

Brain-inspired global-local learning incorporated with neuromorphic computing 复 现报告

摘要

本文提出了一种神经形态全局-局部协同学习模型，将误差驱动的全局学习与神经科学启发的局部学习相结合，以增强多样化学习场景下的学习能力。通过引入一种类脑的元学习范式和一个可微分的脉冲神经模型，结合神经元动力学和突触可塑性，该模型能够元学习局部可塑性，并接收自上而下的监督信息，实现多尺度学习。此外，我们通过加入对抗训练进一步增强模型的鲁棒性和泛化能力。模型在少样本学习、持续学习和神经形态视觉传感器中的容错学习等任务中表现出显著优于单一学习方法的性能。

关键词：Local Plasticity; Global Plasticity; SNN; Hybrid Plasticity

1 引言

现有的大多数神经形态模型是通过单一的反向传播或者由单一的神经科学向导的局部可塑性 (Local Plasticity LP) 建立而成的。其中局部相关驱动学习 (LP) 是基于神经科学的生物机制，通过突出前后神经元异步脉冲活动触发，具有低延迟，高能效的特点但是整体性能不如现有的方法。反向传播是指全局误差驱动学习，即 Global Plasticity (GP)，通过计算整个网络的误差，将误差信号通过各层反向传播，调整神经元之间的权重。这种方法现在广泛应用于卷积神经网络 (CNN) [10]，循环神经网络 (RNN) [11] 等模型中。

1.1 选题背景

受益于深度学习的最新进展 [12] [13] [15]，反向传播以及其许多变体以应用于训练以脉冲神经网络 (SNN) 等为代表的神经形态模型，并且在某些特定任务中表现出了良好的准确性。

然而 GP 方法和 LP 方法虽然各有优势，但是在复杂的学习任务中，单一方法难以胜任，因此开发能够综合两者优势的混合学习模型 (hybrid model) 具有重要的研究意义，但存在技术难点，需要解决脉冲电路的差异以及权重更新行为的不同等。

1.2 选题意义

该选题提出的方法能够推进计算的前沿发展，文章将 GP 和 LP 结合并突破现有的模型局限，提出了一种基于脉冲的混合可塑性模型 (Hybrid Plasticity)，基于大脑启发的元学习

范式 [6], 结合差分脉冲动力学模型和参数化 LP, 并且提供了 GP 和 LP 的通用, 灵活的融合机制。推动神经形态计算的进一步发展, 使其可以应用到高能效和低延迟的任务中。让神经形态模型在复杂的学习任务中表现出更强的鲁棒性和适应性。通过模拟生物大脑的学习过程, 促进跨学科领域的融合, 推动神经科学与人工智能的交叉研究。

1.3 选题依据

混合学习的方式 (结合 GP 和 LP) 能够提高大模型的计算效率、增强鲁棒性、提升适应性, 并支持更复杂的学习任务, 尤其在硬件加速与并行化方面具备优势。文章的提出为神经形态大模型的高效训练和应用提供了新的思路, 尤其是在处理大规模、动态复杂任务时, 能够在保证高性能的同时实现低功耗和高效计算。因此, 选择了这篇文章进行学习复现。

2 相关工作

近年来, 关于全局-局部混合模型的研究不断增多, 研究主要关注如何结合全局误差驱动学习与局部可塑性来提升学习效率和表现。

2.1 三因子学习规则

三因子学习规则 [3] [7] 描述了突触可塑性的一个通用框架, 结合了突触前, 突触后和第三因素 (如神经递质等物质), 代表了自上而下的监督信号, 规则启发了许多基于 SNN 的算法学习的研究, 并利用这一规则将监督误差或奖励作为 “第三因素”, 进行调节局部权重更新, 并成功应用于图像分类、强化学习 (RL) 任务和概率推理任务等。

2.2 元学习

元学习 (Meta-learning) 也是相关的一个研究方向 [4] [1], 主要关注通过学习如何学习来持续改进学习能力。然而, 现有的元学习研究大多仅聚焦于改进基于全局梯度 (GP) 的单一模型, 未能有效结合局部可塑性 (LP)。如何同时优化这两种学习机制并将其结合, 仍是一个重要的开放问题。

2.3 优化局部可塑性 (LP)

1990 年代早期, 研究者提出了一种通过监督信号优化局部可塑性 (LP) 的框架 [7], 并在后续的研究中将这一框架扩展到大型非脉冲神经网络 (non-SNN), 在少样本学习和无监督学习等任务中取得了较高的表现。然而, 由于缺乏有效和可配置的混合机制, 全局-局部学习的潜力未能得到充分发挥, 尤其在一些关键学习能力和计算效率上仍有待提升。

2.4 对抗性样本

进行代码复现时使用了 FGSM (fast gradient sign method) [8] 方法, 一种基于梯度生成对抗样本 [2] 的算法, 通过利用模型的梯度信息, 快速生成对抗样本, 使模型的预测结果发生错误, 使用对抗样本对模型进行训练, 提高模型的鲁棒性, 防止被对抗样本攻击。

3 本文方法

文章提出了一种基于脉冲的混合可塑性模型，旨在将局部可塑性（LP）和全局学习（GP）结合起来，利用神经科学实验中的启示和多尺度学习机制，创建一个协同学习回路。

3.1 神经调节机制与多尺度学习

大脑的海马区局部神经回路可以收到多种自上而下的调节信号控制 [5]，这些信号通过作用于突触来调节突触可塑性行为，包括学习率、更新极性和可塑性整合等。这表明神经调节因子可以被看作是影响突触可塑性的特殊类型的元学习参数，并且这些调节因子在学习过程中扮演着重要角色。神经调节因子和突触可塑性之间存在时间尺度上的耦合关系，神经调节因子在不同时间尺度上与突触活动相互作用，从而实现更复杂的行为模式 [14]。因此，学习过程中，神经调节因子和突触之间的学习可能发生在不同的空间和时间尺度上，形成多尺度的学习机制。

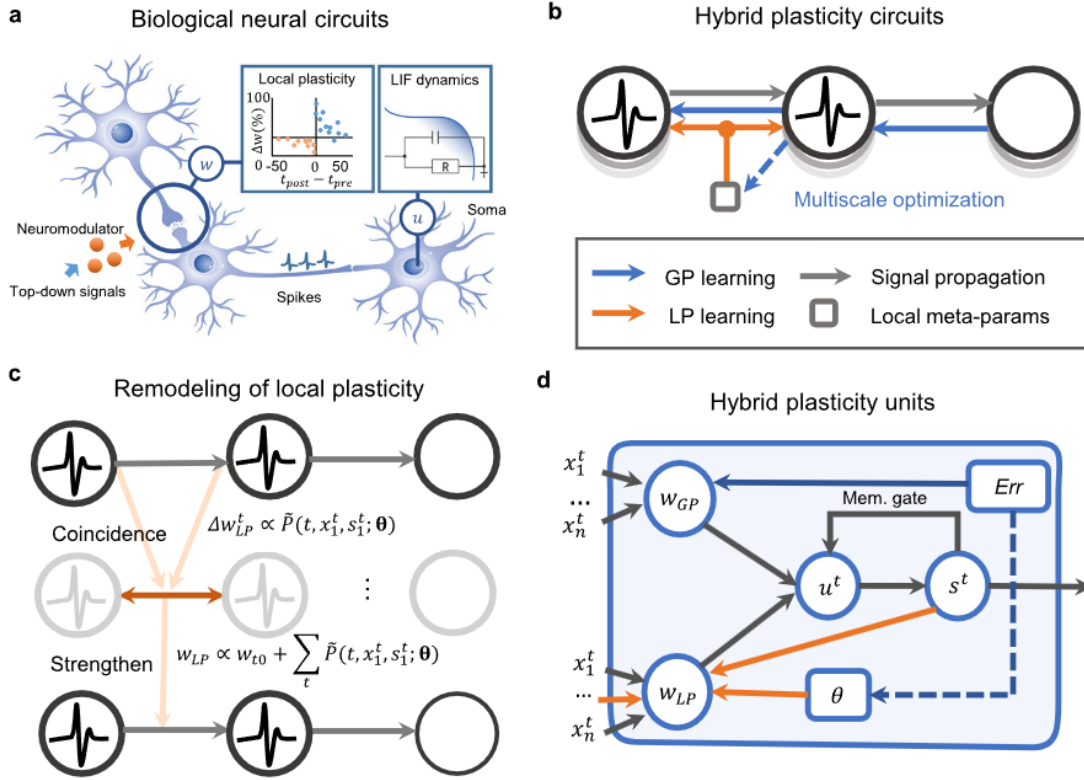


图 1. 混合协同学习模型的图示 a: 生物突触可塑性机制与神经元动态的示意图;b: 多尺度元学习范式;c: 参数化生物学短期可塑性;d:HP 单元的结构

3.2 元参数建模与双层优化

为了实现这种多尺度学习机制，通过将局部可塑性（LP）的超参数建模为一组元参数，涵盖内容如公式1所示

$$\theta = \{Local \ Learning \ Rate, \beta_i, Other \ Hyperparameters\} \quad (1)$$

这些元参数控制突触权重的更新行为。研究提出将这些元参数 θ 作为权重 w 的上层变量，形成一个双层优化问题，将突触权重的学习与神经调节因子的优化分开进行。在此框架下， θ 和 w 的学习过程被解耦，分别在不同的优化循环中进行。这使得可以采用超参数优化技术来优化 θ ，从而实现对局部学习规则和权重的联合优化。

3.3 引入脉冲动态和全局学习

为了将 LP、全局学习和多样化的脉冲动态整合到一个统一的时间信用分配问题中，研究基于膜电位动态和离子通道动态导出了一个可微分的信号传播方程，如公式2所示，以处理脉冲神经元的断点动态。

$$\frac{du_i(t)}{dt} = -u_i(t) + \sum_{j=1}^{n_i} w_{ij}(t)s_j(t) \quad (2)$$

其中 $u_i(t)$ 表示神经元 i 在时刻 t 的膜电位。它是神经元的激活程度或响应值，决定了神经元是否会发放脉冲， $s_j(t)$ 表示神经元 j 在时刻 t 的脉冲输出，通常表示为脉冲数量。这个方程的含义是，神经元 i 的膜电位变化是由两个部分决定的：一部分是它自身的衰减 $-u_i(t)$ ，另一部分是它从其他神经元接收到的输入信号。其突触权重的更新通常是基于 Hebbian 学习规则，如公式3所示

$$\tau_w \frac{dw_{ij}(t)}{dt} = w_{g_{ij}} - w_{ij}(t) + P(t, \text{pre}_j(t), \text{post}_i(t), \theta) \quad (3)$$

其中由于 $P(*)$ 对于局部学习的建模规则是通用的，文章提供了一个 Hebbian 规则的变体，如公式4所示

$$P \triangleq k_{\text{corr}} s_j(t) (\rho(u_i(t)) + \beta_i) \quad (4)$$

其中 k_{corr} 是权重超参数， $\rho(x)$ 是有界非线性函数，根据并发的突触前放电和突触后膜活动来更新权重方程3，可以得到公式5

$$w_{ij}(t) = w_{ij}(t_{n0})e^{-\frac{t-t_{n0}}{\tau_w}} + \int_{t_{n0}}^t P(x, \text{pre}_j(x), \text{post}_i(x), \theta)e^{-\frac{t-x}{\tau_w}} dx \quad (5)$$

由于本文提出的 HP 方法使用潜在的轨迹而不是平衡状态进行计算，因此对初始参数 $w_{ij}(t_{n0})$ 的依赖性非常重要。因此假设 HP 方法可以通过修改某个时间 t_{n0} 的相位权重值 $w_{ij}(t_{n0})$ 来进行监督学习，从而进行自上而下调制信号的形式。将上述方程带入到膜电位方程 $S_j(t)$ 中可以得到公式6

$$\tau_u \frac{du_i}{dt} = -u_i(t) + \sum_{j=1}^{l_n} s_j(t)w_{ij}(t_{n0})e^{-\frac{t-t_{n0}}{\tau_w}} + \sum_{j=1}^{l_n} \left(\int_{t_{n0}}^t P(x, \text{pre}_j(x), \text{post}_i(x), \theta)e^{-\frac{t-x}{\tau_w}} dx \right) s_j(t) \quad (6)$$

为了使用连续动力学与反向传播相兼容，添加了欧拉方法并带入 $P(x)$ 表达式4，整理得到公式7

$$\begin{cases} u_i^l(t_{m+1}) = (1 - s_i^l(t_m))(1 - k_u)u_i^l(t_m) + k_u \sum_{j=1}^{l_n} s_j^{l-1}(t_m)w_{ij}^l(t_0)e^{-\frac{t_m-t_0}{\tau_w}} \\ \quad + k_u \sum_{j=1}^{l_n} s_j^{l-1}(t_m) \sum_{t_f^i, t_f^j < t_m} k_{\text{corr}}^l H_j^{l-1}(t_m - t_f^j) \times (\rho(u_i^l(t_m)) + \beta_i)e^{-\frac{t_m-t_f^j}{\tau_w}}, \\ s_i^l(t_m) = H(u_i^l(t_m) - v_{\text{th}}) \end{cases} \quad (7)$$

其中 $k_u \triangleq \frac{dt}{\tau_u}$ 上索引的 l 表示第 l_{th} 层, $H(x)$ 是 Heaviside 函数确定的发射函数, 随后通过使用记忆门和适当的导数近似方法, 研究能够有效处理脉冲神经元的不连续动态, 如公式8所示。

$$\begin{cases} u_i^l(t_m) = (1 - s_i^l(t_{m-1}))(1 - k_u)u_i^l(t_{m-1}) + k_u \sum_{j=1}^{l_n} (w_{ij}^l(t_{n0})e^{-\frac{t_m-t_0}{\tau_w}} + \alpha_i^l P_{ij}^l(t_m))S_j^{l-1}(t_m) \\ P_{ij}^l(t_m) = P_{ij}^l(t_{m-1})e^{-\frac{dt}{\tau_w}} + \eta_j^l S_j^{l-1}(t_m) \left(\rho(u_i^l) \beta_i^l \right) \\ s_i^l(t_m) = H(u_i^l(t_m) - v_{th}) \end{cases} \quad (8)$$

这里 dt 表现时间步长, α^l 表示控制局部模块的影响, η_j^l 用来控制局部学习率。因此在这里元模块的 θ^l 公式可以划分为一组参数 $\{\alpha^l \eta^l \beta^l\}$

为了解决分类输出, 文章采用了 one-hot 编码, 用输出层的 N 个神经元来表示分类结果, 并将不同的脉冲编码方案合并到一个通用框架, 进行损失计算, 由函数 C 表示9

$$C(w, \theta) \triangleq C(y, \sum_{m=1}^T Z_{t_m} q(u^{n_l}(t_m))) \quad (9)$$

其中 y 是真实值, n_l 是层数, T 是模拟窗口, C 是分类损失。

研究采用反向传播通过时间 (BPTT) 算法作为全局学习方法来训练脉冲神经网络。由于 LP 具有基于相关性的独立更新方式, 直接将局部模块与全局学习结合会导致难以收敛, 因此提出了通过参数建模策略, 将局部突触增量转化为一个与突触前后活动及局部超参数 θ 相关的时间函数, 确保局部权重更新能有效集成到整体优化框架中, 最终可得到表达式10

$$\nabla_{\theta_k} \tilde{C} = \sum_{\pi_i \in \mathcal{I}_s} \nabla_{\theta_k} C_{\pi_i}^{\text{val}}(\mathbf{w}^*, \theta_k) \approx \sum_{\pi_i \in \mathcal{I}_s} \nabla_{\theta_k} C_{\pi_i}^{\text{val}}(\mathbf{w}_k - \xi \nabla_{\mathbf{w}} C_{\pi_i}^{\text{train}}(\mathbf{w}_{k-1}, \theta_{k-1}), \theta_k), \quad (10)$$

3.4 损失函数优化

文章首先定义了一个综合的损失函数 E11, 包含显示的分类损失 C 和由局部模块生成的损失 E_{in}

$$E \triangleq C(t, \mathbf{x}, y; \lambda_1 \mathbf{w}_{GP,t}, \lambda_2 \mathbf{w}_{LP,t}, \boldsymbol{\theta}) + \lambda_3 E_{in}(t, \{\mathbf{f}^{pre}, \mathbf{f}^{post}\}, \mathbf{w}^l, \boldsymbol{\theta}^l, n_l|_{l=2}), \quad (11)$$

其中 $C(t, x, y)$ 是显式的分类损失, w_{GP} 和 w_{LP} 分别是全局可塑性和局部可塑性的权重, E_{in} 是隐式损失, 表示局部模块对网络优化的影响。

3.4.1 隐式损失分析

在容错学习中, 局部权重增量 Δw_{ij}^l 被看作是隐式损失函数 E_{in} 对权重的导数。假设局部权重的更新遵循 Hebbian 规则12

$$\frac{\partial E_{in}}{\partial w_{ij}^l} \propto \Delta w_{ij}^l \propto s_{l-1}^{j,t} \rho(u_i^l) \quad (12)$$

其中, $\rho(u_i^l) = H(u_i^l - v_{th})$ 是 Heaviside 函数, 用于描述神经元是否激活, 通过这个式子, 可以得到隐式损失函数 E_{in} 13

$$E_{in} = - \sum_{t=1}^T \sum_{l=2}^{n_l} s_{l-1}^{T,t} w_l^t s_l^t \quad (13)$$

其中, $H(u_i^l - v_{th})$ 导数为零, 表示只有当神经元激活时, 损失才会改变。这个损失函数与 HAM (Heaviside Activation Models) 中使用的能量函数相似。通过优化能量表面, 局部模块帮助网络对先前的关联模式进行编码, 使得网络能够适应局部最小值, 从而更好地应对输入模式的变化。

3.4.2 少样本学习中隐式损失分析

在少样本学习中, 局部模块利用先前学习到的类间关系来加速收敛。在给定训练样本集 $D = \{(x_k, y_k)\}_{k=1}^N$, 构造一个类似于 Hebbian 矩阵 [9] 的权重 w_{LP} 14

$$w_{LP} = \sum_{k=1}^K \eta_k y_k x_k^T \quad (14)$$

其中, η_k 是归一化的学习率, c_k 是类的样本均值。当查询样本 \tilde{x} 输入时, 局部模块通过内积计算出增强信息 $I_{LP}(\tilde{x})$ 15

$$I_{LP}(\tilde{x}) = \sum_k y_k (c_k^T \tilde{x}) \quad (15)$$

这种增强信息帮助网络根据查询样本与各类样本均值之间的相似性来加速分类。

3.4.3 网络输出的膜电位

在模型中, 网络的输出由 GP 和 LP 两部分加权得到 16

$$u_{nl}^i(t) = \lambda_1 I_{GP, i}(t_m) + (1 - \lambda_1) I_{LP} \quad (16)$$

其中, I_{GP} 和 I_{LP} 分别表示生成过程和局部模块的影响。在训练初期, LP 模块的影响较大, 从而帮助网络更好地利用先前模式的信息进行分类

3.4.4 最终损失计算

为了最小化损失函数 L , 网络需要在类内样本之间保持较小的距离, 而在类间样本之间保持较大的距离, 从而优化分类性能 17。

$$\begin{aligned} L(\mathbf{u}^l, \mathbf{y}) &= \sum_{q \neq i} (\lambda_1 I_{GP, q} + (1 - \lambda_1) (\mathbf{c}_q^T \tilde{\mathbf{x}}))^2 + (1 - ((\lambda_1 I_{GP, i} + (1 - \lambda_1) (\mathbf{c}_i^T \tilde{\mathbf{x}})))^2 \\ &= \sum_{q \neq i} (\lambda_1 I_{GP, q} + (1 - \lambda_1) \mathbf{c}_q^T \tilde{\mathbf{x}})^2 + (1 - \mathbf{c}_i^T \tilde{\mathbf{x}} - \lambda_1 ((I_{GP, i} - \mathbf{c}_i^T \tilde{\mathbf{x}})))^2, \end{aligned} \quad (17)$$

4 复现细节

4.1 与已有开源代码对比

文章源码在 <https://github.com/yjwu17/Spike-Based-Hybrid-plasticity-Network.git> 可以获取到, 本文在复现文章原有的基础上对代码的鲁棒性进行了增强, 增加了对抗网络模块, 使用 FGSM, 源码如算法 1 所示

Algorithm 1 FGSM Attack

Model: $f(\cdot)$, Criterion: $\mathcal{L}(\cdot, \cdot)$, Images: \mathbf{x} , Labels: \mathbf{y} , Hebb1: **hebb1**, Hebb2: **hebb2**, Epsilon: ϵ
Adversarial Images: \mathbf{x}'

Step 1: One-hot encode the labels

$\mathbf{y}' \leftarrow \text{torch.zeros}(\text{batch_size}, 10).\text{scatter_}(1, \mathbf{y}.\text{view}(-1, 1), 1)$

Step 2: Enable gradient computation for images $\mathbf{x}.\text{requires_grad_}()$

Step 3: Forward pass through the model to obtain outputs $\mathbf{out} \leftarrow \text{model}(\mathbf{x}, \mathbf{hebb1}, \mathbf{hebb2})$

Step 4: Compute the loss $\text{loss} \leftarrow \mathcal{L}(\mathbf{out}, \mathbf{y}')$

Step 5: Perform backward pass to compute gradients $\text{loss.backward}()$

Step 6: Generate adversarial images using FGSM $\mathbf{x}' \leftarrow \mathbf{x} + \epsilon \times \text{sign}(\mathbf{x}.\text{grad})$

Step 7: Return the adversarial images \mathbf{x}'

通过将模型的输出预测增加一个微小的扰动值,生成对抗样本进行训练。通过引入 FGSM,对训练的损失函数的输入梯度进行训练,设置一个小的扰动,使用该梯度来生成对抗样本。如公式18所示

$$x' = x + \epsilon \cdot \text{sign}(\nabla_z L(u^l, y)) \quad (18)$$

4.2 实验环境搭建

项目实验环境采用 RTX 3060Ti 显卡,在 Python 环境下,使用图像分类数据集 (MNIST、Fashion-MNIST 和 CIFAR10) 和神经形态数据集 (CIFAR10-DVS and DVS-gesture) 进行训练和测试。

4.3 界面分析与使用说明

将项目部署到本地后,根据工程项目中的 requirement.txt 进行配置环境,随后找到 main_fashion_hp_gan.py 脚本进行运行即可。

4.4 创新点

本文的创新性在于成功地将局部可塑性 (local plasticity) 和全局可塑性 (global plasticity) 相结合,构建了一个更加灵活且适应性更强的模型。通过这一融合,模型不仅能够实现更高效的学习能力,同时也在能耗方面表现出色,实现了低能耗且高效的执行。特别是在神经网络的训练和推理过程中,通过优化可塑性机制,有效提升了模型在多种任务中的表现,尤其是在动态环境中的适应性。

在模型的复现和实验过程中,进一步引入了对抗样本 (adversarial samples) 来增强模型的鲁棒性。对抗样本的加入,使得原本容易受到干扰的脉冲神经网络在面对输入扰动时表现得更加稳健。通过在 SNN 中加入对抗训练,不仅提升了模型在面对不同攻击方式时的防御能力,还增强了其在复杂场景中的稳定性和泛化能力。对抗训练的引入有效地增强了模型的鲁棒性,减少了由于数据噪声或对抗扰动导致的性能下降,从而为模型的进一步优化奠定了基础。

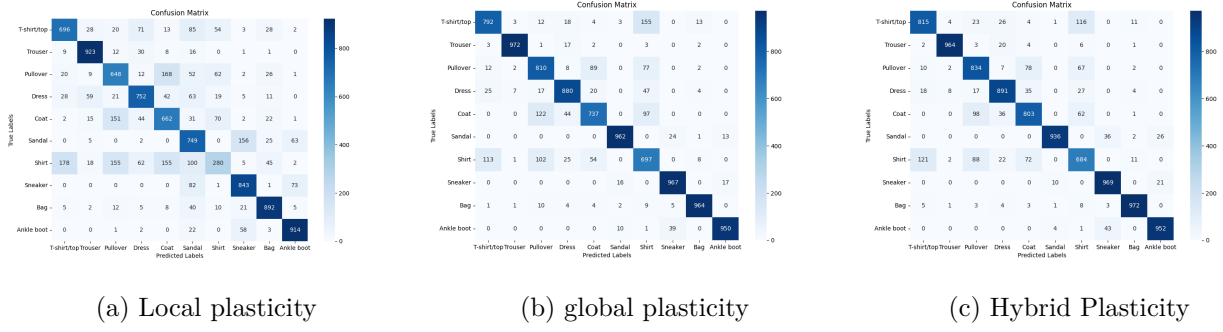


图 2. 不同的训练方法的 SNN 在没有噪声干扰的混淆矩阵分类情况

5 实验结果分析

本部分对实验所得结果进行分析，详细对实验内容进行说明，实验结果进行描述并分析。通过对文章代码复现，可以观察到文章所提出的混合可塑性模型的方法在准确率上明显优于仅进行局部可塑性或仅进行全局可塑性训练的效果，如表1所示

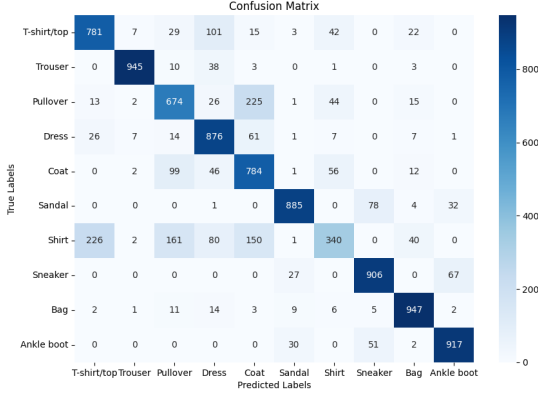
表 1. LP,GP 和 HP 在不同强度噪声干扰下准确率对比

Noise intensity	Local plasticity	global plasticity	Hybrid Plasticity
0.00	73.59	87.31	88.20
0.01	72.57	81.45	82.65
0.02	71.02	76.38	78.70
0.03	69.92	71.52	74.71
0.04	68.40	67.94	72.56
0.05	66.73	64.24	69.60
0.06	65.74	61.18	67.68

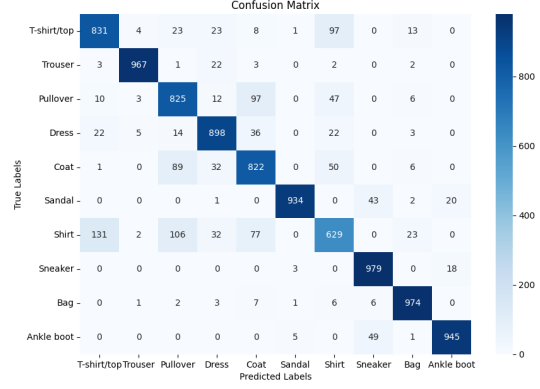
图3中给出了在 LP，GP 和 HP 三种训练方法下在没有噪声干扰的情况的测试集混淆矩阵分类情况。

文章所提出来的方法以及 LP 和 GP 当噪声强度增高时，准确率都会有一定程度的下降，基于这个问题，在进行复现代码时进行了修改，引入了对抗样本。进行了将扰动系数设为 0.1 以及 0.01 时的测试，测试数据的混淆矩阵如图所示。

可以分析得出对抗样本的扰动系数过大时，对整个网络具有负面效果。最终我们选定将扰动系数设置为 0.01 训练效果如表2所示



(a) 扰动系数 0.1



(b) 扰动系数 0.01

图 3. 不同的扰动系数下的混淆矩阵

表 2. 原文提出方法与增加对抗样本后方法的准确率

Noise intensity	Origin	Our
0.00	88.20	88.04
0.01	82.65	87.35
0.02	78.70	86.54
0.03	74.71	86.27
0.04	72.56	85.37
0.05	69.60	85.10
0.06	67.68	84.72

可以从表中观察到通过增加对抗样本后，整个模型的鲁棒性有了显著的提升，面对不同强度的噪声，都可以将准确性保持在一个较高的水准。

6 总结与展望

本文提出了一种基于脉冲神经网络（SNN）的混合协同学习模型，通过引入受海马可塑性机制启发的脑启发元学习范式，结合了基于局部学习的脉冲神经网络和全局监督信号的元学习方法，创新性地将局部学习参数视为可通过全局监督信号优化的元参数。这种协同学习策略可以快速适应不同的学习场景，并在多个顺序分类任务中显著提升性能。并且在进行复现时引入了对抗样本，进一步提升了模型的鲁棒性。

本研究为协同学习模型和神经形态计算的结合开辟了新路径，展示了其在多任务学习、高效能耗计算和硬件实现方面的潜力。这一方向的持续发展有望推动人工智能算法与硬件的协同进化，并为下一代类脑智能系统奠定基础。

参考文献

- [1] Maria Elena Acevedo-Mosqueda, Cornelio Yáñez-Márquez, and Marco Antonio Acevedo-Mosqueda. Bidirectional associative memories: Different approaches. ACM Computing Surveys (CSUR), 45(2):1–30, 2013.
- [2] Naveed Akhtar, Ajmal Mian, Navid Kardan, and Mubarak Shah. Advances in adversarial attacks and defenses in computer vision: A survey. IEEE Access, 9:155161–155196, 2021.
- [3] Craig H Bailey, Maurizio Giustetto, Yan-You Huang, Robert D Hawkins, and Eric R Kandel. Is heterosynaptic modulation essential for stabilizing hebbian plasticity and memory. Nature Reviews Neuroscience, 1(1):11–20, 2000.
- [4] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. Advances in neural information processing systems, 31, 2018.
- [5] György Buzsáki and Kenji Mizuseki. The log-dynamic brain: how skewed distributions affect network operations. Nature Reviews Neuroscience, 15(4):264–278, 2014.
- [6] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. Ieee Micro, 38(1):82–99, 2018.
- [7] Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: experimental support of neo-hebbian three-factor learning rules. Frontiers in neural circuits, 12:53, 2018.
- [8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.
- [9] Dmitry Krotov and John J Hopfield. Unsupervised learning by competing hidden units. Proceedings of the National Academy of Sciences, 116(16):7723–7731, 2019.
- [10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.
- [11] Zachary Chase Lipton. A critical review of recurrent neural networks for sequence learning. arXiv Preprint, CoRR, abs/1506.00019, 2015.
- [12] Hesham Mostafa. Supervised learning based on temporal coding in spiking neural networks. IEEE transactions on neural networks and learning systems, 29(7):3227–3235, 2017.
- [13] Nitin Rathi and Kaushik Roy. Diet-snn: Direct input encoding with leakage and threshold optimization in deep spiking neural networks. arXiv preprint arXiv:2008.03658, 2020.

- [14] Katharina Anna Wilmes and Claudia Clopath. Inhibitory microcircuits for top-down plasticity of sensory representations. Nature communications, 10(1):5055, 2019.
- [15] Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. Neural computation, 30(6):1514–1541, 2018.