Few-shot Class Incremental Learning Survey

秦镛淳

2022 年 12 月 3 日

摘要

本文调研了目前较有影响力和较新的增量学习方法，并调研了增量学习在小样本领域的应用。

1 Introduction

增量学习是机器学习的研究热点之一, 以图像分类问题为例, 增量学习的目标是能够不断在系统中添加新的可以被分类的类别。初期的研究人员关注于 Task Incremental Learning (task-IL), 即在完成对新添加类别的训练后只在新类别上进行测试, 而不考虑在旧类别上的性能。这样的形式并不贴近现实中增量学习的应用场景，因而近年来研究人员更多的关注 Class Incremental Learning (class-IL), 在测试时候会在模型所见过的所有类别上进行测试, 而不仅仅在当前的若干类别上, 因此本文也主要关注 class-IL 领域的相关工作。

小样本学习 (Few-shot Learning) 是机器学习中另一个重要问题。虽然更多的数据往往能够有效提高机器学习系统的性能, 但受制于特定领域有限的数据量, 小样本学习也具有重要意义。近年一些研究将小样本学习和增量学习相结合, 提出了 Few-shot Class Incremental Learning (FSCIL) 的概念。

增量学习面对的最主要问题被称为灾难性遗忘 (catastrophic forgetting)，这是指在训练对新类的分类时，对旧有类别的分类准确率大幅下降，就好像是遗忘了对这些类别的分类能力。通常在 class-IL 中，数据集是以序列形式出现，并且对每个数据集，模型仅能见到一次，因此若不能较好的处理遗忘问题，会使得最终结果大幅偏向新类而失去对旧类的分类能力。此外另一个重要的问题是模型对于新类别的学习能力，既要保持对旧类别的记忆，又要能够学习到新的类别，这需要研究者进行权衡。通常在增量学习中，将 Fine-tunning 方法作为 baseline，因为这种方法是最初很自然的想法并且容易陷入遗忘，与之对比可以反映出模型在解决遗忘问题方面的进展。此外，有研究 [1] 提出，可以将联合训练作为增量学习系统的理论上限，即把新旧数据集中所有的数据收集起来，一起用于重新训练一个随机初始化的网络。这种方法显然可以避免遗忘，但是其巨大的计算开销使得它不可能作为一种真正实用的方法。

参考综述 [2] 的分类方法，可以将当前的增量学习方法大致分为三种路线：**1.** 基于正则化的方式 (Regularization-based)，通过对数据或模型做正则化，缓解灾难性遗忘。**2.** 基于重演的方法 (Rehearsal approaches)，通过保存部分旧类别的数据，使得在学习新类同时可以对旧类别进行复习，从而缓解遗忘。**3.** 基于误差校正的方法 (Bias-correction)，通过对输出的 bias 作矫正，缓解遗忘问题。本文在讨论 class-IL 和 FSCIL 的若干方法时也基本延续这种分类方法。

2 Class Incremental learning

2.1 Regularization approaches

基于正则化的方法主要特点是对模型施加一定的约束，使其不容易遗忘旧类别的信息。这一方法通常不要求额外存储旧类别的实例, 而是通过知识蒸馏使模型能够回顾旧类的信息。Learning without Forgetting (LwF)[3] 首次提出了基于知识蒸馏的方法, 它的损失函数由两部分组成: 交叉熵损失函数和蒸馏损失函数。其中交叉熵衡量模型在全部类别上的分类准确率，蒸馏则是用模型上一时刻（未添加新类别）对某一输入的 logits 作为监督信息，引导模型的优化。在实际中，LwF 需要保存上一时刻的模型参数，而不需要保存过去的训练数据，这种训练方式符合 class-IL 的一般形式（以序列形式输入的数据集），同时带来的额外存储空间需求也可以接受。作者的想法可以理解为，想要模型不发生遗忘，就需要让模型能够回顾过去的信息，而使用过去模型的 logits 做蒸馏就是一种在不存储旧类别数据条件下的合理方法。

Learning without Memorizing (LwM)[4] 对 LwF 做了一定的改进，在 LwM 中，注意力图也被作为蒸馏的监督信息用于更新模型。作者使用了 Grad-CAM 算法生成输入图片的自注意力图, 并且用教师网络的注意力训练学生网络。具体实现方面，就是在 LwF 的交叉熵损失函数和蒸馏损失函数的基础上再加上一个注意力蒸馏损失（Attention distillation loss）。

基于 LwF 的另一种改进是 Deep Model Consolidation (DMC)[5] 。作者观察到一种现象：旧类别使用的蒸馏监督实际上是一种较弱的监督信息, 而新类被直接使用 ground trurh, 是一种很强的监督信息，由此带来的不匹配是灾难性遗忘的主要原因，于是作者提出可以把两者的监督信息都统一为蒸馏信息。具体操作如下：**1,** 仅仅在新数据上训练一个分类器 **2,** 将新老分类器的 logits 拼接 (concat), 作为蒸馏的目标。蒸馏时用的无标签数据, 不属于以前的训练集 **3,** 对拼接后 logits 按照新类和老类分开做归一化，作者说这样可以使信息对称（属于同一个特征空间）。和 LwF 对比，主要不同在于蒸馏时候：LwF 训练新类时是对模型做 fine-tunning，而 DMC 的学生网络是随机初始化的。LwF 的新类直接拿 ground truth 做 label，DMC 对训练的 logits 做了归一化，减少了新类旧类监督信息之间的差异，一致性（对称性）更好。虽然新的分类器较好的挖掘了分类新类所需要的信息，但是融合时仍然只使用 logits 作为监督信息，很难说这部分知识有没有被利用起来（对比 LwM 使用了注意力图）。此外这一方法并不适合 FSCIL，因为很难在小样本数据集上训练出一个性能尚可的分类器。

2.2 Rehearsal approaches

基于重演的方法其主要特点在于需要开辟额外的一块内存用于保存具有代表性的旧类别样本。在增量学习中通常需要对旧类别做回顾以缓解遗忘问题，正则化方法通过施加蒸馏约束实现回顾, 而重演方法通过保存最具代表性的样本, 并在增量学习过程中输入模型训练来缓解遗忘。随着训练轮次的增加，新增类别数量也在不断提高，如果约束每一种类别有固定的保存样本数量则所需内存也会不断提高, 虽然这种提高是线性的, 但通常也不会被相关研究所采纳。最常见的一种方法是限制总的可用内存大小，在每轮训练时删除旧类别的部分样本，并为新类别添加样本。由此引出了重演方法的两大关键问题：筛选具有代表性的样本和对保留的数据集做更新。iCaRL[6] 是第一种基于重演的算法，其基于欧氏距离比较样本和所属类别原型向量的距离来作为其代表性的度量，其中类别原型由该类别所有类的特征向量的均值计算而来。iCaRL 设定一个固定大小集合（通常为 20000 个样本，后续实验也延续这一设定）用于保存旧的训练样例，采用的分类器是最邻近分类头, 直接比较模型输出和各个原型向量的距离。训练时分为三步：**1,** 用分类损失（CE）

和蒸馏损失对模型进行更新 **2,** 去除保存的旧类别中的多余样例 **3,** 构建新类别的保存样例。本文提出了一种数据集更新的方式 nearest-exemplars-mean (NEM)，也就是按照距离类原型的距离作为优先级，删除旧样例时优先去除较远的样本，添加新样本时优先加入离原型较近的样本，随着类别总数增加始终保持所有类别被保存下来的样本数一致。和 LwF 使用输出 logits 作为监督信息相比，iCaRL 保存下来的样本具有更强的监督性，这也是其性能超越 LwF 的可能原因之一。

IL2M[7] 提出一种观点，在基于重演的方法中继续使用 logits 作为蒸馏损失会使得模型性能降低，因为蒸馏学习要求教师网络经过充分的训练，但这在 class-IL 中通常难以保证。作者主要关注到增量学习中存在着不平衡的问题，提出了两条假设：**1,** 类别在首次见到时得到了最精确的建模 **2,** 模型倾向于对最近见过的类别给与较高的权重（容易过拟合到当前类别）。以在预留一个图片库保存训练图片的同时，作者也把模型首次见到某类别时其原型向量记录下来，在训练过程中用这一信息作为监督，因为在后续训练中原型向量会改变，首次训练时原型向量代表最精确建模（假设 1)。输出时候因为模型会给予新类别较高权重，因此直接对新类别的输出结果做加权将其减少，所减少的权重是以往和当前类别在当时全部原型中所占比重的比值。作者将这种新旧类别之间的权重差异称作模型的偏差 (bias)，这种偏差作者认为是系统性的，因此可以通过计算新旧类别的比重加以消除。

2.3 Bias-correction approaches

End-to-End Incremental Learning (EEIL)[8] 是较早提出增量学习中误差校正概念的算法，作者观察到无论是 LwF 还是 iCaRL，在训练以后模型还是会倾向于给与新加入类别较大的置信度，于是作者提出了 balanced fine-tunning 方法，在正常训练以后，用相同数量的新旧类 finetunning 模型，可以纠正在输出上的偏差。

BiC[9] 也是一种基于偏差矫正的方法，作者观察到增量学习中新旧类别存在严重的 bias，提出假设，bias 来源于两方面：**1,** 新类和旧类数量上的巨大差异 **2,** 随着类别的增多，相似的类别也在增多，模型很难分辨其中的差异。作者设计了实验，验证了 bias 主要来源于最后一层全连接层，于是很自然地想到在输出后面再加上一层全连接，按照正常的分类损失 + 蒸馏损失训练模型以后，将除了最后的线性层以外的参数全部冻结，单独训练最后的线性层。和 IL2M 类似地，BiC 也是希望通过一个线性变换纠正新旧类别之间的 bias，区别在于 BiC 的线性变换由可学习的参数组成，而 IL2M 的线性变化是一种手工指定的变化，其实是一种先验知识。

3 Few-shot class-IL

4 Future Work

目前主要有两种思路：

CNN+ 拓扑结构，采用原型分类头。每个类可以对应为一个原型向量，将若干原型向量视作一个图，在增量学习过程中保存拓扑结构。手工设计的拓扑度量不一定会很好，也许可以考虑使用图神经网络，相当于是对图做特征抽取。这种方法的好处在于增量学习过程中可以保持参数量完全不变，用原型向量之间的拓扑结构来解释遗忘问题感觉更为合理。

CNN+linear+sofamax 分类头，这种方法在原型学习过程中不可避免的会增加线性层的参数，同时因为用到 softmax 函数，所以新增类可能会影响到之前类别的置信度计算，相比而言基于拓

扑结构的增量学习较好的避免了这一问题。目前想到的方法是用旧类学出来的特征组合成分类新类所需的特征，然后考虑如何将新旧特征做融合 (类似于 DMC 的方法)。

参考文献

[1] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12183–12192, 2020.

[2] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: survey and performance evaluation on image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[3] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[4] P. Dhar, R. V. Singh, K.-C. Peng, Z. Wu, and R. Chellappa, "Learning without memorizing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5138–5146, 2019.

[5] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, and C.-C. J. Kuo, "Class-incremental learning via deep model consolidation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1131–1140, 2020.

[6] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

[7] E. Belouadah and A. Popescu, "Il2m: Class incremental learning with dual memory," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 583–592, 2019.

[8] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 233–248, 2018.

[9] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 374–382, 2019.