Entropy

文章持续学习的序贯贝叶斯推断

Samue l Kessler [1,*]、Adam Cobb [3], Tim G. J. Rudner 2、Stefan Zohren [1] 和 Stephen J. Roberts [1]

[1] 牛津大学工程科学系，牛津 OX2 6ED，英国；skessler@robots.ox.ac.uk (S.K.)；zohren@robots.ox.ac.uk (S.Z.)；sjrob@robots.ox.ac.uk (S.J.R.)[2] 牛津大学计算机科学系，牛津 OX1 3QG，英国；tim.rudner@cs.ox.ac.uk (T.G.J.R.)[3] SRI International，美国弗吉尼亚州阿灵顿 22209；adam.cobb@sri.com* 通讯作者：skessler@robots.ox.ac.uk

arXiv:2301.01828v2 [cs.LG] 9 Jul 2023

摘要： 序贯贝叶斯推断可用于持续学习，以防止对过去任务的灾难性遗忘，并在学习新任务时提供信息性先验。我们重新审视序贯贝叶斯推断，并评估在贝叶斯神经网络中，将先前任务的后验分布用作新任务的先验能否防止灾难性遗忘。我们的首项贡献是使用哈密顿蒙特卡洛进行序贯贝叶斯推断。我们通过在哈密顿蒙特卡洛样本上拟合密度估计器来近似后验分布，并将该后验分布作为新任务的先验。我们发现，这种方法无法防止灾难性遗忘，凸显了在神经网络中进行序贯贝叶斯推断的难度。随后，我们研究了序贯贝叶斯推断和持续学习的简单解析示例，并指出模型设定错误这一问题；即使进行精确推断，它也可能导致持续学习性能欠佳。此外，我们还讨论了任务数据不平衡如何导致遗忘。基于这些局限性，我们认为，相较于依赖对贝叶斯神经网络权重进行序贯贝叶斯推断，我们需要对持续学习生成过程建立概率模型。最后，我们提出了一种名为原型贝叶斯持续学习的简单基线方法；在类别增量持续学习计算机视觉基准测试中，该方法的表现可与最佳的贝叶斯持续学习方法相媲美。

关键词： 持续学习；终身学习；序贯贝叶斯推断；贝叶斯深度学习；贝叶斯神经网络

check for updates

C引文： Kessler, S.; Cobb, A.; Rudner, T.G.J.; Zohren, S.; Roberts, S.J. 关于S序贯贝叶斯推断C持续学习。 *Entropy***2023**, *1*, 0. https://doi.org/

A学术编辑：Irad E. Ben-Gal a和 Amichai Painsky

R 收稿：2023年5月1日 修订：2023年5月24日 接收：2023年5月28日 发表：

版权： © 2023 作者所有。被许可方MDPI，瑞士巴塞尔。本文为开放获取文章，依据知识共享署名（CC BY）许可协议（https://creativecommons.org/licenses/by/ 4.0/）的条款和条件分发。

1. 引言

持续学习（CL）的目标是找到一种预测器，使其能够学习解决一系列新任务，同时不丧失解决先前已学习任务的能力。神经网络（NN）持续学习的一项关键挑战在于，在基于梯度学习新任务的过程中，先前已学习任务的模型参数会被“覆盖”，从而导致先前习得的能力发生灾难性遗忘 [1,2]。一种持续学习方法依赖于递归应用贝叶斯定理，并将贝叶斯神经网络（BNN）中的权重后验分布用作新任务的先验 [3]。然而，获取神经网络权重的完整后验分布在计算上代价高昂，因此我们通常需要采用拉普拉斯方法 [4] 或变分推断 [5,6] 等近似方法来获得神经网络权重后验分布。

执行贝叶斯持续学习时，采用近似的贝叶斯神经网络后验分布进行序贯贝叶斯推断，而非使用真实后验分布 [7–12]。如果考察在贝叶斯神经网络权重后验分布上进行变分近似的序贯贝叶斯推断，其效果与单纯使用随机梯度下降（SGD）学习新任务相比，几乎没有提升。我们将在第2.2节进一步阐述这一观点。因此，如果我们能够获得真实的贝叶斯神经网络权重后验分布，这是否足以通过序贯贝叶斯推断避免遗忘?

本文的贡献在于重新审视贝叶斯持续学习。（1）在实验方面，我们使用真实的贝叶斯神经网络权重后验分布进行序贯贝叶斯推断。具体而言，我们采用贝叶斯推断方法的黄金标准——哈密顿蒙特卡洛（HMC）[13]。我们对 HMC 样本进行密度估计，并在 HMC 采样过程中，将得到的近似后验密度用作下一个任务的先验。令人惊讶的是，尽管使用了来自真实后验分布的样本，我们的 HMC 持续学习方法与近似推断方法（变分持续学习，Nguyen 等人 [9]）相比并无明显优势。（2）因此，我们考虑了一个简单的解析示例，并指出，即使采用设定错误的模型进行精确推断，仍然可能导致遗忘。（3）我们从数学上证明，在某些假设下，任务数据不平衡会导致贝叶斯神经网络发生遗忘。（4）我们提出了一种用于持续学习的新概率模型，并表明，通过显式建模数据的生成过程，我们能够取得良好的性能，无需依赖对神经网络权重进行递归贝叶斯推断来避免遗忘。我们提出的模型——典型贝叶斯持续学习 (ProtoCL)——在概念上简单、可扩展，并且在类别增量学习设置下，其性能可与最先进的贝叶斯持续学习方法相媲美。

2. 背景

2.1. 持续学习问题

持续学习（CL）是一种学习设定，在这种设定下，模型必须依次学习一组任务并进行预测，同时保持在所有先前学过的任务上的性能。在持续学习中，模型依次接触 T 个任务，记为 $\mathcal{T}_t$ 对于 $\mathrm{t} = 1, \ldots, \mathrm{T}$。每个任务 $\mathcal{T}_t$ 都由一个数据集 $\mathcal{D}_t = \{(\mathrm{x}_i, \mathrm{y}_i)\}^{N_t}\mathrm{i}{=}1$ 构成，模型需要通过该数据集学习进行预测。更一般地，任务由不同的条件数据分布和边缘数据分布元组表示，即 $\{\mathrm{p}_t(\mathrm{y}|\mathrm{x}), \mathrm{p}_t(\mathrm{x})\}$。完成任务$\mathcal{T}_t$后，模型将无法再访问训练数据集，但其在所有任务 $\mathcal{T}_i$ 对于 $\mathrm{i} \leq \mathrm{t}$ 上的性能仍会持续接受评估。我们将预测器分解为 $\mathrm{g} = \mathrm{h} \circ \mathrm{f}$，使得 $\hat{}\mathrm{y} = \mathrm{g}(\mathrm{x})$。我们将 f 定义为一个嵌入函数，映射为 $\mathrm{f}: \mathcal{X} \to \mathcal{Z}$；将 h 定义为一个映射至输出的分类头，映射为 $\mathrm{h}: \mathcal{Z} \to \mathcal{Y}$。一些持续学习方法会为每个任务使用一个独立的分类头 $\{\mathrm{h}_i\}^T\mathrm{i}{=}1$；这类方法称为多头方法，而使用单个分类头的方法称为单头方法。

2.2. 贝叶斯持续学习

我们考虑这样一种设定：任务数据在时间步 $t = 1, 2, \ldots, T$ 依次到达。在第一个时间步 $t = 1$，也就是对于任务 $\mathcal{T}_1$，模型接收第一个数据集 $\mathcal{D}_1$，并学习所有 $(\boldsymbol{x}_i, y_i) \in \mathcal{D}_1$ 的条件分布 $p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta})$（其中 i 表示 $\mathcal{D}_1$ 中一个数据点的索引）。我们将参数 $\boldsymbol{\theta}$ 视为具有针对 $\mathcal{T}_1$ 的先验分布 $p(\boldsymbol{\theta})$。因此，对于测试点 $\boldsymbol{x}_1^* \in \mathcal{D}_1$，后验预测分布为：

$$p(y_1^*|\boldsymbol{x}_1^*, \mathcal{D}_1) = \int p(y_1^*|\boldsymbol{x}_1^*, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}_1) d\boldsymbol{\theta}. \tag{1}$$

我们注意到，计算该后验预测分布需要 $p(\boldsymbol{\theta}|\mathcal{D}_1)$。对于 $t = 2$，持续学习模型需要拟合所有 $(\boldsymbol{x}_i, y_i) \in \mathcal{D}_1 \cup \mathcal{D}_2$ 的分布 $p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta})$。新测试点 $\boldsymbol{x}_2^* \in \mathcal{D}_1 \cup \mathcal{D}_2$ 的后验预测分布为：

$$p(y_2^*|\boldsymbol{x}_2^*, \mathcal{D}_1, \mathcal{D}_2) = \int p(y_2^*|\boldsymbol{x}_2^*, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}_1, \mathcal{D}_2) d\boldsymbol{\theta}. \tag{2}$$

因此，必须更新后验分布，以反映这一新的条件分布。我们可以反复应用贝叶斯法则，计算后验分布 $p(\boldsymbol{\theta}|\mathcal{D}_1, \ldots, \mathcal{D}_T)$，如下：

$$p(\boldsymbol{\theta}|\mathcal{D}_1, \ldots, \mathcal{D}_{T-1}, \mathcal{D}_T) = \frac{p(\mathcal{D}_T|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}_1, \ldots, \mathcal{D}_{T-1})}{p(\mathcal{D}_T|\mathcal{D}_1, \ldots, \mathcal{D}_{T-1})}. \tag{3}$$

在持续学习设置中，我们无法再访问以前的训练数据集；不过，通过反复应用贝叶斯法则（公式 3），我们可以逐步将过去任务的信息纳入参数 $\boldsymbol{\theta}$。在 $t = 1$ 时，我们可以访问 $\mathcal{D}_1$，参数的后验分布为：

$$\log p(\boldsymbol{\theta}|\mathcal{D}_1) = \log p(\mathcal{D}_1|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log p(\mathcal{D}_1). \tag{4}$$

在 $t = 2$ 时，我们需要 $p(\boldsymbol{\theta}|\mathcal{D}_1, \mathcal{D}_2)$ 来计算公式（2）中的后验预测分布。然而，我们已无法访问 $\mathcal{D}_1$。根据贝叶斯法则，后验分布可写为：

$$\log p(\boldsymbol{\theta}|\mathcal{D}_1, \mathcal{D}_2) = \log p(\mathcal{D}_2|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}|\mathcal{D}_1) - \log p(\mathcal{D}_2|\mathcal{D}_1), \tag{5}$$

其中我们利用了 $\mathcal{D}_2$ 与 $\mathcal{D}_1$ 在给定 $\boldsymbol{\theta}$ 时的条件独立性。需要注意的是，似然 $p(\mathcal{D}_2|\boldsymbol{\theta})$ 仅取决于当前任务数据集 $\mathcal{D}_2$，而先验$p(\boldsymbol{\theta}|\mathcal{D}_1)$ 则编码了来自先前任务的参数知识。因此，我们可以将根据 t 时刻评估得到的后验分布用作在 $t + 1$时学习新任务的先验。根据公式（3），我们要求参数为 $\boldsymbol{\theta}$ 的模型是 $\mathcal{D}_1$ 的充分统计量，即$p(\mathcal{D}_2|\boldsymbol{\theta}, \mathcal{D}_1) = p(\mathcal{D}_2|\boldsymbol{\theta})$，从而使似然在给定 $\boldsymbol{\theta}$ 时与 $\mathcal{D}_1$ 条件独立。这一观察促使我们采用高容量预测器，例如贝叶斯神经网络，它们足够灵活，能够从 $\mathcal{D}_1$ 中学习。

持续学习示例：Split-MNIST

对于 MNIST 数据集，[14], 我们知道，如果训练一个贝叶斯神经网络，通过推断后验分布 $p(\boldsymbol{\theta}|\mathcal{D})$ 并对后验分布进行积分，就能推断测试点上的后验预测分布，见公式（1）。因此，如果将 MNIST 数据集划分为 5 个二分类任务，我们就应该能够使用公式（$p(\boldsymbol{\theta}|\mathcal{D})$ $= p(\boldsymbol{\theta}|\mathcal{D}_1 \ldots, \mathcal{D}_5)$ 3）递归地恢复多任务后验分布。这一问题称为 Split-MNIST，[15], 其中第一个任务是对数字 $\{0, 1\}$进行分类，第二个任务是对数字 $\{2, 3\}$进行分类，依此类推。

我们可以定义三种不同的持续学习设置 [16–18]。如果允许持续学习智能体在预测时使用任务标识符，τ 这种情形称为任务增量。例如，标识符 τ 可用于选择不同的输出头，见第 2.1节。这种情形与公式（3）中概述的序贯贝叶斯推断不兼容，因为进行预测时不需要任务标识符。领域增量学习是另一种情形，在评估期间无法访问 τ，并要求持续学习智能体针对每个任务在相同的输出空间中进行分类；例如，对于 Split-MNIST，所有任务的输出空间均为 $\{0, 1\}$，因此相当于在偶数和奇数之间进行分类。领域增量学习与采用伯努利似然的序贯贝叶斯推断兼容。第三种情形是类别增量学习，同样无法访问 τ，但智能体需要将每个示例分到对应的类别。例如，对于 Split-MNIST，每个任务的输出空间为 $\{0, \ldots, 9\}$。类别增量学习与采用分类似然的序贯贝叶斯推断兼容。

2.3. 变分持续学习

变分持续学习（VCL；Nguyen 等人 [9]）将公式（3）中的贝叶斯推断问题，简化为对随机神经网络权重分布 $\boldsymbol{\theta}$ 进行一系列近似贝叶斯更新。为此，VCL 将先前任务的变分后验用作新任务的先验。这样，学习解决第一个任务就意味着要找到一个变分分布 $q_1(\boldsymbol{\theta}|\mathcal{D}_1)$，使相应的变分目标最大化。对于后续任务，先验选为 $q_1(\boldsymbol{\theta}|\mathcal{D}_1)$，目标则变为学习一个变分分布 $q_2(\boldsymbol{\theta}|\mathcal{D}_2)$，使相应的变分目标最大化

在该先验下。用 $q_t(\boldsymbol{\theta}|\mathcal{D}_{1:t})$ 表示根据多个数据集递归推断出的后验分布，我们可以将第 t 个任务的变分持续学习目标表示为：

$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_t) = \mathbb{D}_{\mathrm{KL}}[q_t(\boldsymbol{\theta})||q_{t-1}(\boldsymbol{\theta}|\mathcal{D}_{1:t-1})] - \mathbb{E}_{q_t}[\log p(\mathcal{D}_t|\boldsymbol{\theta})]. \tag{6}$$

将 VCL 应用于 Split-MNIST 问题时（图 1），我们可以看到，在记住过去的任务方面，单头 VCL 的表现仅略优于 SGD。多头 VCL 的表现更好，尽管这并非序贯贝叶斯推断所要求的，见公式（3）。因此，如果我们能够利用公式（3）递归地构建近似后验分布，为什么单头 VCL 的表现没有优于 SGD？我们推测，这可能是因为使用了后验分布的变分近似，因此我们实际上并未严格执行第 2.2 节所述的贝叶斯持续学习过程。我们将在下一节检验这一假设：通过传播真实的贝叶斯神经网络后验分布，验证能否递归地获得真实的多任务后验分布，从而改进单头 VCL 并防止灾难性遗忘。

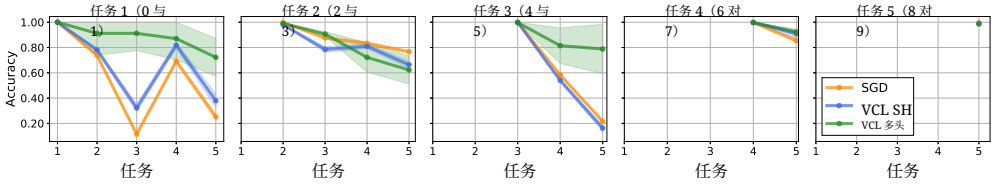

图 1. 采用双层贝叶斯神经网络时，各种持续学习方法在 Split-MNIST 上的准确率；所有准确率均为 10 次运行（使用不同随机种子）的均值和标准差。我们比较了使用 SGD 训练的神经网络（单头）与 VCL。我们考察了单头（ SH ）和多头（ MH ） VCL 变体。

3. 使用哈密顿蒙特卡洛的贝叶斯持续学习

为了对贝叶斯神经网络权重进行推断，我们使用哈密顿蒙特卡洛算法 [13]。随后，我们利用这些样本学习一个密度估计器，并将其用作新任务的先验（我们曾考虑序贯蒙特卡洛，但它无法扩展到我们所研究神经网络所需的维度 [19]。另一方面，哈密顿蒙特卡洛最近已成功扩展到规模相对较小、与本文所研究规模相当的贝叶斯神经网络 [20] 和残差网络模型，但计算成本很高 [21]）。哈密顿蒙特卡洛被视为近似推断的黄金标准，并且可以保证渐近地产生来自真实后验分布的样本（在 NeurIPS 2021 贝叶斯深度学习竞赛中（https://izmailovpavel.github.io/neurips_bdl_competition），目标是找到一种近似推断方法，使其尽可能“接近”哈密顿蒙特卡洛得到的后验样本）。我们使用哈密顿蒙特卡洛得到的 $\boldsymbol{\theta}$后验样本，然后在这些样本上拟合密度估计器，将其用作新任务的先验。这样，我们便可以使用关于 $\boldsymbol{\theta}$的多模态后验分布，而不是像变分持续学习那样使用对角高斯变分后验。更具体地说，为了传递后验分布$p(\boldsymbol{\theta}|\mathcal{D}_1)$，我们使用一个密度估计器，定义为 $\hat{p}(\boldsymbol{\theta}|\mathcal{D}_1)$，在哈密顿蒙特卡洛样本上拟合后验概率密度。对于下一个任务 $\mathcal{T}_2$，我们可以将 $\hat{p}(\boldsymbol{\theta}|\mathcal{D}_1)$ 用作新哈密顿蒙特卡洛采样链的先验，依此类推（见图 2）。用于哈密顿蒙特卡洛采样的密度估计器先验需要满足两个关键条件。第一，它们必须是概率密度函数；第二，它们必须关于输入样本可微。

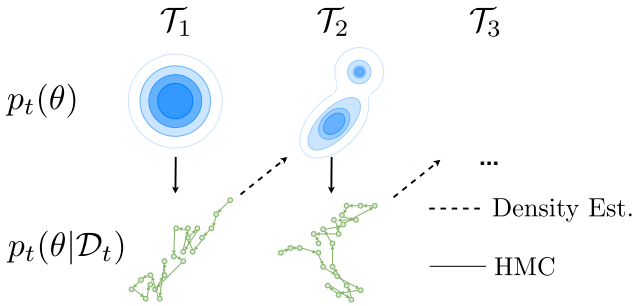

图2。 后验传播过程示意图；先验分布以蓝色表示，位于上排，后验样本位于下排。这是一个两步过程：首先，针对 $\mathcal{T}_1$，使用各向同性高斯先验进行哈密顿蒙特卡洛采样；然后，对后验分布的 HMC 样本进行密度估计，得到 $\hat{p}_1(\theta|\mathcal{D}_1)$。随后，可将此后验分布用作新任务 $\mathcal{T}_2$ 的先验，依此类推。

我们使用一个玩具数据集（图3），其中包含两个类别，输入为 $\boldsymbol{x} \in \mathbb{R}^2$ [22]。每个任务都是一个二分类问题，决策边界会随着每个新任务从左向右延伸。我们训练了一个两层贝叶斯神经网络，隐藏状态大小为 10。我们使用高斯混合模型作为密度估计器，以利用哈密顿蒙特卡洛样本近似后验分布。我们也尝试了灵活性更高的归一化流 [23]；但它们无法稳定地用于哈密顿蒙特卡洛采样（RealNVP 对随机种子的选择非常敏感，学习到的分布所生成的样本无法准确预测当前任务，并且在哈密顿蒙特卡洛采样中用作先验时会导致数值不稳定性）。据我们所知，我们是首个将灵活先验引入哈密顿蒙特卡洛等采样方法的研究者。

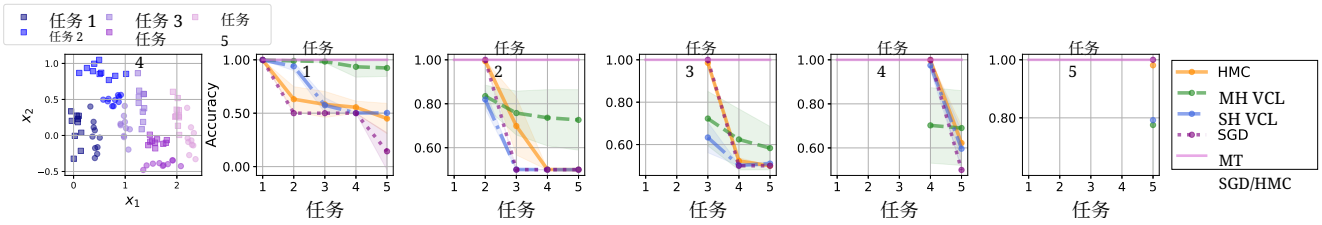

图3。 左侧是玩具数据集，包含 5 个不同的二分类任务，涉及对圆形和正方形进行分类 [22]。此外，还展示了基于 10 个随机种子的持续学习二分类测试准确率。粉色实线表示使用 SGD/HMC、并采用与持续学习实验相同模型的多任务（MT）基线准确率。

在同一多任务数据集上使用哈密顿蒙特卡洛训练贝叶斯神经网络，测试准确率可达 1.0。因此，根据公式（3），最终后验分布适用于持续学习；我们应能通过使用哈密顿蒙特卡洛的递归推断方法，递归地得到多任务后验分布。图3 的结果表明，使用近似多模态后验分布的哈密顿蒙特卡洛无法防止遗忘，且效果不如多头变分持续学习。事实上，多头变分持续学习明显优于哈密顿蒙特卡洛，这表明知识保持并非源于后验分布的传播，而是依赖于任务专用头。对于 $\mathcal{T}_2$，我们使用 $\hat{p}(\boldsymbol{\theta}|\mathcal{D}_1)$ 而非 $p(\boldsymbol{\theta}|\mathcal{D}_1)$ 作为先验，这会使后续所有任务的 HMC 采样产生偏差。下一段将详细说明为确保 HMC 链已收敛、从而能够从真实后验分布中采样而采取的措施。此外，我们还评估了高斯混合模型密度估计器相对于 HMC 样本的保真度。我们还在另一个由五个二分类任务组成的玩具数据集上重复了这些实验，并观察到类似结果，见附录 A。

对于 HMC，我们通过评估链收敛情况和有效样本量（图 A5）来确保采样来自后验分布。有效样本量衡量链中的自相关程度。我们的贝叶斯神经网络的 HMC 链的有效样本量与相关文献中的结果相似 [20]。此外，我们确保高斯混合模型近似后验分布具有多模态特征，且与变分持续学习相比，后验分布更为复杂，并且

对于当前任务，高斯混合模型样本与 HMC 样本产生的结果相当（图 A4）。详情请参阅附录 B。

本节讨论的二维基准测试来自先前的研究，属于领域增量持续学习问题。领域增量设置也比类别增量设置简单 [18]，因此是尝试执行精确序贯贝叶斯推断的良好起点。尽管如此，即使使用被视为贝叶斯深度学习黄金标准的哈密顿蒙特卡洛，我们仍无法在贝叶斯神经网络中执行序贯贝叶斯推断。哈密顿蒙特卡洛与高斯混合模型密度估计能够生成更丰富、更准确且多模态的后验分布。尽管如此，我们仍无法逐步构建多任务后验分布，也无法取得比单头变分持续学习等各向同性高斯后验分布显著更好的结果。这种方法的薄弱之处在于密度估计：高斯混合模型会移除贝叶斯神经网络权重空间后验分布中某些区域的概率质量，而这些区域对新任务至关重要。这充分说明了对贝叶斯神经网络权重后验分布进行建模是多么困难。下一节中，我们将研究序贯贝叶斯推断的另一个解析示例，并探讨模型设定错误和任务数据不平衡如何导致贝叶斯持续学习中的遗忘。

4. 贝叶斯持续学习与模型设定错误

现在，我们考虑一个简单的解析示例，其中可以利用共轭性，以闭式形式执行序贯贝叶斯推断方程（3）。我们考虑这样一个简单的设定：数据点依次在线到达。

观测 $\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_t$ 依次在线到达，且每个观测都由隐变量 $\theta_1, \theta_2, \ldots, \theta_t \sim \mathrm{p}$ 生成，其中 p 是概率密度函数。在时间 t，我们希望使用序贯贝叶斯推断来推断滤波分布 $\mathrm{p}(\theta_t | \mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_t)$ [24]，这与卡尔曼滤波器 [25] 类似。似然为 $\mathrm{p}(\mathrm{y}_t | \theta_t) = \mathcal{N}(\mathrm{y}_t; \mathrm{f}(\cdot; \theta_t), \sigma^2)$，其均值由线性模型参数化：$\mathrm{y}_t = \mathrm{f}(\cdot; \theta_t) + \epsilon$，其中 $\epsilon \sim \mathcal{N}(0, \sigma^2)$ 且 $\mathrm{f}(\cdot; \theta_t) = \theta_t$。我们考虑关于均值参数 θ 的高斯先验，使得 $\mathrm{p}(\theta_0) = \mathcal{N}(\theta_0; 0, \sigma_0^2)$。由于均值的共轭先验也是高斯分布，因此先验和后验分布分别为 $\mathcal{N}(\theta_{t-1}; \hat{\theta}_{t-1}, \hat{\sigma}_{t-1}^2)$ 和 $\mathcal{N}(\theta_t; \hat{\theta}_t, \hat{\sigma}_t^2)$。通过序贯贝叶斯推断，我们可以得到后验参数的闭式更新方程：

$$\hat{\theta}_t = \hat{\sigma}_t^2 \left(\frac{y_t}{\sigma^2} + \frac{\hat{\theta}_{t-1}}{\hat{\sigma}_{t-1}^2} \right) = \hat{\sigma}_t^2 \left(\sum_{i=1}^{t} \frac{y_i}{\sigma^2} + \frac{\hat{\theta}_0}{\hat{\sigma}_0^2} \right), \qquad \frac{1}{\hat{\sigma}_t^2} = \frac{1}{\sigma^2} + \frac{1}{\hat{\sigma}_{t-1}^2}. \tag{7}$$

根据方程（7），后验均值服从高斯分布，且后验均值是在线观测与在线先验的和。因此，后验均值是数据的加权和，后验的最终值不依赖于数据的顺序。我们考虑任务发生变化的情形（这种非平稳性在时间序列文献中称为变点，在图 4A 中 $t = 110$ 处）。具体而言，任务 1 的数据集按照 $\mathcal{N}(-1, \sigma^2)$ 生成，因此我们希望模型能够对该任务进行回归。对于任务 2，数据按照 $\mathcal{N}(1, \sigma^2)$ 生成，因此我们也希望持续学习智能体能够很好地对该任务进行回归。与所有持续学习基准测试一样，我们要求模型保留在过去任务上的性能，并在训练结束时 $t = 220$ 对两个任务都表现同样出色。从图 4A 可以看出，随着数据在线到达且线性模型在线更新，该模型能够很好地对第一个数据集进行回归。然而，从第二个任务的数据中可以看到，线性模型最终会跟踪两个任务的全局均值（方程（7）），而非每个任务各自的均值（图 4A）。当任务数据集不平衡时，这一现象更加明显（图 4B）。

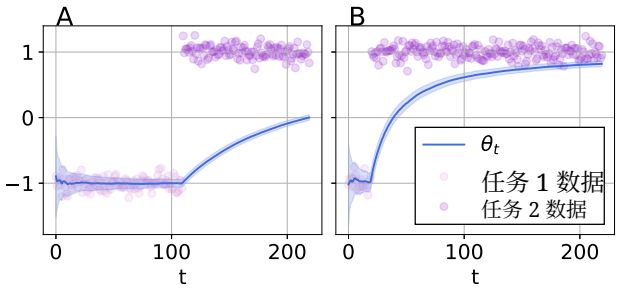

图4. 线性模型滤波分布后验估计，方程（7），其中数据被划分为两个任务。我们研究了两种不同情景，均包含两个任务或变点，数据集分别为平衡和不平衡。在情景 A 中，我们使用任务 1 的数据对线性模型进行 110 次序贯推断更新，然后使用任务 2 的数据再进行 110 次序贯更新。在情景 B 中，任务数据集不平衡。我们使用任务 1 的数据对线性模型进行 20 次序贯贝叶斯更新，然后使用任务 2 的数据进行 200 次更新。

显然，该模型设定错误，因为线性模型无法同时对这两个任务进行回归。更合适的模型是混合模型，它能够对两个任务数据集进行回归。即使进行了精确推断，设定错误的模型仍可能发生遗忘，见图 4。学习第二个任务时，第一个任务上的性能会下降；任务数据集不平衡时，这种现象会更加明显，见图 4B。在哈密顿蒙特卡洛（HMC）的情形下，我们验证了贝叶斯神经网络此前在所有任务上都表现完美。在图 3中，我们使用了设定良好的模型，但在精确序贯贝叶斯推断方面遇到了困难，见方程（3）。在线使用线性模型进行学习时，我们执行的是精确推断，但模型设定错误。区分模型设定错误与精确推断十分重要；据我们所知，持续学习相关文献尚未指出模型设定错误这一局限。此外，只有预先获得所有任务的数据，我们才能确保模型设定良好。因此，在在线持续学习 [26–28], 的情境下，若不对任务分布作出假设，我们就无法确定模型能否在所有过去和未来的任务上表现良好。

5. 序贯贝叶斯推断与不平衡任务数据

神经网络是具有广阔假设空间的复杂模型，因此在解决持续学习问题时，是设定得相当恰当的模型 [29]。然而，在第3节中，我们难以拟合来自哈密顿蒙特卡洛的后验分布样本，以执行序贯贝叶斯推断。

我们继续使用贝叶斯滤波，并假设贝叶斯神经网络的后验分布为具有完全协方差的高斯分布。通过对整个协方差进行建模，我们能够刻画每个权重相对于所有其他权重的变化情况。我们将贝叶斯神经网络中的在线学习解释为滤波 [30]。我们的处理方式与 Aitchison [31],类似，他利用贝叶斯滤波推导出一种优化器。我们考虑图5所示图模型中的推断。目标是在给定单个观测和贝叶斯神经网络权重先验的情况下，推断最优的贝叶斯神经网络权重 $\boldsymbol{\theta}^*_t$ 在 t 。前一时间步的贝叶斯神经网络权重被用作推断贝叶斯神经网络参数后验分布的先验。我们考虑在线设置，即每次观测一个数据点 $(\boldsymbol{x}_t, y_t)$ 。

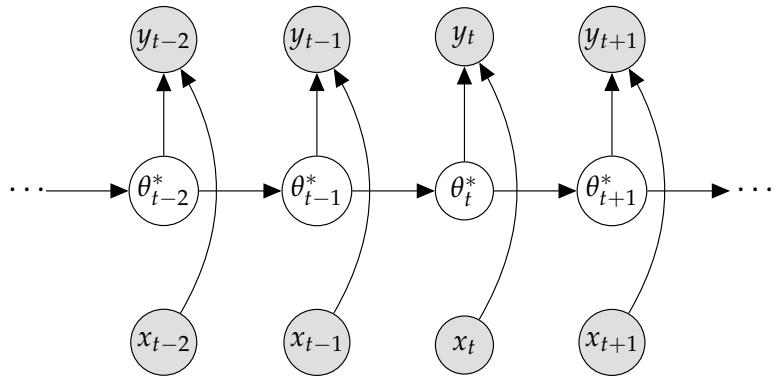

图5。 用于滤波的图模型。灰色节点和白色节点分别表示观测变量和潜变量。

我们不对完全协方差进行建模，而是将每个参数 θ_i 视为所有其他参数的函数 $\boldsymbol{\theta}_{-it}$。我们还假设权重值接近前一时间步的值 [32]。为了根据新的观测和先验得到更新后的贝叶斯神经网络参数方程，我们作出以下两个简化假设。

假设1。 对于输出为 $f(\boldsymbol{x}_t;\ \boldsymbol{\theta})$ 且似然为 $\mathcal{L}(\boldsymbol{x}_t,\ \boldsymbol{y}_t;\ \boldsymbol{\theta})$的贝叶斯神经网络，在 $\boldsymbol{\theta}_t$ 处计算得到的导数为 $\mathbf{z}_t = \partial\mathcal{L}(\boldsymbol{x}_t,\ \boldsymbol{y}_t;\ \boldsymbol{\theta})/\partial\boldsymbol{\theta}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t}$，海森矩阵为 $\boldsymbol{H}$。我们假设数据点 $(\boldsymbol{x}_t,\ \boldsymbol{y}_t)$ 的二次损失形式如下：

$$\mathcal{L}(\boldsymbol{x}_t, y_t; \boldsymbol{\theta}) = \mathcal{L}_t(\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{\theta}^\top \boldsymbol{H}\boldsymbol{\theta} + \mathbf{z}_t^\top \boldsymbol{\theta}, \tag{8}$$

二阶泰勒展开的结果。假设海森矩阵相对于 t 保持不变 o $(\boldsymbol{x}_t,\boldsymbol{y}_t)$ （但不相对于 $\boldsymbol{\theta}$）。

为了构建 $\boldsymbol{\theta}$ 的动力学方程，考虑第 i 个权重的梯度，同时将所有其他参数设为其在 θ^*_{it} 的最优值处的当前估计：

$$\theta^*_{it} = -\frac{1}{H_{ii}}\boldsymbol{H}^\top_{-ii}\boldsymbol{\theta}_{-it}, \tag{9}$$

因为在众数处 $z_{it} = 0$ 。上面的方程表明，最优权重 θ^*_{it} 的动力学取决于其他所有参数 $\boldsymbol{\theta}_{-it}$ 的当前值。 $\boldsymbol{\theta}_{-it}$ 的动力学是一个复杂的随机过程，取决于许多不同变量，例如数据集、模型架构、学习率调度等。

假设 2. 由于无法对 $\boldsymbol{\theta}_-$ 的动力学进行推理，我们假设在下一个时间步，最优权重与前一时间步的权重接近，并且权重 $\boldsymbol{\theta}_{-it}$ 遵循离散化的奥恩斯坦–乌伦贝克过程，其回复速度为 $\vartheta \in \mathbb{R}_+$，噪声方差为 η^2_{-i}：

$$p(\boldsymbol{\theta}_{-i,t+1}|\boldsymbol{\theta}_{-i,t}) = \mathcal{N}((1-\vartheta)\boldsymbol{\theta}_{-it}, \eta^2_{-i}), \tag{10}$$

这意味着最优权重的动态由下式定义：

$$p(\theta^*_{i,t+1}|\theta^*_{i,t}) = \mathcal{N}((1-\vartheta)\theta^*_{it}, \eta^2), \tag{11}$$

其中 $\eta^2 = \eta^2_{-i}\boldsymbol{H}^\top_{-ii}\boldsymbol{H}_{-ii}$.

简单来说，在假设 2 中，我们假设存在一个简约动力学模型，并且根据高斯分布， $\boldsymbol{\theta}_{-i,t}$ 的下一个值与其前一个值接近，这与 Aitchison [31] 的做法类似。

引理1. 根据假设*1* 和 *2* 动力学和似然均服从高斯分布。因此，我们能够通过贝叶斯更新推断最优权重的后验分布；对贝叶斯神经网络进行线性化后，对于一个新的数据点，其均值和方差的后验更新方程为：

$$\mu_{t,post} = \sigma^2_{t,post}\left(\frac{\mu_{t,prior}}{\sigma^2_{t,prior}(\eta^2)} + \frac{y_t}{\sigma^2}g(\boldsymbol{x}_t)\right) \qquad and \qquad \frac{1}{\sigma^2_{t,post}} = \frac{g(\boldsymbol{x}_t)^2}{\sigma^2} + \frac{1}{\sigma^2_{t,prior}(\eta^2)}, \quad (12)$$

其中我们省略了第 i 个参数的记号，后验分布为 $\mathcal{N}(\theta^*t;\ \mu_{t,\ post},\ \sigma^2_{t,\ post})$，且 $g(\boldsymbol{x}_t) = \partial f(\boldsymbol{x}_t;\ \theta^*_{it})\,\partial\theta^*_{it}$ 以及 $\sigma^2_{t,\ prior}$ 是 η^2 的函数。

引理1 的推导见附录1。根据方程（12），我们可以发现，后验均值线性依赖于先验和依赖数据的项，因此其表现将与第4节中的前一个示例类似。在假设1和假设2成立的条件下，如果方程（12）中任务之间存在数据不平衡，那么当当前任务的数据更多时，依赖数据的项将占主导地位。

在第3节中，我们展示了使用当前机器学习工具，对具有小型贝叶斯神经网络的简单持续学习问题执行序贯贝叶斯推断非常困难。通过将贝叶斯推断与模型设定错误区分开来，我们表明，即使进行精确的贝叶斯推断，设定错误的模型仍可能发生遗忘。确保模型设定正确的唯一方法，是证明多任务后验能够针对具体应用产生合理的后验预测分布$p(y|\boldsymbol{x},\ \mathcal{D}) = \int p(y|\boldsymbol{x},\ \mathcal{D},\ \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$。此外，在本节中，我们还表明，在某些假设下，如果任务数据集规模不平衡，就会出现遗忘。

6. 相关工作

随着深度学习的兴起，持续学习领域近来重新焕发活力 [33] 。用于近似序贯贝叶斯推断方程（3）的方法在持续学习的复兴中发挥了开创性作用，并采用了对角拉普拉斯近似 [3,7]。通过对同一层中神经网络权重之间的协方差进行建模，对角拉普拉斯近似得到了改进 [8]。除了拉普拉斯近似，我们还可以使用一种用于序贯贝叶斯推断的变分近似，称为变分持续学习 [9,34]。每个贝叶斯神经网络参数的变分高斯方差可用于对各权重的学习率进行预条件化，并通过剪枝为每个任务创建一个掩码 [10]。采用更丰富的先验也已得到探索 [11,35–38]。例如，可以通过学习新的变分自适应参数，为每个任务学习高斯神经网络权重参数的缩放因子，从而增强特定神经元的贡献 [39]。在线拉普拉斯近似可视为变分持续学习的一种特殊情况，其中 KL 散度项方程（6）经过温度调节，且温度趋于 0 [12]。高斯过程也已应用于持续学习问题，利用诱导点保留先前任务的函数 [40,41]。

在持续学习图像分类基准测试的准确率方面，正则化权重的贝叶斯方法尚未达到基于经验回放的持续学习方法的性能 [42] 。与其正则化高维权重空间，不如正则化任务函数，这是一种更直接的对抗遗忘的方法 [43]。贝叶斯神经网络权重也可以由超网络生成，而超网络只需采用简单的持续学习技术即可防止遗忘 [44]。具体而言，可以利用拉普拉斯近似与高斯过程之间的对偶性，开发一种面向贝叶斯持续学习的函数正则化方法 [45] ，或使用函数空间变分推断 [46,47]。

在下一节中，我们提出一种简单的贝叶斯持续学习基线方法，用于对数据生成的持续学习过程建模，并在低维嵌入空间中执行精确的序贯贝叶斯推断。以往的工作曾尝试通过推断输入与目标的联合分布 $p(\mathbf{x},\mathbf{y})$ 来对数据生成过程建模，并学习生成模型以回放数据、防止遗忘 [48],

以及通过为每个类别学习一个生成模型，并评估给定各类别时输入的似然$p(\mathbf{x}|\mathrm{y})$ [49]。

7. 原型贝叶斯持续学习

我们已经证明，对神经网络参数进行序贯贝叶斯推断非常困难（第 3 节），而且只适用于多任务后验对所有任务都适用的情况。现在我们将说明，建模持续学习问题完整的数据生成过程是一种更有效的方法，并提出一种简单且可扩展的实现方案。具体而言，我们用原型来表示类别 [50,51]，以防止灾难性遗忘。我们将这一框架称为原型贝叶斯持续学习，简称 ProtoCL。这种方法可以看作是 iCarl 的一种概率变体 [51]，；iCarl 为不同类别创建嵌入函数，这些函数仅由类别均值构成，并通过最近邻进行预测。ProtoCL 也与为少样本图像分类开发的小样本学习模型“在线分类概率聚类”相似 [52]。

模型。 ProtoCL 对生成式持续学习过程进行建模。我们考虑由带有狄利克雷先验的分类分布生成的类别 $j \in \{1,\ldots,J\}$：

$$y_{i,t} \sim \mathrm{Cat}(p_{1:J}), \quad p_{1:J} \sim \mathrm{Dir}(\alpha_t). \tag{13}$$

图像通过带有参数$\boldsymbol{w}$的编码器$\boldsymbol{z} = f(\boldsymbol{x};\boldsymbol{w})$映射到嵌入空间。每个类别的嵌入服从高斯分布，其均值具有一个同样服从高斯分布的先验：

$$\boldsymbol{z}_{it}|y_{it} \sim \mathcal{N}(\bar{\boldsymbol{z}}_{yt}, \Sigma_e), \quad \bar{\boldsymbol{z}}_{yt} \sim \mathcal{N}(\boldsymbol{\mu}_{yt}, \Lambda_{yt}^{-1}). \tag{14}$$

模型概览见图6 。为缓解持续学习中的遗忘，ProtoCL 使用过去任务数据的核心集，以原型的形式继续对过去的类别进行区分。类别概率 $\{p_j\}_{j=1}^J$和类别嵌入 $\{\bar{z}_{y_j}\}_{j=1}^J$的后验分布简记为$p(\boldsymbol{\theta})$，其参数为 $\eta_t = \{\alpha_t, \boldsymbol{\mu}_{1:J,t}, \Lambda_{1:J,t}^{-1}\}$。ProtoCL 为每个类别建立原型模型，但不使用任务专属的神经网络参数或模块，例如多头变分持续学习。通过在嵌入空间上建立概率模型，我们可以使用强大的嵌入函数$f(\cdot;\boldsymbol{w})$，而无需对其进行概率参数化，因此，例如，这种方法比变分持续学习更具可扩展性。

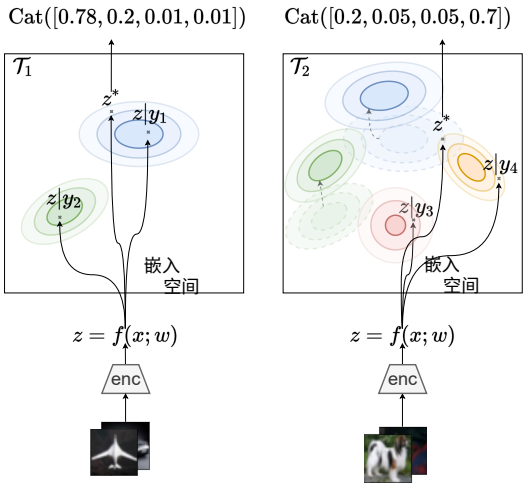

图6。 ProtoCL 概览。

推断。 由于狄利克雷先验与分类分布共轭，原型上的高斯分布同样与原型均值上的高斯先验共轭，因此我们可以通过闭式解计算后验分布，并在观察到新数据时更新参数 η_t

，而无需使用基于梯度的更新。我们通过最大化后验预测分布来优化模型，并使用类别概率上的 Softmax 进行预测。我们对神经网络嵌入函数 $f(\cdot;\boldsymbol{w})$进行基于梯度的学习，并在每次梯度下降迭代中更新参数 η_t，另请参见算法 1。

算法 1 ProtoCL 持续学习

1：输入：任务数据集 $\mathcal{T}_{1:T}$，初始化嵌入函数：f (·; w)，核心集： $\mathcal{M} = \emptyset$。2：对于 $\mathcal{T}_1$ 到 $\mathcal{T}_T$，执行 3：对于 $\mathcal{T}_i \cup \mathcal{M}$ 中的每个批次，执行 4：通过最大化后验预测分布 p(z, y) 方程（18）来优化 f (·; w) 5：通过更新 η，方程（15）–（17），获得关于 θ的后验分布。6：结束循环 7：从 $\mathcal{T}_i$ 中添加随机子集到 $\mathcal{M}$。8：结束循环

序贯更新。我们可以通过分类分布-狄利克雷共轭性，获得狄利克雷后验分布的参数更新：

$$\alpha_{t+1,j} = \alpha_{t,j} + \sum_{i=1}^{N_t} \mathbb{I}(y_t^i = j), \tag{15}$$

其中 N_t 是时间步 t更新期间所见数据点的数量。此外，由于高斯-高斯共轭性，高斯原型的后验分布由下式决定：

$$\Lambda_{y_{t+1}} = \Lambda_{y_t} + N_y \Sigma_\epsilon^{-1} \tag{16}$$

$$\Lambda_{y_{t+1}} \boldsymbol{\mu}_{y_{t+1}} = N_y \Sigma_\epsilon^{-1} \bar{\boldsymbol{z}}_{y_t} + \Lambda_{y_t} \boldsymbol{\mu}_{y_t}, \forall y_t \in C_t, \tag{17}$$

其中 N^y 表示类别 y 的样本数量，且 $\bar{z}^{y_t} = (1/N^y) \sum^{N_y}{}_{i=1} z^{yi}$，详见附录D.2 中的详细推导。

目标函数。 我们优化原型和类别的后验预测分布：

$$p(\mathbf{z}, y) = \int p(\mathbf{z}, y|\boldsymbol{\theta}_t; \eta_t) p(\boldsymbol{\theta}_t; \eta_t) d\boldsymbol{\theta}_t = p(y) \prod_{i=1}^{N_t} \mathcal{N}(\mathbf{z}_{it}|y_{it}; \boldsymbol{\mu}_{y_t,t}, \Sigma_\epsilon + \Lambda_{y_t,t}^{-1}). \tag{18}$$

其中 $p\ (y) = \alpha_y / \sum^J{}_{j=1} \alpha_j$，详见附录 D.3 中的详细推导。随后可以使用基于梯度的优化来优化该目标函数，从而学习原型嵌入函数 $\boldsymbol{z} = f(\boldsymbol{x}; \boldsymbol{w})$。

预测。 对于测试点 $\boldsymbol{x}^*$，选择（对数）后验预测分布最大的类别，其中后验预测分布为：

$$p(y^* = j|\boldsymbol{x}^*, \boldsymbol{x}_{1:t}, y_{1:t}) = p(y^* = j|\boldsymbol{z}^*, \boldsymbol{\theta}_t) = \frac{p(y^* = j, \boldsymbol{z}^*|\boldsymbol{\theta}_t)}{\sum_i p(y = i, \boldsymbol{z}^*|\boldsymbol{\theta}_t)}, \tag{19}$$

详见附录 D.4。

防止遗忘。 由于我们希望保留类别原型，因此使用核心集：来自先前任务的经验。在任务 $\mathcal{T}_t$ 的学习结束时，我们保留一个子集$\mathcal{M}_t \subset \mathcal{D}_t$，并扩充每个新的任务数据集，以确保后验参数 η_t 和原型能够保留先前任务的信息。

类别增量学习。 在这种持续学习设置中，我们不会告诉持续学习智能体正在评估哪个任务，也不会提供任务标识符 τ。因此，我们无法使用任务标识符来选择用于分类测试点的特定分类头。此外，我们要求持续学习智能体识别 Split-MNIST 和 Split-CIFAR10 中的每个类别，$\{0, 。 . . , 9\}$，而不仅仅是像域增量学习中那样识别 $\{0, 1\}$。类别增量学习是一种更通用、更现实且更具挑战性的问题设置，因此值得重点关注，而不是

其他设置，尽管如公式(3)所述，域增量学习也兼容序贯贝叶斯推断。

实现。 对于 Split-MNIST 和 Split-FMNIST，基线方法和 ProtoCL 均使用隐藏状态大小为 200 的两层神经网络。对于 Split-CIFAR10 和 Split-CIFAR100，基线方法和 ProtoCL 使用与 Pan 等人 [22] 类似的四层卷积神经网络，其中包含两个大小为 512 的全连接层。对于 ProtoCL 以及所有依赖回放的基线方法，我们将核心集大小固定为每个任务 200 个点。对于所有 ProtoCL 模型，我们允许学习先验狄利克雷参数，并将其初始值设为 0.7；该值是通过在 MNIST 上使用 ProtoCL 进行随机搜索得到的。ProtoCL 的一个重要超参数是 Split-MNIST 和 Split-FMNIST 的高斯原型嵌入维度，该值设为 128；对于较大的视觉数据集，该值设为 32，是通过网格搜索得到的。

结果。 ProtoCL 在持续学习基准测试上的表现良好，达到或优于 S-FSVI [47]，；S-FSVI 是贝叶斯持续学习方法中最先进的水平，而 ProtoCL 的训练效率高得多，且无需昂贵的变分推断。ProtoCL 能够灵活地扩展到更大型的持续学习视觉基准测试，并取得优于 S-FSVI 的结果。复现所有实验的代码可在此处找到 https://github.com/skezle/bayes_cl_后验分布。我们的所有实验均采用更贴近实际的类别增量学习设置，这比大多数持续学习论文报告的设置更具挑战性，因此表 1 中某些基线方法的结果低于相应论文中的结果。每个任务使用 200 个数据点；参见图 A6，其中分析了 ProtoCL 在 Split-MNIST 基准测试上的表现对核心规模的敏感性。表 2 展示了 ProtoCL 如何扩展到更大型、更具挑战性的持续学习视觉基准测试。与我们考虑的基线方法相比，ProtoCL 展现出有竞争力的性能，同时在同一 GPU 上进行基准测试时，训练时间所需的计算成本仅为其一小部分。

表1。 持续学习视觉基准测试中，类别增量学习 [17]在所有任务上的平均准确率。所有结果均为 10 个随机种子上的均值和标准误差。 * 使用预测熵来决定类别增量学习应使用哪个分类头。

方法	核心集	Split-MNIST	Split-FMNIST
变分持续学习 [9]	✗	33.01 ± 0.08	32.77 ± 1.25
＋ 核心集	✓	52.98 ± 18.56	61.12 ± 16.96
HIBNN * [11]	✗	85.50 ± 3.20	43.70 ± 20.21
FROMP [22]	✓	84.40 ± 0.00	68.54 ± 0.00
S-FSVI [47]	✓	**92.94 ± 0.17**	80.55 ± 0.41
ProtoCL（我们的方法）	✓	**93.73 ± 1.05**	**82.73 ± 1.70**

表2。 持续学习视觉基准测试中所有任务的平均准确率，适用于类别增量学习 [17]。所有结果均为 10 个随机种子的均值和标准误差。 * 使用预测熵来决定类别增量学习应采用哪个分类头。训练时间通过英伟达 RTX3090 GPU 进行基准测试。

方法	训练时间（秒）(↓)	拆分 CIFAR-10（准确率）(↑)
FROMP [22]	1425 ± 28	48.92 ± 10.86
S-FSVI [47]	44434 ± 91	50.85 ± 3.87
ProtoCL（我们的方法）	**384 ± 6**	**55.81 ± 2.10**
		拆分 CIFAR-100（准确率）(↑)
S-FSVI [47]	37355 ± 1135	20.04 ± 2.37
ProtoCL（我们的方法）	**1425 ± 28**	**23.96 ± 1.34**

ProtoCL 的既定目标并非提出一种用于持续学习的全新最先进方法，而是提出一种简单的基线方法，采用不同于权重空间序贯贝叶斯推断的路径。我们能够取得出色的结果，有效缓解遗忘，

具体而言，我们通过对生成式持续学习过程建模，并在类别原型嵌入空间中对少量参数进行序贯贝叶斯推断。我们认为，与尝试对贝叶斯神经网络的权重进行序贯贝叶斯推断相比，对生成式持续学习过程建模是一个值得进一步研究的方向。据我们所知，ProtoCL 可扩展至 Split-CIFAR100 的 10 个任务，这是以往贝叶斯持续学习方法所考虑过的任务数和类别数中的最高水平。

8. 讨论与结论

本文重新审视了序贯贝叶斯推断在持续学习中的应用。我们可以使用序贯贝叶斯方法递归地构建多任务后验分布，见公式（3）。以往方法依赖近似推断，与随机梯度下降相比几乎没有优势。我们在两个简单的持续学习问题中使用哈密顿蒙特卡洛，检验性能不佳是否源于近似推断方案这一假设。哈密顿蒙特卡洛渐近地从真实后验分布中采样；我们使用 HMC 样本上的密度估计器，在哈密顿蒙特卡洛采样过程中将其用作新任务的先验。该密度呈多模态，且对当前任务的刻画准确，但与使用近似后验分布相比并未带来改进。这充分说明，处理贝叶斯神经网络权重后验分布极具挑战性。误差源于密度估计步骤。随后，我们考察了一个序贯贝叶斯推断的解析示例，并进行了精确推断；然而，由于模型设定错误，我们观察到了遗忘。确保模型设定正确的唯一方法，是预先评估其在所有任务上的整体多任务表现。在在线持续学习设置中，这可能无法实现。接着，我们针对贝叶斯神经网络构建了一个解析示例，并在某些假设下证明，如果任务数据存在不平衡，就会导致遗忘。基于这些结果，我们不赞成进行权重空间序贯贝叶斯推断，而主张对生成式持续学习问题进行建模。我们提出了一个名为 ProtoCL 的简单基线方法。ProtoCL 无需复杂的变分优化，并且在真实的类别增量学习设置中取得了与最先进水平相当的结果。

这一结论并不令人意外，因为最新的贝叶斯持续学习论文都依赖多头架构或诱导点/核心集来防止遗忘，而非采用更优的权重空间推断方案。我们的观测与 [53], 近期的理论结论一致，该理论指出，最优的持续学习需要完美记忆。尽管这些结果是在确定性神经网络上展示的，但对于参数集为单一的一组参数的贝叶斯神经网络，同样可以得出这些结果。未来的研究方向包括利用任务数据核心集，高效且准确地近似贝叶斯神经网络的后验分布，以记住先前任务。

作者贡献： S.K. 主导了这项研究，包括构思、开展实验和撰写论文。S.J.R. 协助构思。A.C. 协助完善研究思路，并实现了将密度估计器用作先验的哈密顿蒙特卡洛。T.G.J.R. 为类别增量持续学习实验运行了 S-FSVI 基线方法。T.G.J.R.、A.C. 和 S.J.R. 协助撰写论文。

经费支持： S.K. 感谢牛津-曼彻斯特定量金融研究所提供的经费支持。T.G.J.R. 感谢罗德信托基金、高通以及工程与物理科学研究委员会（EPSRC）提供的经费支持。本材料所述研究工作由美国空军和美国国防高级研究计划局根据合同编号 FA8750-20-C-0002 提供支持。本文所表达的任何观点、研究结果、结论或建议均属于作者个人，不一定反映美国空军和美国国防高级研究计划局的观点。

机构审查委员会声明： 不适用。

数据可用性声明： 所有数据均可公开获取，用于复现全部实验的代码可在此处找到 https://github.com/skezle/bayes_持续学习_后验分布。

致谢： 我们感谢 Sebastian Farquhar、Laurence Aitchison、Jeremias Knoblauch 和 Chris Holmes 与我们进行讨论。我们还要感谢 Philip Ball 在论文撰写方面提供的帮助。

利益冲突：作者声明不存在利益冲突。资助方未参与研究设计、数据收集、分析或解读、稿件撰写，也未参与决定是否发表研究结果。

缩略语
本文稿中使用了以下缩略语：

CL 持续学习 NN 神经网络 BNN 贝叶斯神经网络 HMC 哈密顿蒙特卡洛 VCL 变分持续学习 SGD 随机梯度下降 SH 单头 MH 多头 GMM 高斯混合模型 ProtoCL 原型贝叶斯持续学习

附录A. 玩具高斯数据集

玩具高斯数据集的可视化请参见图 A1 。我们将其用作一个简单的持续学习问题。我们使用哈密顿蒙特卡洛进行后验推断，并将 HMC 样本输入密度估计器，作为新任务的先验，以此评估我们传播真实后验分布的方法。我们为持续学习构造了 5 个二分类问题。每个二分类任务都涉及对相邻的圆形和正方形进行分类，见图 A1。使用一个包含 2 层、每层 10 个神经元的网络，对全部 5 个任务进行多任务学习时，我们的测试准确率达到 1.0。因此，根据公式（3），相同规模的贝叶斯神经网络应当能够通过依次构建后验分布，持续学习全部 5 个二分类任务。

附录 B. 哈密顿蒙特卡洛实现细节

我们将 $\mathcal{T}_1$ 的先验设为 $p_1(\boldsymbol{\theta}) = \mathcal{N}(0, \tau^{-1}\mathbb{I})$，其中 $\tau = 10$。我们对 HMC 链进行 1000 步预热，再采样 10, 000 步，并运行 20 条不同的链，以获取后验分布样本，随后将其输入密度估计器。我们采用 0.001 的步长和 $L = 20$ 的轨迹长度。密度估计过程的更多实现细节请参见附录 C。对于高斯混合模型，我们使用 HMC 样本的留出集来优化分量数。

附录C：密度估计诊断

我们在图A3和A5中提供了图表，以展示 HMC 链确实从后验分布中采样，且已收敛。我们运行了 20 条 HMC 采样链，并针对每个随机种子（共 10 个）随机选取一条链绘图。我们针对 10 个随机种子运行 HMC，并在图3 和A1中汇总结果。后验分布$p(\theta|\mathcal{D}_1)$等通过高斯混合模型近似，并用作第二个任务的先验，依此类推。

我们在图A2 和A4中提供了实证证据，表明密度估计器已拟合后验分布的 HMC 样本。我们展示了高斯混合模型密度估计器的成分数量；该估计器被用作新任务的先验，且这些后验分布均为多模态。我们展示了从高斯混合模型中采样贝叶斯神经网络权重时，贝叶斯神经网络的准确率均能恢复到收敛的 HMC 样本所达到的准确率。20 条链的有效样本量（ESS）衡量样本的相关程度（越高越好）。我们实验报告的 ESS 值与先前使用 HMC 进行贝叶斯神经网络推断的研究结果一致 [20]。

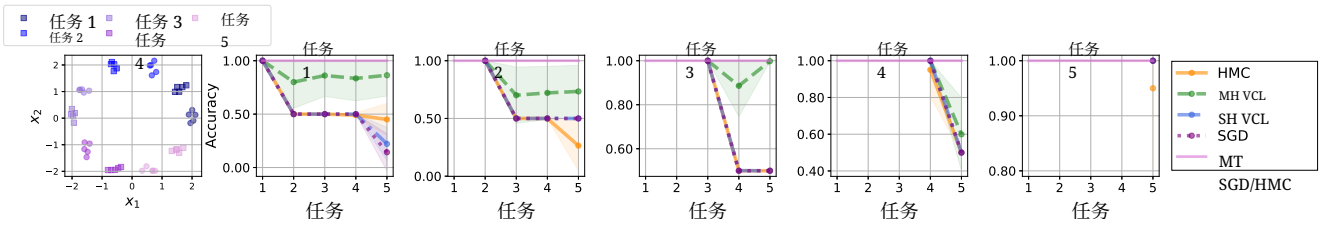

图 A1. 与 [44] 类似的玩具高斯数据集上的持续学习二分类准确率。使用 10 个随机种子。粉色实线表示采用 SGD/HMC 的多任务（MT）基线测试准确率。

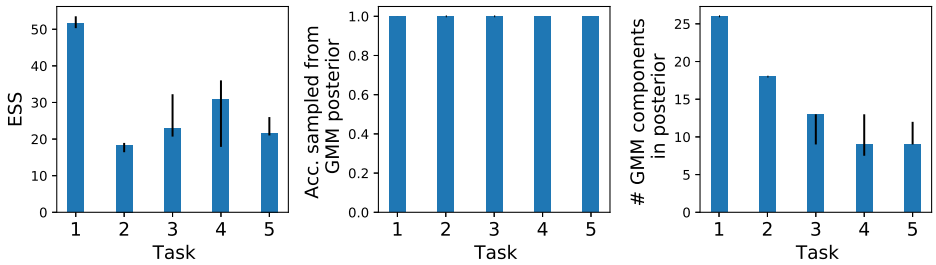

图 A2. 使用拟合于 HMC 生成的后验分布样本的高斯混合模型先验所得的诊断结果；所有结果均基于 10 个随机种子。**左侧，** 所得后验分布 HMC 链的有效样本量（ESS）均高于其他使用 HMC 处理贝叶斯神经网络的研究所报告的数值 [20]。**中间，** 使用高斯混合模型密度估计器中的样本，而非 HMC 样本时，贝叶斯神经网络的准确率。**右侧，** 通过最大化似然，使用 HMC 样本留出集拟合的各个高斯混合模型后验分布的最佳成分数。

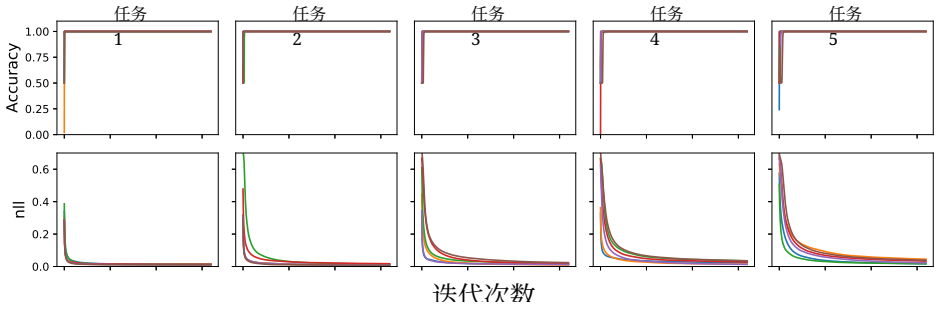

图 A3. 与 Henning 等人类似的玩具高斯数据集上，5 个任务各自进行 10 次不同运行（随机种子）时，每个任务随机抽取的一条 HMC 链（共 20 条）的收敛图 [44]（见图 A1）。我们使用高斯混合模型密度估计器作为以先前任务数据为条件的先验。

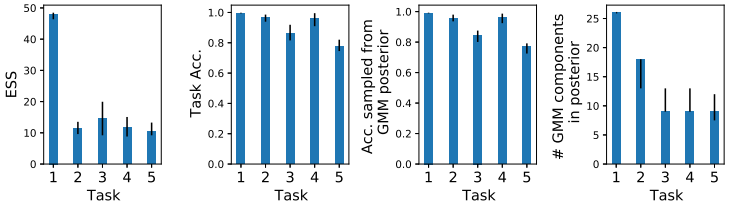

图 A4. 使用高斯混合模型拟合后验 HMC 样本所得的诊断结果；所有结果均来自 Pan 等人玩具数据集上的 10 个随机种子。 [22] （并在图 3中进行了可视化）。**左**，后验分布所得 HMC 链的有效样本量（ESS）均高于其他使用 HMC 处理贝叶斯神经网络的研究所报告的数值 [20]。**中左**：HMC 采样得到的当前任务准确率。**中右**：使用高斯混合模型密度估计器的样本，而非收敛的 HMC 样本时，贝叶斯神经网络的准确率。**右**：通过最大化似然，使用 HMC 样本留出集拟合的各后验高斯混合模型的最佳成分数。

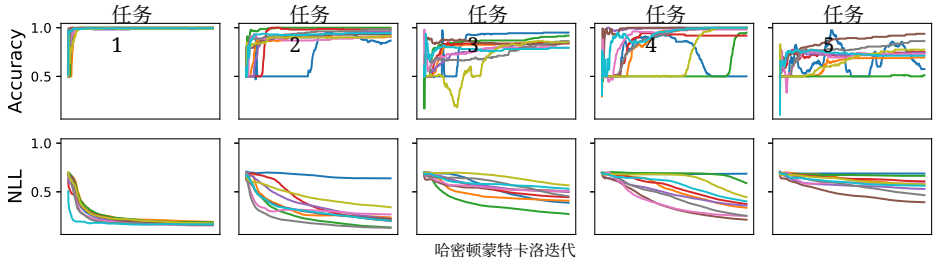

图 A5。对于来自所述玩具数据集的 5 个任务，在 10 个不同随机种子下，每个任务从随机采样的 HMC 链（共 20 条）中得到的收敛图（有关数据的可视化，请参见图 3）。我们使用高斯混合模型密度估计器作为先验。

附录 D. 原型贝叶斯持续学习

ProtoCL 对持续学习的生成过程进行建模，其中新任务由新类别$j \in \{1,.....J\}$组成，总共有J个类别，并且可以使用带有狄利克雷先验的分类分布进行建模：

$$y_{i,t} \sim \text{Cat}(p_{1:J}), \quad p_{1:J} \sim \text{Dir}(\alpha_t). \tag{A1}$$

我们使用神经网络 NN 学习数据的联合嵌入空间，$\boldsymbol{z} = f(\boldsymbol{x};\ \boldsymbol{w})$，其参数为$\boldsymbol{w}$。每个类别的嵌入空间均为高斯分布，其均值的先验同样为高斯分布：

$$\boldsymbol{z}_{it}|y_{it} \sim \mathcal{N}(\bar{\boldsymbol{z}}_{yt}, \Sigma_\epsilon), \quad \bar{\boldsymbol{z}}_{yt} \sim \mathcal{N}(\boldsymbol{\mu}_{yt}, \Lambda_{yt}^{-1}). \tag{A2}$$

通过确保每个类别都有一个嵌入，并使用过去数据的记忆库，我们可以确保嵌入不会漂移。后验参数为 $\eta_t = \{\alpha_t, \boldsymbol{\mu}_{1:J,t}, \Lambda_{1:J,t}^{-1}\}$。

附录 *D.1.* 推断

由于狄利克雷先验与分类分布共轭，且高斯分布与嵌入均值上的高斯先验也共轭，因此我们可以通过闭式解计算后验分布，并在看到新的线上数据时更新参数，而无需使用基于梯度的更新。我们对神经网络嵌入函数$f(\cdot;\boldsymbol{w})$进行基于梯度的学习，其参数为$\boldsymbol{w}$。我们通过最大化数据的对数预测后验来优化模型，并使用类别概率上的 Softmax 进行预测。类别概率 $\{p_j\}_{j=1}^J$ 和类别嵌入$\{\bar{\boldsymbol{z}}_{y_j}\}_{j=1}^J$的后验分布简记为$p(\theta)$，其参数为 $\eta_t = \{\alpha_t, \boldsymbol{\mu}_{1:J,t}, \Lambda_{1:J,t}^{-1}\}$，并在每次梯度下降迭代中以闭式解进行更新。

附录 *D.2.* 序贯更新

我们可以得到后验分布：

$$p(\boldsymbol{\theta}_t|\mathcal{D}_t) \propto p(\mathcal{D}_t|\boldsymbol{\theta}_t)p(\boldsymbol{\theta}_t) \tag{A3}$$

$$= \prod_{i=1}^{N_t} p(\boldsymbol{z}_t^i|y_t^i; \bar{\boldsymbol{z}}_{y_t}, \Sigma_{\epsilon,y_t})p(y_t^i|p_{1:J})p(p_{i:J};\alpha_t)p(\bar{\boldsymbol{z}}_{y_t};\boldsymbol{\mu}_{y_t,t}, \Lambda_{y_t,t}^{-1}) \tag{A4}$$

$$= \mathcal{N}\ （\mu_{t+1}, \Sigma_{t+1}）\ 方向\ （\alpha_{t+1}）\ , \tag{A5}$$

其中，N_t 是在更新t期间观察到的数据点数量。聚焦于分类分布-狄利克雷共轭：

$$\mathrm{Dir}(\alpha_{t+1}) \propto p(p_{1:J};\alpha_t)\prod_{i=1}^{N_t} p(y_t^i;p_{i:J}) \tag{A6}$$

$$\propto \prod_{j=1}^{J} p_j^{\alpha_j-1}\prod_{i=1}^{N_t}\prod_{j=1}^{J} p_j^{\mathbb{I}(y_t^i=j)} \tag{A7}$$

$$= \prod_{j=1}^{J} p_j^{\alpha_j-1+\sum_{i=1}^{N_t}\mathbb{I}(y_t^i=j)}. \tag{A8}$$

因此：

$$\alpha_{t+1,j} = \alpha_{t,j} + \sum_{i=1}^{N_t}\mathbb{I}(y_t^i=j). \tag{A9}$$

此外，由于高斯-高斯共轭性，嵌入的高斯原型的后验分布对于每个类别而言为：

$$\mathcal{N}(\boldsymbol{\mu}_{t+1},\Lambda_{t+1}) \propto \prod_{i=1}^{N_t}\mathcal{N}(z_t^i|y_t^i;\bar{z}_{y_t},\Sigma_\epsilon)\mathcal{N}(\bar{z}_{y_t};\boldsymbol{\mu}_{y_t,t},\Lambda_{y_t}^{-1}) \tag{A10}$$

$$= \prod_{y_t\in\{1,\dots,J\}} \mathcal{N}(z_{y_t}|y_t;\bar{z}_{y_t},\frac{1}{N_{y_t}}\Sigma_\epsilon)\mathcal{N}(\bar{z}_{y_t};\boldsymbol{\mu}_{y_{t+1}},\Lambda_{y_t}^{-1}) \tag{A11}$$

$$= \prod_{y_t\in\{1,\dots,J\}} \mathcal{N}(\bar{z}_{y_t};\boldsymbol{\mu}_{t+1},\Lambda_{y_{t+1}}^{-1}), \tag{A12}$$

其中，N^{y_t}是所有类别 y_t集合$C = \{1,...J\}$中类别的点数。后验分布均值和方差的更新方程为：

$$\Lambda_{y_{t+1}} = \Lambda_{y_t} + N_{y_t}\Sigma_\epsilon^{-1}, \quad \forall y_t \in C_t \tag{A13}$$

$$\Lambda_{y_{t+1}}\boldsymbol{\mu}_{y_{t+1}} = N_{y_t}\Sigma_\epsilon^{-1}\bar{z}_{y_t} + \Lambda_{y_t}\boldsymbol{\mu}_{y_t}, \quad \forall y_t \in C_t. \tag{A14}$$

附录 *D.3. ProtoCL* 目标函数

我们希望优化的后验预测分布为：

$$p(z,y) = \int p(z,y|\boldsymbol{\theta};\eta)p(\boldsymbol{\theta};\eta)d\boldsymbol{\theta}, \tag{A15}$$

其中 $p(\boldsymbol{\theta})$ 表示类别概率 $\{p_j\}^J{}_{j=1}$ 和均值嵌入$\{\bar{z}_{y_j}\}^J{}_{j=1}$的分布，

$$p(z,y) = \int\prod_{i=1}^{N_t} p(z_{it}|y_{it};\bar{z}_{y_t},\Sigma_\epsilon)p(y_{it}|p_{1:J})p(p_{1:J};\alpha_t)p(\bar{z}_{y_t};\boldsymbol{\mu}_{y_t,t},\Lambda_{y_t,t}^{-1})dp_{1:J}d\bar{z}_{y_t} \tag{A16}$$

$$= \int\prod_{i=1}^{N_t} p(z_{it}|y_{it};z_{y_t},\Sigma_\epsilon)p(\bar{z}_{y_i};\boldsymbol{\mu}_{y_t,t},\Lambda_{y_t,t}^{-1})d\bar{z}_{y_t}\underbrace{\int\prod_{i=1}^{N_t} p(y_{it}|p_{1:J})p(p_{1:J};\alpha_t)dp_{1:J}}_{\prod_i p(y_i)=p(y)} \tag{A17}$$

$$= p(y)\prod_{i=1}^{N_t} Z_i^{-1}\int\mathcal{N}(\bar{z}_{y_{it}};\boldsymbol{c},C)d\bar{z}_{y_t} \tag{A18}$$

$$= p(y)\prod_{i=1}^{N_t}\mathcal{N}(z_{it}|y_{it};\boldsymbol{\mu}_{y_t,t},\Sigma_\epsilon+\Lambda_{y_t,t}^{-1}). \tag{A19}$$

其中在方程 (A18) 中，我们在 [54]中使用 §8.1.8。项 $p(y)$ 为：

$$p(y) = \int p(y|p_{1:J})p(p_{1:J};\alpha_t)dp_{1:J} \tag{A20}$$

$$= \int p_y \frac{\Gamma(\sum_{j=1}^J \alpha_j)}{\prod_{j=1}^J \Gamma(\alpha_j)} \prod_{j=1}^J p_j^{\alpha_j - 1} dp_{1:J} \tag{A21}$$

$$= \frac{\Gamma(\sum_{j=1}^J \alpha_j)}{\prod_{j=1}^J \Gamma(\alpha_j)} \int \prod_{j=1}^J p_j^{\mathbb{I}(y=j)+\alpha_j - 1} dp_{1:J} \tag{A22}$$

$$= \frac{\Gamma(\sum_{j=1}^J \alpha_j)}{\prod_{j=1}^J \Gamma(\alpha_j)} \frac{\prod_{j=1}^J \Gamma(\mathbb{I}(y=j)+\alpha_j)}{\Gamma(1+\sum_{j=1}^J \alpha_j)} \tag{A23}$$

$$= \frac{\cancel{\Gamma(\sum_{j=1}^J \alpha_j)}}{\prod_{j=1}^J \Gamma(\alpha_j)} \frac{\prod_{j=1}^J \Gamma(\mathbb{I}(y=j)+\alpha_j)}{\sum_{j=1}^J \alpha_j \cancel{\Gamma(\sum_{j=1}^J \alpha_j)}} \tag{A24}$$

$$= \frac{\prod_{j=1, j\neq y}^J \Gamma(\alpha_j)}{\prod_{j=1}^J \Gamma(\alpha_j)} \frac{\Gamma(1+\alpha_y)}{\sum_{j=1}^J \alpha_j} \tag{A25}$$

$$= \frac{\prod_{j=1, j\neq y}^J \Gamma(\alpha_j)}{\prod_{j=1}^J \Gamma(\alpha_j)} \frac{\alpha_y \Gamma(\alpha_y)}{\sum_{j=1}^J \alpha_j} \tag{A26}$$

$$= \frac{\alpha_y}{\sum_{j=1}^J \alpha_j}, \tag{A27}$$

其中我们使用恒等式 $\Gamma(n+1) = n\Gamma(n)$。

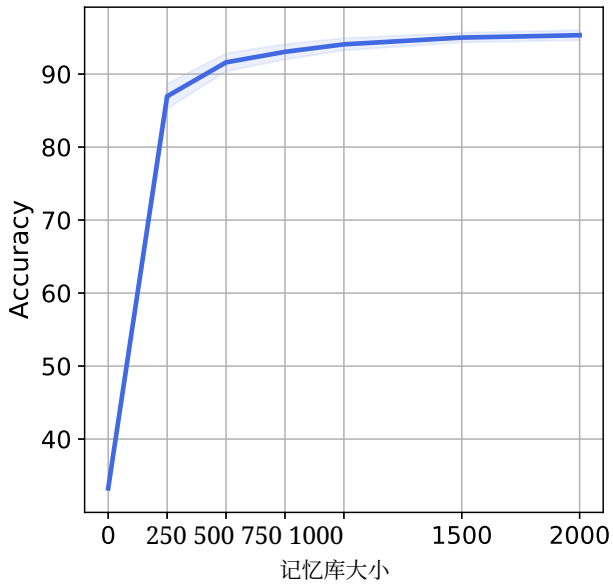

图 A6. 不同记忆库大小下，Split-MNIST 五个任务的平均测试准确率。横轴表示由全部五个任务共享的整个内存缓冲区的大小。准确率为五次不同运行结果的均值和标准差，这些运行使用了不同的随机种子。

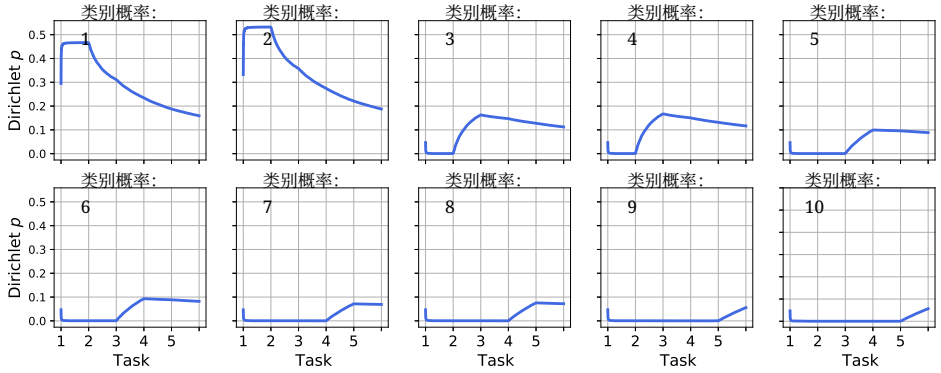

图 A7. ProtoCL 在 Split-MNIST 各任务中每个类别的狄利克雷参数 α_t 的变化。所有α_j 均展示了 10 个随机种子的结果，并附有 ± 1 标准误。训练结束时，所有类别的可能性大致相同，因为我们对所有类别进行了等量训练。

附录 *D.4.* 预测

为测试点 $\boldsymbol{x}^*$ 进行预测：

$$p(y^* = j|\boldsymbol{x}^*, \boldsymbol{x}_{1:t}, y_{1:t}) = p(y^* = j|z^*, \boldsymbol{\theta}_t) \tag{A28}$$

$$= \frac{p(\boldsymbol{z}^*|y^* = j, \boldsymbol{\theta}_t)p(y^* = j|\boldsymbol{\theta}_t)}{\sum_i p(\boldsymbol{z}^*|y^* = i, \eta_t)p(y^* = i|\boldsymbol{\theta}_t)} \tag{A29}$$

$$= \frac{p(y^* = j, \boldsymbol{z}^*|\boldsymbol{\theta}_t)}{\sum_i p(y = i, \boldsymbol{z}^*|\boldsymbol{\theta}_t)}, \tag{A30}$$

其中， $\boldsymbol{\theta}_t$ 是 $(\boldsymbol{x}_{1:t}, y_{1:t})$ 的充分统计量。

防止遗忘。 由于我们希望保留任务特定原型，在完成任务 $\mathcal{T}_t$ 的学习后，我们会从数据中选取一个小子集作为记忆库，以确保后验参数和原型不会发生漂移，参见算法 1。

附录 *D.5.* 实验设置

原型方差 Σ_ϵ 设为对角矩阵，每个原型的方差设为 0.05。原型先验精度 Λ_{yt}同样为对角矩阵，并以随机值初始化，再进行指数变换，以确保序贯更新中的协方差为半正定。参数 $\alpha_j \forall j$ 设为 0.78，该值通过在 MNIST 验证集上进行随机搜索得到。我们还允许在梯度更新步骤中学习 α_j，而不仅仅是在序贯更新步骤中学习（算法 1的第 4、5 行），参见图 A7，了解学习 Split-MNIST 过程中 α_j 在所有类别 j 上的变化。

对于 Split-MNIST 和 Split-FMNIST 基准测试，我们使用一个包含两个 200 单元层的神经网络，并采用 Adam 优化器训练 50 个 epoch。我们通过网格搜索调整学习率、丢弃率和权重衰减系数。嵌入维度设为 128。对于 Split-CIFAR10 和 Split-CIFAR100 基准测试，我们使用与 Pan 等人相同的网络 [22]，该网络由四个卷积层和两个线性层组成。我们使用 Adam 优化器，以 1×10^{-3} 的学习率对每个任务训练网络 80 个 epoch。嵌入维度设为 32。所有实验均在单个 GPU NVIDIA RTX 3090 上运行。

附录E. 序贯贝叶斯估计作为贝叶斯神经网络优化

我们将考虑图 A8A8所示图模型中的推断。我们的目标是在给定观测和前一时刻的贝叶斯神经网络权重的情况下，推断时间t 的最优贝叶斯神经网络权重 $\boldsymbol{\theta}^{*t}$。我们假设权重服从具有完全协方差的高斯后验，因此会对所有权重之间的相互作用进行建模。我们将考虑在线设置，即每次观察一个数据点$(\boldsymbol{x}_t, y_t)$，并且不假设学习过程中数据来自同一任务还是不同任务。

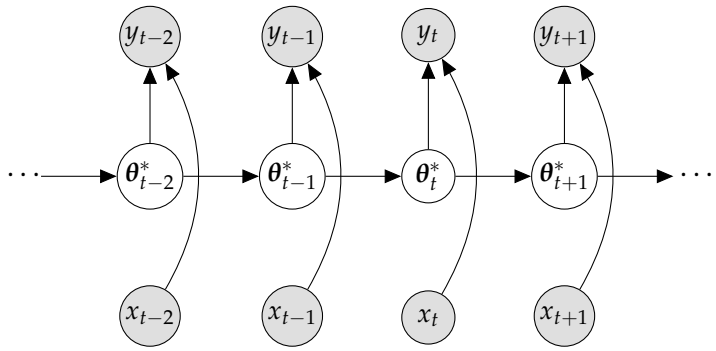

图 A8。 我们在第 5 节中进行推断所依据的图模型。灰色节点表示观测变量，白色节点表示潜变量。

我们将序贯贝叶斯推断问题设定为一个滤波问题，并借鉴 Aitchison [31], 的研究，该研究将神经网络优化视为贝叶斯序贯推断。我们作出一个合理的假设：权重服从具有完全协方差的高斯分布。由于无法对贝叶斯神经网络的完整协方差矩阵进行可行的推理，我们转而考虑第i个参数，并将最优估计 θ^*_{it} 的动态视为所有其他参数 $\boldsymbol{\theta}_{-it}$的函数。每个权重在函数上都依赖于其他所有权重。如果能够获得参数的完整协方差，我们就可以在给定其他所有权重 $\boldsymbol{\theta}_{-it}$取值的情况下，推断未知的最优权重 θ^*_{it} 。然而，由于无法获得完整协方差，另一种方法是根据 $\boldsymbol{\theta}_{-it}$ 的动态推理 θ^*_{it} 的动态，并假设权重的取值接近前一时间步 [32] 的取值，因此我们将该问题表述为一个动力系统。

考虑如下形式的二次损失：

$$\mathcal{L}(\boldsymbol{x}_t, y_t; \boldsymbol{\theta}) = \mathcal{L}_t(\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{\theta}^\top \boldsymbol{H}\boldsymbol{\theta} + \mathbf{z}_t^\top \boldsymbol{\theta}, \tag{A31}$$

我们可以通过简单的泰勒展开得到这一结果，其中$\boldsymbol{H}$ 是海森矩阵，假设它对所有数据点保持不变，但会随参数 $\boldsymbol{\theta}$变化。如果贝叶斯神经网络输出的形式为$f(\boldsymbol{x}_t;\ \boldsymbol{\theta})$，那么在 $\boldsymbol{\theta}_t$处求得的导数为$\boldsymbol{z}_t = \frac{\partial \mathcal{L}_t}{(\boldsymbol{x}\partial\boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t, y; \boldsymbol{\theta})}$。要构造权重的动力学方程，考虑单个数据点的梯度：

$$\frac{\partial \mathcal{L}_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\boldsymbol{H}\boldsymbol{\theta} + \mathbf{z}_t. \tag{A32}$$

如果考虑第i个权重的梯度，并将所有其他参数设为该 r
当前估计值：

$$\frac{\partial \mathcal{L}(\theta_i, \boldsymbol{\theta}_{-i})}{\partial \theta_i} = -H_{ii}\theta_{it} - \boldsymbol{H}_{-ii}^\top \boldsymbol{\theta}_{-it} + z_{ti}. \tag{A33}$$

当梯度设为零时，我们得到 θ_{it} 的最优值，记为 θ^*_{it}：

$$\theta^*_{it} = -\frac{1}{H_{ii}}\boldsymbol{H}_{-ii}^\top \boldsymbol{\theta}_{-it}. \tag{A34}$$

因为在$z_{ti} = 0$ 众数处。上面的方程表明，最优权重 θ^*_{it} 的动态取决于参数 $\boldsymbol{\theta}_{-it}$的所有其他当前值。也就是说， θ^*_{it} 的动态由权重 $\boldsymbol{\theta}_{-it}$的动态所支配。 $\boldsymbol{\theta}_{-it}$的动态是一个复杂的随机过程，取决于许多不同变量。由于推理

其动态难以处理，因此我们改为假设权重 $\boldsymbol{\theta}_{-it}$ 遵循离散化的奥恩斯坦–乌伦贝克过程，回复速度为 $\vartheta \in \mathbb{R}_+$，噪声方差为 η^2_{-i}:

$$p(\boldsymbol{\theta}_{-i,t+1}|\boldsymbol{\theta}_{-i,t}) = \mathcal{N}((1-\vartheta)\boldsymbol{\theta}_{-it}, \eta^2_{-i}), \tag{A35}$$

这意味着最优权重的动力学由以下公式定义

$$p(\theta^*_{i,t+1}|\theta^*_{i,t}) = \mathcal{N}((1-\vartheta)\theta^*_{it}, \eta^2), \tag{A36}$$

其中 $\eta^2 = \eta^2_{-i}\boldsymbol{H}^\top_{-ii}\boldsymbol{H}_{-ii}$。Aitchison [31]也采用了相同的假设。这假设动力学模型是简约的。结合我们的似然:

$$p(y_t|\boldsymbol{x}_t;\boldsymbol{\theta}^*_t) = \mathcal{N}(y_t; f(\boldsymbol{x}_t;\boldsymbol{\theta}^*_t), \sigma^2) \tag{A37}$$

其中 $f(\cdot, \boldsymbol{\theta})$ 是权重为 $\boldsymbol{\theta}$的神经网络预测。现在，我们可以通过对贝叶斯神经网络 [32] 进行线性化，并使用公式（A36）中的转移动力学，为最优权重 θ^*_i 定义一个线性动力系统。因此，我们能够使用类似卡尔曼滤波器的更新来推断最优权重的后验分布 [25]。由于动力学和似然均为高斯分布，先验和后验也均为高斯分布。为简化记号，我们省略索引 i，使得 $\theta^*_{it} = \theta^*_t$:

$$p(\theta^*_t|(\boldsymbol{x},y)_{t-1},\ldots,(\boldsymbol{x},y)_1) = \mathcal{N}(\mu_{t,\text{prior}}, \sigma^2_{t,\text{prior}}) \tag{A38}$$

$$p(\theta^*_t|(\boldsymbol{x},y)_{t},\ldots,(\boldsymbol{x},y)_1) = \mathcal{N}(\mu_{t,\text{post}}, \sigma^2_{t,\text{post}}) \tag{A39}$$

利用转移动力学和先验，我们可以得到闭式更新:

$$p(\theta^*_t|(\boldsymbol{x},y)_{t-1},\ldots,(\boldsymbol{x},y)_1) = \int p(\theta^*_t|\theta^*_{t-1})p(\theta^*_{t-1}|(\boldsymbol{x},y)_{t-1},\ldots,(\boldsymbol{x},y)_1)d\theta^*_{t-1} \tag{A40}$$

$$\mathcal{N}(\theta^*_t;\mu_{t,\text{prior}},\sigma^2_{t,\text{prior}}) = \int \mathcal{N}(\theta^*_t;(1-\vartheta)\theta^*_{t-1},\eta^2)\mathcal{N}(\theta^*_{t-1};\mu_{t-1,\text{post}},\sigma^2_{t-1,\text{post}})d\theta^*_{t-1}. \tag{A41}$$

对 θ^*_{t-1} 进行边缘化后，我们可以得到下一时间步先验分布的更新公式，如下所示:

$$\mu_{t,\text{prior}} = (1-\vartheta)\mu_{t-1,post} \tag{A42}$$

$$\sigma^2_{t,\text{prior}} = \eta^2 + (1-\vartheta)^{-2}\sigma^2_{t-1,\text{post}}. \tag{A43}$$

应用贝叶斯定理即可得到用于求取后验参数 μ_t、post 以及 σ^2_t、post 的更新公式:

$$\log\mathcal{N}(\theta^*_t;\mu_{t,\text{post}},\sigma^2_{t,\text{post}}) \propto \log\mathcal{N}(y_t; f(\boldsymbol{x}_t;\theta^*_t),\sigma^2) + \log\mathcal{N}(\theta^*_t;\mu_{t,\text{prior}},\sigma^2_{t,\text{prior}}), \tag{A44}$$

通过对贝叶斯神经网络进行线性化，使得$f(\boldsymbol{x}_t,\theta_0) \approx f(\boldsymbol{x}_t,\theta_0) + \frac{\partial f(\boldsymbol{x}_t;\theta^*_t)}{\partial\theta^*_t}(\theta^*_t - \theta_0)$并代入方程（A44），即可得到贝叶斯神经网络参数均值后验分布的更新方程:

$$-\frac{1}{2\sigma^2_{t,\text{post}}}(\theta^*_t - \mu_{t,\text{post}})^2 = -\frac{1}{2\sigma^2}(y - g(\boldsymbol{x}_t)\theta^*_t)^2 - \frac{1}{2\sigma^2_{t,\text{prior}}}(\theta^*_t - \mu_{t,\text{prior}})^2 \tag{A45}$$

$$\mu_{t,\text{post}} = \sigma^2_{t,\text{post}}\left(\frac{\mu_{t,\text{prior}}}{\sigma^2_{t,\text{prior}}} + \frac{y}{\sigma^2}g(\boldsymbol{x}_t)\right), \tag{A46}$$

其中 $g(\boldsymbol{x}_t) = \frac{\partial f(\boldsymbol{x}_t;\theta^*_t)}{\partial\theta^*_t}$，高斯后验方差的更新方程为:

$$\frac{1}{\sigma^2_{t,\text{post}}} = \frac{g(\boldsymbol{x}_t)^2}{\sigma^2} + \frac{1}{\sigma^2_{t,\text{prior}}}. \tag{A47}$$

根据更新后的方程，即公式（A46）和公式（A47），我们可以注意到，后验均值线性地依赖于先验和一个额外的、依赖于数据的项。这些方程与第 4 节中的滤波示例类似。因此，在上述某些假设下，贝叶斯神经网络应表现出类似的行为。如果存在任务数据不平衡，那么公式（A46）中的数据项将主导先验项，并可能导致遗忘先前任务。

参考文献

1. McCloskey, M.; Cohen, N.J. 联结主义网络中的灾难性干扰：序贯学习问题。载于《学习与动机心理学》；爱思唯尔：荷兰阿姆斯特丹，1989年；第24卷，第109–165页。2. French, R.M. 联结主义网络中的灾难性遗忘。《认知科学趋势》1999年，3，128–135。3. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; 等。克服神经网络中的灾难性遗忘。《美国国家科学院院刊》2017年，114，3521–3526。https://doi.org/10.1073/pnas.1611835114。4. MacKay, D.J. 反向传播网络的实用贝叶斯框架。《神经计算》1992年，4，448–472。5. Graves, A. 神经网络的实用变分推断。《神经信息处理系统进展》2011年，24，1–9。6. Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; Wierstra, D. 神经网络中的权重不确定性。载于《国际机器学习会议论文集》，PMLR，法国里尔，2015年7月6日至11日；第1613–1622页。7. Schwarz, J.; Czarnecki, W.; Luketina, J.; Grabska-Barwinska, A.; Teh, Y.W.; Pascanu, R.; Hadsell, R. Progress & compress：一种可扩展的持续学习框架。载于《国际机器学习会议论文集》，PMLR，瑞典斯德哥尔摩，2018年7月10日至15日；第4528–4537页。8. Ritter, H.; Botev, A.; Barber, D. 通过在线结构化拉普拉斯近似克服灾难性遗忘。《神经信息处理系统进展》2018年，31，1–11。9. Nguyen, C.V.; Li, Y.; Bui, T.D.; Turner, R.E. 变分持续学习。载于《国际学习表征会议论文集》，加拿大不列颠哥伦比亚省温哥华，2018年4月30日至5月3日。10. Ebrahimi, S.; Elhoseiny, M.; Darrell, T.; Rohrbach, M. 贝叶斯神经网络中的不确定性引导持续学习。载于《IEEE/CVF计算机视觉与模式识别会议研讨会论文集》，美国加利福尼亚州长滩，2019年6月16日至17日；第75–78页。11. Kessler, S.; Nguyen, V.; Zohren, S.; Roberts, S.J. 用于贝叶斯持续学习的层次印度自助餐神经网络。载于《人工智能中的不确定性会议论文集》，PMLR，线上，2021年7月27日至30日；第749–759页。12. Loo, N.; Swaroop, S.; Turner, R.E. 广义变分持续学习。载于《国际学习表征会议论文集》，埃塞俄比亚亚的斯亚贝巴，2020年4月26日至30日。13. Neal, R.M. 使用哈密顿动力学进行马尔可夫链蒙特卡罗。《马尔可夫链蒙特卡罗手册》；Chapman and Hall：美国纽约州纽约，2011年；第113–162页。14. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. 应用于文档识别的基于梯度的学习。《IEEE会刊》1998年，86，2278–2324。15. Zenke, F.; Poole, B.; Ganguli, S. 通过突触智能实现持续学习。载于《国际机器学习会议论文集》，PMLR，澳大利亚悉尼，2017年8月6日至11日；第3987–3995页。16. Hsu, Y.C.; Liu, Y.C.; Ramasamy, A.; Kira, Z. 重新评估持续学习场景：分类方法及强基线的必要性。arXiv 2018，arXiv:1810.12488。17. Van de Ven, G.M.; Tolias, A.S. 持续学习的三种场景。arXiv 2019，arXiv:1904.07734。18. van de Ven, G.M.; Tuytelaars, T.; Tolias, A.S. 三种增量学习类型。《自然机器智能》2022年，4，1185–1197。19. Chopin, N.; Papaspiliopoulos, O.《序贯蒙特卡罗导论》；施普林格：瑞士查姆，2020年；第4卷。20. Cobb, A.D.; Jalaian, B. 使用对称分裂扩展贝叶斯神经网络的哈密顿蒙特卡罗推断。载于《第三十七届人工智能中的不确定性会议论文集》，线上，2021年7月27日至30日；第675–685页。21. Izmailov, P.; Vikram, S.; Hoffman, M.D.; Wilson, A.G.G. 贝叶斯神经网络的后验分布究竟是什么样的？载于《国际机器学习会议论文集》，PMLR，线上，2021年7月18日至24日；第4629–4640页。22. Pan, P.; Swaroop, S.; Immer, A.; Eschenhagen, R.; Turner, R.; Khan, M.E.E. 通过对值得记忆的过去进行函数正则化实现持续深度学习。《神经信息处理系统进展》2020年，33，4453–4464。23. Dinh, L.; Sohl-Dickstein, J.; Bengio, S. 使用 Real NVP 进行密度估计。arXiv 2016，arXiv:1605.08803。24. Doucet, A.; De Freitas, N.; Gordon, N. 序贯蒙特卡罗方法导论。载于《序贯蒙特卡罗方法实践》；施普林格：德国柏林/海德堡，2001年；第3–14页。25. Kalman, R.E. 线性滤波与预测问题的新方法。《基础工程学杂志》1960年3月，82，35–45。26. Aljundi, R.; Lin, M.; Goujaud, B.; Bengio, Y. 面向在线持续学习的基于梯度的样本选择。《神经信息处理系统进展》2019年，32，1–10。27. Aljundi, R.; Kelchtermans, K.; Tuytelaars, T. 无任务持续学习。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，美国加利福尼亚州长滩，2019年6月15日至20日；第11254–11263页。28. De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; Tuytelaars, T. 持续学习综述：在分类任务中抵御遗忘。arXiv 2019，arXiv:1909.08383。

29. Wilson, A.G.; Izmailov, P. 贝叶斯深度学习与泛化的概率视角。神经信息处理系统进展**2020**，*33*，4697–4708。30. Ciftcioglu, Ö.; Türkcan, E. 通过卡尔曼滤波对前馈神经网络进行自适应训练；荷兰能源研究基金会 ECN：荷兰佩滕，1995年。31. Aitchison, L. 贝叶斯滤波统一了自适应与非自适应神经网络优化方法。神经信息处理系统进展 **2020**，*33*，18173–18182。32. Jacot, A.; Gabriel, F.; Hongler, C. 神经切线核：神经网络中的收敛与泛化。神经信息处理系统进展 **2018**，*31*，1–10。33. Thrun, S.; Mitchell, T.M. 终身机器人学习。机器人与自主系统 **1995**，*15*，25–46。34. Zeno, C.; Golan, I.; Hoffer, E.; Soudry, D. 使用在线变分贝叶斯的任务无关持续学习。*arXiv* **2018**，arXiv:1803.10123。35. Ahn, H.; Cha, S.; Lee, D.; Moon, T. 通过自适应正则化实现基于不确定性的持续学习。神经信息处理系统进展 **2019**，*32*，1–11。36. Farquhar, S.; Osborne, M.A.; Gal, Y. 径向贝叶斯神经网络：超越大规模贝叶斯深度学习中的离散支撑集。载于《国际人工智能与统计学会议论文集》，PMLR，线上，2020年8月26日至28日；第1352–1362页。37. Mehta, N.; Liang, K.; Verma, V.K.; Carin, L. 使用权重因子的贝叶斯非参数字典进行持续学习。载于《国际人工智能与统计学会议论文集》，PMLR，线上，2021年4月13日至15日；第100–108页。38. Kumar, A.; Chatterjee, S.; Rai, P. 用于持续学习的贝叶斯结构自适应。载于《国际机器学习会议论文集》，PMLR，线上，2021年7月18日至24日；第5850–5860页。39. Adel, T.; Zhao, H.; Turner, R.E. 使用自适应权重的持续学习（claw）。*arXiv***2019**，arXiv:1911.09514。40. Titsias, M.K.; Schwarz, J.; Matthews, A.G.d.G.; Pascanu, R.; Teh, Y.W. 使用高斯过程进行持续学习的函数正则化。载于 ICLR 论文集，埃塞俄比亚亚的斯亚贝巴，2020年4月26日至30日。41. Kapoor, S.; Karaletsos, T.; Bui, T.D. 用于持续学习的变分自回归高斯过程。载于《国际机器学习会议论文集》，PMLR，线上，2021年7月18日至24日；第5290–5300页。42. Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; Calderara, S. 通用持续学习的暗经验：一种强大而简单的基线方法。神经信息处理系统进展 **2020**，*33*，15920–15930。43. Benjamin, A.; Rolnick, D.; Kording, K. 在函数空间中测量和正则化网络。载于《国际学习表征会议论文集》，加拿大不列颠哥伦比亚省温哥华，2018年4月30日至5月3日。44. Henning, C.; Cervera, M.; D'Angelo, F.; Von Oswald, J.; Traber, R.; Ehret, B.; Kobayashi, S.; Grewe, B.F.; Sacramento, J. 用于持续学习的后验元重放。神经信息处理系统进展 **2021**，*34*，14135–14149。45. Swaroop, S.; Nguyen, C.V.; Bui, T.D.; Turner, R.E. 改进并理解变分持续学习。*arXiv***2019**，arXiv:1905.02099。46. Rudner, T.G.J.; Chen, Z.; Teh, Y.W.; Gal, Y. 贝叶斯神经网络中的可处理函数空间变分推断。载于《第36届神经信息处理系统会议（NeurIPS 2022）论文集》，美国路易斯安那州新奥尔良，2022年11月28日至12月9日。47. Rudner, T.G.J.; Smith, F.B.; Feng, Q.; Teh, Y.W.; Gal, Y. 通过序贯函数空间变分推断实现持续学习。载于《第38届国际机器学习会议论文集》，PMLR，线上，2022年7月18日至24日。48. Lavda, F.; Ramapuram, J.; Gregorova, M.; Kalousis, A. 使用生成模型进行持续分类学习。*arXiv* **2018**，arXiv:1810.10612。49. van de Ven, G.M.; Li, Z.; Tolias, A.S. 使用生成式分类器进行类别增量学习。载于《IEEE/CVF计算机视觉与模式识别会议论文集》，美国田纳西州纳什维尔，2021年6月20日至25日；第3611–3620页。50. Snell, J.; Swersky, K.; Zemel, R. 用于小样本学习的原型网络。神经信息处理系统进展 **2017**，*30*，1–11。51. Rebuffi, S.A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. ICARL：增量分类器与表征学习。载于《IEEE计算机视觉与模式识别会议论文集》，美国夏威夷州檀香山，2017年7月21日至26日；第2001–2010页。52. Harrison, J.; Sharma, A.; Finn, C.; Pavone, M. 无任务的连续元学习。神经信息处理系统进展 **2020**，*33*，17571–17581。53. Knoblauch, J.; Husain, H.; Diethe, T. 最优持续学习需要完美记忆，且是 NP难问题。载于《国际机器学习会议论文集》，PMLR，线上，2020年7月13日至18日；第5327–5337页。54. Petersen, K.B.; Pedersen, M.S. 矩阵手册；丹麦技术大学：丹麦灵比，2008年；第7卷，第510页。

免责声明／出版者说明：所有出版物中所含的声明、观点和数据均仅代表个别
作者及撰稿人的观点，并不代表 MDPI 和／或编辑的观点。MDPI 和／或编辑对因内容中提及的任何观点、方法、说明或产品而造成的人身伤害或财产损失不承担责任。