

# 一种基于随机权神经网络的类增量学习与记忆融合方法

李德鹏<sup>1,2,3</sup> 曾志刚<sup>1,2,3</sup>

**摘要** 连续学习 (Continual learning, CL) 多个任务的能力对于通用人工智能的发展至关重要. 现有人工神经网络 (Artificial neural networks, ANNs) 在单一任务上具有出色表现, 但在开放环境中依次面对不同任务时非常容易发生灾难性遗忘现象, 即联结主义模型在学习新任务时会迅速地忘记旧任务. 为了解决这个问题, 将随机权神经网络 (Random weight neural networks, RWNNs) 与生物大脑的相关工作机制联系起来, 提出一种新的再可塑性启发的随机化网络 (Metaplasticity-inspired randomized network, MRNet) 用于类增量学习 (Class incremental learning, Class-IL) 场景, 使得单一模型在不访问旧任务数据的情况下能够从未知的任务序列中学习 & 记忆融合. 首先, 以前馈方式构造具有解析解的通用连续学习框架, 用于有效兼容新任务中出现的新类别; 然后, 基于突触可塑性设计具备记忆功能的权值重要性矩阵, 自适应地调整网络参数以避免发生遗忘; 最后, 所提方法的有效性和高效性通过 5 个评价指标、5 个基准任务序列和 10 个比较方法在类增量学习场景中得到验证.

**关键词** 连续学习, 灾难性遗忘, 随机权神经网络, 再可塑性启发

**引用格式** 李德鹏, 曾志刚. 一种基于随机权神经网络的类增量学习与记忆融合方法. 自动化学报, 2023, 49(12): 2467–2480

**DOI** 10.16383/j.aas.c220312

## A Class Incremental Learning and Memory Fusion Method Using Random Weight Neural Networks

LI De-Peng<sup>1,2,3</sup> ZENG Zhi-Gang<sup>1,2,3</sup>

**Abstract** The ability to continual learning (CL) on multiple tasks is crucial for the development of artificial general intelligence. Existing artificial neural networks (ANNs) performing well on a single task are prone to suffer from catastrophic forgetting when sequentially fed with different tasks in an open-ended environment, that is, the connectionist models trained on a new task could rapidly forget what was learned previously. To solve the problem, this paper proposes a new metaplasticity-inspired randomized network (MRNet) for the class incremental learning (Class-IL) scenario by relating random weight neural networks (RWNNs) with the relevant working mechanism of biological brain, which enables a single model to learn and remember the unknown task sequence without accessing old task data. First, a general continual learning framework with the closed-form solution is constructed in a feed-forward manner to effectively accommodate new categories emerging in new tasks; Second, a memory-related weight importance matrix is formed by referring to the property of synapses, which adaptively adjusts network parameters to avoid forgetting; Finally, effectiveness and efficiency of the proposed method are demonstrated in the class incremental learning scenario with 5 evaluation metrics, 5 benchmark task sequences, and 10 comparison methods.

**Key words** Continual learning (CL), catastrophic forgetting, random weight neural networks (RWNNs), metaplasticity-inspired

**Citation** Li De-Peng, Zeng Zhi-Gang. A class incremental learning and memory fusion method using random weight neural networks. *Acta Automatica Sinica*, 2023, 49(12): 2467–2480

收稿日期 2022-04-21 录用日期 2022-07-21

Manuscript received April 21, 2022; accepted July 21, 2022  
科技部科技创新 2030 重大项目 (2021ZD0201300), 中央高校基本科研业务费专项资金 (YCJJ202203012), 国家自然科学基金 (U1913602, 61936004), 111 计算智能与智能控制项目 (B18024) 资助  
Supported by National Key Research and Development Program of China (2021ZD0201300), Fundamental Research Funds for the Central Universities (YCJJ202203012), National Natural Science Foundation of China (U1913602, 61936004), and 111 Project on Computational Intelligence and Intelligent Control (B18024)

本文责任编辑 金连文

Recommended by Associate Editor JIN Lian-Wen

1. 华中科技大学人工智能与自动化学院 武汉 430074 2. 华中科技大学人工智能研究院 武汉 430074 3. 图像信息处理与智能控制教育部重点实验室 武汉 430074

人和其他高级动物可以在其一生中学习并记住许多技能, 这种连续学习不同任务的能力被称为连续学习 (Continual learning, CL). 相比之下, 人工神经网络 (Artificial neural networks, ANNs) 在接受新任务训练时, 通常会迅速忘记如何执行之前的任务, 这种现象称为灾难性遗忘<sup>[1]</sup>. 从网络拓扑的角

1. School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074 2. Institute of Artificial Intelligence, Huazhong University of Science and Technology, Wuhan 430074 3. Key Laboratory of Image Processing and Intelligent Control of Education Ministry of China, Wuhan 430074

度来说,遗忘的发生是由于连接主义网络的本质:其信息均被存储在模型参数中,直接在新任务上训练网络会修改连接权值并偏离旧任务,从而不可避免地发生遗忘<sup>[2]</sup>.从数据分布的角度来说,大多数 ANNs 只是为了学习某一特定任务而建立,其中训练与测试数据为独立同分布 (Independently identical distribution, IID).该条件假设了一种静态场景,即数据分布不随时间的推移而发生变化.然而,现实世界中流式数据 (Streaming data) 的不确定性使得预训练模型无法应对非 IID 任务.与此同时,遗忘的发生并不是因为网络容量有限 (同样的网络在交错或联合训练时仍可以学习许多任务)<sup>[3]</sup>,而是源于现实世界中流式数据往往以任务序列的形式先后出现,其中每个任务的数据可能会在一段时间后消失,甚至由于内存限制或隐私问题而无法存储或重新访问<sup>[4]</sup>.因此,研究有效且高效的 CL 方法以克服 ANNs 中的灾难性遗忘具有重要意义,对于通用人工智能的发展也至关重要.

近年来,CL 受到了越来越多的关注<sup>[5-8]</sup>.根据学习过程中如何处理与利用特定于任务的信息,CL 方法可以大致分为以下三类:

1) 扩展方法 (Expansion method)<sup>[9-11]</sup>:这类方法的典型特征为不断增加单独的模型参数以适应新类别,即:模型通过冻结以前的任务参数或者为每个任务分配一个副本,从而逐步为新任务添加额外的网络分支.例如,文献 [12] 通过在新任务的训练过程中扩展网络节点,提出一种可自适应更新的动态网络以增加对新类别的特征表征能力.其虽然可以有效防止遗忘旧任务,但往往会导致网络规模随着任务数量的增加而增加.此外,扩展方法通常受限于“多头” (Multi-heads) 设置,其中每个任务都具备一个专有的输出层,而且测试阶段一般需要提供任务的身份 (Task identities, Task-ID) 才能正确匹配已经学过的、特定于该任务的头 (Head)<sup>[7]</sup>.

2) 重放方法 (Replay method)<sup>[13-15]</sup>:这类工作通过保存部分原始样本或使用生成模型生成伪样本,以在学习新任务时进行联合训练.作为一种经典的重放方法,梯度情景记忆 (Gradient episode memory, GEM)<sup>[16]</sup> 在学习新任务时利用有限的历史数据依次限制每个任务的损失函数不增加.在此基础上,平均梯度情景记忆 (Averaged GEM, A-GEM)<sup>[17]</sup> 和逐层优化梯度分解 (Layerwise optimization by gradient decomposition, LOGD)<sup>[18]</sup> 分别通过考虑所有旧任务的平均损失来放松约束、在不同情景中指定共享和特定于任务的信息以缓解遗忘.文献 [19] 进一步构建基于少数示范性样本的目标检测网络用于低温电子显微镜粒子拾取.重放方法通常依赖旧

任务数据的数量和质量.一方面,由于内存限制或隐私问题可能根本无法存储或重新访问;另一方面,不断训练生成模型也会导致复杂性逐渐增加,而且需要额外注意避免模式崩溃.

3) 基于正则化的方法 (Regularization-based method)<sup>[20-22]</sup>:通过施加惩罚限制重要参数,使其在后续任务的训练过程中不发生较大变化.文献 [23] 首次提出了一种弹性权值巩固 (Elastic weight consolidation, EWC) 方法,该方法利用量化的权值重要性有选择地保护对于旧任务重要的权值来学习新任务.在此基础上,文献 [24] 以在线方式基于损失函数计算权值重要性,并提出了突触智能 (Synaptic intelligence, SI) 算法.文献 [4] 提出的记忆感知突触 (Memory aware synapses, MAS) 可以进一步用于无标签情况.文献 [25] 通过寻找不干扰先前任务权值更新的正交投影,提出了正交权值修改 (Orthogonal weights modification, OWM), 在新任务上训练时,其权值只允许沿着输入空间的正交方向进行变化.文献 [26] 基于知识自蒸馏 (Knowledge self-distillation) 技术设计了一种域增量学习基准并创新性地用于持续人群计数.它们仅向目标函数引入正则化项而不需要扩大网络规模或存储旧任务数据.

然而,现有的 CL 方法为了减轻灾难性遗忘难以同时满足最小化计算、存储和时间需求,往往对这些条件中的两个或全部做出妥协<sup>[15]</sup>.大多数 CL 模型都是在深度神经网络 (Deep neural networks, DNNs) 上实现的,并严重依赖反向传播 (Back propagation, BP) 算法.因此,这些方法不仅训练耗时,而且在任务序列的训练过程中对超参数的设置极为敏感.更为重要的是,它们潜在地需要多次遍历访问 (对应较大的 epoch) 任务数据以获得更好的性能,往往不利于保留旧任务的信息,甚至由于隐私限制可能是不可行的<sup>[16]</sup>.对比之下,生物脑显然已经实现了高效且灵活的连续学习方式.以上基于 DNNs 的 CL 模型与高级动物在学习能力上的显著差距促使我们进一步借鉴生物脑的认知机制.

近年来,具有跨学科特性的类脑智能引起了众多研究人员的关注<sup>[27-29]</sup>.本质上说,类脑智能算法是从神经元、突触和神经环路的基本和涌现特性中获取灵感,从而对其结构、机制或功能进行数学建模,以找到类脑生物特征.这与人工智能密切相关,因为更好地了解生物大脑将进一步构建更加智能的模型.为此,本文将随机权神经网络与生物大脑的相关工作机制联系起来,提出了一种新的再可塑性启发的随机化网络 (Metaplasticity-inspired randomized network, MRNet) 用于类增量学习 (Class incremental learning, Class-IL) 场景.具体地,本文

的主要贡献如下: 1) 以前馈方式开发了一种多层随机权网络结构, 并设计了具备记忆功能的再可塑性矩阵, 用于指导输出权值的更新; 2) 为了有效兼容非 IID 任务以及伴随出现的新类别, 进一步构造了具有解析解的通用 CL 框架, 从而实现学习与记忆融合; 3) 通过设计不同难度和未知顺序的任务序列, MRNet 相比于现有的 CL 方法可以有效且高效地应用于 Class-IL, 表明本文方法进一步拓展了传统机器学习算法在图像分类任务上的分析能力。

与现有工作相比, 所提方法的优势见表 1 所示。具体地: 1) MRNet 利用任务之间共享的随机权值和解析解实现了以前馈方式构建, 只需要访问一次当前任务的数据 (相当于 epoch 始终为 1), 具有易于实现、参数高效、收敛速度快等优点, 是一种更加通用的 CL 方法; 2) MRNet 只需要巩固最后一层的解析解而无需逐层优化模型参数, 无需存储旧数据或构建数据生成器, 也无需增加网络尺寸; 3) 由于不需要调整学习率等超参数, 其模型更新只需要

较少的人为干预且具有较强的任务顺序鲁棒性。

## 1 相关工作

### 1.1 连续学习场景

本节主要介绍 3 种不同的连续学习场景以阐明其难度和普遍性, 即任务增量学习、域增量学习和类增量学习<sup>[14, 18]</sup>。图 1 以手写数字识别数据集 MNIST<sup>[30]</sup> 为例, 将其拆分为 5 个独立的二分类任务。这里仅给出该任务序列的前两个。一个 CL 模型依次在任务  $t$  ( $t = 1, 2, \dots, 5$ ) 上完成训练后需要识别所有见过的  $2t$  个类别 ( $C_1, C_2, \dots, C_{2t}$ )<sup>[24]</sup>。

1) 任务增量学习: 该场景所使用的网络结构具有 Multi-heads 输出层, 这意味着每个任务都有自己的输出单元, 只有匹配某一任务的 Head 才会被激活并作出预测。同时, 网络的其余部分通常在任务之间共享。由于需要不断增加网络尺寸和始终提供被测任务的 Task-ID, 这也是最简单的连续学习场景, 如图 1(a) 所示。

2) 域增量学习: 不同于将知识从旧任务转移到新任务的域自适应, 域增量学习旨在使用单一模型同时兼顾新旧任务的性能。该场景的典型特点是神经网络结构总是固定的, 如输出单元对于所有任务都是相同的, 但输入分布不断变化。同时, 是否需要提供 Task-ID 取决于任务序列的标签是否一致, 如图 1(b) 所示。

表 1 不同类增量学习方法的特性

Table 1 Characteristics of different Class-IL methods

方法	无需多次访问	无需逐层优化	无需数据存储	无需网络扩展
重放	×	×	×	✓
扩展	×	×	✓	×
正则化	×	×	✓	✓
MRNet	✓	✓	✓	✓

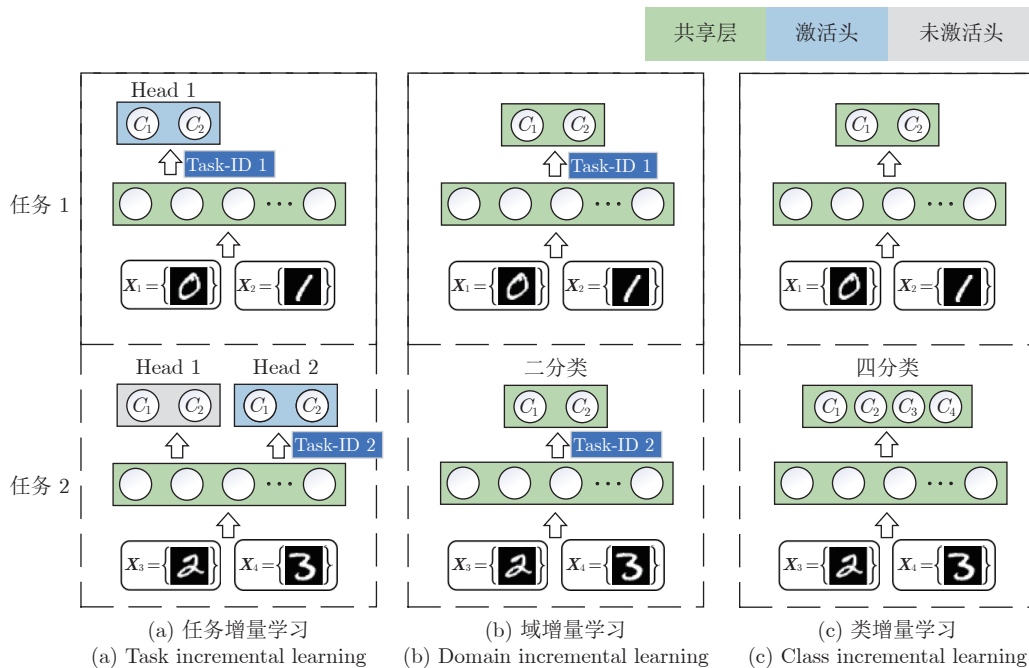


图 1 三种连续学习场景

Fig. 1 Three continual learning scenarios

3) 类增量学习: 要求模型不仅能够识别目前学过的类别, 而且可以适应未来任务中出现的新类别. 相应地, 神经网络的输出节点数与任务序列所包含的类别总数保持一致. 在学习过程中被测任务的身份也不再需要, 这是开放环境中最为常见的场景, 如图 1(c) 所示. 因此, 我们将本文的研究重点集中在最具挑战性和普遍性的 Class-IL 场景中.

## 1.2 类增量学习定义

Class-IL 基本定义如下: 假设给定任务  $t$  ( $t = 1, 2, \dots, T$ ) 的输出类别数为  $C_t$ . 类似传统单任务学习, 类增量学习模型  $\mathcal{M}(\theta_1)$  表示在第 1 个任务上完成训练并能够识别  $C_1$  个类, 其中,  $\theta_1$  为当前模型参数. 不同的是,  $\mathcal{M}(\theta_t)$  ( $t \geq 2$ ) 的目标是在完全不访问或者部分访问旧任务数据等情况下 (见表 1) 学会如何执行新任务并且不忘记旧任务<sup>[15, 21]</sup>, 从而需要累计识别  $\sum_{t=1}^T C_t$  个类. 这意味着测试过程中的样本可能来自迄今为止所学过的任务  $1 \sim t$  中的任何一个或多个, 这可以通过预分配  $C \geq \sum_{t=1}^T C_t$  个输出节点来容纳所有可能出现的类别.

## 1.3 随机权神经网络

上世纪 90 年代, 文献 [31] 提出的随机向量函数链接网络 (Random vector functional link networks, RVFLNs), 以及文献 [32] 提出的具有随机权值的浅层网络结构为随机权神经网络 (Random weight neural networks, RWNNs) 的发展奠定了基础<sup>[33-35]</sup>. 作为一类随机化前馈神经网络, RWNNs 的典型特征在于网络参数 (输入权值和偏置) 仅在给定的区间内随机产生, 因此只需解析求解输出权值. 由于 RWNNs 易于实现且收敛速度快, 从而受到了广泛的关注<sup>[35-37]</sup>. 近年来, 具有代表性的 RWNNs 包括: RVFLNs 的改进版<sup>[38]</sup>、随机配置网络 (Stochastic configuration networks, SCNs)<sup>[39]</sup> 和宽度学习系统 (Broad learning system, BLS)<sup>[40]</sup> 等. 相比 DNNs, 它们的学习形式更加符合生物脑功耗低、实时性快等特点<sup>[34, 41-42]</sup>.

## 1.4 再可塑性

在神经科学领域, 突触可塑性 (Synaptic plasticity) 表示学习与记忆功能的重要细胞机制, 其中长时程增强与长时程抑制是突触可塑性的两种主要表现形式. 再可塑性 (Metaplasticity) 作为突触可塑性的一种高级形式, 它表示突触可塑性的可塑性, 即突触活动的过往史 (Activity history) 对后续的突触可塑性产生的影响, 这表明突触的可塑性依赖于当前的突触“状态”<sup>[43-45]</sup>. 再可塑性一般发生在给

予一个前期刺激 (Priming stimulation) 或经历之后, 且这个前期刺激的影响能够持续存在, 导致后续长时程增强的诱导阈值明显改变. 受再可塑性的启发, 我们将根据先前任务节点之间的连接权值来度量模型参数重要性<sup>[4]</sup> (见第 2.3.2 节).

## 2 类增量连续学习模型

### 2.1 问题描述

无论是 DNNs 还是 RWNNs, 其本质上都是针对某一特定任务而建立的联结主义模型, 并假设所有样本满足 IID 以确保训练与测试数据均服从同一概率分布, 从而形成单一、固定的映射关系. 然而, 该假设忽略了开放环境中流式数据的不可预见性: 一方面, 未来数据的分布一旦发生变化, 就会导致映射关系集之间的覆盖或者干扰; 另一方面, 新任务的出现往往伴随着新类别, 使得其在后续训练过程中难以兼容标签分布的变化. 因此, 当它们在类增量学习场景中依次面向不同任务时都会遭受灾难性遗忘的影响. 具体地, 在算法层面所面临的问题概括如下.

1) 由 DNNs 建立的 CL 模型均是通过 BP 算法实现的, 在任务序列的学习过程中对超参数 (例如学习率、epoch、mini-batch 等) 的设置极为敏感. 为此, 通常会选择较小的学习率和较大的 epoch 以获得更好的性能, 但会导致耗时的训练过程且针对当前任务优化的超参数往往不再适用于过去和未来的任务. 更为重要的是, 在不访问旧任务数据的情况下多次遍历新任务数据, 不仅不利于保留旧任务的信息而且可能由于隐私限制根本无法满足.

2) 作为一种替代性方法, 尽管 RWNNs 随机初始化输入权值和偏置并解析评估输出权值避免了误差反传, 但是仍无法直接面向任务序列. 首先, 仅随机初始化的连接权值难以保证增量式地提取具有判别性的特征; 其次, 特定于任务的输出权值仅考虑了当前任务的信息, 导致其只适用于传统的单任务学习; 最后, 输出权值的更新缺乏理论依据与指导, 即无法实现学习与记忆融合. 因此, 需要进一步的改进.

### 2.2 思想方法

MRNet 的目标是在求解新任务的解析解时, 将其引导至新旧任务的公共低误差区, 而不是只关注某一特定任务. 例如, 过参数化 (Over-parameterization) 使得新任务的解可以无限接近旧任务, 不同可训练参数的最终设置可以产生相同水平的学习性能, 任务序列间的相似性以及现有 CL 方法的研究进展都为 MRNet 的可行性提供了保证<sup>[7, 23, 46]</sup>. 其核心思想是首先在旧任务上评估输出权值的重要性,

然后在新任务上训练时仅更改对于旧任务而言不重要的部分, 从而保证模型在新旧任务上的性能.

MRNet 网络结构如图 2 所示, 主要包括一个全局的预训练特征提取器、随机初始化的扩展输入层和隐含层. 其输入既可以是原始数据  $\mathbf{X}$  也可以是特征向量  $\mathbf{X}_f$ . 在这种设置下, 全局网络仅在复杂任务序列时使用且可以是任何架构. 为了简化符号, 本文使用  $\mathbf{X}$  来表示两者. 假设给定任务  $t$  ( $t = 1, 2, \dots, T$ ) 的训练样本  $\mathbf{D}_t = \{(\mathbf{X}_t, \mathbf{Y}_t) | \mathbf{X}_t \in \mathbf{R}^{N_t \times M_t}, \mathbf{Y}_t \in \mathbf{R}^{N_t \times C_t}\}$ , 其中  $\mathbf{X}_t$  为输入,  $\mathbf{Y}_t$  为输出,  $N_t$  为样本个数,  $M_t$  和  $C_t$  分别为输入模式和输出类别数. 例如, 在图像分类任务中,  $M_t$  表示图像的大小,  $C_t$  表示图像的独热编码 (One-hot coding) 标签.

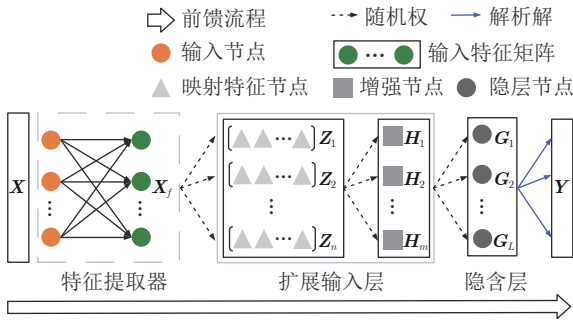


图 2 用于连续学习的 MRNet 结构  
Fig.2 MRNet architecture for CL

从算法实现的角度来说, 训练过程包括阶段 1: 初始化 MRNet; 阶段 2: 学习与记忆融合. 具体地, 第 2.3.1 节首先通过第 1 个任务随机初始化 1 个固定大小的基网络, 并在后续的任务中保持不变; 然后, 第 2.3.2 节设计了具备记忆功能的再可塑性矩阵用于指导输出权值的连续更新; 当第 2 个任务出现时, 第 2.4.1 节构造了目标函数, 并创新性地推导出解析解, 第 2.4.2 节进一步将其推广至任意长的任务序列; 最后, 第 2.5 节给出了上述两阶段训练的算法实现步骤.

## 2.3 初始化 MRNet

### 2.3.1 传统单任务学习

一个基本的 MRNet 包括  $n$  组映射特征节点 (每组  $k$  个节点) 和  $m$  组增强节点 (每组 1 个节点). 其中映射特征节点用于对原始输入样本或特征向量进行特征处理; 增强节点则作为网络额外的输入. 不同于 BLS 及其变体<sup>[40, 42]</sup>, MRNet 简化了从扩展输入层到输出层的直接连接用于匹配判别信息, 因为从先前任务中学习到的判别信息对于后续任务而言往往是不具有判别性的<sup>[11]</sup>, 具体见第 3.4 节消融实验. 任务 1 中,  $\mathbf{D}_1 = \{(\mathbf{X}_1, \mathbf{Y}_1) | \mathbf{X}_1 \in \mathbf{R}^{N_1 \times M_1}, \mathbf{Y}_1 \in$

$\mathbf{R}^{N_1 \times C_1}\}$  的训练过程与传统 RWNNs 的单任务学习类似. 如图 2 所示, 首先第  $i$  组 ( $i = 1, 2, \dots, n$ ) 映射特征节点的输出可以表示为

$$\mathbf{Z}_i = \phi_i(\mathbf{X}_1 \mathbf{W}_{f_i} + \beta_{f_i}) \in \mathbf{R}^{N_1 \times k} \quad (1)$$

第  $j$  个 ( $j = 1, 2, \dots, m$ ) 增强节点的输出为

$$\mathbf{H}_j = \xi_j(\mathbf{Z}^n \mathbf{W}_{e_j} + \beta_{e_j}) \in \mathbf{R}^{N_1} \quad (2)$$

其中,  $\mathbf{W}_{f_i} \in \mathbf{R}^{M_1 \times k}$ ,  $\beta_{f_i} \in \mathbf{R}^k$ ,  $\mathbf{W}_{e_j} \in \mathbf{R}^{nk}$ ,  $\beta_{e_j} \in \mathbf{R}$  分别为在给定区间, 如  $[-1, 1]$  内随机生成的第  $i$  组映射特征节点和第  $j$  个增强节点的输入权值和偏置, 并使用下标  $f$  和  $e$  加以区分;  $\phi_i(\cdot)$  和  $\xi_j(\cdot)$  均表示常用的激活函数;  $\mathbf{Z}^n = [\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_n] \in \mathbf{R}^{N_1 \times nk}$  和  $\mathbf{H}^m = [\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_m] \in \mathbf{R}^{N_1 \times m}$  分别表示将  $n$  组  $\mathbf{Z}_i$  和  $m$  个  $\mathbf{H}_j$  拼接在一起得到的节点集合. 然后, 得到扩展输入层的输出  $\tilde{\mathbf{X}}_1 = [\mathbf{Z}^n, \mathbf{H}^m] \in \mathbf{R}^{N_1 \times (nk+m)}$ . 最后, 具有  $L$  个隐含层节点的 MRNet 输出函数可以表示为

$$f_{\mathbf{W}}(\tilde{\mathbf{X}}_1) = \sum_{l=1}^L \mathbf{W}_l \psi_l(\tilde{\mathbf{X}}_1; \mathbf{W}_{h_l}, \beta_{h_l}) \quad (3)$$

其中,  $\mathbf{W}_l \in \mathbf{R}^{C_1}$  表示输出权值  $\mathbf{W} \in \mathbf{R}^{L \times C_1}$  的第  $l$  行;  $\psi_l(\cdot)$  可以是 ReLU 或者 Sigmoid 激活函数;  $\mathbf{W}_{h_l} \in \mathbf{R}^{nk+m}$  和  $\beta_{h_l} \in \mathbf{R}$  为相应的隐含层随机参数, 用下标  $h$  加以区分.

为了方便, 本文指定  $\mathbf{G}_l = \psi_l(\tilde{\mathbf{X}}_1; \mathbf{W}_{h_l}, \beta_{h_l})$ , 得到隐含层输出矩阵  $\mathbf{A}_1 = [\mathbf{G}_1, \dots, \mathbf{G}_L, \dots, \mathbf{G}_L] \in \mathbf{R}^{N_1 \times L}$ . 进一步地, 式 (3) 可以表示为  $\mathbf{Y}_1 = \mathbf{A}_1 \mathbf{W}$ . 其中输出权值  $\mathbf{W}$  可以通过 Moore-Penrose 广义逆运算<sup>[47]</sup> 得到, 即  $\mathbf{W} = \mathbf{A}_1^+ \mathbf{Y}_1$ . 然而, 这种求解方法往往并不适用于处理较大规模数据, 如图像分类任务. 为此, 这里采用式 (4) 给出的另一种替代方法, 即

$$\mathbf{A}_1^+ = \lim_{\lambda \rightarrow 0} (\lambda \mathbf{I} + \mathbf{A}_1^T \mathbf{A}_1)^{-1} \mathbf{A}_1^T \quad (4)$$

式 (4) 通过引入一个额外的常数约束  $\lambda$  并在矩阵  $\mathbf{A}_1^T \mathbf{A}_1$  对角线上添加一组接近于 0 的正数来近似  $\mathbf{A}_1^+$ . 它不仅可以在一定程度上避免过拟合, 而且可以保证矩阵在病态条件 (Ill condition) 下近似解的存在.

在学习完任务 1 中的  $C_1$  个类别后, 模型  $\mathcal{M}(\theta_1)$  的输出权值  $\mathbf{W}$  可以通过式 (5) 计算, 即

$$\mathbf{W} = (\lambda \mathbf{I} + \mathbf{A}_1^T \mathbf{A}_1)^{-1} \mathbf{A}_1^T \mathbf{Y}_1 \quad (5)$$

因此, 式 (5) 为仅通过任务 1 得到的优化权值 (这里  $\mathbf{W} = \mathbf{W}_1^*$ ). 进一步地, 预测误差矩阵为

$$\mathbf{E}_1 = \mathbf{A}_1 \mathbf{W} - \mathbf{Y}_1 \quad (6)$$

其中,  $\mathbf{E}_1 = [e_{1,1}, \dots, e_{1,p}, \dots, e_{1,N_1}]^T$ ,  $e_{1,p} \in \mathbf{R}^{C_1}$  对应于任务 1 中第  $p$  个训练样本  $(\mathbf{X}_{1,p}, \mathbf{Y}_{1,p})$  的残

差. 需要指出的是,  $\mathbf{W}$  将在后续任务的学习中不断更新.

### 2.3.2 构造再可塑性矩阵

本节为 MRNet 在连续学习过程中如何更新输出权值提供理论依据与指导. 根据第 2.2 节可知, 模型参数重要性发挥着类似突触再可塑性的作用. 其中, 每个输出权值的再可塑性越高, 其重要性越低, 对旧任务性能的影响也就越小, 因此可以优先改变以兼容新任务的分布; 相反, 输出权值的再可塑性低则表示其重要性高, 那么这部分理想上应保持不变, 从而保留模型对旧任务的记忆. 因此, 我们围绕节点之间的连接权值度量其重要性, 并用来巩固 MRNet 的输出权值.

对于任意的任务  $t$  (这里以任务 1 为例), 首先利用交叉熵损失 (Cross-entropy loss) 函数  $\ell(\hat{\mathbf{Y}}_1, \mathbf{Y}_1)$  保证模型  $\mathcal{M}(\theta_1)$  所建立的映射关系与任务 1 中数据  $\mathbf{D}_1 = (\mathbf{X}_1, \mathbf{Y}_1)$  之间分布的近似, 其中,  $\hat{\mathbf{Y}}_1 = \mathbf{A}_1 \mathbf{W}$ ; 然后, 通过对输出权值  $\mathbf{W}$  添加无穷小更新量  $\Delta \mathbf{W}$  观察损失函数的变化, 可以近似表示为

$$\ell(\mathbf{A}_1(\mathbf{W} + \Delta \mathbf{W}), \mathbf{Y}_1) - \ell(\mathbf{A}_1 \mathbf{W}, \mathbf{Y}_1) \approx \mathbf{g}_1 \Delta \mathbf{W}^T \quad (7)$$

其中,  $\mathbf{g}_1 = \frac{\partial \ell(\hat{\mathbf{Y}}_1, \mathbf{Y}_1)}{\partial \mathbf{W}}$  表示梯度信息. 进一步地, 通过对所有无穷小更新量  $\Delta \mathbf{W}$  求和, 累积梯度以获得经过任务 1 解空间整个轨迹上的损失变化 (路径积分)<sup>[24]</sup>. 这相当于计算式 (8), 即

$$\mathcal{F}_1 = \left( \frac{\partial \ell(\hat{\mathbf{Y}}_1, \mathbf{Y}_1)}{\partial \mathbf{W}} \right) \left( \frac{\partial \ell(\hat{\mathbf{Y}}_1, \mathbf{Y}_1)}{\partial \mathbf{W}} \right)^T \quad (8)$$

实际上, 根据文献 [48] 中描述的特性, 式 (8) 中一阶导数的平方是式 (9) 给出的计算耗时的海森矩阵 (Hessian matrix) 的等价且有效替代方法, 即

$$\mathcal{F}_1 \approx \frac{\partial^2 \ell(\hat{\mathbf{Y}}_1, \mathbf{Y}_1)}{\partial \mathbf{W}^2} \quad (9)$$

本质上来说, 式 (9) 中二阶导数反映了  $\ell(\hat{\mathbf{Y}}_1, \mathbf{Y}_1)$  本身相对于  $\mathbf{W}$  中每个元素的变化率. 相应地,  $\mathcal{F}_1$  中较小的元素值表明其对损失函数的影响不大; 反之亦然, 从而实现了对旧任务记忆的保留. 因此, 我们将具备记忆功能的  $\mathcal{F}_1$  命名为再可塑性矩阵并用于 MRNet 输出权值的自适应更新.

需要指出的是: 1) MRNet 在任务 1 上完成网络结构初始化, 并在接下来的任务中保持不变; 后续 MRNet 的初始化均指在前一个任务上持续获得再可塑性矩阵  $\mathcal{F}_t$  ( $t = 1, 2, \dots, T-1$ ). 2) 式 (8) 虽然涉及了梯度计算, 但仅仅是一阶且不需要梯度信息反传. 因此, 以上过程仍然是高效且易于实现的. 3) 一旦完成  $\mathcal{F}_t$  的计算, 相应任务的数据  $\mathbf{D}_t$  不

再保留. 4) 除非另作说明, 下文中出现仅下标不同的同一个变量均表示在特定任务  $t$  下的相同含义.

## 2.4 学习与记忆融合

本节详细介绍如何将再可塑性矩阵用于 MRNet 连续学习过程以实现输出权值的有效更新, 包括: 1) 针对任务 2 构造目标函数并创新性地推导出解析解; 2) 进一步推广至任意长的任务序列.

### 2.4.1 处理两个任务

当任务 2 出现时, 我们利用  $\mathcal{F}_1 = \mathbf{P}_1 \odot \mathbf{P}_1$  指导输出权值  $\mathbf{W}$  中的元素有选择性地接近或偏离最近一次得到的  $\mathbf{W}_1^*$ , 其中, 运算符  $\odot$  表示逐元素积, 即矩阵对应位置元素相乘、所得矩阵维度不变. 于是, MRNet 的目标函数可以表示为下列一般性问题

$$\begin{aligned} \min_{\mathbf{W}} \quad & \frac{1}{2N_2} \sum_{p=1}^{N_2} \|\mathbf{e}_{2,p}\|^2 + \frac{\lambda_1}{2} \sum_{l=1}^L \|\mathbf{P}_{1,l} \odot (\mathbf{W}_l - \mathbf{W}_{1,l}^*)\|^2 \\ \text{s.t.} \quad & \mathbf{f}_{\mathbf{W}}(\tilde{\mathbf{X}}_{2,p}) = \mathbf{Y}_{2,p} - \mathbf{e}_{2,p}, \forall p \end{aligned} \quad (10)$$

其中,  $\lambda_1$  控制着模型在学习任务 2 时对任务 1 的保护程度;  $\mathbf{W}_{1,l}^* \in \mathbf{R}^{C_1}$  表示仅在任务 1 上根据式 (5) 得到的优化权值  $\mathbf{W}_1^*$  的第  $l$  行;  $\mathbf{P}_{1,l} \in \mathbf{R}^{C_1}$  表示相应的输出权值重要性. 注意到式 (10) 中第 1 项负责任务 2 的学习, 而任务 1 的所有记忆都包含在第 2 项中. 因此, MRNet 鼓励其输出权值  $\mathbf{W}$  的解空间围绕先前任务获得的  $\mathbf{W}_1^*$ .

式 (10) 可以通过 KKT (Karush-Kuhn-Tucker) 条件<sup>[49]</sup> 求解如下:

$$\begin{aligned} \mathcal{L}(\mathbf{W}, \mathbf{E}_2, \mathbf{v}_2) = & \frac{1}{2N_2} \sum_{p=1}^{N_2} \|\mathbf{e}_{2,p}\|^2 + \\ & \frac{\lambda_1}{2} \sum_{l=1}^L \|\mathbf{P}_{1,l} \odot (\mathbf{W}_l - \mathbf{W}_{1,l}^*)\|^2 - \\ & \sum_{p=1}^{N_2} (\mathbf{f}_{\mathbf{W}}(\tilde{\mathbf{X}}_{2,p}) - \mathbf{Y}_{2,p} + \mathbf{e}_{2,p}) \mathbf{v}_{2,p}^T \end{aligned} \quad (11)$$

其中,  $\mathbf{v}_{2,p} \in \mathbf{R}^{C_2}$  表示拉格朗日乘子并对应任务 2 中第  $p$  个训练样本  $(\mathbf{X}_{2,p}, \mathbf{Y}_{2,p})$ . 分别对式 (11) 中的决策变量  $\mathbf{W}$ ,  $\mathbf{E}_2$  和  $\mathbf{v}_2$  求偏导可以得出

$$\nabla_{\mathbf{W}} \mathcal{L}(\mathbf{W}, \mathbf{E}_2, \mathbf{v}_2) = \lambda_1 \mathcal{F}_1 \odot (\mathbf{W} - \mathbf{W}_1^*) - \mathbf{A}_2^T \mathbf{v}_2 \quad (12)$$

$$\nabla_{\mathbf{E}_2} \mathcal{L}(\mathbf{W}, \mathbf{E}_2, \mathbf{v}_2) = \frac{1}{N_2} \mathbf{E}_2 - \mathbf{v}_2^T \quad (13)$$

$$\nabla_{\mathbf{v}_2} \mathcal{L}(\mathbf{W}, \mathbf{E}_2, \mathbf{v}_2) = -(\mathbf{A}_2 \mathbf{W} - \mathbf{Y}_2 + \mathbf{E}_2) \quad (14)$$

经过整理, 得到

$$\mathbf{A}_2^T \mathbf{A}_2 \mathbf{W} + \gamma_1 \mathcal{F}_1 \odot (\mathbf{W} - \mathbf{W}_1^*) = \mathbf{A}_2^T \mathbf{Y}_2 \quad (15)$$

其中, 标量  $\gamma_1 = \lambda_1 N_2$  表示任务 1 与任务 2 之间的权衡 (Trade-off).

注意到式 (15) 中传统矩阵相乘与逐元素积的共存导致难以推导出输出权值  $\mathbf{W}$ , 为此, 本文进一步结合逆运算的特点, 创新性地提出了一种分块矩阵对角化 (Partitioned matrix diagonalization, PMD) 方法: 1) 分别对矩阵  $\mathbf{W}$ ,  $\mathcal{F}_1$  和  $\mathbf{Y}_2$  按每一列分块, 得到  $\mathbf{W} = [\mathbf{W}_1, \dots, \mathbf{W}_c, \dots, \mathbf{W}_{C_1}]$ ,  $\mathcal{F}_1 = [\mathcal{F}_{1,1}, \dots, \mathcal{F}_{1,c}, \dots, \mathcal{F}_{1,C_1}]$  以及任务 2 的输出  $\mathbf{Y}_2 = [\mathbf{Y}_{2,1}, \dots, \mathbf{Y}_{2,c}, \dots, \mathbf{Y}_{2,C_2}]$ ; 2) 矩阵  $\mathcal{F}_{1,c}$  ( $c = 1, 2, \dots, C_1$ ) 被对角化为  $\mathbf{A}_{1,c} \in \mathbf{R}^{L \times L}$ ; 3) 在学习完任务 2 中的  $C_2$  个新类别后, 模型  $\mathcal{M}(\theta_2)$  输出权值  $\mathbf{W} = [\mathbf{W}_1, \dots, \mathbf{W}_c, \dots, \mathbf{W}_C] \in \mathbf{R}^{L \times C}$  (其中,  $C = C_1 + C_2$ ) 的每一列都可以采用递归的方式独立地解析为

$$\mathbf{W}_c = (\gamma_1 \mathbf{A}_{1,c} + \mathbf{A}_2^T \mathbf{A}_2)^{-1} (\mathbf{A}_2^T \mathbf{Y}_{2,c} + \gamma_1 \mathbf{A}_{1,c} \mathbf{W}_{1,c}^*) \quad (16)$$

#### 2.4.2 面向任务序列

本节将两个任务的情况推广到任意长的 Class-IL. 对于包含  $C_T$  个新类别的任务  $T$  ( $T \geq 2$ ), 下面给出统一的 CL 框架, 即

$$\begin{aligned} \min_{\mathbf{W}} \frac{1}{2N_T} \sum_{p=1}^{N_T} \|\mathbf{e}_{T,p}\|^2 + \\ \frac{1}{2} \sum_{l=1}^L \left\| \sum_{t=1}^{T-1} \sqrt{\lambda_t} \mathbf{P}_{t,l} \odot (\mathbf{W}_l - \mathbf{W}_{T-1,l}) \right\|^2 \\ \text{s.t. } \mathbf{f}_{\mathbf{W}}(\widetilde{\mathbf{X}}_{T,p}) = \mathbf{Y}_{T,p} - \mathbf{e}_{T,p}, \forall p \end{aligned} \quad (17)$$

其中,  $T-1$  表示紧接在  $T$  之前的任务;  $\mathbf{W}_{T-1}$  表示最近一次得到的输出权值;  $\sum_{t=1}^{T-1}$  则表示在任务  $T$  出现之前的所有任务. 类似地, 模型  $\mathcal{M}(\theta_T)$  输出权值  $\mathbf{W} = [\mathbf{W}_1, \dots, \mathbf{W}_c, \dots, \mathbf{W}_C] \in \mathbf{R}^{L \times C}$  (其中,  $C = \sum_{t=1}^T C_t$ ) 的每一列可以解析为

$$\mathbf{W}_c = \left( \sum_{t=1}^{T-1} \gamma_t \mathbf{A}_{t,c} + \mathbf{A}_T^T \mathbf{A}_T \right)^{-1} \left( \mathbf{A}_T^T \mathbf{Y}_{T,c} + \sum_{t=1}^{T-1} \gamma_t \mathbf{A}_{t,c} \mathbf{W}_{T-1,c} \right) \quad (18)$$

其中,  $\gamma_t = \lambda_t N_{t+1}$  表示任务  $t$  与任务  $t+1$  之间的权衡. 同时, 注意到当  $T=2$  时, 式 (18) 等同于式 (16). 进一步地, 当再可塑性矩阵取为单位阵并忽略不存在的项时, 式 (16) 等同于式 (5). 这从理论上证明了单任务学习是 CL 的一种特殊情况.

## 2.5 算法描述

从算法实现的角度来说, MRNet 本质上属于一种特殊的正则化方法, 其仅通过一个二次惩罚项实现对旧任务的记忆. 这是因为最近一次得到的输出权值包含了先前所有见过任务的信息. 一旦在任务  $t$  上完成训练并得到输出权值  $\mathbf{W}_t$  和再可塑性矩阵  $\mathcal{F}_t$ , 该任务的数据  $\mathbf{D}_t$  不再需要存储和访问. 如式 (17) 所示, 此时先前任务上得到的输出权值  $\mathbf{W}_t$  ( $t = 1, 2, \dots, T-2$ ) 均可以被丢弃,  $\sum_{t=1}^{T-1} \sqrt{\lambda_t} \mathbf{P}_{t,c}$  可以在无需 Task-ID 的情况下累积更新. 这意味着 MRNet 的内存随任务数量的增加基本上保持不变. 具体地, 其训练过程包括两个阶段: 阶段 1: 初始化 MRNet; 阶段 2: 学习与记忆融合.

**输入.** 依次给定当前任务  $t$  ( $t = 1, 2, \dots, T$ ) 的训练样本  $\mathbf{D}_t = \{(\mathbf{X}_t, \mathbf{Y}_t)\}$ .

**输出.** 随机初始化参数  $\mathbf{W}_{f_i}, \beta_{f_i}, \mathbf{W}_{e_j}, \beta_{e_j}, \mathbf{W}_{h_l}, \beta_{h_l}$ , 最近一次得到的输出权值  $\mathbf{W}_{T-1}$ , 当前输出权值  $\mathbf{W}_T$ .

**阶段 1.** 初始化 MRNet ( $t = 1$ )

**步骤 1.1.** 根据式 (1) 和式 (2), 计算扩展输入层的输出  $\widetilde{\mathbf{X}}_1 = [\mathbf{Z}^n | \mathbf{H}^m]$ ;

**步骤 1.2.** 根据式 (3), 计算隐含层输出矩阵  $\mathbf{A}_1 = [\mathbf{G}_1, \dots, \mathbf{G}_l, \dots, \mathbf{G}_L]$ ;

**步骤 1.3.** 根据式 (4) 和式 (5), 计算隐含层输出矩阵的逆矩阵  $\mathbf{A}_1^+$  和输出权值  $\mathbf{W}$ , 并令  $\mathbf{W}_1^* = \mathbf{W}$ ;

**步骤 1.4.** 根据式 (8), 计算再可塑性矩阵  $\mathcal{F}_1$  用于 PMD 方法.

**阶段 2.** 学习与记忆融合 ( $t = 2, \dots, T$ )

**步骤 2.1.** 根据式 (1) 和式 (2), 计算扩展输入层的输出  $\widetilde{\mathbf{X}}_t = [\mathbf{Z}^n | \mathbf{H}^m]$ ;

**步骤 2.2.** 根据式 (18), 计算输出权值  $\mathbf{W}$ ;

**步骤 2.3.** 若  $t < T$ , 根据式 (8), 计算再可塑性矩阵  $\mathcal{F}_t$  用于 PMD 方法, 并令  $\mathbf{W}_t = \mathbf{W}$ ;

**步骤 2.4.** 重复步骤 2.1 ~ 2.3, 直至  $t = T$ .

## 3 仿真实验与结果分析

本节将通过 5 个评价指标、5 个基准任务序列和 10 个比较方法验证所提方法的优越性. 首先给出实验设置, 然后在不同难度的 Class-IL 场景中进行实验, 最后进行参数灵敏度分析和消融实验.

### 3.1 实验细节

#### 3.1.1 评价指标

为了全面评估 CL 方法的有效性和高效性, 本

文采用测试精度、跨任务传递知识的能力、整个执行时间和参数存储需求来同时描述模型质量. 假设任务序列的长度为  $T$ , 具体如下.

1) 平均测试分类精度 (Average test accuracy, ACC): 在迄今所有见到过的任务上测试分类精度的平均值, 即

$$ACC = \frac{1}{T} \sum_{t=1}^T R_{T,t} \quad (19)$$

其中,  $R_{T,t}$  表示在任务  $T$  上训练完后, 模型在任务  $t$  上的测试分类精度.

2) 后向迁移 (Backward transfer, BWT)<sup>[10]</sup>: 表示学习完任务  $T$  与学习完任务  $t$  两种情况下的模型分别在任务  $t$  上的测试分类精度之差, 即

$$BWT = \frac{1}{T-1} \sum_{t=1}^{T-1} R_{T,t} - R_{t,t} \quad (20)$$

本指标反映了模型对旧任务知识的保留能力.  $BWT$  为负值意味着模型在遇到新任务后降低了之前任务的性能, 较小的  $BWT$  则意味着灾难性遗忘.

3) 前向迁移 (Forward transfer, FWT)<sup>[50]</sup>: 通过定义 CL 模型相对于独立训练的模型在任务  $t$  上测试分类精度的提高, 即

$$FWT = \frac{1}{T-1} \sum_{t=2}^T R_{t,t} - R_t^{\text{ind}} \quad (21)$$

其中,  $R_t^{\text{ind}}$  表示仅在任务  $t$  上进行训练的单任务学习模型分类精度.  $FWT$  反映了模型能否利用以前学到的知识来提高当前任务的测试分类精度.

除上述三个精度相关的指标 (均越高越好), 我们还引入另外两个指标, 具体如下.

4) 累计运行时间 (Cumulative running time, Time), 单位: s.

5) 模型最终参数量 (Final number of parameters, No. Para.), 单位: MB, 用于综合评估模型质量. 我们注意到, 大多数 CL 方法仅使用 ACC 验证其优越性而忽略最小化计算、存储和时间需求等.

### 3.1.2 数据集

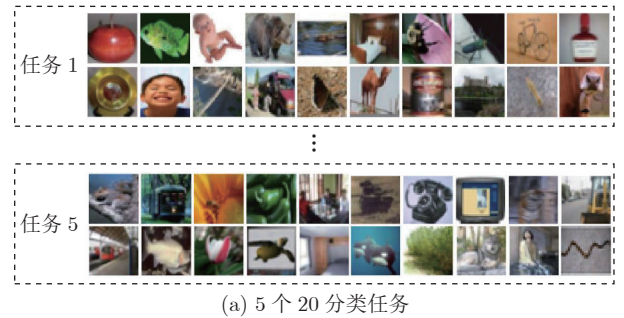
本节按照 Class-IL 场景的通用设置来生成不同难度的任务序列<sup>[14, 25, 51]</sup>, 即分别划分 FashionMNIST, CIFAR-100 和 ImageNet 基准数据集. 每个序列中的任务均独立地来自于原始基准数据集. 我们使用 [DATASET]- $C/T$  表示一个任务序列总共包含  $C$  个类别并被均匀地划分到  $T$  个任务中, 这意味着每个任务都需要学习  $C/T$  个新类别. 下面简述如何生成任务序列.

1) 对于 FashionMNIST, 将 10 个类别划分为 5 个二分类任务, 对应任务序列 FashionMNIST-10/5 如图 3 所示. 模型每学习完一个任务立即丢掉该任务的训练数据 (重放方法除外), 然后等待新任务的出现. 尽管由 5 个简单的二分类任务组成, 传统的 ANNs 学习该任务序列仍具有挑战性.

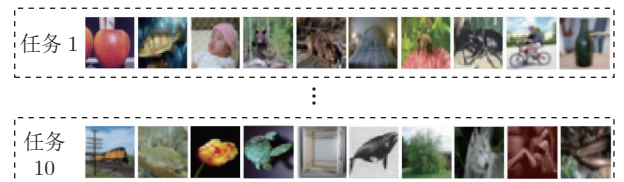


图 3 FashionMNIST-10/5 任务序列  
Fig.3 FashionMNIST-10/5 task sequence

2) 对于 CIFAR-100, 将 100 个类别分别划分为 5 个包含 20 个自然图像类的任务序列 (CIFAR-100/5) 和 10 个包含 10 个自然图像类的任务序列 (CIFAR-100/10), 如图 4 所示. CIFAR-100/5 中包含更多的类, 增加了每个任务的难度; 而 CIFAR-100/10 则通过增加任务序列的长度, 测试 CL 方法的信息保留能力.



(a) Five twenty-class classification tasks



(b) Ten ten-class classification tasks

图 4 CIFAR-100 任务序列  
Fig.4 CIFAR-100 task sequence

3) 对于 ImageNet, 我们使用其子集 ImageNet-200, 让模型分别增量学习不同长度的任务序列. 类似地, 所得到的 ImageNet-200/10 和 ImageNet-200/50 分别需要模型具有连续学习 10 个 20 分类任务和 50 个 4 分类任务的能力.

### 3.1.3 比较方法

本节将本文提出的 MRNet 在 Class-IL 场景中与现有经典和最先进的 7 种 CL 方法相比, 包括 MAS<sup>[4]</sup>, GEM<sup>[16]</sup>, EWC<sup>[23]</sup>, SI<sup>[24]</sup>, OWM<sup>[25]</sup>, IL2M<sup>[15]</sup>,

PCL<sup>[11]</sup>. 为了使得比较更加全面, 我们还与 3 种非 CL 方法进行比较, 包括一种仅适用于单任务学习的 RWNNs, 即 BLS<sup>[40]</sup>, 基于 L2 正则化的微调, 在整个任务序列上离线联合训练 (Jointly training, JT), 其通常被认为是 CL 方法关于评价指标 ACC 的上界.

### 3.1.4 网络参数

1) 网络结构: 为了公平比较, 所有方法对所有划分的任务序列分别使用类似大小的网络结构. 对于 FashionMNIST-10/5, 基于 BP 算法的比较方法使用一个多层感知机 (Multi-layer perceptron, MLP), 即 [784-800-800-10]; BLS 与 MRNet 则分别通过随机初始化固定的基网络 [748 - (10 × 10 + 1500) - 10]、[784 - (10 × 10 + 700) - 800 - 10] (对应  $n = 10$ ,  $k = 10$ ,  $m = 700$ ,  $L = 800$ ). 由于该任务序列较为简单, 故没有使用预训练特征提取器 (见图 2). 对于 CIFAR-100, 一个标准的 ResNet-56 用于特征提取. 我们按照 OWM<sup>[25]</sup> 中的实验设置, 首先使用由整个训练集预训练的特征提取器来分析原始图像, 然后将得到的特征向量依次输入 CL 模型中 (与 FashionMNIST-10/5 实验相似的网络结构). 即 CIFAR-100/5 与 CIFAR-100/10 是增量学习不同类别特征和标签之间的映射关系. 我们将预先训练好的特征提取器同时应用于 MRNet 和所有比较方法. 对于 ImageNet-200, 我们接下来评估 MRNet 是否能够学习具有挑战性且不涉及使用训练集获取预训练特征提取器的新任务<sup>[11]</sup>. 例如, 先以 Tiny-ImageNet-200 作为辅助数据在 ResNet-50 上进行预训练, 然后从 ImageNet 余下的 800 个类别中随机选择 200 个用于划分任务序列. 这符合人类学习的特点, 即我们倾向于为具有挑战性的任务做充分的准备, 而不是没有任何先验知识. 需要指出的是, 当使用预训练特征提取器时, 其在本文所提 MRNet 和所有比较方法中都同样使用.

2) 参数选择: 对于所有基于 BP 算法的比较方法, 我们使用由其作者发布的开源代码或流行的第三方代码, 并进行了大量的调参. 学习率取自 {0.1, 0.01, 0.001}, mini-batch 大小设为 100 (用于 FashionMNIST 和 CIFAR-100 划分的 3 个任务序列) 或 {100 和 20} (用于 ImageNet 划分的 2 个任务序列). 进一步地, 根据 GEM<sup>[16]</sup> 等方法中的实验设置, 我们将简单任务序列 (FashionMNIST) 的 epoch 设置为 1, 即当前任务样本在简单序列中仅允许被访问一次; 对于具有挑战性的任务序列 (CIFAR-100 和 ImageNet) 的 epoch 放宽为不加以限制; 而本文

所提 MRNet 对所有任务序列的 epoch 始终为 1. 对于重放方法, 我们始终保留 4.4 k 个随机样本集 (FashionMNIST) 以及限制重放预算为 2 k 个样本 (CIFAR-100 和 ImageNet); 对于扩展方法, 保证其在学习完所有的任务后与基网络相同规模; 对于正则化方法, 权衡系数针对不同任务序列选自集合 {100, 1000, 10000, 100000}, MRNet 的相关设置见第 3.3 节参数分析.

### 3.1.5 实验协议

在训练过程中, 仅重放方法可以存储和访问少量先前任务的训练数据, 而其他方法只有当前任务  $t$  的训练样本可供模型训练. 在测试过程中, 当模型学完  $T$  个任务时, 测试样本来自迄今为止所有学过的任务  $1 \sim T$ , 且无需告诉模型 Task-ID, 然后计算评价指标 (见第 3.1.1 节). 同时, 我们在每个基准数据集所划分的任务序列 (见第 3.1.2 节) 上独立重复 10 次实验, 并记录这些结果的平均值和标准差. 其中, 每次实验中的任务顺序 (Task ordering) 都是随机分配且未知的, 以验证模型在开放环境中的实用性.

## 3.2 结果与讨论

### 3.2.1 FashionMNIST

表 2 给出了 MRNet 与 10 种经典和最新的比较方法在 FashionMNIST-10/5 任务序列上的连续学习性能. 实验结果通过 5 个评价指标在该任务序列上独立重复 10 次实验获得, 并记录了这些结果的平均值和标准差. 其中, 每次实验所提供的任务顺序均从 5 个 2 分类的任务中随机分配. 作为非 CL 方法, BLS 和 L2 只能针对 IID 任务建立特定的输入输出映射关系, 而且一旦训练完成, 该映射不再发生改变. 其中, 尽管 BLS 在单任务学习中表现良好, 对于任务序列只记得最近一次学习的任务, 而之前的任务被完全遗忘. L2 对待所有模型参数施加同样程度的惩罚, 仍导致模型偏向特定任务的学习.

首先关于 ACC, 我们注意到余下 CL 方法基本都可以在 MNIST-10/5 任务序列上获得较好的性能, 而一旦换成 FashionMNIST-10/5 则呈现出不同程度的下降. 一般来说, 类增量学习对于无需网络结构增加和使用旧任务数据的正则化方法 (EWC, MAS, SI 和 OWM) 来说尤其困难. 扩展 (PCL) 和重放 (GEM 和 IL2M) 方法的性能普遍较好, 很大程度上归因于网络结构的增加和旧数据的使用, 其中 IL2M 是最强基线 (Baseline). 对比之下, MRNet 的 ACC 高出 IL2M 8.46%. 此外, 通过 ACC 的

表 2 连续学习 FashionMNIST-10/5 任务序列对比实验  
Table 2 Comparative experiments on continuously learning FashionMNIST-10/5 task sequence

方法	指标					
	ACC (%)	BWT	FWT	Time (s)	No. Para. (MB)	
非 CL 方法	BLS	19.93±0.22	—	—	8.17±0.24	0.25
	L2	26.55±6.27	—	—	59.12±2.73	1.28
	JT	~ 96.61	—	—	—	—
CL 方法	EWC	34.96±7.62	-0.7248±0.0953	-0.0544±0.0300	69.21±4.10	11.48
	MAS	38.54±3.49	-0.4781±0.0561	-0.2576±0.0548	110.26±1.74	3.83
	SI	56.19±3.21	-0.3803±0.0631	-0.1329±0.0504	67.67±2.25	5.11
	OWM	79.16±1.11	-0.1844±0.0197	-0.0635±0.0078	40.38±7.09	3.18
	GEM	81.98±2.80	-0.0586±0.0654	-0.1093±0.0510	45.73±1.17	1.28
	PCL	82.13±0.61	-0.1385±0.0413	-0.0647±0.0172	348.75±9.83	1.28
	IL2M	84.61±2.95	-0.0712±0.0273	-0.0258±0.0248	44.18±1.34	1.28
	MRNet	93.07±0.74	-0.0458±0.0069	-0.0261±0.0035	11.38±0.29	0.83

标准差可知, 大多数 CL 方法非常容易受序列中任务顺序的影响, 而 MRNet 执行 10 次随机分配的实验都具有一致的 ACC, 且仅次于具有 Multi-heads 设置的 PCL<sup>[11]</sup>. 这意味着本文所提方法通过解析解可以提高任务顺序鲁棒性.

其次, 一个可取的 CL 模型还应该具有较大的 BWT 和 FWT. MRNet 的 BWT 值最接近于 0, 表明其对于旧任务知识的保留能力高于其他方法. 重放方法 (GEM 和 IL2M) 的 BWT 值明显高于其他方法, 这在一定程度上是因为它们在学习新任务时访问旧任务数据. 而当模型保留的旧任务数据非常有限时往往会导致类别不平衡问题. 与此同时, IL2M 进一步存储了历史任务类别的统计信息, 其 FWT = -0.0258 在所有方法中最大. MRNet 的 FWT = -0.0261 意味着其尝试借助先前任务的信息以更好地学习 (至少尽可能不阻碍) 新任务.

最后, MRNet 在 Time 和 No. Para. 上明显优于其他基于 BP 算法的 CL 方法, 表明其具有参数高效、收敛速度快和易于实现等优点. 同时, 这两个指标也在一定程度上反映了模型质量. 然而, 现有 CL 方法大多只关注 ACC 而忽略时间、计算成本和存储需求. 因此, FashionMNIST-10/5 实验通过 5 个评价指标更加全面地验证了本文所提方法的有效性和高效性.

### 3.2.2 CIFAR-100

尽管在 FashionMNIST-10/5 任务序列上的实验结果表明 MRNet 能够有效缓解灾难性遗忘, 但是所建立的 CL 模型仅仅用于依次学习 5 个简单

的 2 分类任务. 本节将面向更复杂的分类任务以及更长的任务序列. 图 5 给出了 Class-IL 场景中不同 CL 方法在 CIFAR-100/5 和 CIFAR-100/10 两个任务序列上的 ACC, 分别需要模型具有连续学习 5 个 20 分类和 10 个 10 分类任务的能力. 结果表明, 本文所提 MRNet 明显优于其他方法. 对于图 5(a), BLS 将所有见过的任务识别为最近一次任务中的类别; L2 在学完任务 2 时也已经开始遭受严重的灾难性遗忘; 除了 OWM, 其余正则化方法 (EWC, MAS 和 SI) 均表现出不同程度的遗忘. MRNet 分别比基于重放的 IL2M 和基于扩展的 PCL 高出 8.30% 和 5.26%. 同样的结论也可以从图 5(b) 中得出.

### 3.2.3 ImageNet

表 3 进一步给出了不同方法在 ImageNet-200 上的实验结果. 相比 FashionMNIST 和 CIFAR-100 任务序列, 该实验中单任务的学习更具有挑战性, 多任务之间的干扰更容易发生. 首先, 虽然 MRNet 在 ImageNet-200/10 任务序列中的 ACC 仅高出 PCL 约 0.1%, 但是后者的参数量比 MRNet 多 36.16% (见表 2), 时间需求比 MRNet 高上百倍; 其次, MRNet 在 ImageNet-200/50 任务序列中的标准差略低于 PCL, 但其 ACC 高出 PCL 3.47% 且明显优于 IL2M 和 OWM; 最后, 我们注意到从 ImageNet-200/10 到 ImageNet-200/50, IL2M 的学习性能显著降低. 这在一定程度上是因为基于重放的 IL2M 允许使用 2 k 个样本大小的缓冲区, 但任务数量的增加导致新旧任务之间的数据不平衡问题越来越明显. 总的来说, 本文方法具有精度高、计算/内存成本低和训练时间快等优点.

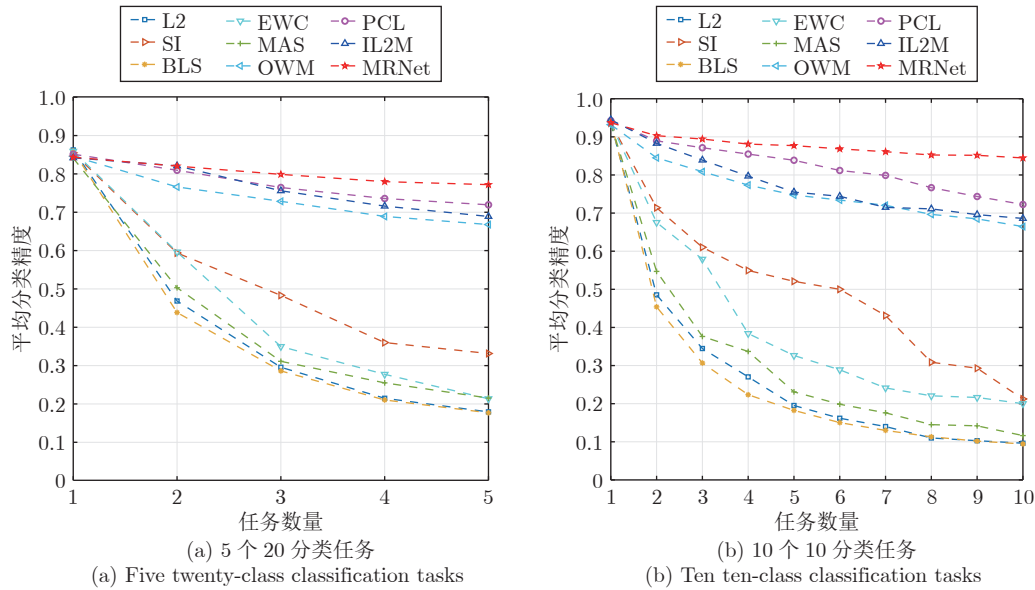


图5 不同方法在 CIFAR-100 任务序列上的分类精度曲线

Fig.5 Classification accuracy curves of different methods on CIFAR-100 task sequence

表3 连续学习 ImageNet-200 任务序列对比实验  
Table 3 Comparative experiments on continuously learning ImageNet-200 task sequence

方法	任务序列	
	ImageNet-200/10	ImageNet-200/50
IL2M	54.13±11.30	47.84±18.85
OWM	55.93±14.29	49.67±20.98
PCL	56.41±9.75	52.46±8.95
MRNet	<b>56.50±9.13</b>	<b>55.93±11.51</b>

### 3.3 参数分析

除了第 3.1.4 节给出的超参数设置, 下面进一步讨论 MRNet 在  $T$  个任务之间引入的权衡系数, 即  $\{\gamma_1, \gamma_2, \dots, \gamma_{T-1}\}$ . 如前所述, MRNet 作为一种再可塑性启发的类增量学习算法, 其关键在于“稳定性”和“可塑性”之间的平衡. 从算法实现的角度

来说, CL 模型不仅需要维持先前任务的信息, 而且需要克服旧任务对新任务的干扰. 为此, 我们从  $1 \leq \lambda_t \leq 10^8$  中选取不同数值在 CIFAR-100/5 任务序列上进行比较实验. 表 4 记录了模型学习完新任务  $t$  ( $t = 1, \dots, 5$ ) 后, 在所有见过的  $t$  个任务上的 ACC, 即  $A_t$ ; 同时, 给出了学习完 5 个任务时模型的 BWT 和 FWT. 评价指标均越高越好. 在此基础上, 我们对不同权衡系数进行了灵敏度分析. 为了简化描述, 定义  $\gamma_t$  的大小表示对旧任务的保护程度, 并依次分为: “无保护”  $\{1, 1, 1, 1\}$ , “欠保护”  $\{1, 1, 1, 1\} \times 10^2$ , “适当保护”  $\{1, 1, 1, 1\} \times 10^4$ , “过保护”  $\{1, 1, 1, 1\} \times 10^6$  和“全保护”  $\{1, 1, 1, 1\} \times 10^8$  五种情况. 1) “无保护”情况: MRNet 仅记得当前学习过的任务, 由于整个过程几乎完全偏向新任务的学习, 使得 FWT 出现了正值, 这意味着模型使用(牺牲)先前学到的知识来提高新任务的性能; 2) “欠保护”情况: 类似于“欠拟合”, 其对旧任务的保护程

表4 权衡系数灵敏度分析  
Table 4 Sensitivity analysis on the trade-off coefficients

保护程度	评价指标						
	$A_1$ (%)	$A_2$ (%)	$A_3$ (%)	$A_4$ (%)	$A_5$ (%)	BWT	FWT
1	84.45	42.88	28.20	20.51	17.45	-0.8420	<b>0.0001</b>
$10^2$	84.45	75.48	68.57	61.54	55.65	-0.3629	-0.0015
$10^4$	84.45	<b>82.33</b>	<b>80.90</b>	<b>78.46</b>	<b>77.86</b>	-0.0615	-0.0253
$10^6$	84.45	71.48	61.37	49.81	41.11	-0.0199	-0.5263
$10^8$	84.45	44.35	31.05	23.29	18.62	<b>0.0003</b>	-0.8270

度非常有限,重点仍然是当前任务;3)“适当保护”情况:该程度的保护适用于本文所提的 MRNet,有效实现了未知任务序列的学习与记忆融合;4)“过保护”情况:学习的重点转向历史任务,导致多任务之间出现干扰;5)“全保护”情况:模型通过限制当前任务的学习以维持历史任务的信息,因此 BWT 出现了正值.

需要指出的是,参数分析的目的是为 MRNet 确定对旧任务的保护程度并提供合适的权衡系数候选集.因此,可以在  $\{1, 1, 1, 1\} \times 10^4$  的基础上对  $\gamma_t$  进行微调.此外,我们注意到基于 BP 算法的正则化方法,如 (Online) EWC<sup>[20, 23]</sup>、SI<sup>[24]</sup> 和 MAS<sup>[4]</sup> 对权衡系数极为敏感,不仅容易受其他超参数(学习率, mini-batch 等)的影响,而且在不同任务序列上的设置也完全不同.换句话说,这些方法对旧任务难以预先设定好保护程度.

### 3.4 消融实验

在第 2.3.1 节提到,MRNet 在随机初始化基网络时简化了从扩展输入层到输出层的直接连接,用于匹配判别信息.如表 5 所示,分别给出了在 FashionMNIST-10/5 任务序列上不使用和使用直连结构的实验结果.可以看出,当 MRNet 带有直连结构时,其仅在第 1 个任务上的分类精度  $A_1$  优于没有直连的情况.这是因为将从扩展输入层到输出层的信息作为额外输入确实有利于增加所提取特征的判别信息,从而出现较高的  $A_1$ .然而,从旧任务中学习到旧类的判别信息在新类之间以及旧类和新类之间往往不具有判别性,这导致其  $A_2 \sim A_5$ , BWT 和 FWT 明显降低,加剧了灾难性遗忘的发生.

表 5 MRNet 结构分析  
Table 5 Analysis on MRNet architecture

有无直连	评价指标						
	$A_1$ (%)	$A_2$ (%)	$A_3$ (%)	$A_4$ (%)	$A_5$ (%)	BWT	FWT
×	98.20	92.58	93.98	93.34	92.61	-0.0199	-0.0560
√	99.87	34.14	33.83	32.01	28.40	-0.1304	-0.1883

## 4 结束语

本文基于随机神经网络建立了统一的连续学习框架,用于有效兼容未来非独立同分布任务以及伴随出现的新类别,并给出了无需梯度下降算法的解析解,包括受启发于突触再可塑性构造了具备记忆功能的权值重要性矩阵用于自适应地调整网络参

数,从而维持对历史任务的记忆.与现有经典和最先进的连续学习方法相比,所提 MRNet 具有参数高效、收敛速度快、人为干预度少等优点.与此同时,它打破了现有连续学习方法需要多次遍历访问(对应较大的 epoch)当前任务的数据以在新任务上获得更好性能的约束,是一种更加具有通用性的学习与记忆融合模型与算法.然而,MRNet 结构包括一个全局特征提取器和特征到端的简单分类器,这表明本文所提方法需要借助预训练模型.因此,我们将在 MRNet 基础上进一步建立端到端的连续学习器.

## References

- McCloskey M, Cohen N J. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*. Elsevier, 1989.
- French R M. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 1999, **3**(4): 128–135
- McClelland J L, McNaughton B L, O'Reilly R C. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 1995, **102**(3): 419–457
- Aljundi R, Babiloni F, Elhoseiny M, Rohrbach M, Tuytelaars T. Memory aware synapses: Learning what (not) to forget. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Munich, Germany: Springer, 2018. 139–154
- Li Z Z, Hoiem D. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, **40**(12): 2935–2947
- Parisi G I, Kemker R, Part J L, Kanan C, Wermter S. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019, **113**: 54–71
- Li Z Z, Hoiem D. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, **44**(7): 3366–3385
- Perkonig M, Hofmanninger J, Herold C J, Brink J A, Pinykh O, Prosch H, et al. Dynamic memory to alleviate catastrophic forgetting in continual learning with medical imaging. *Nature Communications*, 2021, **12**(1): 1–12
- Mallya A, Lazebnik S. Packnet: Adding multiple tasks to a single network by iterative pruning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, USA: IEEE, 2018. 7765–7773
- Rosenfeld A, Tsotsos J K. Incremental learning through deep adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, **42**(3): 651–663
- Hu W P, Qin Q, Wang M Y, Ma J W, Liu B. Continual learning by using information of each class holistically. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021, **35**(9): 7797–7805
- Yang B Y, Lin M B, Zhang Y X, Liu B H, Liang X D, Ji R R, et al. Dynamic support network for few-shot class incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023, **45**(3): 2945–2951

- 13 Shin H, Lee J K, Kim J, Kim J. Continual learning with deep generative replay. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS). Long Beach, USA: Curran Associates, Inc., 2017. 2990–2999
- 14 Ven van de G M, Siegelmann H T, Tolias A S. Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications*, 2020, **11**(1): 1–14
- 15 Belouadah E, Popescu A. IL2M: Class incremental learning with dual memory. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, South Korea: IEEE, 2019. 583–592
- 16 Lopez-Paz D, Ranzato M. Gradient episodic memory for continual learning. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS). Long Beach, USA: Curran Associates, Inc., 2017. 6470–6479
- 17 Chaudhry A, Marc’Aurelio R, Rohrbach M, Ellhoseiny M. Efficient lifelong learning with A-GEM. In: Proceedings of the International Conference on Learning Representations (ICLR). New Orleans, USA: 2019.
- 18 Tang S X, Chen D P, Zhu J G, Yu S J, Ouyang W L. Layer-wise optimization by gradient decomposition for continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, USA: IEEE, 2021. 9634–9643
- 19 Zhang X Y, Zhao T F, Chen J S, Shen Y, Li X M. EPicker is an exemplar-based continual learning approach for knowledge accumulation in cryoEM particle picking. *Nature Communications*, 2022, **13**(1): 1–10
- 20 Schwarz J, Czarnecki W, Luketina J, Grabska-Barwinska A, Teh Y W, Pascanu R, et al. Progress & compress: A scalable framework for continual learning. In: Proceedings of the International Conference on Machine Learning (ICML). Stockholm, Sweden: JMLR, 2018. 4528–4537
- 21 Zhang J T, Zhang J, Ghosh S, Li D W, Tasci S, Heck L, et al. Class-incremental learning via deep model consolidation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). Snowmass, USA: IEEE, 2020. 1131–1140
- 22 Liu X B, Wang W Q. GopGAN: Gradients orthogonal projection generative adversarial network with continual learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023, **34**(1): 215–227
- 23 Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu A A, et al. Overcoming catastrophic forgetting in neural network. *Proceedings of the National Academy of Sciences (PNAS)*, 2017, **114**(13): 3521–3526
- 24 Zenke F, Poole B, Ganguli S. Continual learning through synaptic intelligence. In: Proceedings of the International Conference on Machine Learning (ICML). Sydney, Australia: JMLR, 2017. 3987–3995
- 25 Zeng G X, Chen Y, Cui B, Yu S. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 2019, **1**(8): 364–372
- 26 Gao J Q, Li J Q, Shan H M, Qu Y Y, Wang J Z, Zhang J P. Forget less, count better: A domain-incremental self-distillation learning benchmark for lifelong crowd counting. arXiv preprint arXiv: 2205.03307, 2022.
- 27 Meng Xi, Qiao Jun-Fei, Han Hong-Gui. Soft measurement of key effluent parameters in wastewater treatment process using brain-like modular neural networks. *Acta Automatica Sinica*, 2019, **45**(5): 906–919  
(蒙西, 乔俊飞, 韩红桂. 基于类脑模块化神经网络的污水处理过程关键出水参数软测量. *自动化学报*, 2019, **45**(5): 906–919)
- 28 Nadji-Tehrani M, Eslami A. A brain-inspired framework for evolutionary artificial general intelligence. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, **31**(12): 5257–5271
- 29 Hu B, Guan Z H, Chen G R, Chen C L P. Neuroscience and network dynamics toward brain-inspired intelligence. *IEEE Transactions on Cybernetics*, 2022, **52**(10): 10214–10227
- 30 LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998, **86**(11): 2278–2324
- 31 Pao Y H, Takefji Y. Functional-link net computing: Theory, system architecture, and functionalities. *Computer*, 1992, **25**(5): 76–79
- 32 Schmidt W F, Kraaijveld M A, Duin R P W. Feedforward neural networks with random weights. In: Proceedings of the 11th IAPR International Conference on Pattern Recognition. IEEE Computer Society, 1992. 1–4
- 33 Igel'nik B, Pao Y H. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Transactions on Neural Networks*, 1995, **6**(6): 1320–1329
- 34 Cao W P, Wang X Z, Ming Z, Gao J Z. A review on neural networks with random weights. *Neurocomputing*, 2011, **275**: 278–287
- 35 Zhang L, Suganthan P N. Visual tracking with convolutional random vector functional link network. *IEEE Transactions on Cybernetics*, 2016, **47**(10): 3243–3253
- 36 Dai W, Li D P, Zhou P, Chai T Y. Stochastic configuration networks with block increments for data modeling in process industries. *Information Sciences*, 2019, **484**: 367–386
- 37 Zou Wei-Dong, Xia Yuan-Qing. Virtual machine performance prediction using broad learning system based on compression factor. *Acta Automatica Sinica*, 2022, **48**(3): 724–734  
(邹伟东, 夏元清. 基于压缩因子的宽度学习系统的虚拟机性能预测. *自动化学报*, 2022, **48**(3): 724–734)
- 38 Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: Theory and applications. *Neurocomputing*, 2006, **70**(1–3): 489–501
- 39 Wang D H, Li M. Stochastic configuration networks: Fundamentals and algorithms. *IEEE Transactions on Cybernetics*, 2017, **47**(10): 3466–3479
- 40 Chen C L P, Liu Z L. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Transactions on Neural Networks and Learning Systems*, 2017, **29**(1): 10–24
- 41 Dai Wei, Li De-Peng, Yang Chun-Yu, Ma Xiao-Ping. A model and data hybrid parallel learning method for stochastic configuration networks. *Acta Automatica Sinica*, 2021, **47**(10): 2427–2437  
(代伟, 李德鹏, 杨春雨, 马小平. 一种随机配置网络的模型与数据混合并行学习方法. *自动化学报*, 2021, **47**(10): 2427–2437)
- 42 Gong X R, Zhang T, Chen C L P, Liu Z L. Research review for broad learning system: Algorithms, theory, and applications. *IEEE Transactions on Cybernetics*, 2022, **52**(9): 8922–8950
- 43 Abraham W C, Bear M F. Metaplasticity: The plasticity of synaptic plasticity. *Trends in Neurosciences*, 1996, **19**(4): 126–130

- 44 Wang Shao-Li, Lu Wei. Progress on metaplasticity and its role in learning and memory. *Acta Physiologica Sinica*, 2016, **68**(4): 475–482  
(王韶莉, 陆巍. 再可塑性在学习记忆中作用的研究进展. 生理学报, 2016, **68**(4): 475–482)
- 45 Jedlicka P, Tomko M, Robins A, Abraham W C. Contributions by metaplasticity to solving the catastrophic forgetting problem. *Trends in Neurosciences*, 2022, **45**(9): 656–666
- 46 Sussmann H J. Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural Networks*, 1992, **5**(4): 589–593
- 47 Lancaster P, Tisemenetsky M. *The Theory of Matrices: With Applications*. Elsevier, 1985.
- 48 Kay S M. Fundamentals of statistical signal processing: Estimation theory. *Traces and Emergence of Nonlinear Programming*. Prentice-Hall, Inc, 1993.
- 49 Kuhn H W, Tucker A W. Nonlinear programming. *Traces and Emergence of Nonlinear Programming*. Springer, 2014.
- 50 Pan P, Swaroop S, Immer A, Eschenhagen R, Turner R, Khan M, et al. Continual deep learning by functional regularisation of memorable past. In: Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS). Vancouver, Canada: 2020. 4453–4464
- 51 Verma V K, Liang K J, Mehta N, Rai P, Carin L. Efficient feature transformations for discriminative and generative continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, USA: IEEE, 2021. 13865–13875



**李德鹏** 华中科技大学人工智能与自动化学院博士研究生. 主要研究方向为增量学习, 对抗机器学习, 脑启发神经网络, 计算机视觉.

E-mail: dpli@hust.edu.cn

**(LI De-Peng** Ph.D. candidate at the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His research interest covers incremental learning, adversarial machine learning, brain-inspired neural networks, and computer vision.)



**曾志刚** 华中科技大学人工智能与自动化学院教授. 主要研究方向为神经网络理论与应用, 动力系统稳定性分析, 联想记忆. 本文通信作者.

E-mail: zgzeng@hust.edu.cn

**(ZENG Zhi-Gang** Professor at the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His research interest covers theory and applications of neural networks, stability analysis of dynamic systems, and associative memories. Corresponding author of this paper.)