动态模块化稀疏通用持续学习

Arnav Varma[1], ElaheArani[†1], 2 和 Bahram Zonooz[†1], 2[1]高级研究实验室，四维图新欧洲，荷兰埃因霍温[2]埃因霍温理工大学数学与计算机科学系，荷兰 *arnav.varma@navinfo.eu, e.arani@tue.nl, bahram.zonooz@gmail.com*

† 贡献相同

关键词：动态神经网络、策略梯度、终身学习。

摘要：现实世界的应用通常需要在不断变化的条件下，从数据流中持续学习。尝试从这类非平稳数据中学习时，深度神经网络（DNN）会灾难性地遗忘先前学到的信息。在避免灾难性遗忘的常见方法中，基于回放的方法已被证明有效。然而，由于所有参数都会响应所有任务，这些方法仍容易因任务干扰而发生遗忘。为此，我们从大脑中的稀疏编码获得启发，为基于回放的通用持续学习引入动态模块化与稀疏性（*Dynamos*）。在这一设置中，深度神经网络通过激活相关的神经元子集来学习响应刺激。我们在多个数据集上，依据具有挑战性的持续学习评估协议，展示了 *Dynamos* 的有效性。最后，我们表明，该方法学到的表征具有模块化和专门化特性，同时通过激活相互重叠程度与刺激相似性相对应的神经元子集，保持了可复用性。代码可在 https://github.com/NeurAI-Lab/DynamicContinualLearning 获取。

arXiv:2301.00620v1 [cs.CV] 2 Jan 2023

1 引言

深度神经网络（DNN）在多个应用中已达到人类水平的性能（Greenwald 等人，2021年；Taigman 等人，2014年）。这些网络针对应用中的多个任务进行训练，并假设接收的数据相互独立且服从相同分布（i.i.d.）。通过打乱所有任务的数据，并对应用中每个任务的样本进行平衡和归一化，即可满足这一假设（Hadsell 等人，2020年）。因此，DNN 可以通过将数据的联合分布建模为平稳过程，在这些应用的所有任务上达到人类水平的性能。另一方面，人类能够根据本质上非平稳且连续的观察来构建对世界的认知（French，1999年）。对于终身学习机器人（Thrun 和 Mitchell，1995年）和自动驾驶汽车（Nose 等人，2019年）等许多应用而言，持续从更符合现实的连续、非平稳数据中学习至关重要。然而，在这类持续学习设置中，对连续不断的任务和数据流进行常规的基于梯度的训练，会导致深度神经网络参数发生任务干扰，进而使网络灾难性地遗忘旧任务（McCloskey 和 Cohen，1989年；Kirkpatrick 等人，2017年）。因此，需要能够缓解持续学习中灾难性遗忘的方法。

以往研究致力于应对持续学习中的这些挑战，大致可分为三类。第一类是基于正则化的方法（Kirkpatrick 等，2017；Schwarz 等，2018；Zenke 等，2017），通过惩罚深度神经网络参数的变化来减少任务间干扰。第二类是参数隔离方法（Adel 等，2020），为不同任务分配互不相同的参数子集。最后一类是基于回放的方法（Chaudhry 等，2019），通过对当前样本和存储的先前样本进行联合训练。在这些方法中，基于正则化和参数隔离的方法通常需要额外信息（例如测试时的任务标识和训练期间的任务边界），或允许网络不受限制地增长。这些要求不符合通用持续学习（GCL）的期望（Delange 等，2021；Farquhar 和 Gal，2018），因此这些方法不适用于通用持续学习。

尽管基于回放的方法优于其他类别的方法，也符合 GCL 期望，但它们仍会因深度神经网络参数中的任务干扰而遭受灾难性遗忘，因为所有参数都会对所有样本和任务作出响应。这一问题或许可以通过以下方式解

决：在基于回放的方法中引入针对特定任务或样本的参数隔离。然而，值得注意的是，与参数隔离方法不同，大脑中的模块化和稀疏性并非静态不变。有证据表明，大脑会以动态且稀疏的方式响应刺激，不同模块或神经元子集会对不同刺激作出“动态”响应（Graham and Field，2006年）。在稳态情境下，深度学习领域已通过模块门控（Veit and Belongie，2018年）、提前退出（Li等人，2017年；Hu等人，2020年）和动态路由（Wang等人，2018年）等机制，以及鼓励神经激活稀疏性的训练损失（Wu等人，2018年），探索了动态稀疏地响应刺激的优势。这些研究发现，经过动态预测训练的深度神经网络也会学会对不同输入作出不同响应。此外，学习得到的深度神经网络会根据输入的相似性、难度和分辨率等任务特征对参数进行聚类（Wang等人，2018年；Veit and Belongie，2018年），这表明其具有动态模块化。因此，我们假设，将基于回放的方法与动态稀疏性和模块化相结合，有望在符合 GCL 期望的同时，以更符合生物学实际的方式进一步缓解灾难性遗忘。

为此，我们提出动态模块化与稀疏性（*Dynamos*），这是一种通用持续学习算法，将基于回放的方法与动态模块化和稀疏性相结合。具体而言，我们力求实现三个目标：借助专用模块对输入作出动态且稀疏的响应、保持良好的性能，以及减少灾难性遗忘。为实现对输入的动态稀疏响应，我们在深度神经网络中定义多个智能体，每个智能体负责根据某卷积层的输入，动态地将该层滤波器的激活值置零。如果网络预测准确，智能体选择移除激活值（即产生稀疏响应）的动作便会得到奖励；而若所选动作导致预测不准确，则会受到严厉惩罚。智能体还利用原型损失学习专用特征。为减少遗忘并保持良好性能，我们维护一个大小固定的记忆缓冲区，用于存储先前见过的样本。网络会在当前样本之外，使用先前样本重新训练，从而既维持其在当前和先前任务上的性能，也确保对刺激作出的当前响应与先前响应保持一致。*Dynamos*在多个持续学习数据集上、采用多种评估协议（包括通用持续学习）时，均展现出良好的性能。此外，我们的方法对相似输入表现出相似且重叠的响应，对不相似输入则表现出截然不同的响应。最后，我们证明，该方法能够模拟人类中观察到的试次间变异性（Faisal 等人，2008年；Werner and Mountcastle，1963年）。

2 相关工作

深度学习研究通过动态神经网络，探索了人脑中观察到的动态组合性和稀疏性：针对不同刺激，激活不同的神经元子集或不同的子网络（Bengio et al., 2015; Bolukbasi et al., 2017）。这一点可通过提前退出（Hu et al., 2020）、经由专家混合或多个分支实现动态路由（Collier et al., 2020; Wang et al., 2022），以及模块门控（Wang et al., 2018）来实现。提前退出可能会迫使深度神经网络在较早的层中学习特定特征，从而损害性能（Wu et al., 2018），因为众所周知，深度神经网络的早期层会学习通用特征（Yosinski et al., 2014）。另一方面，动态路由需要针对新任务不断增添专家，这可能导致网络不受约束地增长；或者需要初始化一个规模更大的深度神经网络，并设置与预期任务数量相对应的分支（Chen et al., 2020）。与此同时，研究表明，采用嵌入小型门控网络的标准深度神经网络，具有门控机制的动态网络在独立同分布训练中能够取得良好性能（Veit and Belongie, 2018; Wu et al., 2018; Wang et al., 2018）。这些门控网络会根据模块或深度神经网络的输入，为每个模块输出一个离散的保留/丢弃决策。由于这一操作不可微分，通常会在每个模块中使用Gumbel-Softmax近似（Veit and Belongie, 2018; Wang et al., 2018），或采用通过策略梯度训练的智能体（Wu et al., 2018; Sutton and Barto, 2018），以实现反向传播。然而，与后者不同，Gumbel-Softmax近似会导致推理和训练时前向传播激活之间存在不对称性（Wang et al., 2018）。此外，这些方法并不适用于持续学习。

近期研究尝试为持续学习场景构建动态网络（Chen 等人，2020年；Abati 等人，2020年），在这类场景中，数据以更贴近现实的顺序方式到达。InstAParam（Chen 等人，2020年）、随机路径选择（RPS）（Rajasegaran 等人，2019年）和专家混合（MoE）（Collier 等人，2020年）从每个层中的多个并行块开始，并在这个大型网络中寻找

特定于输入或任务的路径。不过，这需要在训练前预先知道要学习的任务数量。更重要的是，动态网络在独立同分布训练中借助门控机制展现出的良好性能表明，初始化一个大型网络或许并无必要。与此不同，MNTDP（Veniat 等人，2021年）、LMC（Ostapenko 等人，2021年）和 CCGN（Abati 等人，2020年）从标准架构开始，并通过增加单元来响应新数据或新任务。在这些方法中，MNTDP 和 LMC 构建特定于任务的网络，同一任务的所有输入都会引发相同的响应，因此无法对刺激做出真正动态的响应。相比之下，CCGN 会动态组合卷积滤波器来响应刺激，为每个卷积滤波器使用特定于任务的向量，并利用任务边界冻结频繁激活的滤波器。然而，这会导致网络无节制地增长，而且在缺少任务边界时无法奏效，因此不适用于通用持续学习。

因此，我们提出一种通用的持续学习方法，通过使用策略梯度训练的强化学习智能体，诱导出动态模块化与稀疏性（Dynamos）。

3 方法

人类能够从世界固有的非平稳、序列式观察中持续学习，而不会发生灾难性遗忘；即使没有关于待执行任务或新任务到来的监督，也能始终保持有限的记忆。这涉及多个方面，其中包括在当前与先前观察之间建立多尺度关联（Goyal 和 Bengio，2020），以及对刺激作出稀疏且动态的响应（Graham and Field，2006）。前者关乎对过往经验的巩固，并确保已学到的经验能够引发相似的响应。后者则关乎动态组合可用的专门神经模块子集，以响应刺激，并仅重用先前学到的相关信息。这也避免了抹除与当前刺激无关、却与过往经验相关的信息。现在，我们提出一种动态、稀疏且模块化的通用持续学习方法，利用深度神经网络模拟这些过程。

3.1 对刺激的动态、模块化与稀疏响应

为实现对输入的动态、模块化和稀疏响应，我们使用一个深度神经网络 F，并通过一种策略在每一层组合可用模块的子集，以响应输入到该层的数据。具体而言，我们使用一个卷积神经网络，通过策略梯度使其能够自适应地丢弃激活值中的部分通道（Sutton 和 Barto，2018年；Williams，1992年）。

考虑具有l^{th} 个输出通道c_l的卷积层 $\forall l \in \{1,2,..L\}$，其中$L$是网络中卷积层的总数。卷积层的输入由智能体模块处理，并输出动作$a_l \in \{0,1\}^{c_l}$；每个动作表示决定丢弃（动作 = 0）或保留（动作 = 1）卷积层输出中相应的通道。智能体模块使用自注意力网络获得维度为c_l的通道级注意力向量v_l，再通过概率层将其转换为“动作概率”。随后，从c_l维伯努利分布中采样得到动作选择策略；

$$\begin{aligned} p_l &= \sigma(v_l) \\ \pi_l(a_l) &= \prod_{i=1}^{c_l} p_{l,i}^{a_{l,i}} (1-p_{l,i})^{(1-a_{l,i})}, \end{aligned} \tag{1}$$

其中，$p_l \in (0,1)^{c_l}$ 是概率层 σ的输出，而 π_l 是策略函数。卷积层的最终输出是动作与卷积输出的逐通道乘积。在 CNN 的每个卷积层中均采用这种策略形式，因此总共有L 个智能体。卷积层中智能体的整体结构如图1所示。

这些智能体通过奖励函数获得奖励：在准确预测的同时丢弃通道。对于输入到用于分类的深度神经网络X 以及标签Y:

$$\begin{aligned} Z, V &= F(X), V = [v_1 | v_2, ... | v_L] \\ \hat{Y} &= \arg\max Z, \end{aligned} \tag{2}$$

其中Z 表示逻辑值。现在，第l 层中保留的激活或通道比例由 $^1_{cl}\sum^{c_l}_{i=1}a_{l,i}$确定。因此，对于每层的目标激活保留率或“保留率”$kr$，奖励函数如下:

$$R_l(X,Y) = \begin{cases} -(kr - \frac{1}{c_l}\sum_{i=1}^{c_l} a_{l,i})^2, & \text{if } \hat{Y} = Y \\ -\lambda(kr - \frac{1}{c_l}\sum_{i=1}^{c_l} a_{l,i})^2, & \text{otherwise.} \end{cases} \tag{3}$$

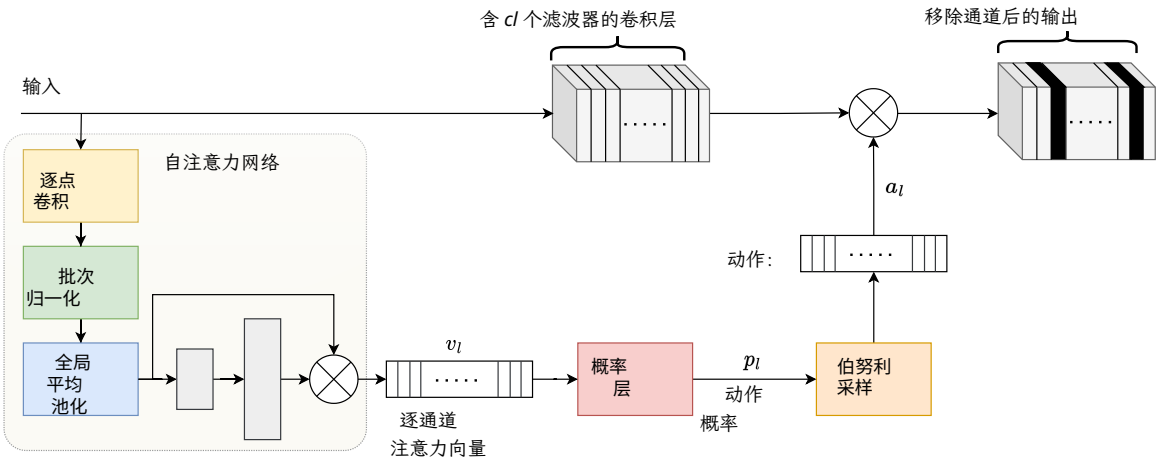


图1：*Dynamos* 在l^{th} 卷积层的动态稀疏响应机制概览。被涂黑的激活值已移除。智能体（底部路径）的自注意力网络使用逐点卷积匹配输出通道，并通过全局平均池化得到一个通道长度的展平向量。该向量经过一个带有单个隐藏层和 Sigmoid 激活函数的多层感知机处理，再与原始的通道长度表示相乘，从而得到逐通道自注意力向量。

因此，当深度神经网络的预测正确时，每个智能体都会因丢弃足够多的激活，使其符合对应卷积层的“保留率”而获得奖励。然而，当预测错误时，每个智能体都会因此受到惩罚，惩罚力度由常数惩罚因子 λ 缩放。奖励函数通过依赖预测的正确性而具有全局性，这也促使智能体之间进行协调。按照 REINFORCE（Williams，1992年），所有智能体$t = 1,2,...L$的损失为：

$$\begin{aligned} L_R(X,Y) &= \mathbb{E}_l\mathbb{E}_\pi[-R_l(X,Y)\log\pi_l(a_l)] \\ &= \mathbb{E}_l\mathbb{E}_\pi[-R_l(X,Y)\log\prod_{i=1}^{c_l} p_{l,i}a_{l,i} \\ &\qquad + (1-p_{l,i})(1-a_{l,i})] \qquad (4) \\ &= \mathbb{E}_l\mathbb{E}_\pi[-R_l(X,Y)\sum_{i=1}^{c_l}\log[p_{l,i}a_{l,i} \\ &\qquad + (1-p_{l,i})(1-a_{l,i})]]. \end{aligned}$$

尽管这些智能体与该损失共同确保了深度神经网络产生稀疏且动态的响应，但它们并未明确促使神经模块像人类那样进行组合专门化。由于深度神经网络中被激活的逐通道“模块”直接取决于逐通道注意力向量，因此我们最终对这些向量应用一种专门化损失，并将其称为原型损失。具体而言，对于分类，在任意一个输入批次中，我们将属于同一类别的向量拉近，同时将属于不同类别的向量推远。这会使不同类别的输入使用不同子集的逐通道模块，从而对不同类别的输入采取不同的模块组合。当与足够高的“保留率”结合时，这会促进重叠，进而复用相关的既有学习信息（例如，将对应于已学习类别的通道复用于新观察到的类别），并因此促使模块学习通用特征。对于输入批次X 及其对应的标签Y，以及相应的拼接逐通道注意力向量批次V （公式 2），原型损失定义如下：

$$L_P(X,Y) = \frac{1+\Sigma_{(V_1,V_2)\in V^2:Y_1=Y_2} MSE(V_1,V_2)}{1+\Sigma_{(V_1,V_2)\in V^2,Y_1\neq Y_2} MSE(V_1,V_2)}, \quad (5)$$

其中，MSE 指均方误差估计器。请注意，我们仅对预测正确的样本应用此损失。

3.2 多尺度关联

如前所述，人类用于减轻遗忘的机制之一，是在当前经验与先前经验之间建立多尺度关联。

秉持这一目标，我们遵循近期基于复习的方法（Buzzega 等人，2020年；Riemer 等人，2019年），这些方法符合 GCL，并在训练期间使用记忆缓冲区存储此前见过的示例及其响应。缓冲区采用蓄水池采样（Vitter，1985年）进行更新，从而有助于近似迄今为止见过的样本分布（Isele 和 Cosgun，2018年）。然而，我们只考虑预测正确的批次样本子集，将其加入记忆缓冲区

。这些缓冲区样本与新样本一起通过深度神经网络进行回放，并通过损失将当前响应与存储的先前响应关联起来，从而使响应随时间保持一致。

令M 表示记忆缓冲区，D_T 表示当前任务流，分别从中采样批次（X_M，Y_M，Z_M，V_M）和（X_t，Y_t）。其中，Z_M 和V{v37>分别是在最初观测X_M 时保存的逻辑值和通道级注意力向量。在任务T 期间，当前响应和先前响应对应的一致性损失计算如下：

$$\begin{aligned} Z_M^{'}, V_M^{'} &= F(X_M) \\ L_C(Z_M, Z_M^{'}) &= \mathbb{E}_{X_M}[\|Z_M - Z_M^{'}\|_2^2] \\ L_C(V_M, V_M^{'}) &= \mathbb{E}_{X_M}[\|V_M - V_M^{'}\|_2^2]. \end{aligned} \tag{6}$$

除了施加一致性损失外，我们还在记忆样本上约束准确率、动态稀疏性和模块化。因此，我们有四组损失：

- 当前样本和记忆样本上的任务性能损失，以确保当前任务和先前任务的正确性。对于分类，我们使用交叉熵损失（L_{CE}）。
- 当前样本和记忆样本上的奖励损失（公式 4），以确保当前任务和先前任务的动态模块化与稀疏性。
- 当前样本和记忆样本上的原型损失（公式 5），以确保模块在当前任务和先前任务上的专门化。
- 用于当前样本与先前样本之间多尺度关联的一致性损失（公式 6）。

综合以上各项，总损失为：

$$\begin{aligned} L_{\text{total}} = & L_{CE}(X_B, Y_B) + \gamma L_r(X_B) \\ & + \beta[L_{CE}(X_M, Y_M) + \gamma L_r(X_M)] \\ & + \alpha L_C(Z_M, Z_M^{'}) + \alpha_p L_C(V_M, V_M^{'}) \\ & + w_p[L_P(X_B, Y_B) + L_P(X_M, Y_M)]. \end{aligned} \tag{7}$$

赋予损失 α、 α_p、 β、w_P 和 γ 的权重，以及公式 3 中误分类惩罚（λ）和保留率（kr），均为超参数。请注意，我们在训练开始时采用预热阶段，在此期间既不使用记忆缓冲区，也不使用智能体。这相当于在这一阶段仅使用交叉熵损失进行训练，同时冻结智能体。这样可以在智能体开始搜索解决方案时，为其提供更好的搜索空间。我们将上述方法称为动态模块化与稀疏性——*Dynamos*。

4 实验详情

数据集。我们展示了 MNIST（LeCun 等人，1998年）和 SVHN 的序列变体结果，分别为 Seq-MNIST 和 Seq-SVHN（Netzer 等人，2011年）。Seq-MNIST 和 Seq-SVHN 分别将各自的数据集划分为 5 个任务，每个任务包含 2 个类别。此外，为了测试 *Dynamos* 在通用持续学习中的适用性，我们还使用 MNIST-360 数据集（Buzzega 等人，2020年）。

架构。 我们采用基于 ResNet-18（He 等人，2016）结构的网络，移除其四个模块中靠后的两个，并将每个卷积层的滤波器数量从 64 减少到 32。初始卷积层的规模缩减为 3×3，以适应更小的图像。基线实验中，我们没有使用任何智能体。对于我们的方法，虽然所有卷积层都可以使用智能体，但我们只在第二个模块中使用智能体。我们基于近期研究作出这一选择：研究发现，较早的层几乎不会发生遗忘（Davari 等人，2022），具有很强的可迁移性（Yosinski 等人，2014），即使是在通过动态模块化学习时，也会用于大多数样本（Abati 等人，2020）。我们在智能体中使用带温度层的 Sigmoid 函数作为概率层，并将 0.5 设为选择动作（即推理期间选择通道）的阈值。温度用于调节自注意力层输出的范围，确保用于选择动作的采样概率不会过小，并选择足够多的激活以支持学习。每个实验所用的具体网络结构，包括智能体的自注意力网络，可见附录中的表3和表4。

设置。 所有方法均在 PyTorch 1.6 中的 Mammoth 仓库[1] 实现，并在 Nvidia V100 GPU 上进行训练。每项实验对应的超参数见附录表 5。我们始终将保留比率维持在高于 1/任务数 的水平，以便学习具有重叠性、-可复用性和通用性的模块。除非另有说明，概率层的 Sigmoid 激活函数温度保持为 0.15。

[1]https://github.com/aimagelab/mammoth/

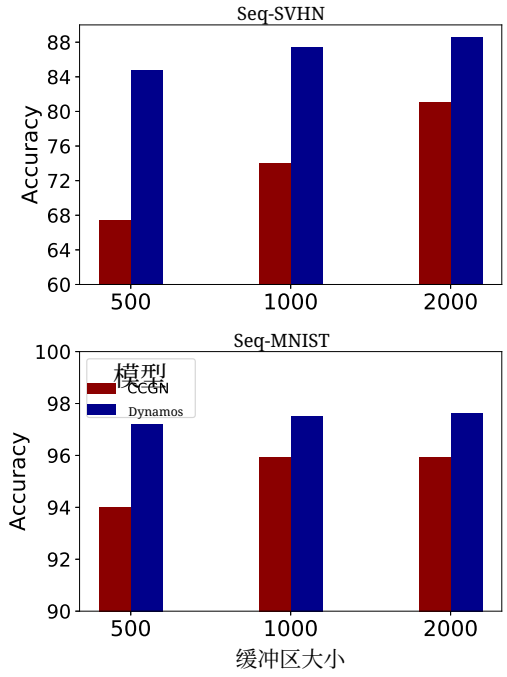


图2：类别增量学习协议下的定量结果。结果为三个随机种子的平均值。CCGN 数值取自原始论文。具体准确率见表 2。

5 结果

我们将根据两种标准评估协议对 *Dynamos* 进行评估，这些协议均遵循通用持续学习的核心要求。

5.1 类别增量学习（CIL）

类别增量学习（CIL）指的是一种评估协议：将互斥的类别集合依次呈现给网络，并且测试时不提供任务身份，这满足了通用持续学习（GCL）的核心要求（Farquhar 和 Gal，2018年）。我们将其与条件卷积门控网络（CCGN）（Abati 等人，2020年）进行比较，该网络同样会动态组合卷积滤波器以实现持续学习。从图2可以看出，在所有缓冲区大小下，*Dynamos* 在 Seq-MNIST 和 Seq-SVHN 数据集上的准确率都更高。然而，CCGN 要求每个卷积层中的每项任务都配备一个单独的任务向量，导致训练期间模型不受限制地增长；而我们在训练过程中维持了有界的记忆占用。此外，与 CCGN 不同，我们在训练期间不利用任务边界或验证集。因此，在类别增量学习中，*Dynamos* 的表现优于此前动态组合式持续学习的最先进方法，同时在训练期间保持了有界的记忆占用。

5.2 通用持续学习（GCL）

迄今为止，我们已在 CIL 协议下观察了 *Dynamos* 。与 CIL 不同，现实世界的数据流没有明确的任务边界，同一数据可能会以不同分布（例如不同姿势）再次出现。参照（Buzzega 等人，2020年），我们使用 MNIST-360 来近似这一设定，其中各任务的数字（即类别）存在重叠，会以不同旋转角度（即不同分布）再次出现，并且每个样本在训练期间恰好出现一次。这用于验证模型是否符合 GCL 期望（Farquhar 和 Gal，2018年；Delange 等，2021年）。我们通过逐步从*Dynamos*中移除动态模块化和多尺度关联，研究二者的影响。当二者都不使用时，学习采用标准的基于梯度的训练，不采用任何应对遗忘的策略。移除动态模块化后，所采用的学习策略构成我们的基线，此时不使用智能体，从而将公式 7 中的总训练损失简化为：

$$L_{\text{base}} = L_{CE}(X_B, Y_B) + \beta L_{CE}(X_M, Y_M) + \alpha L_C(Z_M, Z_M^{'}). \tag{8}$$

表 1 显示，在所有缓冲区大小下，*Dynamos* 的表现都优于基线，证明动态模块化对通用持续学习有利。此外，当多尺度关联也被移除、不使用缓冲区时，深度神经网络会发生灾难性遗忘。因此，*Dynamos*适用于通用持续学习，其动态模块化性能优于基线。我们假设，动态模块化能够通过自适应地重复利用先前学习到的相关信息，使通用持续学习中模糊的任务边界更易于处理；在此情况下，这些信息对应于学习到的滤波器。

6 模型特征

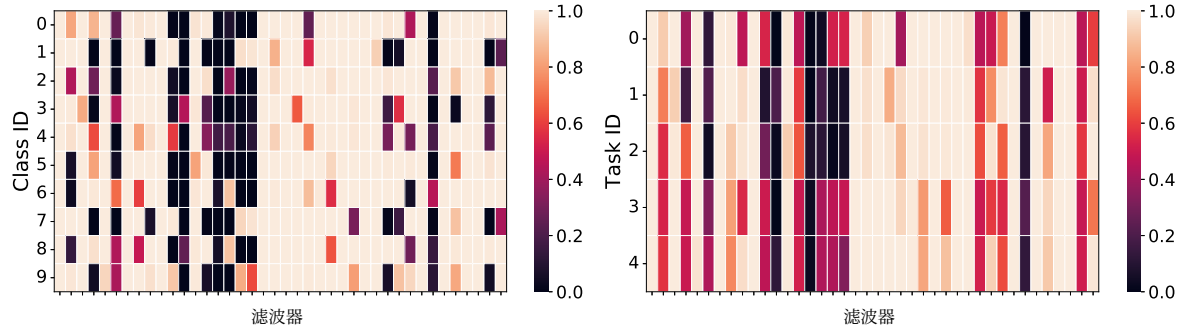
现在，我们来分析*Dynamos*的一些特征和优势。本节的所有实验均使用我们在 Sequential-MNIST 上训练的模型，缓冲区大小为 500。

6.1 动态模块化与组合性

人类会对刺激表现出模块化和专门化的反应（Meunier等，2010年），并以动态且稀疏的

表 1：不同缓冲区大小下的通用持续学习结果。所有结果均为五个随机种子的平均值。

多尺度关联	动态模块化	缓冲区大小		
		100	200	500
✓	✓	**64.418** ± 4.095	**79.638** ± 2.853	**90.519** ± 0.737
✓	✗	61.192 ± 3.072	75.364 ± 1.259	88.150 ± 0.888
✗	✗		18.712 ± 0.690	



(a) 按类别　　(b) 按任务

图3：滤波器在测试集上针对各任务和类别的激活率。为便于可视化，w e只查看最后40个滤波器。完整的可视化内容见附录（图7）。

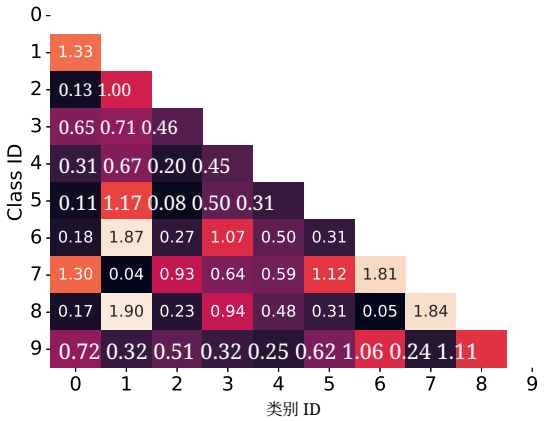


图4：测试集上类别对激活率的詹森-香农散度（×100）。

方式响应输入（Graham and Field，2006年）——我们在深度神经网络学习任务序列时，通过动态移除卷积层的通道激活，使其具备了这一能力。因此，我们在图3中考察了每个神经元（滤波器）放电率在不同任务和类别下的倾向。

可以看出，*Dynamos* 学习到了任务和类别的软分离，这分别体现在每个滤波器按任务和按类别统计的放电率上。这与静态方法形成对比，后者的所有滤波器都会对所有样本作出反应。图3a进一步表明，这使模型能够为相似样本学习相似的激活模式。例如，MNIST 数字对1和7、6和8在形状上有一些相似之处，其激活模式／放电率也相似。这可能是因为模型能够动态地复用和丢弃学习到的滤波器，使深度神经网络对相似输入作出相似反应，并根据样本间的相似性划分其响应。此外，动态复用滤波器的能力使深度神经网络能够为不相似的样本和类别学习相互重叠的激活模式，而不是使用完全不同的激活模式。这也有助于模型在无需扩展深度神经网络容量或在初始化时配置更大容量的情况下学习任务序列，这与用于持续学习的静态参数隔离方法形成对比。

按照 Abbasi 等人（2022）的做法，我们在图4中使用詹森-香农散度（JSD）量化最后一层中每对类别激活率之间的重叠程度。JSD 越低，表示重叠程度越高。类别对（1，7）的 JSD 最低（这两个数字看起来都像竖线），而 ~ $\frac{1}{15}$ 是类别对之间的平均 JSD，~ $\frac{1}{42}$ 是重叠程度最低的类别对（1，8）的 JSD（1 是一条线，8 由环组成）。根据公式 1，层中的滤波器依据通道级注意力向量 v_L（见公式 2）进行激活；对于同一类别的样本，原型损失（公式 5）会使这些向量彼此靠近，而对于不同类别的样本，则会使它们彼此远离。我们在测试集上对这些 v_L 进行 t-SNE 可视化，如图——

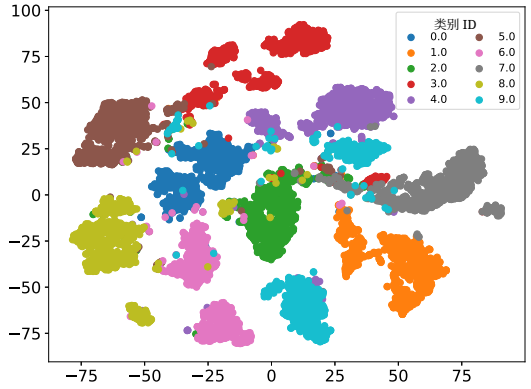


图 5：基于所有类别的通道级自注意力向量学习得到的类别原型在测试集上的 t-SNE 可视化。

图5所示，并观察到属于同一类别的样本聚集在一起，证实了原型损失的有效性。此外，视觉上相似类别的簇彼此靠近，这与前文观察到的 JSD 和类别级激活率一致。类别间的相似性也体现在数字 9 形成多个簇：一个簇显示它与数字 6（环形）和 1（线条）相似，另一个簇则显示它与 7（线条）和 4（线条 +环形）相似。最后，我们观察到一些样本散落在所属类别簇之外，并与其他簇重叠，这可能表明这些特定样本在视觉上更接近其他数字。不过请注意，这些相似样本和类别分布在不同任务中，这解释了为什么图3b中任务对之间激活模式的相似度低于图3a中类别对之间的相似度。

因此，*Dynamos* 能够学习模块化且专业化的单元，根据与先前学习示例的相似程度，实现激活值的输入自适应动态分离与重叠。我们还认为，形状相似的数字所产生的激活值重叠，表明模型学习到了通用特征。

6.2 试次间变异性

众所周知，大脑在不同试次中的响应存在变异性（Faisal 等人，2008年；Werner 和 Mountcastle，1963年）。对于相同的刺激，神经元响应的具体情况可能因试次而异，而大多数传统深度神经网络并不具备这种行为。在我们的方法中，我们通过在每个卷积层使用伯努利采样而非阈值化来做出保留/丢弃决策，从而模拟大脑的这一特性。在图 6 中，我们展示了深度神经网络最后一个卷积层在四次试验中对同一示例的响应变异性。我们只选取预测正确的响应，

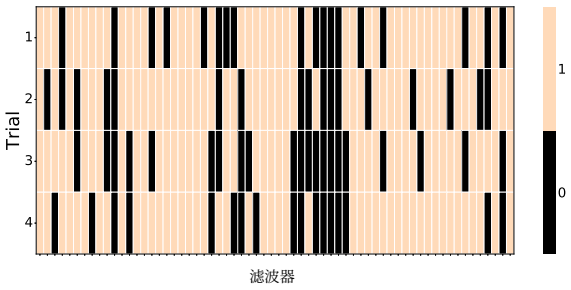


图 6：*Dynamos* 对相同输入的响应的试次间变异性。

这些预测都是正确的。可以看出，每次试验都会引发深度神经网络不同的响应。此外，尽管响应存在差异，但也有一些相似之处。有些滤波器会反复处于未使用状态，而另一些滤波器则在每次试验中都会被使用。这表明，动力模块 还可以模拟大脑中观察到的试次间变异性。

7 结论与未来工作

我们提出了*Dynamos*，一种通用持续学习方法，用于模拟大脑中观察到的对刺激作出的动态模块化且稀疏的响应。Dynamos 利用策略梯度，奖励根据输入自适应地移除卷积层的通道激活值，从而产生动态且稀疏的响应。为进一步促进模块化，同一类别样本对应的各卷积层通道级自注意力向量会被拉近，而不同类别样本对应的向量则会被推远；随后，这些向量用于采样相应通道的保留/丢弃决策。借助记忆缓冲区，我们在先前响应与当前响应之间强制施加多尺度一致性，以防止遗忘。在类别增量学习（CIL）和通用持续学习（GCL）协议下进行评估时，Dynamos 在多个数据集上的表现优于以往的基线方法。Dynamos 以自适应方式利用学习到的滤波器子集，使相似输入产生相似且重叠的响应，而不同输入则产生不同的响应。我们量化了类别间重叠的程度，并表明类别的语义相似性（例如 MNIST 中的数字 1 和 7）会体现为更高的表征重叠。我们还对通道级注意力向量进行了可视化，发现它们按类别聚类，且语义相似类别的簇彼此靠近或相互重叠。最后，我们还展示了该方法的能力

能够模仿大脑中观察到的试次间变异性：相同输入通过不同的“响应”（即激活值）获得相同输出。因此，我们认为这项工作是迈向动态模块化和通用持续学习的一步。

参考文献

Abati, D., Tomczak, J., Blankevoort, T., Calderara, S., Cucchiara, R., and Bejnordi, B. E.（2020）。用于任务感知持续学习的条件通道门控网络。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第3931–3940页。Abbasi, A., Nooralinejad, P., Braverman, V., Pirsiavash, H., and Kolouri, S.（2022）。神经激活值零空间中的稀疏性与异质性随机失活，用于持续学习。*arXiv*预印本 *arXiv:2203.06514*。Adel, T., Zhao, H., and Turner, R. E.（2020）。采用自适应权重的持续学习（CLAW）。载于第八届国际学习表征会议，*ICLR 2020*，埃塞俄比亚亚的斯亚贝巴，*2020*年*4*月*26*日至*30*日。OpenReview.net。Bengio, E., Bacon, P.-L., Pineau, J., and Precup, D.（2015）。神经网络中的条件计算可加速模型。*arXiv*预印本 *arXiv:1511.06297*。Bolukbasi, T., Wang, J., Dekel, O., and Saligrama, V.（2017年）。用于高效推理的自适应神经网络。载于国际机器学习会议，第527–536页。PMLR。Buzzega, P., Boschini, M., Porrello, A., Abati, D., and Calderara, S.（2020年）。通用持续学习的暗经验：一种强大而简洁的基线。神经信息处理系统进展，33：15920–15930。Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M.（2019年）。利用A-GEM实现高效终身学习。载于第七届国际学习表征会议，*ICLR 2019*，美国路易斯安那州新奥尔良，*2019*年*5*月*6*日至*9*日。OpenReview.net。Chen, H.-J., Cheng, A.-C., Juan, D.-C., Wei, W., and Sun, M.（2020年）。通过实例感知参数化缓解在线持续学习中的遗忘。神经信息处理系统进展，33：17466–17477。Collier, M., Kokiopoulou, E., Gesmundo, A., and Berent, J.（2020年）。采用协同训练的路由网络，用于持续学习。*arXiv*预印本 *arXiv:2009.04381*。Davari, M., Asadi, N., Mudur, S., Aljundi, R., and Belilovsky, E.（2022年）。探究监督式与无监督持续学习中的表征遗忘。*arXiv*预印本 *arXiv:2203.13381*。Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T.（2021年）。持续学习综述：应对分类任务中的遗忘。*IEEE*模式分析与机器智能汇刊。

Faisal, A. A., Selen, L. P., and Wolpert, D. M.（2008年）。神经系统中的噪声。《自然综述：神经科学》, 9(4):292–303。Farquhar, S. and Gal, Y.（2018年）。迈向稳健的持续学习评估。持续学习：一种强化学习方法研讨会，国际机器学习会议。French, R. M.（1999年）。联结主义网络中的灾难性遗忘。《认知科学趋势》, 3(4):128–135。Goyal, A. and Bengio, Y.（2020年）。深度学习高阶认知的归纳偏置。*arXiv*预印本 *arXiv:2011.15091*。Graham, D. J. and Field, D. J.（2006年）。新皮层中的稀疏编码。《神经系统的演化》, 3:181–187。Greenwald, N. F., Miller, G., Moen, E., Kong, A., Kagel, A., Dougherty, T., Fullaway, C. C., McIntosh, B. J., Leow, K. X., Schwartz, M. S., et al.（2021年）。利用大规模数据标注和深度学习，以人类水平的性能对组织图像进行全细胞分割。《自然·生物技术》, pages 1–11。Hadsell, R., Rao, D., Rusu, A. A., and Pascanu, R.（2020年）。拥抱变化：深度神经网络中的持续学习。《认知科学趋势》, 24(12):1028–1040。He, K., Zhang, X., Ren, S., and Sun, J.（2016年）。用于图像识别的深度残差学习。载于《*IEEE*计算机视觉与模式识别会议论文集》, pages 770–778。Hu, T., Chen, T., Wang, H., and Wang, Z.（2020年）。三重优势：通过启用输入自适应推理，同时提升准确率、鲁棒性和效率。载于国际学习表征会议。Isele, D. and Cosgun, A.（2018年）。用于终身学习的选择性经验回放。载于人工智能促进协会人工智能会议论文集, volume 32。Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.（2017年）。克服神经网络中的灾难性遗忘。《美国国家科学院院刊》, 114(13):3521–3526。LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P.（1998年）。应用于文档识别的基于梯度的学习。《*IEEE*会刊》, 86(11):2278–2324。Li, X., Liu, Z., Luo, P., Change Loy, C., and Tang, X.（2017年）。并非所有像素都相同：通过深层级联实现难度感知语义分割。载于《*IEEE*计算机视觉与模式识别会议论文集》, pages 3193–3202。McCloskey, M. and Cohen, N. J.（1989年）。联结主义网络中的灾难性干扰：序列学习问题。载于《学习与动机心理学》, volume 24, pages 109–165。爱思唯尔。Meunier, D., Lambiotte, R., and Bullmore, E. T.（2010年）。脑网络的模块化与层级模块化组织。《神经科学前沿》, 4:200。Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y.（2011年）。利用无监督特征学习识别自然图像中的数字。Nose, Y., Kojima, A., Kawabata, H., and Hironaka, T.（2019年）。一种利用卷积神经网络从实时图像中进行转向控制在线学习的车道保持系统研究

载于*2019*年第*34*届国际电路/系统、计算机与通信技术会议（*ITC-CSCC*）, pages 1–4。IEEE。Ostapenko, O., Rodriguez, P., Caccia, M., and Charlin, L.（2021年）。通过局部模块组合实现持续学习。神经信息处理系统进展, 34:30298–30312。Rajasegaran, J., Hayat, M., Khan, S. H., Khan, F. S., and Shao, L.（2019年）。面向持续学习的随机路径选择。神经信息处理系统进展, 32。Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., and Tesauro, G.（2019年）。通过最大化迁移、最小化干扰来实现无遗忘的学习。载于第七届国际学习表征会议，*ICLR 2019*，美国路易斯安那州新奥尔良，*2019*年*5*月*6*日至*9*日。OpenReview.net。Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R.（2018年）。Progress & compress：一种可扩展的持续学习框架。载于国际机器学习会议, pages 4528–4537。PMLR。Sutton, R. S. and Barto, A. G.（2018年）。强化学习：导论。麻省理工学院出版社。Taigman, Y., Yang, M., Ranzato, M., and Wolf, L.（2014年）。DeepFace：缩小人脸验证与人类水平性能之间的差距。载于《*IEEE*计算机视觉与模式识别会议论文集》, pages 1701–1708。Thrun, S. and Mitchell, T. M.（1995年）。终身机器人学习。《机器人与自主系统》, 15(1-2):25–46。Veit, A. and Belongie, S.（2018年）。具有自适应推理图的卷积网络。载于《欧洲计算机视觉会议（*ECCV*）论文集》, pages 3–18。Veniat, T., Denoyer, L., and Ranzato, M.（2021年）。利用模块化网络和任务驱动先验实现高效持续学习。载于国际学习表征会议。Vitter, J. S.（1985年）。蓄水池随机采样。*ACM*数学软件汇刊（*TOMS*）, 11(1):37–57。Wang, X., Yu, F., Dou, Z.-Y., Darrell, T., and Gonzalez, J. E.（2018年）。SkipNet：在卷积网络中学习动态路由。载于《欧洲计算机视觉会议（*ECCV*）论文集》, pages 409–424。Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., and Pfister, T.（2022年）。学习提示以实现持续学习。载于《*IEEE/CVF*计算机视觉与模式识别会议论文集》, pages 139–149。Werner, G. and Mountcastle, V. B.（1963年）。感觉系统中中枢神经活动的变异性及其对感觉事件中枢反映的影响。《神经生理学杂志》, 26(6):958–977。Williams, R. J.（1992年）。简单的统计梯度跟踪用于联结主义强化学习的算法。机器学习, 8(3):229–256。Wu, Z., Nagarajan, T., Kumar, A., Rennie, S., Davis, L. S., Grauman, K., 和 Feris, R.（2018年）。Blockdrop：残差网络中的动态推理路径。载于*IEEE*计算机视觉与模式识别会议论文集，第8817–8826页。Yosinski, J., Clune, J., Bengio, Y., 和 Lipson, H.（2014年）。深度神经网络中的特征可迁移性有多强？神经信息处理系统进展，27。Zenke, F., Poole, B., 和 Ganguli, S.（2017年）。通过突触智能实现持续学习。载于国际机器学习会议，第3987–3995页。PMLR。

附录

表 2：图2中 CCGN 和 Dynamos 对应的类别增量学习准确率。

缓冲区 Size	模型	Seq-SVHN	Seq-MNIST
500	CCGN	67.45	94.01
	Dynamos	84.815	97.19
1000	CCGN	73.99	95.94
	Dynamos	87.38	97.51
2000	CCGN	81.02	95.94
	Dynamos	88.54	97.57

表3：输入特征形状为$B \times C_{in} \times H \times W$、输出特征形状为$B \times C_{out} \times 1 \times 1$的 Agent架构，其中$B$ 为批次大小，C_{in} 为输入中的通道数，C_{out} 为输出所需的通道数，该数量与对应卷积层所需的保留/丢弃操作数相同。

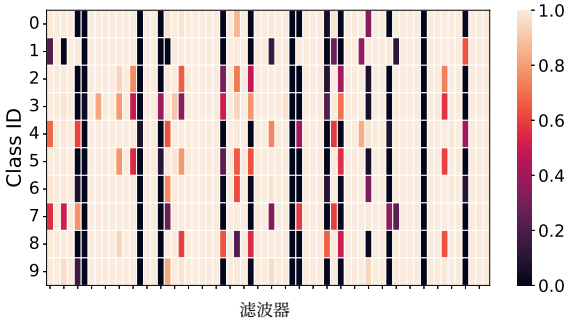
操作	**输入大小**	**输出大小**
逐点卷积	$B \times C_{in} \times h \times w$	$B \times C_{out} \times h \times w$
平均池化	$B \times C_{out} \times h \times w$	$B \times C_{out} \times 1 \times 1$
重塑	$B \times C_{out} \times 1 \times 1$	$B \times C_{out}$
线性层	$B \times C_{out}$	$B \times C_{out}/16$
ReLU	$B \times C_{out}/16$	$B \times C_{out}/16$
线性层	$B \times C_{out}/16$	$B \times C_{out}$
重塑	$B \times C_{out}$	$B \times C_{out} \times 1 \times 1$
Sigmoid 函数。	$B \times C_{out} \times 1 \times 1$	$B \times C_{out} \times 1 \times 1$

表4：实验中使用的架构。对于不包含动态组合性的基线实验，我们不使用“Agent”分支。Conv(k, n, s, p)表示卷积层，其卷积核大小为k、滤波器数量为n、步幅为s、填充为p。BN表示批归一化。Linear(M, N)表示线性层，其输入维度为M，输出维度为N。Agent(C_{in}, C_{out}, τ)表示Agent子网络，其输入通道为C_{in}，输出通道为C_{out}，概率层中Sigmoid函数的温度为 τ（见图1、第4节）。智能体输出的动作与卷积层输出进行逐元素乘法，若存在批归一化，则在其应用之后进行；若存在ReLU激活函数，则在其应用之前进行。有关Agent架构的完整说明，请参见表3。类别数 指待预测的类别数量。

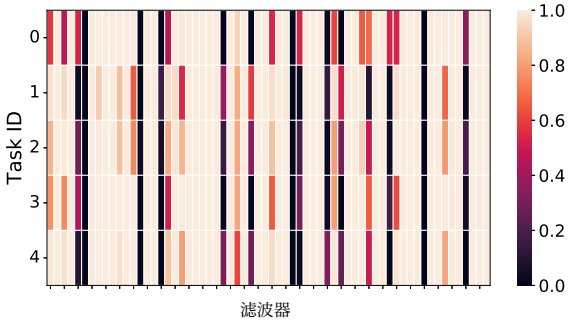
组件	**主分支**	**残差分支**	**智能体分支**
Conv1	卷积(3, 32, 1, 1)，批归一化，*ReLU*	—	—
Block1	卷积(3, 32, 1, 1)，批归一化，*ReLU* 卷积(3, 32, 1, 1)，批归一化，*ReLU*	恒等映射 恒等映射	智能体(64，64，0。15) 智能体(64，64，0。15)
模块2	卷积(3, 32, 1, 1), 批归一化, *ReLU* 卷积(3, 32, 1, 1), 批归一化, *ReLU*	卷积(1, 64,2, 0), 批归一化 卷积(1, 64,2, 0), 批归一化	智能体(64, 64, 0.15) 智能体(64, 64, 0.15)
分类器	线性层(64, 类别数)	—	—

表 5：Dynamos 中所有数据集的超参数。

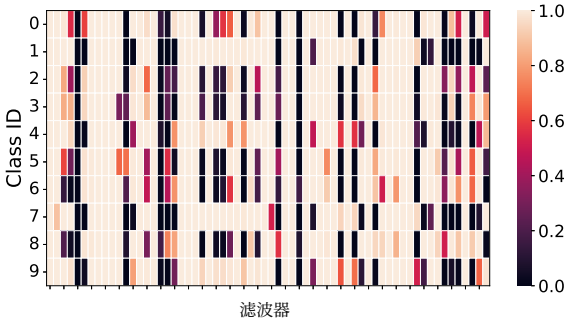
数据集	缓冲区 Size	lr	训练轮数	预热步数 轮次/迭代 次数	批次 Size	记忆 批次 Size	α	β	α_P	λ	γ	w_P	kr
Seq-MNIST	500	0.07	1	10 次迭代	10	10	0.2	2.0	0.2	500	0.5	0.3	0.7
	1000	0.07	1	10 次迭代	10	10	0.1	2.5	0.2	200	0.7	0.5	0.7
	2000	0.07	1	10 次迭代	10	10	0.5	3.0	0.2	200	0.5	0.5	0.7
SVHN	500	0.07	70	10 个周期	16	16	2.0	3.0	1.0	500	1.0	0.5	0.7
	1000	0.07	70	10 个周期	16	16	2.5	2.0	0.2	500	1.0	0.5	0.7
	2000	0.07	70	10 个周期	16	16	2.5	2.0	0.2	500	1.0	0.5	0.7
MNIST-360	100	0.07	1	10 它	16	16	0.2	1.0	0.1	200	0.5	0.5	0.7
	200	0.07	1	10 它	16	16	0.2	1.5	0.1	200	1.0	0.5	0.7
	500	0.07	1	10 它	16	16	0.1	1.5	0.1	200	0.3	0.3	0.7



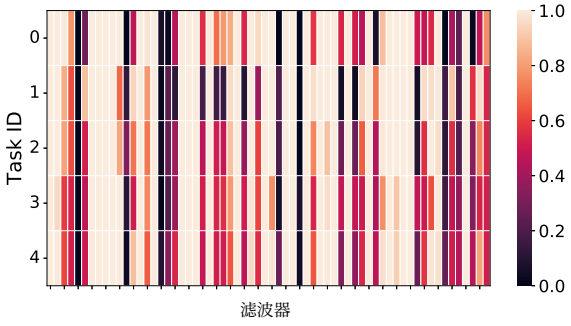
（a）第1层按类别划分的激活



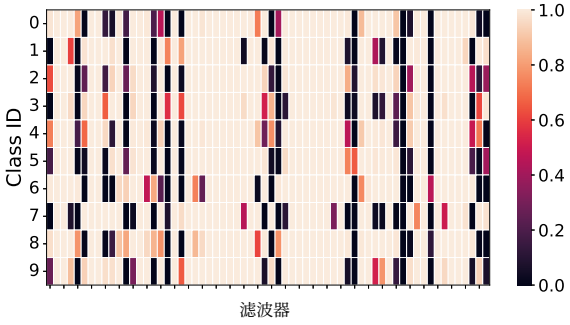
（b）第1层按任务划分的激活



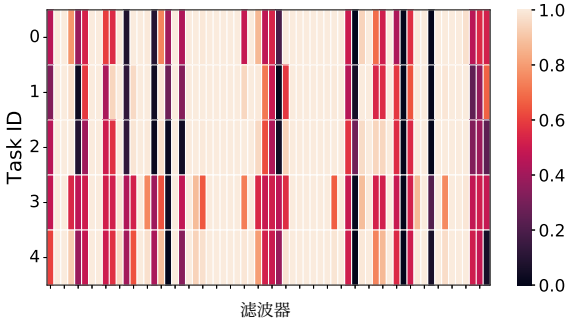
（c）第2层按类别划分的激活



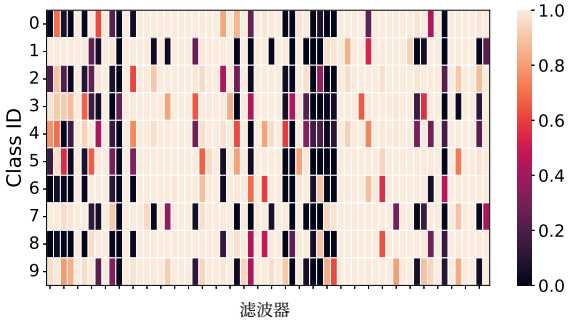
（d）第2层按任务划分的激活



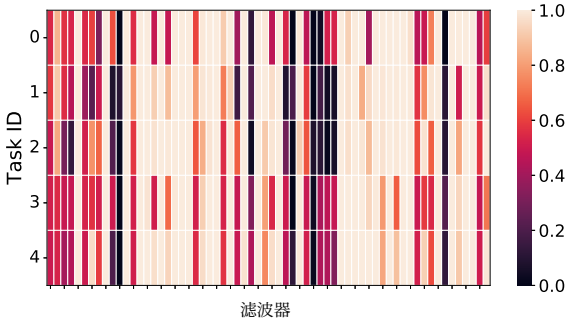
(e) 第3层按类别划分的激活值



(f) 第3层按任务划分的激活值



(g) 第4层按类别划分的激活值



(h) 第4层按任务划分的激活值

图7：模块2中每个卷积层的各个滤波器针对 MNIST 任务和类别的滤波器激活率。仍然可以观察到任务和类别之间存在重叠激活，这表明它们具有相似性。例如，1 和 7 的响应仍然非常相似。