍这些基本概念。

2.1.1　脉冲神经元模型

SNN 的神经元模型通常基于生物神经元的电生理学特性而进行的设计，并进行了适当的简化。常见的脉冲神经元模型包括 Hodgkin-Huxley 模型、Izhikevich 模型和 Leaky Integrate-and-Fire (LIF) 模型。这些模型各有特点，应用场景视需求而定。由于本文的工作采用 LIF 模型，所以会详细介绍该模型，对另外两种简略介绍。

Hodgkin-Huxley 模型是三者中最为精确的神经元模型，它详细模拟了神经元的膜电位和离子通道的电流动态。该模型通过一组四个常微分方程来描述神经元的动态行为，考虑了钠离子（Na+）、钾离子（K+）和泄漏电流，如图2-1所

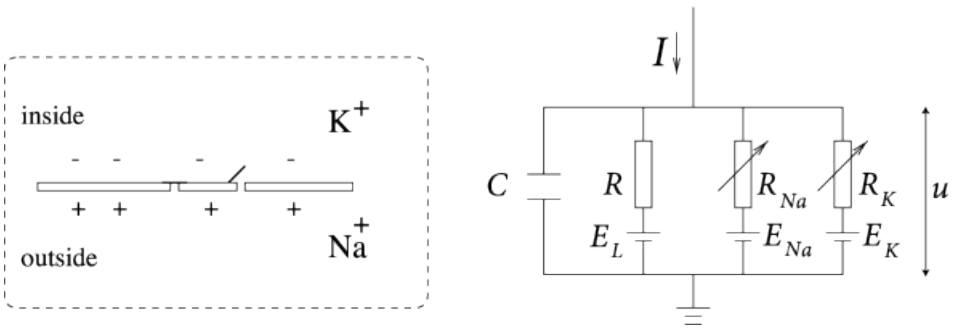

图 2-1　Hodgkin-Huxley 模型离子通道建模示意图

示。膜电位 V(t) 的变化由以下方程描述：

$$C_m\frac{dV}{dt} = -(I_{N_a} + I_K + I_{Leaky}) + I_{ext}. \tag{2-1}$$

其中 C_m 是膜电容，I_{N_a}，I_K，I_{Leaky} 分别是钠、钾和泄漏电流，I_{ext} 是外部输入电流。三个离子电流的具体形式如下：

$$I_{N_a} = g_{Na}m^3h(V - E_{N_a}), \tag{2-2}$$

$$I_K = g_K n^4(V - E_K), \tag{2-3}$$

$$I_{Leaky} = g_L(V - E_L). \tag{2-4}$$

其中 g_{N_a}，g_K，g_{Leaky} 分别是钠离子通道、钾离子通道和泄漏电流通道的电导，E_{N_a}，E_K，E_{Leaky} 为三个通道的平衡电位，m 和 n 为钠、钾通道打开的概率（介于 0 和 1 之间），h 为通道关闭的概率。三个概率的公式由三个微分方程给出：

$$\frac{dm}{dt} = \alpha_m(1 - m) - \beta_m m, \tag{2-5}$$

$$\frac{dn}{dt} = \alpha_n(1 - n) - \beta_n n, \tag{2-6}$$

$$\frac{dh}{dt} = \alpha_h(1 - h) - \beta_h h. \tag{2-7}$$

其中 α_m，α_n，α_h，β_m，β_n，β_h 是和膜电压相关的速率常数，一般由实验观察总结出的经验性数值。

Izhikevich 模型较 HH 模型更为简化明了，它只用了两个微分方程和少数几个参数便实现了对不同类型的神经元放电行为的较为真实的模拟。描述膜电位 v

和恢复变量（控制膜电位的回复速度）u 的微分方程和重置公式如下：

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I, \tag{2-8}$$

$$\frac{du}{dt} = a(bv - u), \tag{2-9}$$

$$\text{if } v > 30mV, \text{ then } v \leftarrow c, \ u \leftarrow u + d. \tag{2-10}$$

其中 I 是外部输入电流，a 是恢复变量的时间尺度，b 是膜电位与恢复变量之间的耦合强度，c 是膜电位 v 达到阈值时的重置值，d 是膜电位达到阈值时恢复变量 u 增加的一个固定值。这些参数设置不同的数值大小，会让神经元处于不同的运行状态，比如参数 a 较大时，为适应性放电行为，神经元会呈现逐渐减缓的脉冲发放频率；当参数 b 较大时，为爆发放电状态，神经元会在短时间内发放多个脉冲，之后静息一段时间。Izhikevich 模型能够通过调整这几个参数来模拟生物神经元多种不同的放电模式，是一种较为简约高效的脉冲神经元模型。

下面重点介绍 LIF 神经元模型，这是一种较为简化但运用最广泛的脉冲神经元模型。它最早可以追溯到 1907 年 Lapicque 用电流刺激青蛙腿的实验[35]，通过调节电流的幅度和持续时间来观察青蛙腿的抽搐时间，并得出结论：脉冲神经元可被视为一个由电容 C 和电阻 R 组成的低通滤波器电路，现在的 LIF 神经元依然是基于这种 RC 电路建模形式的。LIF 神经元的核心思想是随着时间的推移膜电位会累积输入电流，并且会发生泄漏（即膜电位会随着时间逐渐衰减），当膜电位达到既定阈值时，神经元会发放脉冲，随后膜电位会被重置。顾名思义，Leaky Integrate-and-Fire 神经元的整个行为流程可以拆分为三个部分：1. 整合：当收到外界的电流输入（current input），神经元膜电位（membrane potential）在泄露的同时会根据输入电流进行累积，2. 发放：当目前的膜电位超过发放阈值（threshold），神经元会激活，向下一层释放一个脉冲（spike）；3. 重置：发放脉冲后，膜电压会更新为重置电位（reset potential）。

LIF 神经元的公式有两个版本：连续时间版本和离散时间版本。微分方程形式的连续时间版本膜电压公式如下：

$$\text{Integrate：} \ \tau_m \frac{dV}{dt} = -(V(t) - V_{rest}) + R_m I(t). \tag{2-11}$$

其中 $\tau_m = R_m C_m$ 表示膜时间常数，R_m 为膜电阻，C_m 为膜电容，V_{rest} 为静息电位。当膜电位 $V(t)$ 达到或超过发放阈值时，神经元发放脉冲，随后膜电位被瞬时重置，并进入短暂的不应期（Refractory Period），此过程用公式表达如下：

$$\text{Fire}：\ \text{If } V(t) \geqslant V_{th},\ \text{then } S(t) = 1. \tag{2-12}$$

$$\text{Reset}：\ V(t) = V_{reset}. \tag{2-13}$$

这里的 $S(t)$ 表示当前时刻发放的脉冲，V_{reset} 表示重置电位。

由于微分方程形式的表达式不便于实验模拟，实验仿真中默认使用以下离散时间形式的表达式：

$$\text{Integrate}：\ V(t+1) = V(t) + \frac{1}{\tau_m}[-(V(t) - V_{rest}) + R_m I(t)]. \tag{2-14}$$

$$\text{Fire and Reset}：\ \text{If } V(t) \geqslant V_{th},\ \text{then } S(t) = 1,\ \ V(t) = V_{reset}, \tag{2-15}$$

$$\text{else } S(t) = 0.$$

LIF 模型的数学形式源自 Hodgkin-Huxley 模型的简化，通过忽略 Na^+/K^+ 等离子通道动态并线性化膜电位变化，在计算效率与生物合理性之间取得平衡，因此本工作全部采用了 LIF 模型。

2.1.2　脉冲编码方式

脉冲编码是脉冲神经网络中信息表征的核心机制，决定了外部输入刺激如何有效地转化为适配 SNN 的数据形式——离散的脉冲事件。目前有多种不同的编码策略，如频率编码、时间编码、德尔塔调制编码[36]等，下面一一介绍：

频率编码（Rate Coding）：频率编码是最常见的脉冲编码方式，它通过脉冲发放的频率来表示信息的强度。在这种方式中，神经元的输出频率（即单位时间内发放脉冲的次数）与输入信号的强度成正比。具体来说，输入信号越强，神经元发放脉冲的频率就越高。

泊松编码（Poisson Coding）[37]：泊松编码是频率编码的一个具体实现形式，它通过泊松过程来模拟随机的输入脉冲序列，使得某个时间窗口内，脉冲序列的平均发放率符合频率编码的定义。

延迟编码（Latency Coding）：延迟编码使用脉冲之间的时间间隔来编码信息，即输入信号的强度决定了脉冲之间的间隔时间。输入信号越强，脉冲之间的时间间隔越短。

首个脉冲时间编码（First Spike Timing Encoding）[38]：首个脉冲时间编码和延迟编码同属于时间编码的范畴。顾名思义，在这种编码方式中，信息通过第一个脉冲的发放时刻来表示。输入信号越大，第一个脉冲的发放时间越早。

相位编码（Phase Coding）：相位编码通过脉冲在周期性信号中的相对位置（即相位）来表示信息。具体来说，信息通过脉冲的发放相对于某个周期信号的相位（如正弦波周期）进行编码。

种群编码（Population Coding）[39]：种群编码是一种更高阶的编码方式，是通过多个神经元群体的共同活动来表示信息。每个神经元可能只携带少量信息，但通过联合多个神经元的活动（例如，发放的脉冲数量、频率、时间间隔等）可以共同表达一个更复杂的信息。

直接编码（Direct Coding）[40]：直接编码其实是没有显式的编码环节，直接将原始物理世界的连续值数据输入脉冲神经网络中，让网络的第一层直接处理数据，起到隐式编码层的作用。这种做法的好处是尽可能多地保留原始数据信息，一些最新研究表明直接编码的实验精度可以超越常见的频率编码。

2.1.3 SNN 学习算法之一：STDP

脉冲时间依赖可塑性（Spike-Timing-Dependent Plasticity, STDP）是一种突触可塑性规则，其核心思想是突触强度的调整取决于突触前神经元与突触后神经元动作电位（脉冲）的时间顺序。如图2-2所示，若突触前神经元在突触后神经元之前放电（即时间差 $\Delta_t = t_{post} - t_{pre} > 0$），突触权重增强（长时程增强, LTP）。若突触后神经元先于突触前神经元放电（$\Delta_t < 0$），突触权重减弱（长时程抑制, LTD）。这一特性体现了神经系统的赫布学习法则，模拟了神经元之间的突触可塑性过程。STDP 的权重更新规则可以使用如下形式的函数来描述：

$$\Delta W = \begin{cases} A_+ \cdot e^{-\Delta_t/\tau_+} & \text{if } \Delta t > 0, \\ -A_- \cdot e^{-\Delta_t/\tau_-} & \text{if } \Delta t < 0. \end{cases} \tag{2-16}$$

其中 A_+ 和 A_- 分别是用于表示突触增强和抑制的幅度常数，τ_+ 和 τ_- 分别是正向和负向时间常数，决定了 STDP 学习规则的时间窗口宽度。

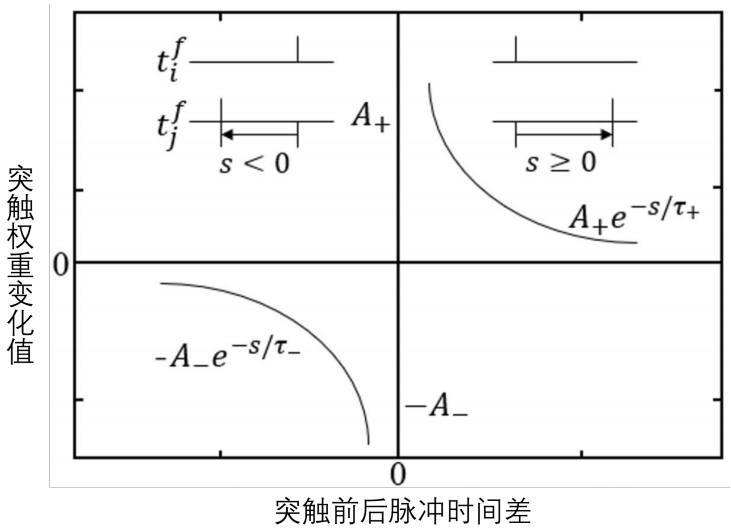

图 2-2　脉冲时序依赖可塑性学习窗口和 STDP 曲线[23]

STDP 的学习机制在生物神经系统中有着广泛的实验证据，尤其在大脑皮层和海马体等区域。神经科学研究表明，当一个神经元的突触前脉冲和突触后脉冲之间的时间间隔较短时，神经元之间的协同活动会得到增强，从而提高神经元之间的信息传递效率。反之，当脉冲时间间隔较长时，神经元之间的连接会减弱。这个过程被认为是学习和记忆的一个关键机制。

STDP 还存在几种变体拓展：1. 抑制性突触 STDP：突触的权重可遵循反向规则（后脉冲先于前脉冲导致增强）；2.Triplet-STDP：三元组 STDP 是两个突触后脉冲和一个突触前脉冲的三重态相互作用，这种多个神经元之间的相互作用变得更加复杂且细致，能够捕捉到更多的时序信息；3.STDP-R：奖励驱动的 STDP 在传统的 STDP 基础上加入了神经调质（如多巴胺）作为全局奖励信号来调节突触的可塑性，允许网络根据外部奖励来加强或削弱某些突触连接。这种变体结合了强化学习的元素，使得 SNN 能够在有监督的环境中进行学习；4.Asymmetric STDP：非对称 STDP 在突触前后脉冲时间差的不同方向上使用不同的学习强度，即对于 LTP 和 LTD 使用不同的幅度和时间常数，以加强某种类型的学习过程。

STDP 的优点是作为一种无监督学习算法，不需要大量标签数据；且 STDP 遵循了“用进废退”的赫布学习思想，在传统赫布学习基础上引入了时间依赖性，更加符合生物中的神经可塑性现实，具有较高的生物学可解释性。STDP 的

局限性在于，基于 STDP 算法的 SNN 网络结构仅限于浅层网络，深度一般不超过 4-5 层，主要应用一般为最简单的 MNIST 数据集[41]。这是因为 STDP 这种缺乏全局损失监督的学习规则在深层网络中难以协调多层的信息，难以进行稳定的权重优化和有效的特征学习。

2.1.4 SNN 学习算法之二：替代梯度下降法

替代梯度下降法（Surrogate Gradient Descent）是一种用于脉冲神经网络训练的优化方法，旨在解决脉冲神经元模型中激活函数不可微的问题。传统的反向传播 BP 算法要求激活函数必须是连续可微的，而 SNN 中的脉冲神经元通常采用阶跃函数，使得经典的反向传播方法无法直接应用。因此，替代梯度下降法应运而生，它通过使用一个平滑的“代理”梯度函数来代替原始激活函数的梯度，从而实现神经网络的训练。

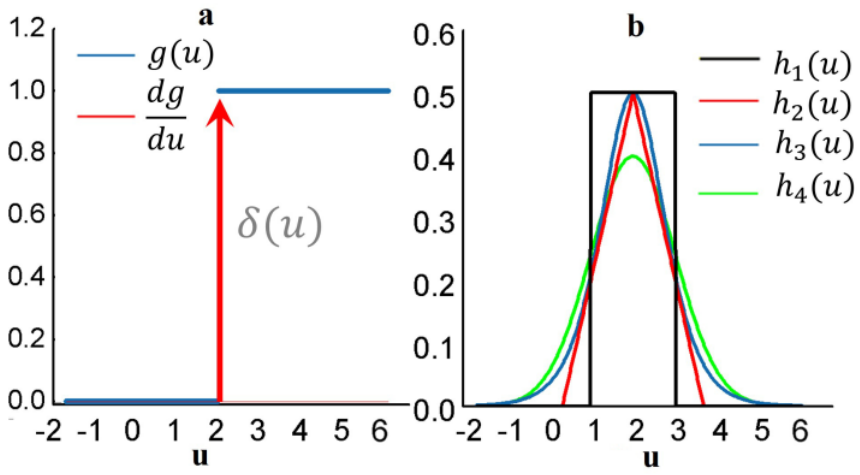

图 2-3 替代脉冲函数的不同代理函数选择示意图[42]

代理函数在脉冲激活函数的不可微点附近近似其导数，从而可以计算出近似的梯度，并通过这些梯度更新网络权重。在实际应用中，替代函数通常是选择一个平滑的函数（如 sigmoid、tanh 等），图2-3展示了不可微的脉冲函数和几种可微的替代函数的导数的示意图，常见的选择有如下几种，指数函数近似：

$$\sigma'(u) = \alpha e^{-\alpha|u-v_{th}|}. \tag{2-17}$$

Sigmoid 近似：

$$\sigma(u) = \frac{1}{1 + e^{-\alpha(u - v_{th})}}, \tag{2-18}$$

$$\sigma'(u) = \alpha\sigma(u)(1 - \sigma(u)). \tag{2-19}$$

矩形函数近似：

$$\sigma'(u) = \begin{cases} \frac{1}{2\gamma} & \text{if } |u - v_{th}| \leqslant \gamma, \\ 0 & \text{otherwise .} \end{cases} \tag{2-20}$$

通过将原本不可微的激活函数替换为一个平滑的代理函数，替代梯度下降法允许使用梯度下降法进行训练。具体来说，脉冲神经网络中的损失函数 $\boldsymbol{L}$ 可以表示为神经元的输出与目标之间的误差。利用代理梯度法，损失函数相对于网络权重的梯度可以通过链式法则计算：

$$\begin{aligned} \frac{\partial L}{\partial W} &= \sum_t \frac{\partial L}{\partial S(t)} \cdot \frac{\partial S(t)}{\partial u(t)} \cdot \frac{\partial u(t)}{\partial W} \\ &\approx \sum_t \frac{\partial L}{\partial S(t)} \cdot \sigma'(u) \cdot \frac{\partial u(t)}{\partial W}. \end{aligned} \tag{2-21}$$

(损失梯度 * 代理导数 * 膜电位梯度)

替代梯度下降法优缺点比较明显。优点是与标准优化算法兼容，由于代理梯度函数是平滑的，替代梯度下降法可以与标准的优化算法（如 SGD、Adam 等）结合使用，从而加速训练过程。缺点是梯度消失问题，即虽然代理函数是平滑的，但在某些情况下，代理梯度的梯度值可能变得非常小，导致梯度消失问题，影响网络的训练效果；另一个缺点是引入了近似误差，尤其在深层神经网络中，这些误差逐渐积累进而影响 SNN 的模型性能。

2.1.5　SNN 学习算法之三：ANN-to-SNN

ANN-to-SNN 是一种间接训练脉冲神经网络的有效方法，由于其可以利用人工神经网络训练结果，所以在深度脉冲神经网络中取得了较好的结果。该方法将一个训练好的人工神经网络经过特殊处理和参数迁移转换成对应的脉冲神经网络，通过使用脉冲神经网络的激活频率拟合人工神经网络的输出，使得脉冲神经

网络可以取得与人工神经网络一致的数据拟合能力。

ANN-to-SNN 算法的流程分为以下几个步骤：1.ANN 训练：首先在标准框架（如 PyTorch、TensorFlow）中训练一个 ANN，使用 ReLU 等非负激活函数以确保激活值与 SNN 的脉冲发放率兼容。训练完成后，固定 ANN 的权重和结构。2. 权重与参数迁移：将训练好的 ANN 权重直接迁移到 SNN 中，为确保脉冲发放率与 ANN 激活值匹配，需对权重和阈值进行归一化处理。例如，通过最大激活值对权重进行缩放：

$$W' = \frac{W}{\lambda}. \tag{2-22}$$

其中 W' 表示转换后的 SNN 权重，W 表示原始 ANN 的权重，$\lambda = max(|W|)$ 表示 ANN 中某一层的最大激活值。3. 推理和微调：在 SNN 上进行推理，通过调整时间窗口长度、阈值和输入编码策略，优化脉冲发放率与 ANN 激活值的一致性。若精度不足，可采用微调策略进一步提升性能。

由于 ANN-to-SNN 方法无法实现从 ANN 到 SNN 的等效转化，难免有性能损失，所以研究者们提出一系列阈值均衡算法来减小这种损失，主要方法有以下几种：1. 基于神经元模型优化的阈值平衡，这类方法专注于优化脉冲神经元的发放机制，例如将膜电压重置从硬复位方式调整为软复位方式，再如根据输入分布动态调整发放阈值，这些手段有助于减少脉冲发放率与 ANN 激活值之间的偏差。2. 基于网络结构和权重的阈值平衡，这类方法主要是优化 ANN 到 SNN 转换过程中的网络结构和权重，以减少性能损失，例如权重归一化、添加残差连接等，可以减小转化过程带来的误差。

ANN-to-SNN 方法有如下优点：兼容现有的 ANN 架构，可以将现有的 ANN（如 VGG、ResNet 等）转换为 SNN，而无需重新设计网络架构；相比于前两种训练算法，更适用于深层网络和复杂任务（如 CIFAR-100、ImageNet 等），具有接近 ANN 的实验精度。ANN-to-SNN 方法的缺点是：需要较多的时间步，推理时的计算开销大；由于转化前的 ANN 是时间上静态的，转换过程又不涉及时间编码，所以 ANN-to-SNN 难以利用 SNN 固有的时间动态特性，在时序任务场景下效果有限；ANN-to-SNN 方法不是一种直接的、SNN 自身的训练算法，生物学可解释性低，ANN 训练结束后权重便固定，缺少突触可塑性机制。总而言之，ANN-to-SNN 方法可被视为一种关注精度的工程优化方案，而非真正的生物神经

计算模拟。

2.2　非 BP 式局部学习规则相关技术

以 BP 算法为代表的全局误差算法一直是神经网络领域的主流学习算法。然而在脉冲神经网络的场景下，BP 算法会遭遇脉冲激活函数不可微带来的反向传播困难问题，尽管有替代梯度下降法等算法一定程度上化解了 SNN 难训练的问题，但是替代函数的引入本身会带来近似误差，这种误差在深度 SNN 中会随着网络层数逐渐积累，最终有可能加重梯度消失的问题；而且基于 BP 的训练算法会遭遇“死神经元”问题（Dead Neuron Problem）[43]，即当神经元没有发放脉冲时就不会发生学习。因此，有必要考虑除了反向传播之外的训练方法，探寻更适合 SNN 的学习算法。

2.2.1　反馈对齐算法

BP 算法是生物学不可信的，BP 算法的前馈、反馈权重是对称的，反馈权重等于前向权重的转置。然而神经科学研究表明，大脑中存在广泛的反馈连接（Feedback Connections），这些反馈连接有别于皮层中的前馈连接，两者不共用一套突触连接系统，具有解剖学意义上的独立性。大脑中的前馈连接负责把原始感官信息传输到更高级的皮层区域，整合构建复杂的信息表征；而反馈连接从高级皮层区域的较深层投射到低级皮层区域的较浅层，反馈连接传递的是高层次的认知、预测，帮助低级皮层对输入信息进行调节修正。

既然生物神经网络不依赖全局反向传播的精确梯度传递，那么人工神经网络是否也能通过单独的反馈路径接收到训练信号？实验表明，在深度网络中，从输出层到路径中较早的神经元的直接相连的反馈路径足以实现误差驱动的学习，这就是所谓的反馈对齐（Feedback Alignment，FA）[44]类算法，该方法中反馈权重是随机的，并非前馈路径的复用。这些反馈连接采取类似从高级皮层区域到低级皮层区域的直接“自上而下”的皮层-皮层连接的形式，和通过链式法则逐层传递误差信号的方式有较大不同。

接下来将通过数学表达式形式化一下 BP 算法和 BP 的替代品——反馈对齐方法，以便更直观地了解两者的区别在哪里。假设有一个多层前馈神经网络，第

l 层的权重为 W_l，上一层的激活向量为 h_{l-1}，激活函数为 $f(\cdot)$，本层的输出由如下公式给出：

$$h_l = f(a_l) = f(W_l h_{l-1} + b_l). \tag{2-23}$$

利用链式法则，

$$\Delta W_l = -\eta \frac{\partial L}{\partial a_l} \frac{\partial a_l}{\partial W_l}. \tag{2-24}$$

注意到 $a_l = W_l h_{l-1} + b_l$，则有 $\frac{\partial a_l}{\partial W_l} = h_{l-1}$。记 $\delta_l = \frac{\partial L}{\partial a_l}$ 表示局部误差信号项，于是得到：

$$\Delta W_l = -\eta \delta_l h_{l-1}^T. \tag{2-25}$$

误差信号项在不同层的表示不相同，在输出层 L 中，通过损失函数对 a_L 的梯度来获得误差向量 e，即 $e = \frac{\partial L}{\partial a_L} = \delta_L$。于是输出层 L 的权重更新公式为：

$$\Delta W_L = -\eta e h_{L-1}^T. \tag{2-26}$$

对于隐藏层 $l(1 \leqslant l \leqslant L)$，需要通过将输出误差向量 e 通过反馈路径传播来计算误差信号 δ_l，误差信号项由下一层传播来的误差信息和突触后权重的转置以及激活的导数共同表示：

$$\delta_l = \frac{\partial L}{\partial a_l} = (W_{l+1}^T \delta_{l+1}) \odot f^{'}(a_l). \tag{2-27}$$

于是隐藏层 l 权重更新公式为：

$$\Delta W_l = -\eta((W_{l+1}^T \delta_{l+1}) \odot f^{'}(a_l)) h_{l-1}^T. \tag{2-28}$$

仔细观察一下这个权重更新公式，它可以被视为一个类赫布学习规则。它由两部分组成：突触前活动即前一层输出信息 h_{l-1}^T，由后续层传递的误差信息和激活门控函数调整得到的类似突触后活动的项 $(W_{l+1}^T \delta_{l+1}) \odot f^{'}(a_l)$，这暗示反向传播的更新可视为一种广义赫布规则。

可以注意到在 BP 的权重更新公式中，需要使用第 $l+1$ 层的前馈权重 W_{l+1} 来计算第 l 层的权重改变量 ΔW_l，也就是说反向传播中使用的突触权重与正向计

算中使用的突触权重完全相同，BP 存在权重传输问题（或者叫权重对称问题）。此外还有更新锁定问题，即计算 δ_l 之前需要首先计算 δ_{l+1}，前者对后者有依赖关系，这妨碍了并行更新。基于这些观察，研究者们提出了反馈对齐类算法，来改进 BP 的这些不足之处。

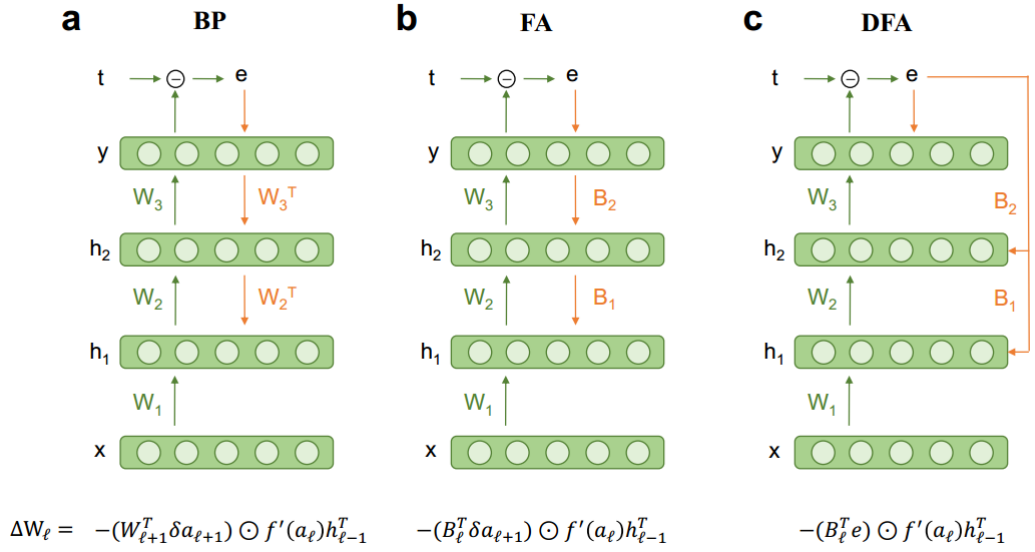

图 2-4　BP 算法、FA 算法和 DFA 算法原理对比图[45]

如图2-4所示，反馈对齐算法在 δ_l 的计算过程中用固定的随机矩阵 B_{l+1} 代替 W_{l+1}^T，来获得每一层的误差信号：

$$\delta_L^{\mathrm{FA}} = \frac{\partial L}{\partial a_L} = e, \tag{2-29}$$

$$\delta_l^{\mathrm{FA}} = \frac{\partial L}{\partial a_l} = (B_{l+1}\delta_{l+1}^{\mathrm{FA}}) \odot f'(a_l). \tag{2-30}$$

因此隐藏层 l 权重更新公式为：

$$\begin{aligned} \Delta W_l^{\mathrm{FA}} &= -\eta \delta_l^{\mathrm{FA}} h_{l-1}^T \\ &= -\eta((B_{l+1}\delta_{l+1}^{\mathrm{FA}}) \odot f'(a_l)) h_{l-1}^T. \end{aligned} \tag{2-31}$$

如上述公式所示，计算 δ_l^{FA} 时，用 B_{l+1} 代替前向权重 W_{l+1}^T 可以解决权值迁移问题；但是后向锁定问题仍然存在，因为在 $\delta_{l+1}^{\mathrm{FA}}$ 的计算完成之前依旧无法进行 δ_l^{FA} 的计算。在基准数据集上的实验结果表明，反馈对齐 FA 算法可以在一些不太复杂的场景下成功训练神经网络，证实了训练多层神经网络不需要精确的前向权

值，并且只需要用固定的随机权重来训练。此外，实验观察到随着训练的进行，前向权重 W_{l+1}^{T} 和随机反馈矩阵 B_{l+1} 的余弦相似度夹角在逐渐缩小，这意味着每个前向权重随着学习进度与其对应的反馈权重对齐。

由于反馈对齐的顺序计算方式无法避免更新锁定问题，直接反馈对齐（Direct Feedback Akignment,DFA）[46]被提出，见图2-4。在 DFA 中，输出层的误差信号通过随机反馈权值直接传播到所有隐藏层，即每一个隐藏层都接受统一的误差向量 e。DFA 方法中的误差信号公式如下所示：

$$\delta_{L}^{\mathrm{DFA}}=\frac{\partial L}{\partial a_{L}}=e, \tag{2-32}$$

$$\delta_{l}^{\mathrm{DFA}}=\frac{\partial L}{\partial a_{l}}=(B_{l+1}e)\odot f^{'}(a_{l}). \tag{2-33}$$

相应的隐藏层 l 权重更新公式为：

$$\begin{aligned}\Delta W_{l}^{\mathrm{DFA}}&=-\eta\delta_{l}^{\mathrm{DFA}}h_{l-1}^{T}\\&=-\eta((B_{l+1}e)\odot f^{'}(a_{l}))h_{l-1}^{T}.\end{aligned} \tag{2-34}$$

可以注意到，DFA 算法的第 l 层的误差信号直接使用了输出层误差向量 e，不需要上游层的误差信号 $\delta_{l+1}^{\mathrm{DFA}}$，因此不存在反向锁定问题，可以有效地并行训练。

在常见数据集上的实验表明，DFA 算法的性能表现好于 FA 算法，尽管仍然没有超过 BP 算法。最近，一些基于 DFA 的变体工作也陆续出现，例如稀疏直接反馈对齐（Sparse Direct Feedback Akignment, SDFA）通过在反馈矩阵中引入稀疏性，从而相比于传统 DFA 减少了反馈权重的内存；再如前向直接反馈对齐算法（Forward Direct Feedback Alignment, FDFA）[47]通过前向传播过程中的随机扰动计算方向导数，并用方向导数和动量机制更新随机反馈矩阵，使其逼近输出层对隐藏层的真实导数。FDFA 算法在较复杂数据集（如 Tiny ImageNet）上表现优于传统 DFA 算法，更加接近 BP 算法的精度。

也有一些工作尝试把直接反馈对齐类算法用于训练脉冲神经网络[48-50]，并取得了成功。这样做的原因主要有两个：1.SNN 难以梯度反向传播的困境，和 DFA 算法免于误差反向传播的特点正好不谋而合，所以两者的结合是自然的。2.DFA 方法不需要维护与前向路径严格匹配的反馈权重，从而简化了模型设计

和实现，在一些神经形态硬件上，这种不依赖精确权重传输的方法更易于实现和部署，有利于在低功耗、高并行性的神经形态硬件上进行高效训练，比较契合SNN 的硬件部署场景。

DFA 算法及其衍生的变体算法作为 BP 算法的替代品，是迈向生物学可信的类脑算法的重要一步。它摆脱了严格的权重对称要求，相比于反向传播算法更容易部署在神经形态硬件上；它不受更新锁定的限制，误差信号不需要经过传统的层层传递，而是直接从输出层反馈到各层，这种解耦的方式在一定程度上实现训练的并行化；它还提高了算法在生物学上的合理性，也为神经网络训练方法的多样化提供了全新的思路。

2.2.2　分层局部学习相关算法

除了反馈对齐类算法，还有其他替代 BP 算法的可行方案，这些方法不需要全局误差信号，而是让深层网络的每一层进行局部信息处理和局部权重更新。虽然这些方法各自之间也有显著差异，但鉴于它们的共同特性，在这里还是将它们归结在一起，统称为分层局部学习（Layer-wise Local Learning）。广义的分层局部学习可以包含以下研究成果：分层预训练、目标传播、赫布学习、平衡传播（Equilibrium Propagation）和两次前向传播（Forward-Forward Algorithm）等，下面主要对前三者做介绍。

分层局部学习方法可以追溯到 Hinton 等人于 2006 年在深度信念网络（Deep Belief Network, DBN）中提出的无监督逐层预训练技术。DBN 是一种由多层受限玻尔兹曼（Restricted Boltzmann Machine, RBM）堆叠而成的网络模型，其核心思想是通过无监督的逐层预训练和有监督的微调，解决深层神经网络训练中的梯度消失问题。具体而言，逐层预训练发挥的作用是：逐层贪心训练每一个 RBM 模块，使用对比散度方法调整参数值，目标是重构输入数据，让 RBM 学习到输入数据的有效特征表示，这种无监督分层预训练为接下来的有监督微调优化做铺垫，提升了网络性能。

目标传播（Target Propagation, TP）是一种旨在替代传统反向传播的局部学习方法，其核心思想在于为每一层构建一个局部目标来指导权重更新。该算法的主要步骤是：首先进行正常的前向传播，然后利用每一层的近似逆映射将上层的目标转化为本层的局部目标，接着计算当前层输出和这个局部目标的差值作为

局部误差信号，最后在该信号的指导下进行权重更新。由于近似逆映射往往不是完美的，这会导致局部目标的精确度有偏差，为了解决这个问题，需要引入差分项进行校正，这便是 Bengio 等人于 2015 年提出的差分目标传播（Difference Target Propagation, DTP）[51]，见下图2-5。

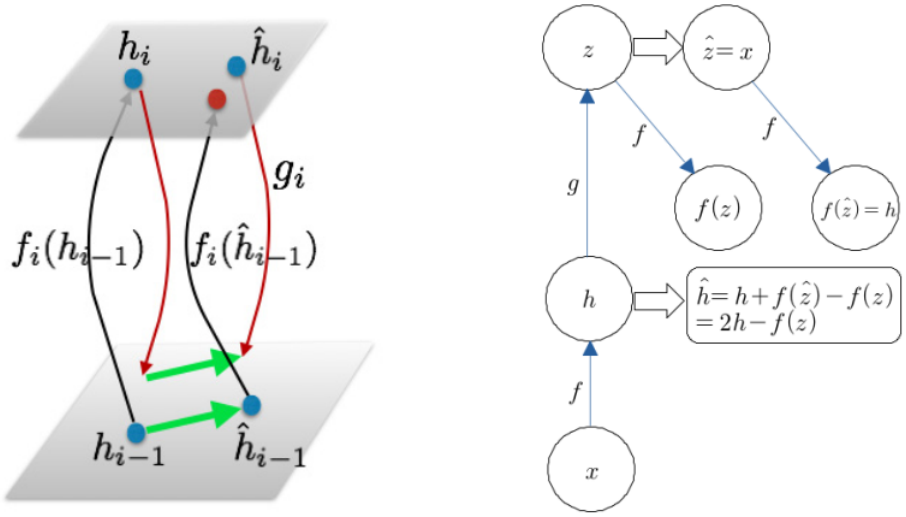

图 2-5 差分目标传播原理示意图[51]

下面是 DTP 算法的主要符号公式，对于输出层，设定全局目标为 $\tilde{h}_L$，对于隐藏层 l，需要计算局部目标 $\tilde{h}_l$，这个过程需要先计算正向映射 f_l 近似逆映射 g_l，即：

$$g_l(f_l(h_{l-1})) \approx h_{l-1}, \tag{2-35}$$

$$\tilde{h}_{l-1} = g_l(\tilde{h}_l). \tag{2-36}$$

在原始的 TP 中，每一层的局部误差信号由下式得出：

$$\begin{aligned} e_{l-1} &= \tilde{h}_{l-1} - h_{l-1} \\ &= g_l(\tilde{h}_l) - h_{l-1}. \end{aligned} \tag{2-37}$$

而在引入了差分项的 DTP 中，有如下差分公式：

$$\tilde{h}_{l-1} = h_{l-1} + g_l(\tilde{h}_l) - g_l(h_l). \tag{2-38}$$

因此 DTP 中的局部误差为：

$$\begin{aligned} e_{l-1} &= h_{l-1} + g_l(\widetilde{h}_l) - g_l(h_l) - h_{l-1} \\ &= g_l(\widetilde{h}_l) - g_l(h_l). \end{aligned} \tag{2-39}$$

差分校正使得 DTP 在面对不完美逆映射时能获得更稳定和准确的局部目标，从而相比于 TP 提高整个网络训练的效果。目标传播和差分目标传播均试图将全局 BP 的信用分配问题分解为每层独立的局部误差问题，弥补了全局梯度回传不稳定的问题。

赫布学习（Hebbian Learning）是一种经典的局部学习规则，其基本思想为：每个突触权重的更新仅依赖于其连接的两个神经元的信息，如果突触前神经元反复地参与激活突触后神经元，那么它们之间的突触联系就会被增强。最基本的赫布学习规则表达式如下：

$$\Delta W_{ij} = \eta \cdot x_i \cdot y_j \tag{2-40}$$

其中，x_i 表示第 i 个突触前神经元的活动，y_j 表示第 j 个突触后神经元的活动，η 为学习率。上式表明，当突触前后神经元激活状态一致时，突触连接会加强；当突触前后神经元活动有一个为 0 时，突触权重不发生更新。

在原始赫布学习规则的基础上，衍生出了一些变体和改进，例如 Oja 规则（Oja's Rule），该变体是为了解决原始赫布学习权重容易无限增长的问题，因此增添了一个权重衰减项，其公式为：

$$\Delta W_{ij} = \eta \cdot (x_i \cdot y_j - y_j^2 \cdot W_{ij}) \tag{2-41}$$

值得注意的是，前文介绍的 STDP 也是赫布学习的一种变体，它是一种对时间敏感的更精细的赫布学习机制。此外，还有三因素赫布学习规则（Three-factor Hebbian Learning Rule），即在传统的“突触前 + 突触后”双因子基础上，加入了一个调制因子作为第三个学习触发条件，例如多巴胺水平、奖励信号等，使得学习规则相较于原始赫布学习能服务于更复杂的任务。其可以表示为如下形式：

$$\Delta W = F(\text{pre}, \text{post}, \text{third-factor}, W) \tag{2-42}$$

2.2.3 信息瓶颈学习框架

在前述分层局部学习方法小节中，通过分层预训练、目标传播以及赫布学习等技术可以实现各层独立的局部更新，从而摆脱了全局误差反向传播的限制。然而，这些方法是从如何使用局部损失局部地进行参数优化的角度考虑的，缺少对不同层之间的表示关系和不同层学习到的特征的差异的考量。而信息论的知识可以弥补上述不足，近年来，信息论被越来越多地用于神经网络的的训练和可解释性分析中，其中最具代表性的方法当属信息瓶颈。

信息瓶颈（Information Bottleneck, IB）[52]是信息论中的一个重要概念，最早由 Tishby 等人在 2000 年提出。信息瓶颈的核心思想是通过压缩输入数据，去除不必要的冗余信息，同时保留其中对预测任务最相关的信息。为了介绍信息瓶颈，先从信息论中的一些最基本的概念讲起。

信息熵指一个事件的发生所提供的信息量，它起到了度量一个事件不确定性的作用。假设一个事件 X 的发生概率为 p，随机变量 X 服从概率分布 $p(x)$，则该事件的自信息量为 $I(x) = -\log p(x)$。概率越小，其意外性越高，自信息量也越大。基于自信息量，信息熵可以被定义为随机变量所有可能取值的信息量的平均值（或者说连续随机变量的信息量的期望）。对于随机变量 X，$p(x)$ 表示 X 取值为 x 的概率，那么信息熵定义如下：

$$H(X) = -\sum_{x} p(x)\log p(x). \tag{2-43}$$

由于生活中经常出现一个事件的发生会影响另一个事件发生概率变化的情况，接下来考虑两个随机变量 X 和 Y 的条件信息熵。条件信息熵表示的是在已知随机变量 Y 的条件下，随机变量 X 的不确定程度，可以简称为条件熵，表达式如下：

$$\begin{aligned} H(X|Y) &= -\sum_{y} p(y) \sum_{x} p(x|y)\log p(x|y) \\ &= -\sum_{x,y} p(x,y)\log p(x|y). \end{aligned} \tag{2-44}$$

互信息是两个随机变量之间的相互依赖关系的度量，它是由前述信息熵和条件

熵作差定义的：

$$I(X;Y) = H(X) - H(X|Y). \tag{2-45}$$

也可以等价地表示为：

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \tag{2-46}$$

互信息衡量了知道 Y 后减少了 X 的不确定性的多少，即从 Y 获取的关于 X 的信息量的多少。互信息越大，说明两者之间的依赖关系越强；若两个随机变量独立，则 $I(X;Y) = 0$。

接下来介绍和信息瓶颈密切相关的率失真理论。率失真理论（Rate-Distortion Theory）是指在允许信息一定失真的情况下，如何用最低的比特率表示信息。该理论可以建模成：在给定失真度 D 的条件下，找到最小的比特率 R 来表示信息。

$$R(D) = \min_{p(\hat{x}|x)} I(X;\hat{X}) \; s.t. \; \mathbb{E}[d(X,\hat{X})] \leqslant D. \tag{2-47}$$

其中 $\hat{X}$ 是原始信息的重建信息，$d(X,\hat{X})$ 是失真度量，D 是可接受的最大失真度。当给定失真度 D 越大，所需的比特率 R 就越小; 相反，当失真度越小，用 $\hat{X}$ 表示 X 会越精确，但所需比特率会更大。

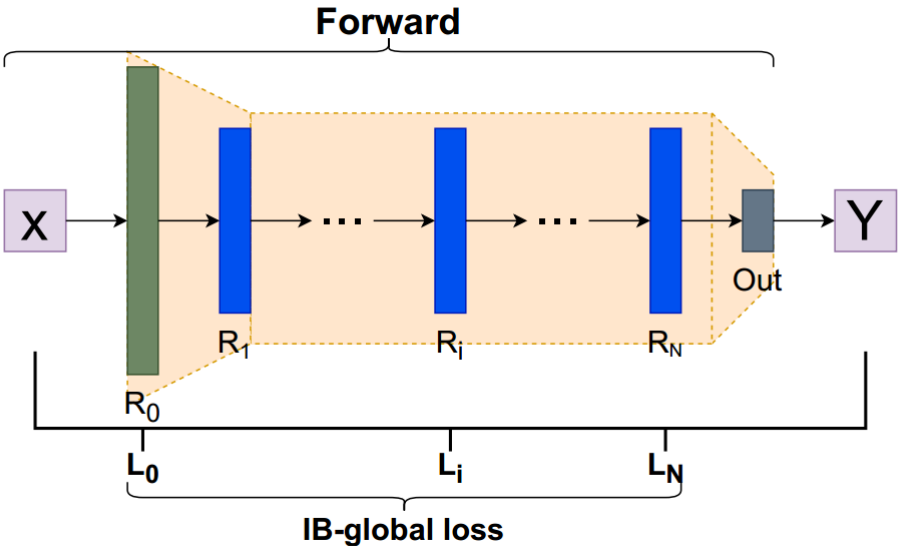

图 2-6　信息瓶颈理论示意图[53]

有了以上的基础概念，进一步可以引入信息瓶颈（Information Bottleneck, IB）：信息瓶颈方法旨在从输入 X 中提取出对预测目标 Y 最为重要的信息，同时尽量去除与 Y 无关的冗余信息，其本质是在隐藏层 Z 上达到一种信息压缩与信息保留之间的平衡，整个过程由图2-6呈现。信息瓶颈的目标表达式为：

$$\min_{p(z|x)} I(X;Z) - \beta I(Z;Y). \tag{2-48}$$

其中 $I(X;Z)$ 表示输入 X 和中间特征表示 Z 之间的互信息，$I(Z;Y)$ 表示中间特征表示 Z 和输出 Y 之间的互信息，前者起到压缩输入数据的作用，后者保留对输出的预测能力，而 β 为权衡系数，在信息压缩和信息保留之间取得平衡。信息瓶颈的目的在于压缩掉与任务无关的冗余信息，同时保留与最终输出最密切相关的信息。信息瓶颈技术在神经网络的可解释性和优化领域发挥了一定作用：有理论认为深度神经网络的训练过程可以拆解为信息拟合和信息压缩两部分，和信息瓶颈的思想很契合。

信息瓶颈可以作为局部损失函数来为网络中间层提供一种约束，从而提升网络的泛化能力，为设计新的局部损失函数提供了启发。已有一些工作尝试将信息瓶颈用于脉冲神经网络的学习[53]，证实了信息瓶颈理论可以替代 BP 算法有效训练脉冲神经网络。

2.3 本章小结

本章节主要介绍后续工作用到的基础知识，或者涉及的相关工作，主要分为两部分：脉冲神经网络的介绍、非 BP 局部学习相关技术的介绍。在脉冲神经网络的介绍小节中，回顾了脉冲神经元模型、脉冲编码和三种主流的脉冲神经网络训练算法的基础知识。在非 BP 局部学习相关技术小节中，主要讨论了包括直接反馈对齐、差分目标传播、信息瓶颈、生物启发的赫布学习等方法，它们都是全局误差传播 BP 算法的替代算法，这些算法一定程度上拥有生物学可解释性、并行化、对神经形态硬件友好的特点，适合与 SNN 相结合。本章节的介绍为后续第三、四章奠定了理论基础。

第三章　脉冲神经网络混合损失局部学习方法

本章提出了一种用局部损失分层训练脉冲神经网络的模型，并采用多种不同原理、类型的局部损失函数的结合，实验出性能最好的组合搭配。作为非 BP 式算法，在常规图像分类数据集上取得超越 BP 式替代梯度下降法的性能。

3.1　背景和动机

无论在 ANN 还是 SNN 中，用于分类的神经网络通常使用全局交叉熵损失进行训练，预测误差从输出层逐层反向传播到隐藏层，也就是通常所说的 BP 算法的方式。然而，BP 算法被认为是生物学不可信的，具体而言有以下几个问题：

关于 BP 的生物不可信性的其中一个的方面是“权重传输”问题（Weight Transport Problem），即在误差信号的反向传播期间需要获知前向传播的权重值的依赖问题。在 BP 算法中，每层的权重更新项由输出损失函数的梯度向量确定，该梯度是从输出层到输入层顺序计算的，在反向传输的计算过程中需要使用相同的前向权重，这种前向和后向权重矩阵的完全对称结构在生物神经网络中是没有被观察的，大脑不会形成对称的反向连接。因此“权重传输”问题也可以被称为“权重对称”问题（Weight Symmetry Problem）。

BP 的生物不可信性的另一个的表现是“更新锁定”问题（Update Locking Problem），即在前向和后向传递完成之前，隐藏层权重不能及时更新。这种后向锁定防止了权重更新的并行化，阻止了跨层的并行学习，这在生物学上是不可信的。且从工程学的角度来看，BP 算法难以执行同步计算，与大规模的模型并行计算不兼容，计算效率低下。

此外，BP 的生物不可信性还表现为以下方面：反向传播过程中，网络活动被冻结。相反，在生物大脑中神经活动在可塑性变化期间不被冻结，并且通过反馈连接传播的信号可以同时影响前向传播的神经活动，从而导致它们的增强或抑制。信号传输问题，BP 算法的参数更新取决于所有的下游神经元节点，而生

物突触仅依赖与它们相连接的前后神经元的局部信号进行学习。BP 算法需要存储所有中间激活和梯度，内存占用巨大，这在生物神经网络中是不现实的。

已经有些工作尝试通过引入随机后向权重来避免上述权重传输、更新锁定等问题，但这种做法在较大数据集（如 ImageNet）上的扩展性较差。另一方面，基于权重扰动的方法直接将损失信号发送回权重连接，因此不需要任何后向权重。在前向传递中，网络向突触连接添加一个轻微的扰动，然后权重更新乘以损失的负变化，权重扰动以前被认为是 BP 的生物学上合理的替代方案。但是缺点也很明显，扰动法采样效率低，需要进行多次前向传播达到近似梯度的效果；扰动法收敛速度也慢，需要更多的迭代才能达到和梯度计算相同的优化效果。

使用局部损失可能是迈向更生物合理的深度学习的一步，因为全局错误不必传播回隐藏层。全局目标可以直接投影回隐藏层。局部损失函数可以使训练更快，记忆效率更高，更并行，生物学上更合理。

在本工作中，将会说明通过对具有局部生成误差的隐藏层进行逐层训练可以避免更新锁定和权重传输等问题。局部损失函数不依赖于全局生成误差，梯度不反向传播到之前的隐藏层，并且隐藏层权重可以在向前传递期间更新。当隐藏层的权重已经更新时，梯度和激活不再需要保存在内存中。这降低了训练深度脉冲神经网络时的内存需求。虽然分别训练了所有层，但本地生成的损失使得能够一次一个地对训练层进行更精确的训练，这可以进一步减少存储器占用，并且还减少训练时间。

3.2　基于分层局部混合损失的脉冲神经网络训练算法设计

3.2.1　分层辅助分类器和预测损失

为了方便接下来的讨论便利和符号的一致性，首先确定本工作任务场景下的 SNN 模型的选取和公式定义。本工作选择 LIF 神经元模型，采用直接编码的脉冲编码方式，在膜电压重置时选择软重置的方式。本工作中的 SNN 模型表达式如下：

$$u_i^l[t] = e^{-1/\tau}(u_i^l[t-1] - v_{th}s_i^l[t-1]) + \sum_{j=1}^{N_{l-1}} w_{ij}^l s_j^{l-1}[t], \tag{3-1}$$

$$s_i^l[t]=\begin{cases}1 & \text{if } u_i^l[t]\geqslant v_{th},\\ 0 & \text{otherwise .}\end{cases} \tag{3-2}$$

其中，$u_i^l[t]$ 和 $s_i^l[t]$ 分别表示在时间步 t 时第 l 层中的神经元 i 的膜电位和可能发放的脉冲，τ 是时长常数，$e^{-1/\tau}$ 起到了衰减率的效果，w_{ij}^l 是从前一层中的第 j 个神经元到当前层中的第 i 个神经元的权重连接，v_{th} 是神经元的脉冲发放阈值，减去 $v_{th}s_i^l[t-1]$ 这一项体现的是膜电压的软重置方式。

在本工作的分层局部学习方案中，每个隐藏层都会有自己的损失函数以产生局部误差信号，指导网络权重更新。具体设计如下：脉冲神经网络中的每个层会接受 T 个时间步的输入，每个时间步都有概率生成脉冲序列，脉冲序列中包含从这一层的输入提取的特征信息。这些特征信息会被后续层继续提炼新的特征，最终在输出层做出对不同类别标签的预测。那么，这些脉冲序列特征在当前层理应蕴含对真实标签的一定预测能力。然而，SNN 当前层的输出脉冲序列的张量维度可能和真实标签的维度不一致，无法直接对两者计算误差。

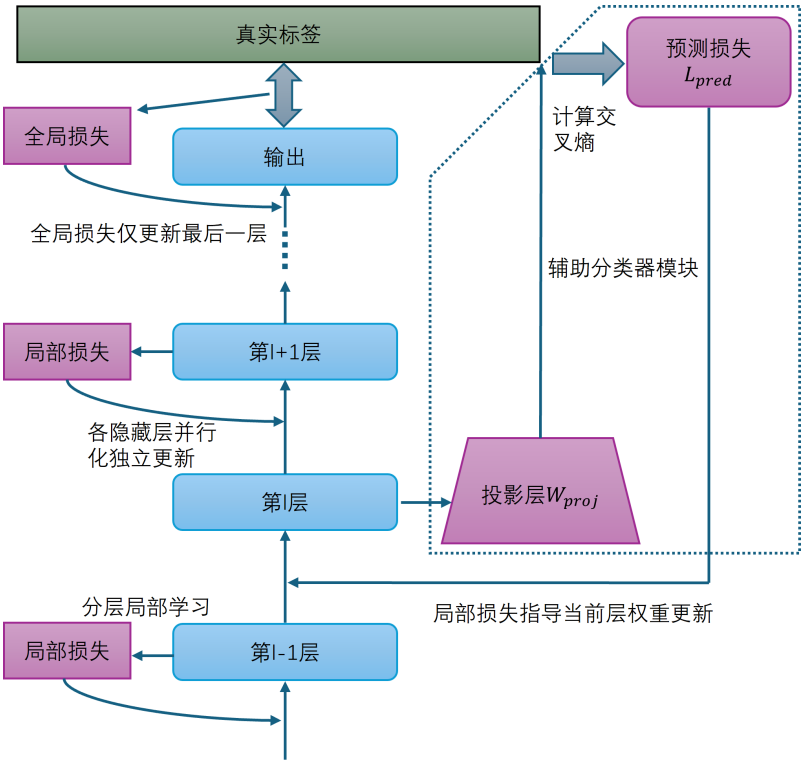

图 3-1　SNN 分层辅助分类器和局部预测损失示意图

基于以上考量，可以为脉冲神经网络的每一隐藏层增添一个辅助分类器（Auxiliary Classifier），如图3-1的虚线框出部分，也可以叫辅助分类层，因为

它的形式就是一个简单的线性映射。该辅助分类器把当前层的输出脉冲序列作为输入，映射到对整个网络最终标签的预测。将辅助分类器的输出预测和真实标签计算交叉熵损失，然后对辅助分类器进行单层的梯度反传，进而为脉冲神经网络的当前层提供了误差指导信息，整个计算更新流程如图3-1右半部分所示。辅助分类器的形式如下：

$$F(s^l[t]) = W^l_{proj} Flatten(s^l[t]) = y^l[t]. \tag{3-3}$$

其中，$F(\cdot)$ 是一个映射函数，因为辅助分类层本质上就是一个当前层脉冲输出到最终标签的映射。映射函数由两部分复合而成：$Flatten(\cdot)$ ：$\mathbb{R}^{B\times C\times H\times W} \rightarrow \mathbb{R}^{B\times(C\cdot H\cdot W)}$ 是一个展平函数，它能让某一隐藏层任意形状的特征图改变维度以适配线性映射的输入；W^l_{proj} ：$\mathbb{R}^{(C\cdot H\cdot W)\times N}$ 是一个权重矩阵，其高等于分类任务类别数 N，其宽等于当前层输出经过展平操作后的特征维度。$y^l[t]$ 是当前层辅助分类器当前时间步做出的对最终分类标签的预测。那么，就可以顺理成章地计算预测标签 $y^l[t]$ 和真实标签 y 的交叉熵损失，这样就得到了 SNN 当前隐藏层的辅助分类器的预测损失（Predict Loss）函数：

$$L_{pred} = \sum_{t=1}^{T} CrossEntropy(y,\ y^l[t]). \tag{3-4}$$

根据神经元动力学，每个时间步的神经元状态和过去时间步的神经元状态可以共同影响损失 L，这种复杂的时间依赖关系会降低训练效率，需要 BPTT（Backpropagation Through Time）算法处理。为了简单起见，此处忽略时间依赖性，转而采用时间局部性，即分别对每个时间步 t，可以对辅助分类器应用单层梯度下降来评估损失函数相对于辅助分类层和隐藏层两者的权重的导数，如下所示：

$$\Delta W^l_{proj}[t] = -\eta \frac{\partial L_{pred}[t]}{\partial y^l[t]} \frac{\partial y^l[t]}{\partial W^l_{proj}}, \tag{3-5}$$

$$\Delta W^l[t] = -\eta \frac{\partial L_{pred}[t]}{\partial y^l[t]} \frac{\partial y^l[t]}{\partial s^l[t]} \frac{\partial s^l[t]}{\partial u^l[t]} \frac{\partial u^l[t]}{\partial W^l}. \tag{3-6}$$

辅助分类器可以使用真实标签在空间意义上局部训练每个隐藏层，各层训练互

不打扰，可以做到并行运算；它还能做到在每个时间步实时连续调整网络权重。辅助分类器仅在训练阶段发挥作用，推理阶段不会启用它。

3.2.2 相似度匹配损失

分层辅助分类器可以有效地训练脉冲神经网络，然而辅助分类器中的单个映射层的学习能力有限，它提供的误差信号不如全局误差传播的那么准确，实验结果也表明了这一点：只靠辅助分类器训练的脉冲神经网络的准确率逊色于同结构的替代梯度下降法训练的脉冲神经网络。这促使本研究寻找其他的思路，相似度匹配损失便是一个可行的代替思路。

相似度匹配损失（Similarity Matching Loss）的想法源自于对样本数据内部的相似性结构的利用，之前的一些工作与此有紧密关系，包括度量学习、表征相似性分析（Representational Similarity Analysis，RSA）、无监督聚类、线性判别分析（Linear Discriminant Analysis，LDA）等。例如在神经科学中的表征相似性分析，通过计算不同刺激引发的神经活动模式之间的相关性，了解不同类别刺激在神经信号中的表征差异，从而评估神经表征是否稳定。再如机器学习中的线性判别分析，寻找这样一个投影空间，使得原本样本中的同类在投影空间中的投影点尽可能接近，而不同类的投影点尽可能远。即扩大组间方差，缩小组内方差的方式，达到降维和分类的目的。

基于前述相关研究的启发，有如下思考：样本之间的相对位置关系隐含着对标签分类任务有用的信息，通过相似性度量可以揭示样本空间的结构。可以利用这一点，通过扩大样本中间特征的类间差异，缩小样本中间特征的类内差异，从而让中间特征的表征能力、判别性得到增强。

考虑到标签能够直观表明哪些数据点属于同一类别，那么可以构造一个标签相似度矩阵——通常用 one-hot 编码的标签矩阵 $Y = [y_1, \cdots, y_n]$ 构造得到，其中 y_i 是样本 i 的 one-hot 标签，矩阵中的元素在同类别数据间为 1、不同类别间为 0。同样地，对于一个批次的样本在隐藏层 l 的中间特征表示 $S^l = [s^l_{:,1}, \cdots, s^l_{:,n}]$，其中 $s^l_{:,i}$ 为当前 batch 中第 i 个样本在当前隐藏层 l 的激活。这里没有直接使用 s^l_i 来表示前述的量，因为 s^l_i 已被用来表示 SNN 当前隐藏层 l 中的第 i 个神经元的脉冲值，为了避免混淆，所以此处采用 $s^l_{:,i}$ 的形式以示区分。于是可以定义下

述优化目标：

$$\min_{\theta} ||Sim(E_{\theta}(S^{l})) - Sim(Y)||_{F}^{2}. \tag{3-7}$$

其中 $Sim(\cdot)$ 是度量样本之间相近程度的相似度距离；标签 Y 是固定的，因此标签的相似度矩阵 $Sim(Y)$ 也是固定的；E_{θ} 表示脉冲编码层，将隐藏层脉冲特征 S 映射为可学习的嵌入表示，θ 是可调整参数，调整它可以使上式的值减小；$||\cdot||_{F}$ 表示 Frobenius 范数，它是一种衡量矩阵大小的矩阵范数。

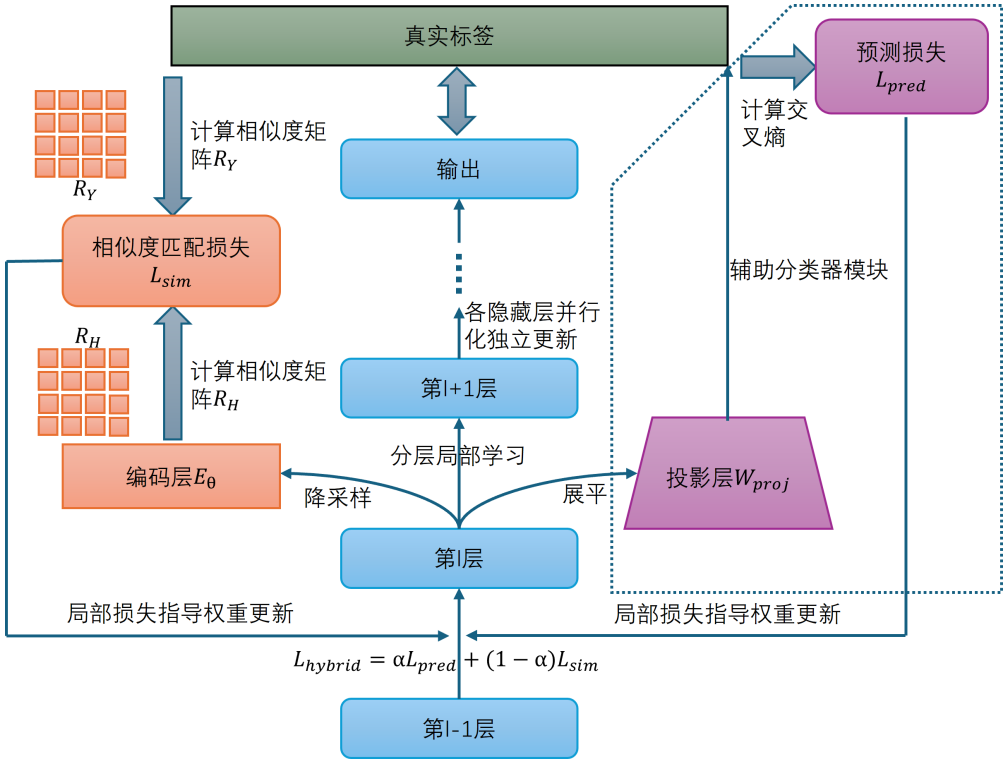

图 3-2　SNN 混合损失分层局部学习原理图

上述优化目标可以使得标签 Y 的样本间的相似度关系矩阵，和隐藏层脉冲特征 S 的样本间的相似度关系矩阵更加接近，也就是说强制网络学习到的表示在相似度空间结构上和标签蕴藏的相似度空间结构一致，从而使得使得隐层特征具备更明显的类别区分性。因此可以自然地定义相似度匹配损失：

$$L_{sim} = \sum_{t=1}^{T} ||Sim(E_{\theta}(S^{l}[t])) - Sim(Y))||_{F}^{2}. \tag{3-8}$$

$Sim(\cdot)$ 作为一种相似度算子，这里选用余弦相似度，即 Cosine 相似度。$Sim(Y)$

便是标签的余弦相似度矩阵，其元素 s_{ij} 为：

$$s_{ij} = s_{ji} = \frac{y_i y_j}{||y_i||_2 ||y_j||_2}. \tag{3-9}$$

这样就定义好了相似度匹配损失，如图3-2左半部分所示，它可以让网络训练过程的中间特征尽可能和最终标签保持一致的空间位置结构，从而增强网络对最终分类目标的判别能力。

3.2.3 其他类型可选损失

实验表明，无论是辅助分类器中的预测损失 L_{pred}，还是相似度匹配损失 L_{sim}，它们各自单独作为局部损失分层训练脉冲神经网络时，性能都没能超越全局误差反向传播的替代梯度下降法。这促使本研究继续思考如何进一步提升分层局部学习的性能。

首先可以想到尝试其他类型的分层局部损失，比如自编码器的重构损失（Reconstruction Loss）。具体做法是，为网络的某一隐藏层附加上一个解码器，让解码器重构当前层的输入，这能约束网络每一层的学习保存尽量多有用的输入信息。网络的正向传播阶段已经相当于一个局部编码器：

$$h_l = F_l(h_{l-1}). \tag{3-10}$$

其中 h_{l-1} 是前一层的输出，$F_l(\cdot)$ 表示这一层的正向变换，也可视为一个编码器。记 $G_l(\cdot)$ 为每一层单独附加的解码器，它目标是让输出 h_l 重构成输入 h_{l-1}。于是，这层的重构损失表达式可以写成：

$$\begin{aligned} L_{recon} &= ||G_l(h_l) - h_{l-1}||_2^2 \\ &= ||G_l(F_l(h_{l-1})) - h_{l-1}||_2^2. \end{aligned} \tag{3-11}$$

除此之外，还可以联想到了对比学习损失（Contrastive Loss），也可以作为本算法中的局部损失。简单地讲，对比学习损失是一类用于拉近同类样本、推远不同类样本的损失函数，包括基于样本对的经典对比损失和基于 softmax 的多样

本对比损失（如 InfoNCE Loss 和 SupCon Loss）。这里仅使用最经典的对比损失，具体做法是对网络每一隐藏层的某一批次的样本，计算两两样本间的脉冲发放率的欧几里得距离，构建欧式距离矩阵 $\boldsymbol{D}$，其中矩阵元素 $\boldsymbol{D}_{i,j}$ 表示样本 i 和 j 之间的欧几里得距离。定义好距离矩阵 $\boldsymbol{D}$ 后，可以得出以下对比损失函数表达式：

$$L_{contrastive} = \sum_{i,j}[y_{i,j}D_{i,j}^2 + (1-y_{i,j})\max(0, m - D_{i,j})^2] \tag{3-12}$$

其中，$y_{i,j} \in \{0,1\}$，若 i 和 j 属于同一类，则 $y_{ij}=1$；若 i 和 j 属于不同类，则 $y_{ij}=0$；m 为 margin，表示异类样本间的最小距离要求。对于正样本对（$y_{i,j}=1$），上述对比损失的效果是最小化同类样本的距离 $D_{i,j}^2$，让它们在特征空间中靠近；对于负样本对（$y_{i,j}=0$），效果为最大化不同类样本的距离，至少达到 m。

3.2.4　多损失组合方案

然而，以上任意一种单独的损失类型作为局部损失训练脉冲神经网络，在实验中也无法超越常规的全局误差反向传播方法。不妨尝试改变思路，转而对不同损失函数做组合，也即混合损失（Hybrid Loss）方案。特别地，该方案按照 loss 尺度一致原则，设定加权系数 $\alpha,\beta,\gamma,\lambda$ 和 δ，按照系数对上述几种局部损失做加权组合，使得各不同损失分量在混合损失中尺度大致相当：

$$\begin{aligned} L_{predsim} &= \alpha L_{pred} + (1-\alpha)L_{sim} \\ &= \alpha \sum_{t=1}^{T} CrossEntropy(y,\ y^l[t]) + (1-\alpha)\sum_{t=1}^{T} ||Sim(E_\theta(S^l[t])) - Sim(Y)||_F^2. \end{aligned} \tag{3-13}$$

$$\begin{aligned} L_{predrecon} &= \beta L_{pred} + (1-\beta)L_{recon} \\ &= \beta \sum_{t=1}^{T} CrossEntropy(y,\ y^l[t]) + (1-\beta)||G_l(F_l(h_{l-1})) - h_{l-1}||_2^2. \end{aligned} \tag{3-14}$$

$$\begin{aligned} L_{predcontrast} &= \gamma L_{pred} + (1-\gamma)L_{contrast} \\ &= \gamma \sum_{t=1}^{T} CrossEntropy(y,\ y^l[t]) \\ &\quad + (1-\gamma)\sum_{i,j}[y_{i,j}D_{i,j}^2 + (1-y_{i,j})\max(0, 1 - D_{i,j})^2]. \end{aligned} \tag{3-15}$$

$$\begin{aligned} L_{triple} =& \lambda L_{pred} + \delta L_{sim} + (1-\lambda-\delta)L_{contrast} \\ =& \lambda \sum_{t=1}^{T} CrossEntropy(y,\ y^l[t]) + \delta \sum_{t=1}^{T} ||Sim(E_\theta(S^l[t])) - Sim(Y)||_F^2 \\ &+ (1-\lambda-\delta) \sum_{i,j} [y_{i,j} D_{i,j}^2 + (1-y_{i,j}) \max(0, 1-D_{i,j})^2]. \end{aligned} \tag{3-16}$$

以上公式没有列举出所有组合情形，只是列举了有代表性的几个。对于混合损失中的 L_{pred} 和 L_{sim} 分量，含有时间项 t，它们是实时计算的；而对于混合损失中的 L_{recon} 和 $L_{contrast}$ 分量，是在所有时间步结束后只进行单次计算，这里面有计算开销层面的考量：因为对比损失 $L_{contrast}$ 中，假如每个时间步计算一次特征的欧氏距离矩阵 D，计算复杂度是比较大的，所以采用在最终时间步对脉冲发放率仅计算一次度量矩阵的方式。

通过观察可以发现，对于损失 $L_{predsim}$，它的两个分量都是在时间维度求和得来的，所以可以做到对每个时刻进行实时在线更新，不需要存储整个脉冲序列的中间状态，这有利于减少内存的负担。损失 $L_{predsim}$ 在 t 时刻的实时局部权重更新公式为：

$$\Delta W^l[t] = -\eta \frac{\partial L_{predsim}[t]}{\partial s^l[t]} \frac{\partial s^l[t]}{\partial u^l[t]} \frac{\partial u^l[t]}{\partial W^l}. \tag{3-17}$$

在计算混合损失损失 $L_{predsim}$ 时，如果对存储资源的要求进行一定程度的放宽，且需要进一步降低计算代价，可以将实时更新的方式改为最终时间步更新。为此，需要改变预测损失和相似度匹配损失的定义，不再是逐时间步计算并求和的形式，而是对所有时间步的脉冲序列 $s^l[t], t = 1, 2, \cdots, T$，计算脉冲发放率 r^l：

$$r^l = \frac{1}{T} \sum_{t=1}^{T} s^l[t]. \tag{3-18}$$

在最终时间步对脉冲发放率 r^l，分别计算预测损失和相似度匹配损失，这两种损失相应地变为：

$$L_{pred} = CrossEntropy(y,\ W_{proj}^l Flatten(r^l)). \tag{3-19}$$

$$L_{sim} = ||Sim(E_\theta(r^l) - Sim(Y))||_F^2. \tag{3-20}$$

不同类型的局部损失函数的加权组合，相比单独的某种局部损失函数，性能会更好。这是因为单独某种局部损失函数从单一的角度对隐藏层进行优化：辅助分类器的预测损失是从使得该层的输出尽可能地对目标标签有正确的判别区分能力的角度对网络进行优化；相似性匹配损失是从使得该层不同样本在样本空间中的相似度结构尽量和目标标签的不同样本的相似度结构保持一致的角度进行优化；而重构损失是从使得该层尽可能多地保存原始输入中的有用信息的角度进行优化的。不同角度的优化目标的结合很有可能优势互补、扬长避短。

算法3.1展示了选择实时更新或最终时间步更新情形下，混合局部损失 $L_{predsim}$ 监督脉冲神经网络的某一隐藏层的权重更新的算法流程：

算法 3.1 SNN 分层局部损失算法使用混合损失 $L_{predsim}$ 时隐藏层 l 的更新

输入: 前一层的脉冲 S^{l-1}，当前层膜电压初始化值 U^l，需要学习的权重 W^l
参数: 学习率 η，损失比例系数 α，时间步 T
输出: 使用混合损失更新后的权重 W^l
for $t = 1$ to T **do**
    根据公式3-1和3-2计算 t 时刻的膜电压 $U^l[t]$ 和脉冲值 $S^l[t]$
    if 实时更新 **then**
        根据公式3-4和3-8分别计算 t 时刻的局部损失 $L_{pred}[t]$ 和 $L_{sim}[t]$
        用系数 α 加权得到混合损失 $L_{predsim}[t] = \alpha L_{pred}[t] + (1-\alpha)L_{sim}[t]$
        权重更新：$W^l[t] \leftarrow W^l[t-1] - \eta \frac{\partial L_{predsim}[t]}{\partial W^l}$
    end if
end for
if 最终时间步更新 **then**
    根据公式3-19和3-20分别计算基于脉冲发放率 r^l 的局部损失 L_{pred} 和 L_{sim}
    用系数 α 加权得到混合损失 $L_{predsim}$
    权重更新：$W^l \leftarrow W^l - \eta \frac{\partial L_{predsim}}{\partial W^l}$
end if

3.3　实验和分析

本节根据 3.2 小节中介绍的脉冲神经网络分层局部损失训练算法的技术原理，在常规图像分类数据集上开展实验，以验证算法的有效性和优越性。

3.3.1　数据集介绍

本工作在常见的图像分类数据集 MNIST、Fashion-MNIST、CIFAR-10 上进行实验。其中 MNIST 数据集包含手写数字（0-9），共 10 类，尺寸为 28×28 像

素的灰度图像，训练集 60000 张图像，测试集 10000 张图像，是图像分类领域经典的入门数据集。

Fashion-MNIST 数据集包含各种服装和鞋帽的图片，包括 T 恤、牛仔裤、套衫、裙子、外套、凉鞋、衬衫、运动鞋、包、短靴，共 10 类。与 MNIST 相同，为 28×28 像素的灰度图像，数据量也相同，训练集 60000 张图像，测试集 10000 张图像。但相比于 MNIST，Fashion-MNIST 图片内容更复杂，具有更多的纹理和结构信息。

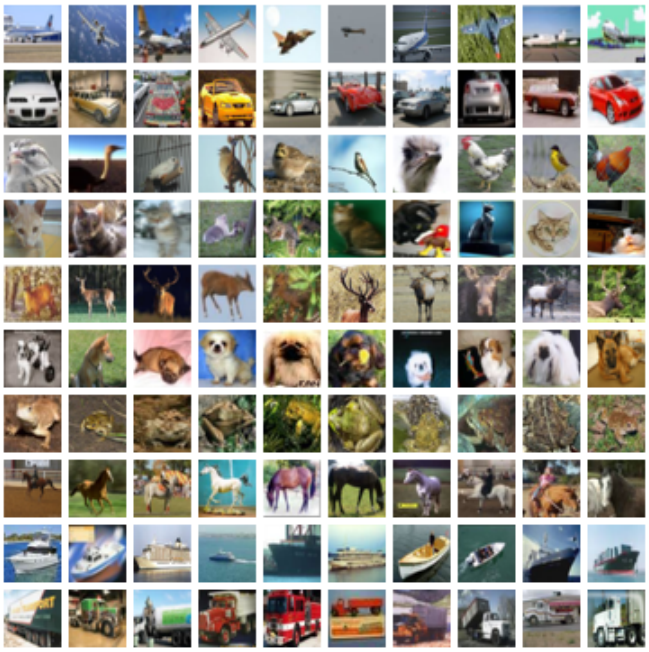

图 3-3 CIFAR-10 数据集

CIFAR-10 数据集包含 10 类物体的彩色图像，如飞机、汽车、鸟、猫、鹿、狗、青蛙、马、船、卡车，如图3-3所示。图像为 32 ×32 像素 3 个 RGB 通道的彩色图像，训练集 50000 张图像，测试集 10000 张图像。CIFAR-10 数据集背景更加复杂、噪声较多，物体也更加多样化，比前两种数据集更有挑战性。

3.3.2 实验设置

本实验采用多层感知机 MLP 结构（即传统的全连接神经网络）和卷积神经网络 CNN 结构的 SNN 测试所提出的算法，其中 MLP 的结构为 784-1024-10。CNN 网络采用 8 层 VGG 结构，主要由 3×3 卷积层（步幅为 1，边缘填充为 1）和 2×2 最大池化层组成，整个网络为 conv 128-conv 256-pool-conv 256-conv 512-

pool-conv 512-pool-conv 512-pool-fc 1024-fc 10。对于 MNIST 数据集，采用 MLP 结构和浅层 CNN 结构的脉冲神经网络；而对 Fashion-MNIST 和 CIFAR-10 数据集，只采用 VGG 结构的网络。

所有实验均设置 batch size=64，采用 ADAM 优化器进行训练。加权因子 β 是根据不同类型损失的绝对大小手动调整的，比如在 $L_{predsim}$ 中，因为根据前文的定义及实际实验中的观测，预测损失和相似度匹配损失的原始大小差了近两个数量级，所以设置 $\beta = 0.01$，让两种损失在总损失中占比相当，避免其中一种损失占主导而掩盖另一种损失的作用的情况发生。

由于本研究提出了多种不同的损失和它们的组合，实验时会对不同的局部损失方案保持网络结构、超参数、丢弃率、学习率、批次大小、训练周期等变量保持一致，来探索最好的损失组合。本实验涉及的所有参数设置详见下表3-1：

表 3-1　SNN 分层局部损失实验网络参数设置

参数名称	设置
优化器	Adam
训练轮数 epoch	100(MNIST),200(Fashion-MNIST),400(CIFAR-10)
初始学习率	0.0005
学习率衰减率	0.25
batch size	64
dropout 率	0.1(MNIST,MLP),0.2,0.25(CIFAR-10)
膜电压衰减率	1/e
脉冲发放阈值	1.0
时间步	10
比例系数	$\alpha = 0.99, \beta = 0.5, \gamma = \lambda = 0.95, \delta = 0.01$
MLP 结构	784-1024-10
VGG 结构	128C3-256C3-P2-256C3-512C3-P2-512C3-P2-512C3-P2-FC1024-FC10

其中，学习率采用阶梯式衰减；轮数 epoch 在相同数据集不同损失组合情况保持一致，跨数据集采用不同的训练轮数；比例系数选取的原则是使不同损失项对总损失贡献相当。

此外，实验时会对数据集进行常规的数据预处理，包括对所有数据集的图像进行归一化处理。对 CIFAR-10 数据集还会做数据增强处理，包括随机裁剪（RandomCrop）和随机水平翻转（RandomHorizontalFlip）两种数据增强手段。

3.3.3 对比实验

MNIST 数据集上的实验，设置初始学习率为 $5e-4$，MLP 结构的网络丢弃率为 0.1，CNN 结构的网络丢弃率为 0.2，训练周期 $epoch=100$。实验结果见下表：

表 3-2 MNIST 数据集实验结果

工作	训练算法	编码方式	网络结构	时间步	测试集准确率
[41]	STDP(unsupervised)	rate code	MLP(784-6400-10)	10	95.0%
[42]	Spatio-temporal BP	rate code	MLP(784-800-10)	10	98.89%
[48]	Global Feedback + STDP	temporal code	MLP(784-800×3-10)	10	98.6%
[31]	STDP-based BP	rate code	CNN	/	97.2%
[34]	DECOLLE(local learning)	/	CNN	10	97.51%
[54]	TSLL(local learning)	direct code	CNN	10	99.35%
[55]	surrogate gradient BP	rate code	CNN	10	99.26%
SNN-HLL	局部损失 $L_{predsim}$	direct code	MLP(784−1024-10)	10	**97.72%**
SNN-HLL	局部损失 $L_{predsim}$	direct code	CNN	10	**99.35%**

在表3-2中，呈现了分层局部混合损失算法 SNN-HLL 和其他常见的脉冲神经网络工作中的算法在 MNIST 数据集上的性能对比。该表中参考对比的工作包括无监督的 STDP 算法[41]、有监督的 surrogate gradient BP 算法[42][55]、有监督的生物学合理的局部学习算法[34][54]等。对于无监督的算法，会在表格中加以标记，没有标记则为有监督算法。对于全连接结构的简单网络，本工作提出的 SNN-HLL 算法效果显著好于无监督的 STDP 算法，在 Diehl 等人的工作[41]中，他们将一个投票层加在网络末尾进行类别分配，相当于常规网络最后的 softmax 层的作用，达到了 95.0% 的准确率。对于卷积结构的网络，替代梯度下降类算法中的经典工作[42]采用时空反向传播算法，在时间和空间两个维度计算并传播替代梯度，达到了 98.89% 的准确率，另一个此类工作[55]达到了 99.26% 的准确率；而和本算法思路同属于分层局部学习范畴的 DECOLLE 算法[34]和 TSLL 算法[54]分别在 MNIST 上达到了 97.51% 和 99.35% 的准确率。以上的这些参照工作均没有超过 SNN-HLL 算法的 99.35% 的准确率，表明了本工作提出的算法对训练脉冲神经网络的有效性。

Fashion-MNIST 数据集上的实验，设置初始学习率为 $5e-4$，VGG8 结构的网络丢弃率为 0.2，训练周期 $epoch=100$，辅助分类器的输入维度为 1024。实验结果见下表：

表 3-3　Fashion-MNIST 数据集实验结果

工作	训练算法	编码方式	网络结构	时间步	测试集准确率
[48]	Global Feedback + STDP	temporal code	MLP	10	89.05%
[32]	Implicit Differentiation	rate code	MLP	5	90.25%
[34]	DECOLLE(local learning)	/	CNN	10	90.75%
[54]	TSLL(local learning)	direct code	CNN	10	92.56%
SNN-HLL	局部损失 $L_{predsim}$	direct code	VGG-like	10	**92.11%**
SNN-HLL	局部损失 $L_{predsim}$	direct code	VGG-like(2x)	10	**93.46%**

在表3-3中，给出了分层局部混合损失算法和其他常见的脉冲神经网络工作中的算法在 Fashion-MNIST 数据集上的性能对比。其中，VGG(2x) 表示将网络中的卷积通道数乘 2。在 Fashion-MNIST 数据集上做实验的工作相对较少，很难找到和本工作中采用的网络层数相近的工作。显而易见，本工作采用的 VGG 结构的网络比参照工作中的浅层 CNN 网络深度更深、性能更好，所以 DECOLLE 和 TSLL 分别只有 90.75% 和 92.56% 的准确率，而本方法取得了 93.36% 的准确率，这是自然的、不足为奇的。

CIFAR-10 数据集上的实验，设置初始学习率为 $5e-4$，VGG8 结构的网络丢弃率为 0.25，训练周期 $epoch=400$，辅助分类器的输入维度为 2048。实验结果见下表：

表 3-4　CIFAR-10 数据集实验结果

工作	训练算法	编码方式	网络结构	时间步	测试集准确率
[42]	Spatio-temporal BP	rate code	VGG8	12	90.53%
[56]	surrogate gradient BP	rate code	VGG9	100	90.45%
[56]	surrogate gradient BP	rate code	ResNet11	100	90.95%
[57]	surrogate gradient BP	rate code	VGG9	25	90.50%
[58]	ANN-to-SNN	temporal code	ResNet20	2048	91.42%
[32]	Implicit Differentiation	rate code	VGG8	30	92.08%
[34]	DECOLLE(local learning)	/	VGG8	10	74.70%
[54]	TSLL(local learning)	direct code	VGG8	10	89.22%
SNN-HLL	局部损失 $L_{predsim}$	direct code	VGG8	10	**90.77%**
SNN-HLL	局部损失 $L_{predsim}$	direct code	VGG8(2x)	10	**91.44%**

在表3-4中，展示了分层局部混合损失算法和其他常见的脉冲神经网络工作中的算法在 CIFAR-10 数据集上的性能对比。CIFAR-10 数据集上的对照工作的实验都采用了监督学习，且网络结构深度较类似，参考意义较大。这些对照工作主要分两类：surrogate gradient BP 算法[42][56]；具有生物学合理性性的非 BP 算

法[32][34][54]。从上表的数据中可以得知，基于替代梯度的反向传播方法大约能达到 90.5% 至 91.0% 的准确率，但时间步普遍超过 10 个时间步；而生物学合理的非 BP 算法中，基于隐式微分的平衡传播方法[32]以 30 个时间步获得 92.08% 的表格中最高准确率，和本工作的 SNN-HLL 算法类似的局部学习算法 DECOLLE[34]和 TSLL[54]逊色一些，分别取得了 74.70% 和 89.22% 的准确率。本算法在使用 pred-sim 混合损失时，在同等 10 个时间步的条件下，用 VGG8 架构即取得 90.77% 的准确率，较 TSLL 提升 1.55 个百分点。当拓展网络容量（VGG8(2x)）后，最好能提升至 91.44% 的准确率。这一表现不仅超越表格中所有的替代梯度方法（最佳 90.95%），更以微弱优势超过了需 2048 个时间步的 ANN-to-SNN 转换方法[58]（91.42%）。本文提出的脉冲神经网络混合损失局部学习算法在 CIFAR-10 数据集上的实验结果超越了绝大多数被列出的对照工作，足以证明该算法性能具有较强的竞争力。

3.3.4 消融实验

为了比较不同局部损失，以及它们的组合的性能差异，考虑在 CIFAR-10 数据集上做消融实验，即考虑几种单独损失、各种损失的两两组合、甚至是三种损失的组合等情况的实验。除了损失函数不同外，实验的所有参数设置保持一致：全都采用 LIF 神经元模型，都使用直接编码的方式，都是 VGG8 的网络结构，时间步也都是 10。结果详见下表3-5：

表 3-5 CIFAR-10 数据集不同损失组合消融实验结果

损失组合名称	损失 1 及系数	损失 2 及系数	损失 3 及系数	测试集准确率
预测损失 L_{pred}	$1.0 * L_{pred}$	/	/	87.30%
相似度损失 L_{sim}	$1.0 * L_{sim}$	/	/	87.92%
对比损失 $L_{contrast}$	$1.0 * L_{contrast}$	/	/	78.25%
预测重构损失 $L_{pred-recon}$	$0.5 * L_{pred}$	$0.5 * L_{recon}$	/	81.15%
相似度重构损失 $L_{sim-recon}$	$0.01 * L_{sim}$	$0.99 * L_{recon}$	/	81.54%
预测对比损失 $L_{pred-contrast}$	$0.95 * L_{pred}$	$0.05 * L_{contrast}$	/	78.85%
相似度对比损失 $L_{sim-contrast}$	$0.2 * L_{sim}$	$0.8 * L_{contrast}$	/	60.30%
预测相似度损失 $L_{pred-sim}$	$0.99 * L_{pred}$	$0.01 * L_{sim}$	/	**90.77%**
三重损失 $L_{pred-sim-contrast}$	$0.95 * L_{pred}$	$0.01 * L_{sim}$	$0.04 * L_{contrast}$	89.91%

上表中有一些值得注意的点：1. 没有单独的重构损失 L_{recon} 实验，这是因为自编码器的重构损失属于无监督学习的范畴，它主要关注输入数据的重构效

果，而不是任务相关的判别信息，所以单独用它作为每一层的局部损失，最终在分类任务中的效果并不好，需要结合其他类型的损失才行。2. 相似度匹配损失和对比损失按比例系数复合得到的相似度对比损失 $L_{sim-contrast}$ 的测试集准确率（60.30%）显著低于其他损失及组合，这有可能是因为相似度匹配损失 L_{sim} 和对比损失 $L_{contrast}$ 的原理比较类似，它们都关注特征空间中的每个样本相对于其他样本的相对位置关系以及整个特征空间的结构，这样的组合是有些冗余的。3. 引入重构损失或对比损失并未进一步提升性能，甚至在部分组合中导致性能下降。这并不意味着它们是无用的组件，例如单独的对比损失 $L_{contrast}$ 的准确性为 78.25%，说明它确实在驱动表示学习，只不过在当前任务中并非表现最佳的组件，它更多起到了表示约束的作用，而非作为一个主导损失直接优化分类目标。这些尝试表明在脉冲神经网络局部损失方案中，不同的损失设计效果差异明显，启发我们再未来的工作中进一步探索更好的损失类型及其组合方案，这是一个开放性的问题。

从表3-5中可以发现，单一种类的分层局部损失性能难以超越全局反向传播，但当两种损失（预测损失和相似度匹配损失）按一定比例系数结合起来得到预测相似度损失 $L_{pred-sim}$，可以达到和全局 BP 一样好的性能。这是因为预测损失和相似度匹配损失能够“优势互补”，它们分别从不同的角度对隐藏层进行了优化：预测损失使用局部分类器预测当前层的输出是否能够正确分类，给出了局部的分类误差信号，目标是让每个隐藏层能学到对分类任务有用的特征，使得该层的输出尽可能地接近目标标签，其侧重点是直接提升分类任务的准确性；而相似度匹配损失用于生成局部的相似度误差信号，促使同类样本的特征更接近，而不同类样本的特征更加区分开来，优化了特征空间的分布结构，侧重于让隐藏层学到更好的特征表示。两种损失功能不同，两者组合起来有助于提升网络的泛化能力和优化效果，相得益彰。

3.4　本章小结

本章节提出了一种基于混合局部损失分层训练的脉冲神经网络学习算法 SNN-HLL，通过结合多种局部损失函数（如辅助分类预测损失、相似度匹配损失、对比损失等），在图像分类任务中实现了超越 SNN 领域常规算法替代梯度

下降法的性能。该方法通过逐层生成局部误差信号，避免了全局反向传播（BP）的生物学不可信问题（如权重对称、更新锁定、内存占用），以及克服了替代梯度带来的误差累积和梯度消失的缺点，同时支持并行训练和实时权重调整。实验表明，合理设计的多损失组合策略（如预测损失 + 相似度匹配损失的协同使用）能有效整合不同优化角度（分别对应判别能力、结构相似性约束），为脉冲神经网络的隐藏层提供全面的局部更新指导信号，相比于单一损失显著提升了模型性能。本章的 SNN-HLL 算法是一种非 BP 算法，在取得较高准确率的同时，避免了常规 BPTT 算法的大量梯度计算，在生物学合理性-精度-计算开销三方面都表现不错，为脉冲神经网络的训练算法提供了新的思路和有竞争力的解决方案。

第四章　脉冲神经网络 HSIC 瓶颈三因素赫布学习方法

本章提出了一种和上一章不同思路的脉冲神经网络非 BP 局部学习算法，用信息瓶颈理论的变体——希尔伯特-施密特独立性准则瓶颈度量脉冲神经网络不同层之间的相关性（依赖性），从而构建一种压缩冗余信息，保留预测能力的脉冲神经网络分层局部学习框架。并使用该算法在常规的图片分类数据集上对 SNN 进行了实验，取得了接近全局 BP 类算法的性能，且对噪声扰动具有一定鲁棒性。

4.1　背景与动机

探寻人脑的工作机理，以及开发类脑的神经网络学习算法一直是类脑科学和神经计算的关注重点。脉冲神经网络作为一种生物启发的新型神经网络，其训练算法和学习机制的创新往往会在生物可解释性、计算能效等方面寻求平衡。

在 SNN 的几种主要训练算法中，以 STDP 为代表的局部可塑性学习规则是生物启发式的，遵循脑科学中对大脑突触的观察结果，仅根据局部神经元脉冲信息进行突触权重调整，是一种无监督生物学合理的学习方式。然而由于 STDP 没有使用全局监督信号，其性能上限十分有限，在复杂任务中的性能通常远低于基于反向传播的监督学习方法。而基于 BP 的监督学习算法的缺点已经在前文 3.1 节中详细介绍，此处不再赘述。

是否能设计一种脉冲神经网络的学习算法，既有类似局部可塑性规则的生物学合理性，又能利用全局监督信号提升网络的学习表征能力？带着这样的问题，本研究尝试参考神经科学的相关研究进展，以期在理论层面获取学习机制上的灵感启发：从微观层面，大脑的突触学习确实遵循了局部可塑性规则，例如 STDP 规则；而从宏观层面，大脑中是存在全局调制信号的，例如多巴胺等神经递质的奖励信号，有点类似强化学习中的奖励机制。然而，大脑中并没有观察到全局反向传播，也就是说大脑中的全局调制信号并非是通过链式法则计算最终损失对于每一层的梯度（即逐层反向传播）的方式，而是有其他多种选择（比如

多巴胺、乙酰胆碱等神经递质），大脑的学习机理是复杂的、多尺度的、局部和全局混合的，目前神经科学还没有统一的理论，研究者们参考大脑的学习机制，也不是为了在生物细节上的模仿，而是为了获得学习理念上的启发。

生物神经系统中局部可塑性和全局调制信号协同调控的学习模式，为脉冲神经网络的学习算法设计提供了具有启示意义的范例，启发研究者也设计局部和全局结合的学习算法。关于如何使用全局信号，强化学习，随机投影（反馈对齐）、直接监督（正如在第三章做的那样）等都是一些可行方案，不过在本工作不打算使用这些已有的“全局调制”方法，而是将目光聚焦到信息论中的一些方案。因为神经网络本质上可以视为一个分层非线性的信息处理系统，从网络的输入层一直到输出层不断进行信息的压缩和变换，以便提炼出有用的特征。从信息论的视角审视神经网络乃至脉冲神经网络，便可以找到一种全新的学习范式——信息瓶颈及其变体 HSIC 瓶颈。

4.2　基于 HSIC 瓶颈三因素赫布学习的脉冲神经网络训练算法设计

4.2.1　随机变量独立性判定

在前文 2.2.3 节中，已经对信息瓶颈做了介绍。脉冲信号在 SNN 不同层之间的传递，可以看作一个信息链条：后一层产生脉冲信号只依赖于前一层发放的的脉冲信号。从输入的脉冲编码 X，到中间层的脉冲 Z，一直到最终目标输出 Y，特征的表示越来越“精炼”，即丢弃冗余无用的信息，只保留有利于任务目标的有用信息。换言之，神经网络乃至脉冲神经网络的训练过程可视为在两个主要任务之间权衡：一是信息压缩，即去除冗余特征；二是特征拟合，即增强对目标输出的判别能力。因此可以使用信息瓶颈的方法指导脉冲神经网络的学习。

为从信息论角度刻画 SNN 中层间信息的压缩与保留，首先回顾一下信息瓶颈的基本形式化目标：

$$\min_{p(z|x)} I(X;Z) - \beta I(Z;Y), \tag{4-1}$$

$$\text{其中} I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}. \tag{4-2}$$

可以发现在公式(4-2)中，若 $I(X;Y)=0$，说明 $p(x,y)=p(x)p(y)$，也就是两个变

量是相互独立的，否则是相关的。而互信息 $I(X;Y)$ 的计算是需要知晓随机变量 X 和 Y 的概率密度函数的，这就导致一个问题：在实际数据集上，通常很难获取数据的真实分布估计，这就使得互信息计算很困难，间接导致了信息瓶颈的计算困难。那么，互信息需要的分布估计难以获得，可以转而寻求一种替代方案，互信息的代替方案主要有基于变分近似的变分信息瓶颈 (Variational Information Bottleneck，VIB) 方法和基于核的方法，后者也就是 HSIC 瓶颈。

由于互信息 $I(X;Y)$ 是衡量两个随机变量相互依赖关系的指标，其缺陷在于对概率分布的显式依赖。为了突破这一限制，需要重新从数学的角度审视依赖性或者独立性的定义。在正式引入 HSIC 瓶颈之前，先回顾一下如何判断两个随机变量的独立性，一个常见的条件是：随机变量 X 和 Y 独立，当且仅当对任意可测集合 $A \in \sigma(X)$ 和 $B \in \sigma(Y)$，有：

$$P(X \in A, Y \in B) = P(X \in A) \cdot P(Y \in B). \tag{4-3}$$

这里的 $\sigma(X)$ 和 $\sigma(Y)$ 是 X 和 Y 生成的 σ 代数，这一判定条件表明，任何关于 X 的事件和任何关于 Y 的事件相互独立。这并不是唯一的判定条件，还存在其他的判定条件，如函数表征的独立性判断：随机变量 X 和 Y 独立，当且仅当对于任意有界可测的函数 f 和 g，有：

$$E[f(X)g(Y)] = E[f(X)]E[g(Y)]. \tag{4-4}$$

这一定义不再依赖准确的分布估计，而是转化为任意可测函数的期望匹配问题。从工程实践的角度，可以选取足够多的测试函数 f 和 g 来评估 X 和 Y 之间的依赖关系。尽管在理论上要求所有可测函数对都满足上式才能严格等价于独立性，但在实际使用中，通过对丰富的函数族进行近似测试，也能较好地判断变量之间是否独立。值得一提的是，当选取 $f = I_A$ 和 $G = I_B$，其中 I_A 和 I_B 表示示性函数时，有：

$$E[I_A(X)I_B(Y)] = P(X \in A, Y \in B) = P(X \in A) \cdot P(Y \in B) = E[I_A(X)] \cdot E[I_B(Y)]. \tag{4-5}$$

这就退化为了前面的事件定义的独立性。

现在可以根据公式(4-4)，定义交叉协方差算子 $c[f,g]$：

$$c[f,g] = E_{(x,y)\sim p(x,y)}[f(x)g(y)] - E_{x\sim p(x)}[f(x)]E_{y\sim p(y)}[g(y)]. \tag{4-6}$$

为了量化多个函数对独立性的整体判别能力，可以引入以下总协方差指标 L_h：

$$L_h = \sum_{f,g}(c[f,g])^2. \tag{4-7}$$

指标 L_h 的含义为：选取足够多的函数 f 和 g，看 L_h 与 0 是否足够接近，足够接近就意味着两个随机变量的独立性高而相关性弱。这样就获得了一个判定独立性的方法，进一步对 $(c[f,g])^2$ 进行展开计算：

$$\begin{aligned}(c[f,g])^2 =&(E_{(x,y)\sim p(x,y)}[f(x)g(y)])^2 + (E_{x\sim p(x)}[f(x)]E_{y\sim p(y)}[g(y)])^2\\ &- 2E_{(x,y)\sim p(x,y)}[f(x)g(y)]E_{x\sim p(x)}[f(x)]E_{y\sim p(y)}[g(y)]\\ =&E_{(x_1,y_1)\sim p(x,y),(x_2,y_2)\sim p(x,y)}[f(x_1)g(y_1)f(x_2)g(y_2)]\\ &+ E_{x_1\sim p(x),x_2\sim p(x),y_1\sim p(y),y_2\sim p(y)}[f(x_1)g(y_1)f(x_2)g(y_2)]\\ &- 2E_{(x_1,y_1)\sim p(x,y),x_2\sim p(x),y_2\sim p(y)}[f(x_1)g(y_1)f(x_2)g(y_2)].\end{aligned} \tag{4-8}$$

可以发现每一项都是求 $f(x_1)g(y_1)f(x_2)g(y_2)$ 的期望，区别仅在于采样分布不同。

现在的问题是如何选取函数 f 和 g，才能让 L_h 发挥判定随机变量独立性的功能呢？实际上，正如公式(4-7)所示，选取足够丰富的函数族并非易事。如果所选函数族过于简单，就无法捕捉到变量间高阶的、各种可能的依赖模式；另一方面，如果在所有可测函数上穷举，又在计算上不可行。那么如何构造一族既足够丰富，又具备良好计算性质的函数空间呢？这就需要引入核函数的工具了。

4.2.2　核函数的选取

核函数在统计学和机器学习中扮演着重要的角色。数学上，任意一个满足正定条件的核函数，都唯一对应一个再生核希尔伯特空间（Reproducing Kernel Hilbert Space, RKHS），这是一个以核函数为内积结构的函数空间。在 RKHS 中，对于任意函数 f 属于该空间，以及任意输入 x，函数在该点的取值可以表示为与

核函数的内积形式，其中 $\mathcal{H}$ 表示由核函数 K 所诱导的 RKHS：

$$f(x) = \langle f, K(\cdot, x) \rangle_{\mathcal{H}}, \tag{4-9}$$

这表明任意函数在任意点处的取值可以通过与对应核函数的内积“再生”得到，这就是“再生核”名字的由来。

核函数能将数据从低维空间映射到高维的再生核希尔伯特空间，且无需在高维空间中进行复杂运算，而是通过一种隐式的计算方式在原始空间计算。核函数是一种二元函数：$K : X \times X \to \mathbb{R}$，满足：

$$K(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle_{\mathcal{H}}. \tag{4-10}$$

其中，$\phi : X \to \mathcal{H}$ 是将输入空间 X 非线性映射到 RKHS 空间 $\mathcal{H}$ 的特征映射，$\langle \cdot, \cdot \rangle$ 是 $\mathcal{H}$ 中的内积。得益于核方法，能够在不显式构造 ϕ 的情况下，仅通过核函数 K 在输入空间中完成高维计算。根据 Mercer 定理，若核函数是连续、对称和正定的，那么存在一组正交特征函数 $\phi_i(x)$ 以及对应的非负特征值 λ_i，使得核函数可展开为：

$$K(x, y) = \sum_{i=0}^{\infty} \lambda_i \phi_i(x) \phi_i(y). \tag{4-11}$$

上述展开公式表明，核函数可以看作其特征值和特征函数构成的一组正交基在 L^2 空间上的无穷级数展开。

常见的核函数有：线性核函数、余弦相似度核函数、高斯核函数、拉普拉斯核函数和 Sigmoid 核函数。线性核函数的表达式为：

$$K(x_1, x_2) = x_1^T x_2. \tag{4-12}$$

余弦相似度核函数的表达式为：

$$K(x_1, x_2) = \frac{x_1^T x_2}{||x_1||_2 ||x_2||_2}. \tag{4-13}$$

高斯核函数的表达式为：

$$K(x_1, x_2) = \exp\left(-\frac{||x_1 - x_2||_2^2}{2\sigma^2}\right). \tag{4-14}$$

拉普拉斯核函数的表达式为：

$$K(x_1, x_2) = \exp\left(-\frac{||x_1 - x_2||_2}{\sigma}\right). \tag{4-15}$$

Sigmoid 核函数的表达式为：

$$K(x_1, x_2) = \tanh\left(\alpha x_1^T x_2 + c\right). \tag{4-16}$$

假设已经有定义在 $X \times X$ 上的核函数 K_1，那么可以计算出其特征值 $\lambda_i (i = 1, 2 \cdots)$ 和对应的特征函数 $\phi_i (i = 1, 2 \cdots)$；同样地，定义在 $Y \times Y$ 上的核函数 K_2，那么可以计算出其特征值 $\mu_i (i = 1, 2 \cdots)$ 和对应的特征函数 $\psi_i (i = 1, 2 \cdots)$。特征函数 ϕ_i 和 ψ_i 其实构成了核函数所隐式定义的特征空间中的一组正交基。接下来将公式(4-7)中的 f 和 g 换成 ϕ_i 和 ψ_i，并乘上各自的特征值作为权重，就变成了以下公式：

$$L_h = \sum_{i,j} \lambda_i \mu_j (c[\phi_i, \psi_j])^2. \tag{4-17}$$

结合 $(c[\phi_i, \psi_j])^2$ 的展开公式，得到：

$$\begin{aligned} L_h =& E_{(x_1,y_1)\sim p(x,y),(x_2,y_2)\sim p(x,y)} \sum_{i,j} \lambda_i \mu_j \phi_i(x_1)\psi_j(y_1)\phi_i(x_2)\psi_j(y_2) \\ &+ E_{x_1\sim p(x),x_2\sim p(x),y_1\sim p(y),y_2\sim p(y)} \sum_{i,j} \lambda_i \mu_j \phi_i(x_1)\psi_j(y_1)\phi_i(x_2)\psi_j(y_2) \\ &- 2E_{(x_1,y_1)\sim p(x,y),x_2\sim p(x),y_2\sim p(y)} \sum_{i,j} \lambda_i \mu_j \phi_i(x_1)\psi_j(y_1)\phi_i(x_2)\psi_j(y_2). \end{aligned} \tag{4-18}$$

将公式(4-11)代入，就有：

$$\begin{aligned} L_h =& E_{(x_1,y_1)\sim p(x,y),(x_2,y_2)\sim p(x,y)} K_1(x_1, x_2) K_2(y_1, y_2) \\ &+ E_{x_1\sim p(x),x_2\sim p(x),y_1\sim p(y),y_2\sim p(y)} K_1(x_1, x_2) K_2(y_1, y_2) \end{aligned}$$

$$-2E_{(x_1,y_1)\sim p(x,y),x_2\sim p(x),y_2\sim p(y)}K_1(x_1,x_2)K_2(y_1,y_2). \tag{4-19}$$

这其实已经得到了希尔伯特-施密特独立性准则（Hilbert-Schmidt independence criterion，HSIC）的雏形。上述推导构成了 HSIC 的核心思想，它提供了一种基于核函数度量随机变量间统计依赖性的有效方式。这种度量变量依赖性的统计量可以用来评估脉冲神经网络中不同层特征之间的关联程度，为 SNN 中的信息瓶颈学习范式提供了一种替代互信息的可实现路径。

4.2.3 HSIC 瓶颈

经过以上两小节的推导，由公式(4-19)，可以自然地给出 HSIC 的公式定义：

$$\begin{aligned}HSIC(X,Y)=&E_{x_1,y_1,x_2,y_2}[K_1(x_1,x_2)K_2(y_1,y_2)]+E_{x_1,x_2}[K_1(x_1,x_2)]E_{y_1,y_2}[K_2(y_1,y_2)]\\&-2E_{x_1,y_1}E_{x_2}[K_1(x_1,x_2)]E_{y_2}[K_2(y_1,y_2)].\end{aligned} \tag{4-20}$$

观察这个定义，HSIC 试图衡量两个随机变量 X 和 Y 的独立性，从联合分布 P_{XY} 中独立抽取两个样本对 (x_1,y_1) 和 (x_2,y_2)，样本 x_1 和 x_2 相当于是从边缘分布 P_X 中独立抽取的，是满足独立同分布的，同理，样本 y_1 和 y_2 也是独立同分布采样的。并利用这两个独立抽样样本对来构造相应的期望项，捕捉随机变量内部的结构差异，从而测量出两个随机变量在高维特征空间中的依赖性（或独立性）。等号右边的第一项是联合分布项，对联合分布下两个样本对进行比较，反映了联合相似性；第二项是边缘分布项，分别对两个边缘分布计算相似性，反映了无依赖假设下的相似性结构；第三项是修正项。这三项的设计体现了核方法对高阶、非线性依赖的敏感性，这是其优于传统线性相关指标的根本原因。

公式(4-20)给出的是无限样本的期望表达式，对于有限样本的情形：假设有 m 对样本，则上述 HSIC 表达式可以写成有限样本的求和形式：

$$\begin{aligned}HSIC(X,Y)=&\frac{1}{m^2}\sum_{i,j}K_1(x_i,x_j)K_2(y_i,y_j)+\frac{1}{m^2}\sum_{i,j}K_1(x_i,x_j)\frac{1}{m^2}\sum_{k,l}K_2(y_k,y_l)\\&-\frac{2}{m^3}\sum_{i,j,l}K_1(x_i,x_j)K_2(y_i,y_l).\end{aligned} \tag{4-21}$$

可以看到上式等号右侧的第一项只有两个求和指标 i 和 j，意味着只要找两对数据点即可，所有情况数是 $O(m^2)$ 的；第二项有四个求和指标 i,j,k,l，看上去似乎更麻烦，实则不然，第二项其实是两个独立的部分的乘积，复杂度其实还是 $O(m^2)$ 的；而第三项有三个求和指标 i,j,l，x_i 与 y_i 共享下标，而 x_j 与 y_l 都是单独的下标，因此计算复杂度为 $O(m^3)$。然而在脉冲神经网络的实际训练中，尤其是在深层脉冲神经网络和大规模样本的情况下，$O(m^3)$ 的计算开销将成为负担。因此，为了形式的统一简洁和运算复杂度的降低，对第三项做如下近似：

$$\sum_{i,j,k} K_1(x_i,x_j)K_2(y_i,y_l) \approx \frac{1}{m}\sum_{i,j,k,l} K_1(x_i,x_j)K_2(y_k,y_l). \tag{4-22}$$

这样就有：

$$\begin{aligned} HSIC(X,Y) \approx & \frac{1}{m^2}\sum_{i,j} K_1(x_i,x_j)K_2(y_i,y_j) + \frac{1}{m^2}\sum_{i,j} K_1(x_i,x_j)\frac{1}{m^2}\sum_{k,l} K_2(y_k,y_l) \\ & - \frac{2}{m^4}\sum_{i,j,k,l} K_1(x_i,x_j)K_2(y_k,y_l) \\ = & \frac{1}{m^2}\sum_{i,j} K_1(x_i,x_j)K_2(y_i,y_j) - \frac{1}{m^4}\sum_{i,j} K_1(x_i,x_j)\sum_{k,l} K_2(y_k,y_l). \end{aligned} \tag{4-23}$$

在有限样本情形下完成近似处理后，再次恢复为无限情形的期望形式，即：

$$HSIC(X,Y) \approx \left(E_{(x_1,y_1)}E_{(x_2,y_2)} - E_{x_1}E_{x_2}E_{y_1}E_{y_2}\right)[K_1(x_1,x_2)K_2(y_1,y_2)]. \tag{4-24}$$

通过一系列推导，最终得到了 $HSIC(X,Y)$ 的一个可用版本，不要掌握随机变量的分布估计，只需要一个合适的核函数，就能用有限的样本数据计算出两个随机变量的依赖程度。如果 $HSIC(X,Y)$ 的值较大，说明随机变量 X 和 Y 的依赖性较高；反之，如果 $HSIC(X,Y)$ 的值较小甚至为 0，说明随机变量 X 和 Y 的依赖性较低甚至是相互独立的。注意，这里的依赖性和相关性还不一样，常见的皮尔逊相关系数（Pearson Correlation Coefficient）度量的是两个变量之间的线性相关程度，而依赖性是个更广泛的概念，它可以捕捉任意的关系，无论是线性关系还是非线性关系。因此，HSIC 瓶颈的依赖性比常规的各种相关系数、相似性指标更加全面。

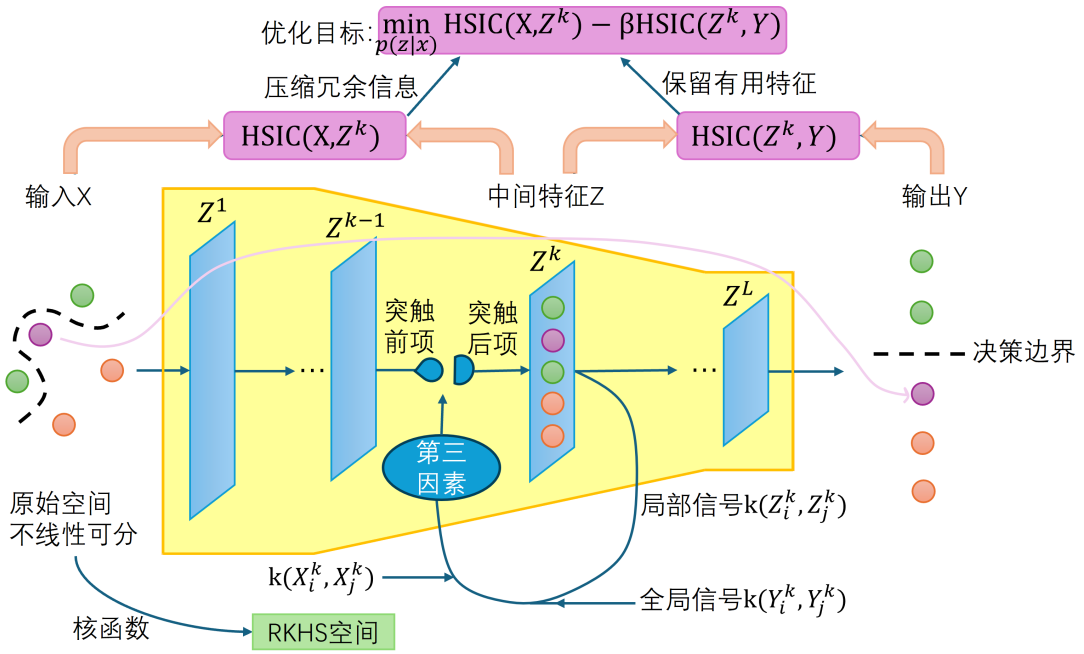

图 4-1　SNN 的 HSIC 瓶颈三因素赫布学习原理图

基于上述推导，可以在信息瓶颈范式中用 $HSIC(X,Y)$ 项替换掉原本难以估计的互信息项 $I(X;Y)$，从而构造出以下更具可计算性的 HSIC 瓶颈优化目标：

$$\min_{p(z|x)} HSIC(X,Z^k) - \beta HSIC(Z^k,Y), k = 1,2,\ldots,L. \tag{4-25}$$

图 4-1 展示了脉冲神经网络的 HSIC 瓶颈三因素 Hebbian 学习训练算法整体原理框架示意图。该方法以多层脉冲神经网络为基础，将信息瓶颈思想作用于每一层中间的脉冲值特征 $Z^k, k = 1,2,\cdots,L$，构建包含两项的优化目标以实现信息的最优传递：其中第一项 $HSIC(X,Z^k)$ 为正项，效果是通过缩减 X 和 Z^k 的依赖性，压缩 X 到 Z^k 的冗余相关信息；第二项 $-\beta HSIC(Z^k,Y)$ 为负项，最小化优化目标即最大化这一项，作用是通过增强 Z^k 到 Y 的依赖性，从而尽可能多地保留 Z^k 对 Y 的判别性。该图左下角部分表达的含义为：HSIC 利用核函数将原始空间中的变量映射到再生核希尔伯特空间（Reproducing Kernel Hilbert Space, RKHS），使得在该特征空间中通过线性算子即可度量原始空间变量间复杂的非线性统计依赖关系，这一核映射机制构成了 HSIC 理论框架的基础。该图中间部分展现了本算法中的突触赫布学习类规则，不光有局部突触信号，还用到了全局调制信号，在下一小节中会做详细介绍。

4.2.4　三因素赫布学习规则

接下来，本小节将推导上述 HSIC 瓶颈优化目标是如何作为局部信号指导脉冲神经网络的更新的。首先，需要给出本工作中所使用的脉冲神经网络表达式：

$$U^{l,t} = \lambda(U^{l,t-1} - S^{l,t-1}V_{th}) + W^l S^{l-1,t}, \tag{4-26}$$

$$S^{l-1,t} = \theta(U^{l-1,t} - V_{th}). \tag{4-27}$$

其中，$U^{l,t}$ 表示时间步 t 时第 l 层的膜电压，$S^{l,t}$ 表示时间步 t 时第 l 层可能发放的脉冲，$\theta(x)$ 为 Heaviside 函数，当 $x \geqslant 0$ 时，$\theta(x) = 1$，否则为 0。λ 为衰减系数，V_{th} 为脉冲发放阈值电位，W^l 为 SNN 第 l 层的网络连接权值。观察上面的 SNN 表达式，有时间 t 和层数 l 两个指标，即每一层某一时间步的膜电压，不光与前一层同一时刻的脉冲有关，还和当前层上一时刻的膜电压状态和是否发放脉冲有关。对这种在空间和时间两个维度上有递归关系的神经网络，一般要使用 BPTT 算法（Back-propagation Through Time），梯度计算量会比较大。且生成脉冲的 Heaviside 函数的不可微特性会导致计算梯度的困难，哪怕是使用代理函数替代脉冲函数，也会引入误差。

根据式(4-26)和(4-27)，其实可以得到某一时刻当前层脉冲和前一层脉冲之间的关系，即：

$$S^{l,t} = \theta\left(\lambda(U^{l,t-1} - S^{l,t-1}V_{th}) + W^l S^{l-1,t} - V_{th}\right). \tag{4-28}$$

记 $b^{l,t-1} = \lambda(U^{l,t-1} - S^{l,t-1}V_{th}) - V_{th}$，则上式变为：

$$S^{l,t} = \theta\left(W^l S^{l-1,t} + b^{l,t-1}\right). \tag{4-29}$$

这样就将脉冲神经网络中相邻两层某一时刻的脉冲信息的关系表示出来了。令 $Z^{l,t} = S^{l,t}$，于是可以定义 SNN 当前层的 HSIC 瓶颈损失：

$$L^l_{HSIC} = \sum_t HSIC(X, Z^{l,t}) - \beta HSIC(Z^{l,t}, Y), l = 1, 2, \ldots, L. \tag{4-30}$$

下面推导计算局部损失 L^l_{HSIC} 对权重的导数，有：

$$\frac{\partial L^l_{HSIC}}{\partial W^l} = \sum_t \frac{\partial (HSIC(X, Z^l) - \beta HSIC(Z^l, Y))}{\partial W^l}. \tag{4-31}$$

对于 HSIC 统计量，考虑有限情形的 HSIC 表示，于是有：

$$\begin{aligned}
&\frac{\partial (HSIC(X, Z^{l,t}) - \beta HSIC(Z^{l,t}, Y))}{\partial W^l} = \\
&\frac{1}{m^2}\sum_{ij} k(x_i, x_j)\frac{\partial k(z_i^{l,t}, z_j^{l,t})}{\partial W^l} - \frac{1}{m^2}\sum_{ql} k(x_q, x_l)\frac{1}{m^2}\sum_{ij}\frac{\partial k(z_i^{l,t}, z_j^{l,t})}{\partial W^l} \\
&- \beta\frac{1}{m^2}\sum_{ij} k(y_i, y_j)\frac{\partial k(z_i^{l,t}, z_j^{l,t})}{\partial W^l} + \beta\frac{1}{m^2}\sum_{ql} k(y_q, y_l)\frac{1}{m^2}\sum_{ij}\frac{\partial k(z_i^{l,t}, z_j^{l,t})}{\partial W^l}.
\end{aligned} \tag{4-32}$$

为了形式上的简洁，定义 $\tilde{k}(a_i, a_j) = k(a_i, a_j) - \sum_{ql} k(a_q, a_l)/m^2$，则公式(4-32)变成以下更简约的形式：

$$\frac{\partial (HSIC(X, Z^{l,t}) - \beta HSIC(Z^{l,t}, Y))}{\partial W^l} = \frac{1}{m^2}\sum_{ij}\left(\tilde{k}(x_i, x_j) - \beta\tilde{k}(y_i, y_j)\right)\frac{\partial k(z_i^{l,t}, z_j^{l,t})}{\partial W^l}. \tag{4-33}$$

对于核函数，选用特定核函数，能让上式呈现出显式的 Hebbian 学习的形式，在选取 Gaussian 核函数（见公式(4-14)）的情况下，有：

$$\begin{aligned}
\frac{\partial k(z_i^{l,t}, z_j^{l,t})}{\partial W^l} &= \frac{\partial}{\partial W^l} exp\left(-\frac{||z_i^{l,t} - z_j^{l,t}||_2^2}{2\sigma^2}\right) \\
&= -\frac{k(z_i^{l,t}, z_j^{l,t})}{\sigma^2}(z_i^{l,t} - z_j^{l,t})\frac{\partial (z_i^{l,t} - z_j^{l,t})}{\partial W^l} \\
&= -\frac{k(z_i^{l,t}, z_j^{l,t})}{\sigma^2}(z_i^{l,t} - z_j^{l,t})(\frac{\partial z_i^{l,t}}{\partial W^l} - \frac{\partial z_j^{l,t}}{\partial W^l}).
\end{aligned} \tag{4-34}$$

于是每一层的 HSIC 瓶颈损失提供的误差信号，会以如下形式作用于权重更新：

$$\Delta W^l \propto -\sum_t\sum_{ij}\frac{k(z_i^{l,t}, z_j^{l,t})}{\sigma^2}\left(\tilde{k}(x_i, x_j) - \beta\tilde{k}(y_i, y_j)\right)(z_i^{l,t} - z_j^{l,t})(\frac{\partial z_i^{l,t}}{\partial W^l} - \frac{\partial z_j^{l,t}}{\partial W^l}). \tag{4-35}$$

上式显然表明，对 SNN 的每个时间步都需计算一次 HSIC 瓶颈损失。而在实际运用中，为减小计算代价，可以用脉冲发放率 $r^l = \frac{1}{T}\sum_{t=1}^{T} S^{l,t}$ 来替换原本对所有时间步求和的形式。相邻层的脉冲发放率可以近似看作有以下关系：$r^l = F\left(W^l, r^{l-1}\right)$，其中 F 是一个非线性映射。于是公式(4-35)可以写成：

$$\begin{aligned}\Delta W^l &\propto -\sum_{ij}\frac{k(r_i^l, r_j^l)}{\sigma^2}\left(\tilde{k}(x_i, x_j) - \beta\tilde{k}(y_i, y_j)\right)(r_i^l - r_j^l)\left(\frac{\partial r_i^l}{\partial W^l} - \frac{\partial r_j^l}{\partial W^l}\right) \\ &= -\sum_{ij}\frac{k(r_i^l, r_j^l)}{\sigma^2}\left(\tilde{k}(x_i, x_j) - \beta\tilde{k}(y_i, y_j)\right)(r_i^l - r_j^l)\left(F'(r_i^{l-1})r_i^{l-1} - F'(r_j^{l-1})r_j^{l-1}\right).\end{aligned} \tag{4-36}$$

对这个更新规则的形式进行分析，它可以分解为三个部分：$r_i^l - r_j^l$ 为当前层第 i 个样本和第 j 个样本的脉冲发放率之差，为突触后信息；$F'(r_i^{l-1})r_i^{l-1} - F'(r_j^{l-1})r_j^{l-1}$ 为前一层的脉冲发放率 r^{l-1} 和导数项 $F'(W^l, r^{l-1})$ 之积在两个样本 i 和 j 之间的差值；$\frac{k(z_i^k, z_j^k)}{\sigma^2}\left(\tilde{k}(x_i, x_j) - \beta\tilde{k}(y_i, y_j)\right)$ 为每层特定的第三因素。因此本算法是符合三因素赫布学习规则的，神经科学上的三因素分别指突触前活动（presynaptic activity）、突触后活动（postsynaptic activity）、第三因素——一种误差信号或全局调制信号（a global modulatory factor），类似生物中多巴胺等神经递质的作用方式。如图4-1所示，本方法引入模拟突触三因素学习机制的结构：以突触前项 Z^{k-1}、突触后项 Z^k 活动为基础，并引入由核函数生成的调制信号，该信号综合了局部特征间的相似度，输出目标相似度和原始数据的相似度，这种设计对应了神经科学中的“Hebbian + 调制信号”范式，实现了生物启发的三因素赫布学习规则。

算法4.1展示了 SNN 每一层的 HSIC 瓶颈损失 L_{HSIC} 的计算步骤，以及其监督 SNN 各层权重更新的算法流程：

4.3　实验和分析

4.3.1　数据集和实验设置

本工作使用的数据集和第三章的工作相同，也是 MNIST、Fashion-MNIST（如图4-2所示）两个图像分类数据集，分别用于手写体识别、服饰分类任务。相关数据集介绍详见第三章，此处不再赘述，仅简述与实验数据相关的处理方式：

算法 4.1 SNN 的 HSIC 瓶颈训练算法流程伪代码

输入: 输入数据 X，对应标签 Y，需要学习的权重 $\{W^l\}_{l=1}^{L}$
参数: 学习率 η，平衡参数 β，时间步长 T
输出: 更新后的权重 $\{W^l\}_{l=1}^{L}$
for 每个 mini-batch B **do**
    根据高斯核函数公式4-14，计算当前批次输入 X 和及其真实标签 Y 的核矩阵 K_X 和 K_Y
    for $l = 1$ to L **do**
        for $t = 1$ to T **do**
            根据公式4-26和4-27计算第 l 层 t 时刻的膜电压 $U^{l,t}$ 和脉冲值 $S^{l,t}$
        end for
        对所有时间步的脉冲值 $S^{l,t}$ 计算平均脉冲发放率 r^l，作为中间特征 Z^l
        根据公式4-14计算 Z^l 的核矩阵 K_{Z^l}
        根据公式4-23和核矩阵 K_X,K_Y 以及 K_{Z^l}，分别计算 $HSIC(X, Z^l)$ 和 $HSIC(Z^l, Y)$
        计算当前层的 HSIC 损失 $L_{HSIC} = HSIC(X, Z^l) - \beta HSIC(Z^l, Y)$
        根据公式4-32计算当前层梯度 $\nabla_{W^l} L_{HSIC}$
        更新当前层权重 $W^l \leftarrow W^l - \eta \nabla_{W^l} L_{HSIC}$
    end for
end for

需要对数据集 MNIST、fashion-MNIST 进行归一化预处理。

以下是实验参数设置。对于核函数，实验中主要采用高斯核函数，因为高斯核能诱导出显式的三因素赫布学习规则形式的更新公式，相比其他核函数更具有生物学可信度。高斯核函数的宽度参数 σ 取 5.0。关于 HSIC 瓶颈，采用的是近似后的 HSIC 损失，而非原始的 HSIC 损失，因为前者比后者表现更好，HSIC 瓶颈中的平衡参数 β 设置为 2，为经验选择结果。

关于网络结构，主要有两种：MLP 和 VGG-like 的卷积网络。其中 MLP 为三层全连接网络，每一层有 1024 个神经元。MLP 的实验主要用于简单的 MNIST 数据集，使用 SGD 优化器训练 100 个 epoch，丢弃率为 0.1。VGG-like 的卷积网络用于全部几种数据集，使用 AdamW 优化器训练 400 个 epoch，丢弃率为 0.1，其具体网络结构为：conv128-conv256-maxpool-conv256-conv512-maxpool-conv512-maxpool-conv512-maxpool-fc1024-fc10，batch size 为 64。无论是全连接层还是卷积层，每一层都遵循以下的顺序进行运算：对上一层的脉冲序列进行线性或卷积运算 → 批归一化 BN（可选）→dropout（可选）→HSIC 瓶颈损失计算 →HSIC 损失指导局部权重更新。

由于后续的实验涉及多种不同的核函数的对比实验，以及有噪声数据情形

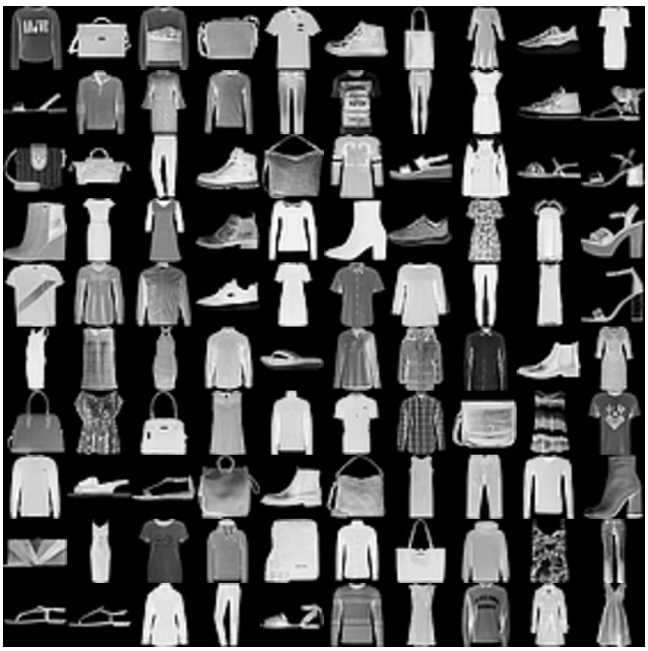

图 4-2　Fashion-MNIST 数据集

相较于无噪声数据情形的比较实验，进行这些实验时将通过控制变量，保证网络结构和各种实验超参数设置保持一致。本实验涉及的所有实验参数设置详见下表4-1所示：

表 4-1　SNN 的 HSIC 瓶颈三因素赫布学习算法实验网络参数设置

参数名称	设置
优化器	SGD(MLP),AdamW(VGG)
训练轮数 epoch	400
初始学习率	0.0005
学习率衰减率	0.25
batch size	64
dropout 率	0.1
膜电压衰减率	0.8(MNIST),0.75(Fashion-MNIST)
脉冲发放阈值	0.45(MNIST),0.6(Fashion-MNIST)
时间步	10
信息瓶颈平衡参数	$\beta = 2$
核函数宽度参数	$\sigma = 5.0$(高斯核 + 拉普拉斯核)
MLP 结构	784-1024-10
VGG 结构	128C3-256C3-P2-256C3-512C3-P2-512C3-P2-512C3-P2-FC1024-FC10

4.3.2 对比实验

在实验中，可以观察到本工作中的脉冲神经网络分类性能对膜电压衰减率 λ 和发放阈值电位 V_{th} 两个参数较敏感，这两个参数大小的绝对值和其相对大小关系对最终结果的影响较显著，对不同数据集都会单独设置特定的参数值。

对于 MNIST 数据集，分别实验了 MLP 和 VGG 两种结构的网络。膜电压衰减率 λ 和发放阈值电位 V_{th} 分别设置为 0.8 和 0.45 时效果比较好。具体结果见下表4-2：

表 4-2 MNIST 数据集实验结果

工作	训练算法	编码方式	网络结构	时间步	测试集准确率
[41]	STDP(unsupervised)	rate code	MLP(784-6400-10)	10	95.0%
[48]	Global Feedback + STDP	temporal code	MLP(784-800×3-10)	10	98.6%
[42]	Spatio-temporal BP	rate code	MLP(784-800-10)	10	98.89%
[42]	Spatio-temporal BP	rate code	CNN	10	99.42%
[59]	Feedback Alignment	rate code	CNN	10	99.01%
[31]	STDP-based BP	rate code	CNN	/	97.2%
[55]	surrogate gradient BP	rate code	CNN	10	99.26%
[34]	DECOLLE(non-BP)	/	CNN	10	97.51%
[53]	IB(non-BP)	rate code	CNN	3	98.96%
SNN-HBH	HSIC bottleneck(non-BP)	direct code	CNN	10	**99.24%**

在表4-2中，呈现了脉冲神经网络 HSIC 瓶颈局部训练算法 SNN-HBH 和其他常见的脉冲神经网络工作中的算法在 MNIST 数据集上的性能对比。该表中参考对比的工作可以分为 BP 式的和非 BP 式的：BP 式的算法包括 surrogate gradient BP 的工作[42][55]，和混合方法的工作[31]；非 BP 式的方法又可以分为两种：传统的局部可塑性学习框架的工作[48][59][34]和信息瓶颈学习框架的工作[53]。替代梯度下降法的工作[42][55]在 MNIST 上的准确率分别为 99.42% 和 99.26%，和它们相比，本研究的 HSIC 瓶颈方法实现了 99.24% 的准确率，仅略低于前者，在不依赖全局反向传播的前提下，展现出极具竞争力的性能。非 BP 式的反馈对齐算法训练 SNN 的工作[59]的实现了 99.01% 的准确率，和他们相比，本工作的方法精度更高。相较于同样采用信息瓶颈技术训练脉冲神经网络的工作[53]，本算法的性能也更有优势，这是因为他们使用的是传统信息瓶颈 IB 方法，且使用了一些损失函数来替换互信息 $I(Z;X)$，回避了对互信息的准确建模。而本章的 HSIC 瓶颈方法，通过核函数构建了更有效的依赖性度量，从而实现了对互信息的较为

精准的估计。

对于 Fashion-MNIST 数据集上的实验，实验采用 VGG 结构的 8 层卷积网络。对该数据集，膜电压衰减率 λ 和发放阈值电位 V_{th} 分别设置为 0.75 和 0.6 时效果较好。实验结果见下表4-3：

表 4-3　Fashion-MNIST 数据集实验结果

工作	训练算法	编码方式	网络结构	时间步	测试集准确率
[48]	Global Feedback + STDP	temporal code	MLP	10	89.05%
[32]	Implicit Differentiation	rate code	MLP	5	90.25%
[60]	IB + surrogate gradient BP	temporal code	VGG9	10	90.17%
[53]	IB(non-BP)	rate code	VGG8	3	87.92%
SNN-HBH	HSIC(non-BP)	direct code	VGG8	10	**89.80%**

在表4-3中，展示了脉冲神经网络 HSIC 瓶颈局部训练算法 SNN-HBH 和其他常见的脉冲神经网络工作中的算法在 Fashion-MNIST 数据集上的性能对比。其中，传统信息瓶颈的局部非 BP 式算法[53]在 Fashion-MNIST 上达到了 87.92% 的准确率，本工作的 HSIC 瓶颈的局部非 BP 式算法（SNN-HBH）达到了 89.80% 的准确率，显著优于他们的结果，这和前面 MNIST 数据集上的结果是一致的。值得注意的是，和采用全局替代梯度下降法的信息瓶颈框架[60]相比，SNN-HBH 的准确率（89.80%）略低于其最优结果（90.17%），虽然本算法在精度上稍逊一筹，但是在计算开销、生物学可解释性和神经形态硬件适应性等方面展现出更优的潜力。

综上，在 MNIST 和 Fashion-MNIST 两个标准数据集上，本章提出的脉冲神经网络 HSIC 瓶颈赫布学习训练算法 SNN-HBH 均实现了在非 BP 训练框架下的较好性能，非常接近全局误差 BP 类方法，验证了本方法在保持生物启发性与工程性能之间的良好权衡。值得注意的是，与需要多次误差传播的替代梯度 BPTT 方法不同，SNN-HBH 实现了层级可并行的局部更新，理论上具有更好的硬件适应性与神经形态设备部署潜力。

4.3.3　不同核函数的效果比较实验

HSIC 瓶颈方法的重要组件——核函数的选择是多样化的，除了可以显式推导出类赫布学习规则的高斯核函数外，其余的核函数也是值得尝试的。下表4-4给出了 4.2.2 节中的几种主要核函数（除了线性核函数，因为线性核函数较简单，效

果一般，忽略不计）在 Fashion-MNIST 数据集上的实验对比结果，所有参数保持一致。

表 4-4 不同核函数在 Fashion-MNIST 数据集上的效果对比

核函数	表达式	准确率
高斯核	$exp(-\frac{\|\|x_1-x_2\|\|_2^2}{2\sigma^2})$	89.8%
余弦相似度核	$\frac{x_1^T x_2}{\|\|x_1\|\|_2\|\|x_2\|\|_2}$	88.7%
拉普拉斯核	$exp(-\frac{\|\|x_1-x_2\|\|_2}{\sigma})$	88.8%
sigmoid 核	$tanh(\alpha x_1^T x_2 + c)$	87.5%

由表4-3可见，几种核函数都能在 HSIC 瓶颈局部学习的框架下，有效度量网络特征间的依赖性，从而有效地训练脉冲神经网络。各个核函数之间的效果有些许差异，其中高斯核的效果是最好的；拉普拉斯核的准确率相比高斯核低了1.0%。这两种核函数的区别在于，高斯核是基于 L_2 距离的，而拉普拉斯核是基于 L_1 距离的，因此高斯核的特点是平滑性高、非线性程度更高，而拉普拉斯核则对异常值更鲁棒，总体而言还是高斯核通用性强。余弦相似度核在衡量方向性特征方面具有优势，但其效果略低于欧式距离类核函数。整体来看，高斯核在当前任务中表现出更好的通用性与学习能力。

4.3.4 噪声数据对比实验

为了验证所提出的基于 HSIC 瓶颈的脉冲神经网络训练算法在噪声干扰环境下的鲁棒性，本小节在 MNIST 和 Fashion-MNIST 数据集上引入两种常见类型的输入噪声（高斯噪声与椒盐噪声），并与 SNN 传统的全局替代梯度反向传播算法进行对比实验。该实验的目的在于评估不同训练范式在面临输入扰动时对模型性能的影响，从而探讨基于 HSIC 瓶颈的局部训练方法在实际场景中的鲁棒性优势。

本实验引入两种常见的输入扰动方式进行鲁棒性评估：它们分别是高斯噪声（Gaussian noise）与椒盐噪声（Salt-and-Pepper noise），如图4-3所示。其中，高斯噪声是指对图像像素加入均值为 0、标准差为 $\sigma = 0.4$ 的正态分布噪声；椒盐噪声则以概率 p=0.2 随机将图像像素置为最大值或最小值，模拟传感器故障或信道干扰等离散型噪声干扰场景。这两种噪声分别代表了连续扰动与离散破坏两类典型输入噪声，观察它们对不同训练模型的影响能较全面地评估不同训练方

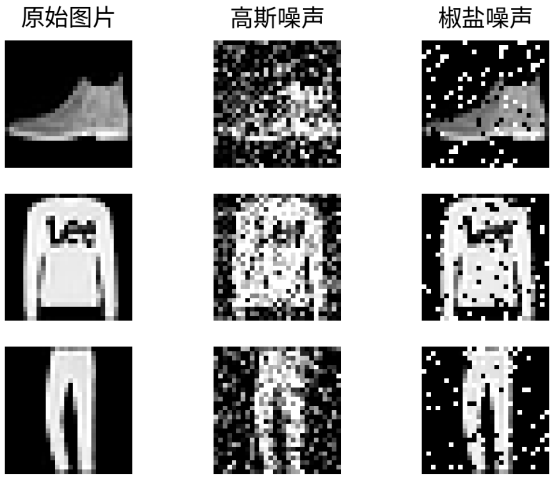

图 4-3　带噪声的 Fashion-MNIST 数据集示意图

式对输入扰动的鲁棒性。实验结果如下表4-5所示：

表 4-5　加噪声干扰的 MNIST 和 Fashion-MNIST 数据集实验结果

数据集	噪声类型	训练算法	准确率下降
MNIST	高斯噪声	全局替代梯度 BP	7.0%
		HSIC 瓶颈赫布学习	4.4%
	椒盐噪声	全局替代梯度 BP	4.1%
		HSIC 瓶颈赫布学习	2.4%
Fashion-MNIST	高斯噪声	全局替代梯度 BP	12.3%
		HSIC 瓶颈赫布学习	8.6%
	椒盐噪声	全局替代梯度 BP	8.5%
		HSIC 瓶颈赫布学习	6.4%

其中，准确率下降值定义为添加噪声前后模型测试准确率的差值，反映噪声对模型性能的干扰程度，下降越小代表鲁棒性越强。实验结果表明，在所有测试条件下，基于 HSIC 瓶颈的局部赫布式学习方法在准确率下降幅度上均显著优于全局替代梯度 BP 方法。例如，在 MNIST 数据集上添加高斯噪声时，传统 BP 方法准确率下降达 7.0%，而 HSIC 方法仅下降 4.4%；在椒盐噪声下，准确率下降分别为 4.1% 和 2.4%。类似地，在更复杂的 Fashion-MNIST 数据集上，HSIC 方法在高斯噪声和椒盐噪声条件下的准确率下降分别为 8.6% 和 6.4%，相较于 BP 的 12.3% 和 8.5% 显著更小。上述结果表明，HSIC 瓶颈训练策略能够有效限制脉冲信号中的冗余信息传播，提升模型对输入扰动的免疫能力，从而在面临噪声时表现出更强的鲁棒性。这也进一步支持了在 SNN 结构中采用基于信息瓶颈学习范式的局部可塑性规则进行训练的可行性与优势。

4.3.5 和相关工作的比较分析

本章的这项工作和前一项工作都是基于这样的一个出发点：之前的高性能脉冲神经网络训练算法是基于全局误差反向传播 BP 算法的，或多或少面临生物合理性低、权重对称、更新锁定、计算能耗大、梯度消失、对神经形态硬件不友好等问题。为了解决或避免上述的问题，研究者尝试了一系列非 BP 式算法，用于 SNN 的训练学习，这些方法正如在第二章相关工作部分中所讲的那样，包括最早的使用随机反馈矩阵传递误差信息的反馈对齐类算法，构造逐层目标值反向传播的目标传播类算法、基于能量模型和隐式微分的平衡传播算法、利用前向信号生成局部误差驱动的前向传播算法等等，以及本论文提出的混合损失分层局部学习算法、HSIC 瓶颈局部学习算法。以上这些算法的特征和性能的比较见下表：

表 4-6 SNN 非 BP 式训练算法对比

方法	局部算法	非权重对称	非更新锁定	异步可并行化	有竞争力的精度
surrogate-BP	✗	✗	✗	✗	✓
FA	✗	✓	✗	✗	✗
DFA	✓	✓	✗	✗	✗
TP	✓	✓	✗	✗	✓
Eprop	✓	✓	✗	✗	✓
STDP	✓	✓	✓	✓	✗
SNN-HLL	✓	✓	✓	✓	✓
SNN-HBH	✓	✓	✓	✓	✓

简单解释一下表4-6，✓ 表示满足某一项，✗ 表示不满足某一项，“有竞争力的精度”指在标准数据集如 Fashion-MNIST 或 CIFAR-10 上达到与 surrogate-BP 相近或略低的准确率。其中 BP 算法（对应 SNN 中的替代梯度下降法）虽然具有较高的模型精度，但在局部算法、非权重对称、非更新锁定、异步可并行化四个方面全都不满足，不是具有生物合理性的低能耗局部训练算法。其他几种对照工作：脉冲时序依赖可塑性（STDP）、反馈对齐算法（FA）、直接反馈对齐算法（DFA）、目标传播算法（TP）、平衡传播算法（Eprop）等只能够部分满足几个条件。例如 DFA 算法能够解决“权重对称”问题，但是其网络某一层的更新依然要等到整个前向传播阶段结束后才可以进行，还是存在“更新锁定”的问题，且该算法无论是在 ANN 还是 SNN 上，性能都和基于 BP 的算法有明显差距，精度较差不具有竞争力。总而言之，这几种可用于脉冲神经网络训练的非 BP 式

算法，它们的生物合理性比全局误差 BP 算法更高，能部分解决 BP 存在的一些问题，但仍不及本文的两个算法，因为本文的混合损失局部学习算法（HLL）和 HSIC 瓶颈学习算法可以满足表中全部条件，即不光具有较好的生物合理性和并行化潜力，而且取得了不错的模型精度。

4.4　本章小结

本章依旧遵循非全局 BP 式的思路，提出了一种全新的基于信息瓶颈的变体——HSIC 瓶颈的局部损失，成功地用它设计出一种新的分层局部脉冲神经网络训练算法——SNN-HBH，采用核函数将样本特征通过核映射投射到高维空间，让数据更有可分性，更容易捕捉到变量间的非线性依赖性。进一步地，用数学推导表明了选取特定的核函数（高斯核函数），可以诱导出具有生物合理性的三因素赫布学习形式的更新规则。在常见的图像分类数据集上取得了接近全局反向传播算法的性能，证实了本算法对训练 SNN 的有效性。此外，还设计了噪声实验，验证了基于 HSIC 瓶颈的学习范式可以在面对噪声干扰时表现出更优的鲁棒性，适用于非线性复杂结构或对噪声敏感的任务场景。本工作的算法在精度、生物可解释性与能效之间取得了良好的平衡。

第五章　总结与展望

脉冲神经网络（SNN）是新一代的仿生神经网络模型，由于其生物可解释性和低功耗的特点而备受关注，逐渐成为当前人工智能和神经计算领域的研究热点。与传统的人工神经网络（ANN）不同，脉冲神经网络通过模拟生物神经元的脉冲发放和信息处理机制，更加贴近大脑神经系统的工作原理。相比于 ANN，SNN 能够实现更低的功耗和更高的计算效率，尤其适合应用于脑机接口、动态视觉感知、神经形态硬件、边缘计算等领域，这在深度学习愈发消耗计算资源的当下，显得意义非凡。

脉冲神经网络的有效训练算法一直是该领域的痛点和难点。因为脉冲发放函数自身的不可微特点，脉冲神经网络较难训练，且性能往往逊色于同样的 ANN，在性能和能耗方面难以两头兼顾，如何找到高性能、低功耗的脉冲神经网络训练算法是一个研究热点。本文针对脉冲神经网络训练算法在生物合理性、计算效率与模型精度之间的平衡难题，提出了两种 SNN 的新型局部非 BP 式学习算法，从而避免了 BP 式算法（如替代梯度下降法）计算开销大、梯度消失、近似误差累积、并行效率低、生物合理性差等缺点。

本文的第一项工作是脉冲神经网络混合损失局部学习算法（SNN-HLL），通过逐层构建辅助分类器，避免了全局梯度计算，仅依赖局部误差信号。该方法通过结合预测损失、相似度匹配损失等多种损失函数，降低了计算开销的同时，还取得了高精度，在图像识别数据集上超过了全局误差传播的 BP 类算法。第二种方法是脉冲神经网络 HSIC 瓶颈三因素赫布学习算法（SNN-HBH），引入信息瓶颈学习框架和希尔伯特-施密特独立性准则（HSIC）这一度量变量依赖性（反面为独立性）的指标来减少冗余信息，保留任务相关特征，形成了一种类生物可塑性的学习规则，性能接近传统梯度下降方法。

尽管本工作在脉冲神经网络的具有生物可解释性的、高性能、低功耗训练算法方向取得了一定的研究进展，但还存在一定的不足之处：1. 算法验证主要基于静态图像分类任务，没有在动态 DVS 数据集上实验，尚未在充分体现脉冲神经

网络处理动态时空信号的核心优势。2. 混合损失函数的组合策略以及核函数的选择仍依赖经验性设计，缺乏理论层面的优化指导。3. 当前实验仅在仿真环境完成，尚未在神经形态硬件上实际部署并验证能效优势，本文的实验结果本质上是计算机模拟的结果，现实物理世界中的实际效果有待进一步探究。

针对以上不足，希望在后续的工作中能够努力达成这些目标：1. 在更复杂的网络结构和更大的数据集上（如 CIFAR-100 和 ImageNet）用本文的算法成功训练脉冲神经网络，不局限于静态数据集，而是在包括神经形态数据集（如 N-MNIST 和 DVS128 Gesture）上广泛实验。2. 拓展本研究中的两种非 BP 式局部学习算法，形成非 BP 式有监督 SNN 训练算法体系，扩大这类方法在 SNN 学习算法领域的影响力和地位。3. 减少算法参数设置中的经验性设计，例如混合损失的系数选取应避免人工干预，而是采用如自适应加权、元学习等方法，自动学习比例系数，更加科学有效。4. 在有条件的情况下，将本文的算法部署到神经形态芯片上，验证算法的能效比优势。

参考文献

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

[2] HEBB D O. The organization of behavior: A neuropsychological theory[M]. Psychology press, 2005.

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

[4] MINSKY M, PAPERT S A. Perceptrons, reissue of the 1988 expanded edition with a new foreword by Léon Bottou: an introduction to computational geometry[M]. MIT press, 2017.

[5] WERBOS P. Beyond regression: New tools for prediction and analysis in the behavioral sciences[J]. PhD thesis, Committee on Applied Mathematics, Harvard University, Cambridge, MA, 1974.

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

[7] LECUN Y, BOSER B, DENKER J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551.

[8] ELMAN J L. Finding structure in time[J]. Cognitive science, 1990, 14(2): 179-211.

[9] HINTON G E, OSINDERO S, TEH Y W. A fast learning algorithm for deep belief nets[J]. Neural computation, 2006, 18(7): 1527-1554.

[10] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.

[11] KINGMA D P, WELLING M, et al. Auto-encoding variational bayes[Z]. 2013.

[12] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[13] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020, 33: 1877-1901.

[14] WANG X, LIN X, DANG X. Supervised learning in spiking neural networks: A review of algorithms and evaluations[J]. Neural Networks, 2020, 125: 258-280.

[15] CRICK F. The recent excitement about neural networks[J]. Nature, 1989, 337(6203): 129-132.

[16] DELBRÜCK T, LINARES-BARRANCO B, CULURCIELLO E, et al. Activity-driven, event-based vision sensors[C]//Proceedings of 2010 IEEE international symposium on circuits and systems. 2010: 2426-2429.

[17] MAASS W. Networks of spiking neurons: the third generation of neural network models[J]. Neural networks, 1997, 10(9): 1659-1671.

[18] DAMPFHOFFER M, MESQUIDA T, VALENTIAN A, et al. Backpropagation-based learning techniques for deep spiking neural networks: A survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023.

[19] HODGKIN A L, HUXLEY A F. A quantitative description of membrane current and its application to conduction and excitation in nerve[J]. The Journal of physiology, 1952, 117(4): 500.

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

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

[22] NUNES J D, CARVALHO M, CARNEIRO D, et al. Spiking neural networks: A survey[J]. IEEE access, 2022, 10: 60738-60764.

[23] NIU L Y, WEI Y, LIU W B, et al. Research Progress of spiking neural network in image classification: a review[J]. Applied intelligence, 2023, 53(16): 19466-19490.

[24] NEFTCI E O, MOSTAFA H, ZENKE F. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks[J]. IEEE Signal Processing Magazine, 2019, 36(6): 51-63.

[25] CAO Y, CHEN Y, KHOSLA D. Spiking deep convolutional neural networks for energy-efficient object recognition[J]. International Journal of Computer Vision, 2015, 113: 54-66.

[26] MEROLLA P A, ARTHUR J V, ALVAREZ-ICAZA R, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface[J]. Science, 2014, 345(6197): 668-673.

[27] DAVIES M, SRINIVASA N, LIN T H, et al. Loihi: A neuromorphic manycore processor with on-chip learning[J]. Ieee Micro, 2018, 38(1): 82-99.

[28] BOI F, MORAITIS T, DE FEO V, et al. A bidirectional brain-machine interface featuring a neuromorphic hardware decoder[J]. Frontiers in neuroscience, 2016, 10: 563.

[29] YAN Z, ZHOU J, WONG W F. Energy efficient ECG classification with spiking neural network[J]. Biomedical Signal Processing and Control, 2021, 63: 102170.

[30] MOZAFARI M, GANJTABESH M, NOWZARI-DALINI A, et al. Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks[J]. Pattern recognition, 2019, 94: 87-95.

[31] TAVANAEI A, MAIDA A. BP-STDP: Approximating backpropagation using spike timing dependent plasticity[J]. Neurocomputing, 2019, 330: 39-47.

[32] XIAO M, MENG Q, ZHANG Z, et al. Training feedback spiking neural networks by implicit differentiation on the equilibrium state[J]. Advances in neural information processing systems, 2021, 34: 14516-14528.

[33] ZHANG Y, INOUE K, NAKAJIMA M, et al. Training Spiking Neural Networks via Augmented Direct Feedback Alignment[J]. ArXiv preprint arXiv:2409.07776, 2024.

[34] KAISER J, MOSTAFA H, NEFTCI E. Synaptic plasticity dynamics for deep continuous local learning (DECOLLE)[J]. Frontiers in Neuroscience, 2020, 14: 424.

[35] BRUNEL N, VAN ROSSUM M C. Lapicque's 1907 paper: from frogs to integrate-and-fire[J]. Biological cybernetics, 2007, 97(5): 337-339.

[36] ESHRAGHIAN J K, WARD M, NEFTCI E O, et al. Training spiking neural networks using lessons from deep learning[J]. Proceedings of the IEEE, 2023, 111(9): 1016-1054.

[37] GILSON M, MASQUELIER T, HUGUES E. STDP allows fast rate-modulated coding with Poisson-like spike trains[J]. PLoS computational biology, 2011, 7(10): e1002231.

[38] TOVEE M J, ROLLS E T. Information encoding in short firing rate epochs by single neurons in the primate temporal visual cortex[J]. Visual cognition, 1995, 2(1): 35-58.

[39] POUGET A, DAYAN P, ZEMEL R. Information processing with population codes[J]. Nature Reviews Neuroscience, 2000, 1(2): 125-132.

[40] KIM Y, PARK H, MOITRA A, et al. Rate coding or direct coding: Which one is better for accurate, robust, and energy-efficient spiking neural networks?[C]//ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2022: 71-75.

[41] DIEHL P U, COOK M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity[J]. Frontiers in computational neuroscience, 2015, 9: 99.

[42] WU Y, DENG L, LI G, et al. Spatio-temporal backpropagation for training high-performance spiking neural networks[J]. Frontiers in neuroscience, 2018, 12: 331.

[43] SHRESTHA S B, ORCHARD G. Slayer: Spike layer error reassignment in time[J]. Advances in neural information processing systems, 2018, 31.

[44] LILLICRAP T P, COWNDEN D, TWEED D B, et al. Random synaptic feedback weights support error backpropagation for deep learning[J]. Nature communications, 2016, 7(1): 13276.

[45] DELLAFERRERA G, KREIMAN G. Error-driven input modulation: solving the credit assignment problem without a backward pass[C]//International Conference on Machine Learning. 2022: 4937-4955.

[46] NØKLAND A. Direct feedback alignment provides learning in deep neural networks[J]. Advances in neural information processing systems, 2016, 29.

[47] BACHO F, CHU D. Low-variance forward gradients using direct feedback alignment and momentum[J]. Neural Networks, 2024, 169: 572-583.

[48] ZHAO D, ZENG Y, ZHANG T, et al. GLSNN: A multi-layer spiking neural network based on global feedback alignment and local STDP plasticity[J]. Frontiers in Computational Neuroscience, 2020, 14: 576841.

[49] LEE J, ZHANG R, ZHANG W, et al. Spike-train level direct feedback alignment: Sidestepping backpropagation for on-chip training of spiking neural nets[J]. Frontiers in neuroscience, 2020, 14: 143.

[50] SHRESTHA A, FANG H, RIDER D P, et al. In-hardware learning of multilayer spiking neural networks on a neuromorphic processor[C]//2021 58th ACM/IEEE Design Automation Conference (DAC). 2021: 367-372.

[51] LEE D H, ZHANG S, FISCHER A, et al. Difference target propagation[C]// Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part I 15. 2015: 498-515.

[52] TISHBY N, PEREIRA F C, BIALEK W. The information bottleneck method[J]. ArXiv preprint physics/0004057, 2000.

[53] GUO S, LIN T. An efficient non-backpropagation method for training spiking neural networks[C]//2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI). 2021: 192-199.

[54] MA C, XU J, YU Q. Temporal dependent local learning for deep spiking neural networks[C]//2021 International joint conference on neural networks (IJCNN). 2021: 1-7.

[55] WU J, CHUA Y, ZHANG M, et al. Deep spiking neural network with spike count based learning rule[C]//2019 International Joint Conference on Neural Networks (IJCNN). 2019: 1-6.

[56] LEE C, SARWAR S S, PANDA P, et al. Enabling spike-based backpropagation for training deep neural network architectures[J]. Frontiers in neuroscience, 2020, 14: 497482.

[57] KIM Y, PANDA P. Revisiting batch normalization for training low-latency deep spiking neural networks from scratch[J]. Frontiers in neuroscience, 2021, 15: 773954.

[58] HAN B, ROY K. Deep spiking neural network: Energy efficiency through time based coding[C]//European conference on computer vision. 2020: 388-404.

[59] ZHANG T, JIA S, CHENG X, et al. Tuning convolutional spiking neural network with biologically plausible reward propagation[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(12): 7621-7631.

[60] YANG S, CHEN B. SNIB: improving spike-based machine learning using nonlinear information bottleneck[J]. IEEE transactions on systems, man, and cybernetics: Systems, 2023, 53(12): 7852-7863.

[61] PRESKILL J. Quantum Computing in the NISQ Era and Beyond[J/OL]. Quantum, 2018, 2: 79(2018-08-06) [2022-10-27]. https://quantum-journal.org/papers/q-2018-08-06-79/. DOI: 10.22331/q-2018-08-06-79.

[62] LILLICRAP T P, SANTORO A, MARRIS L, et al. Backpropagation and the brain[J]. Nature Reviews Neuroscience, 2020, 21(6): 335-346.

[63] JADERBERG M, CZARNECKI W M, OSINDERO S, et al. Decoupled neural interfaces using synthetic gradients[C]//International conference on machine learning. 2017: 1627-1635.

[64] MA W D K, LEWIS J, KLEIJN W B. The HSIC bottleneck: Deep learning without back-propagation[C]//Proceedings of the AAAI conference on artificial intelligence: vol. 34: 04. 2020: 5085-5092.

[65] NØKLAND A, EIDNES L H. Training neural networks with local error signals[C]//International conference on machine learning. 2019: 4839-4850.

[66] POGODIN R, LATHAM P. Kernelized information bottleneck leads to biologically plausible 3-factor hebbian learning in deep networks[J]. Advances in Neural Information Processing Systems, 2020, 33: 7296-7307.

[67] MA C, YAN R, YU Z, et al. Deep spike learning with local classifiers[J]. IEEE Transactions on Cybernetics, 2022, 53(5): 3363-3375.

[68] REN M, KORNBLITH S, LIAO R, et al. Scaling forward gradient with local losses[J]. ArXiv preprint arXiv:2210.03310, 2022.

致　　谢

时光荏苒，日月如梭，硕士研究生三年的时光过得很快，即将画上句号。在完成本论文之际，我怀着复杂的感情，写下这篇致谢，谨向在我求学和成长过程中给予我关心和帮助的亲人、老师和朋友致以最诚挚的感谢，也向这段求学生涯做不舍的道别。

首先，要感谢我的导师申富饶教授以及徐百乐老师。感谢申老师的鞭策和教诲，申老师不但在学习科研中给予了我耐心指引，还在生活中教会了我不少为人处世的道理，令我受益匪浅。感谢申老师在我从数学专业逐步融入人工智能领域的过程中给予的引导，使我顺利完成了从学习者到研究者的过渡转变。感谢您在百忙之中抽出时间，每周和我进行个人讨论交流科研进展，帮助我明确研究方向、持续前行。感谢徐老师在研三一年对我的悉心指导，您在方法细节打磨和论文撰写等方面给予了我很大的帮助，您宽容而严谨的治学风格让我深受启发，让我学会了以专业的视角完善研究工作。

其次，要感谢读研期间相遇的各位同学。感谢 RINC 组的各位成员，感谢 516 实验室的王翔宇、熊昕、易梦军、郭苏涵等师兄师姐平日里耐心解答我的疑问、无私分享经验，感谢和我同一级的杨洪朝、俞诗航、李若彤、刘佩涵四位同学在学习生活中对我的关心和帮助，让我在科研道路上少一些孤单。感谢我的室友曹博文和陈武洋，无论是深夜的畅谈，还是日常的点滴陪伴，都给我这段求学旅程增添了不少温馨与力量。感谢王春力同学在我实习期间对我的照顾和帮助。

接着，要感谢我的家人。感谢我的爸爸和妈妈，你们永远是我坚强的后盾和温暖的港湾。在我迷茫气馁时，是你们给予了我鼓励和支持，助我渡过难关；在我取得进步时，是你们和我一起分享喜悦，为我感到高兴。

最后，也感谢一下自己。感谢自己在迷茫内耗时厘清现状、找寻目标，在遭遇挫折时克服心中的恐惧、勇敢坚持下来。关关难过关关过，一切终将成为回忆。希望自己在未来的道路上更加勇敢独立、更加自信乐观。我们下一段旅程见！

简历与科研成果

基本信息

胡嘉骏，男，汉族，2000 年 3 月出生，江苏省盐城市射阳县人。

教育背景

2022 年 9 月—2025 年 6 月 南京大学人工智能学院 硕士

2018 年 9 月—2022 年 6 月 南京大学数学系 本科

攻读硕士学位期间的发明专利

1. 徐百乐，**胡嘉骏**，申富饶。《一种具有生物合理性的分层局部混合损失脉冲神经网络训练算法》（202510600130.7）

攻读硕士学位期间参与的科研课题

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

学位论文出版授权书

本人完全同意《中国优秀博硕士学位论文全文数据库出版章程》（以下简称“章程”），愿意将本人的学位论文提交“中国学术期刊（光盘版）电子杂志社”在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版，并同意编入《中国知识资源总库》，在《中国博硕士学位论文评价数据库》中使用和在互联网上传播，同意按“章程”规定享受相关权益。

作者签名：胡嘉骏

2025年 5 月 26日

论文题名	具有生物合理性的脉冲神经网络局部非 BP 训练算法研究				
研究生学号	502022370015	所在院系	人工智能学院	学位年度	2025
论文级别	☑ 学术学位硕士 ☐ 专业学位硕士 ☐ 学术学位博士 ☐ 专业学位博士				
作者 Email	xyz@smail.nju.edu.cn				
导师姓名	申富饶				

论文涉密情况：

☐ 不保密

☐ 保密，保密期（_____年____月____日至_____年____月____日）