南京大学

研究生毕业论文
(申请硕士学位)

论文题目 基于原型的类增量学习算法研究

作者姓名 毛乐坤

学科、专业名称 计算机技术

研究方向 人工智能

指导教师 申富饶教授，吴楠副教授

2021年6月1日

学　　号：MF1833046

论文答辩日期：2021 年 5 月 20 日

指 导 教 师：　　**(签字)**

Prototype-based Class Incremental Learning

by

Lekun Mao

Supervised by

Professor Furao Shen,Associate Professor Nan Wu

A dissertation submitted to
the graduate school of Nanjing University
in partial fulfilment of the requirements for the degree of
MASTER
in
Computer Technology

Department of Computer Science and Technology
Nanjing University

May , 2021

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：基于原型的类增量学习算法研究

计算机技术 专业 2018 级硕士生姓名：毛乐坤

指导教师(姓名、职称)：申富饶教授，吴楠副教授

摘　要

随着人工智能的迅猛发展，科研工作者们发现深度神经网络在图片分类、语音识别等问题上有很好的应用效果。例如 2012 年，在有着计算机视觉“世界杯”之称的 ImageNet 图像分类竞赛中，Geoffery E.Hinton 等人凭借卷积神经网络 AlexNet 以超过第二名近 12% 的准确率一举夺得该竞赛冠军。然而，在目前的分类模型训练方法中，人们需要预先收集所有类别的数据，这导致分类模型无法像人类一样做到持续性学习。为了让模型能从动态的环境中集成新的数据，来实现增量式分类学习，需要克服的一个难题就是灾难性遗忘问题。但是目前在类增量学习领域，算法要么无法很好平衡旧知识的记忆和新知识的学习，要么未做到对采样数据的有效利用。本文从网络结构和训练数据两个方面，提出了三种基于原型的类增量学习算法。本文的成果主要体现在如下几个方面：

（1）**提出了基于原型的类增量学习算法 PCRC**。该模型利用原型向量来作为每个类别的代表点，使得提取的特征具有类内紧凑和类间分离的特点，从而降低多个分头网络在预测阶段相互混淆的程度。并且，该模型设计可以直接利用旧数据来提高性能，且无须做任何的修改。

（2）**提出了基于半监督学习的类增量学习算法 SS-PCRC**。在 PCRC 模型中，新类数据既用来进行旧知识的记忆，同时还要进行新知识的学习，这导致模型无法达到一个平衡——要么因为记忆权重较大，导致模型无法进一步学习新类；要么由于学习权重较大，导致明显的遗忘问题。为了缓解数据的不平衡问题，受半监督学习的启发，本文使用无标签数据侧重旧知识记忆，使用新类数据侧重新知识的学习。通过这种方式，进一步增强模型增量式学习的能力。实验表明，该模型只需要少量的辅助数据就能有明显的性能提升。

（3）**提出了平衡化的类增量学习算法 BPCRC**。新类数据中的不同样本在模型记忆旧类知识和学习新类知识方面起到的作用程度不同。我们把数据为关键性数据和非关键性数据，并且强化模型在关键性数据上的学习，减小模型在其他非关键性数据上的参数调整，从而缓解模型在增量学习过程中出现大幅度的参数调整问题。

（4）**模型在水声目标识别场景中的应用**。最后，我们在一个水声目标识别项目中使用 PCRC 算法来解决模型增量式分类舰船噪声的子问题，在采集的实际数据中，通过与 LwF.MC 算法比较，有力地验证了 PCRC 算法在落地方面的优越性。

关键词： 灾难性遗忘；类增量学习；原型

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Prototype-based Class Incremental Learning

SPECIALIZATION: Computer Technology

POSTGRADUATE: Lekun Mao

MENTOR: Professor Furao Shen,Associate Professor Nan Wu

ABSTRACT

With the rapid development of artificial intelligence, researchers have found that deep neural network has a good application in the field of image classification, speech recognition and other issues. In 2012, for example, in the ImageNet image classification competition known as the "World Cup" of computer vision, Geoffrey E. Hinton and others won the championship with the accuracy of nearly 12% over the second place with convolutional neural network AlexNet. However, in the current classification model training method, people need to collect data of all categories in advance, which makes the classification model unable to achieve continuous learning like humans. In order to integrate new data from the dynamic environment and realize incremental classification learning, the **catastrophic forgetting** problem needs to be overcome. However, in the current field of class incremental learning, algorithms either cannot balance the memory of old knowledge and the learning of new knowledge well, or they fail to make effective use of sampled data. In this paper, three prototype based class incremental learning algorithms are proposed from two aspects of network structure and training data:

(1) A prototype based class incremental learning algorithm PCRC is proposed. The model uses the prototype vector as the representative point of each category, so that the extracted features have the characteristics of compactness and separation between classes, thereby reducing the degree of confusion among multi-head network in the prediction stage. Moreover, the network design can directly use old data to improve performance without any modification.

(2) A class incremental learning algorithm SS-PCRC based on semi-supervised learning is proposed. In the PCRC model, the data of new categories are used not only to memorize the old knowledge, but also to learn the new knowledge, which makes the model unable to achieve a trade-off ——either because the memory weight is large, the model can not further learn new classes; or because the learning weight is large, it leads to obvious forgetting problem. In order to alleviate the problem of data imbalance, inspired by semi-supervised learning, this paper uses unlabeled data to focus on the memory of old knowledge, and uses new class data to focus on the learning of new knowledge. In this way, the model's ability is further enhanced. Experiments show that the model only needs a small amount of auxiliary data can have significant performance improvement.

(3) A balanced class incremental learning algorithm BPCRC is proposed. Different samples in new class data play different roles in model memory of old class knowledge and learning of new class knowledge. We take the data as key data and non critical data, and strengthen the learning of the model on the key data, reduce the parameter adjustment of the model on other non critical data, so as to alleviate the large-scale parameter adjustment problem in the incremental learning process.

(4) The application of the model in underwater object recognition scene. Finally, we apply PCRC algorithm to an underwater acoustic recognition project to solve the sub problem of incremental classification of ship noise. In the actual data collected, by comparing with the LwF.MC algorithm, it strongly verifies that the PCRC algorithm is superior in landing.

KEYWORDS: Catastrophic Forgetting, Class Incremental Learning, Prototype

目　　次

插图清单

附表清单

第一章　绪论

1.1　类增量学习问题

近几年，深度神经网络的应用得到了快速发展。尽管科研人员通过各种各样的网络结构、损失函数设计，来提高网络的分类精度，但是始终面临一个问题：所有类别的训练数据必须全部预先准备完毕，才能训练出一个满足要求的模型。在这种情况下，所有分类类别都是事先已知的，并且也意味着我们可以同时以任意顺序访问所有类别的训练数据，而这样的条件在实际情况中不成立的。随着计算机视觉更加智能化，我们需要更灵活的方式来应对实际场景中数据的动态性，顺序性。这也意味着，当新类别的训练数据到来时，模型应该能够继续学习，并具有区分所有已知类别的能力。我们把这种学习方式称为类增量学习。

随着人工智能的继续发展，科研人员致力于让部署到智能体上的算法实现更高程度的智能化。在分类场景下，这意味着智能体能够不断根据新的学习资料更新自己的模型，做到在保持之前的分类能力基础之上，增加新类别的学习能力。这个问题便是类增量学习领域需要解决的目标。类增量学习方法赋予智能体不断学习分类的能力，让其可以根据自己周围环境中遇到的新的物体类别，来更新自己的模型。例如智能相册应用，类增量学习可以支持用户自定义感兴趣的类别，在用户逐渐向自己新建类别添加图片的过程中学习到了分辨该类别的能力。这就使得相册模型除了支持软件发布者预先设定的粗粒度的类别，也能满足不同用户的个性化需求。可以看到，类增量学习这一人工智能科研课题，在未来具有一定的商业价值。

1.2　类增量学习算法分类

每当模型遇到新的类别时，从头开始重新训练模型是十分昂贵的。首先是存储成本，对于以后越来越多的数据，如果要存储下来供模型未来使用，那么存储成本就十分巨大。另外，对于模型的训练而言，又要耗费相当多的 CPU，GPU

计算资源。考虑到目前的模型参数中实际上保存了区分之前类别的知识，那么我们可以在预训练模型的基础上，通过算法来实现模型参数增量式调整。参考 Rebuffi[1] 对该研究课题的定义，类增量学习算法需要满足如下三个条件：

1. 对于流数据中在不同时间段出现的不同类别的样本，模型可以一直进行训练；
2. 对于迄今为止出现的所有类别，模型能训练出一个具有区分能力的分类器；
3. 模型训练的计算成本以及存储成本的增加，相对于类别数目的增加，要十分缓慢。

类增量学习分类器的学习过程如图 1-1所示，分类器可以从流数据中连续地学习，并能够区分到目前为止分类器见过的所有类别。

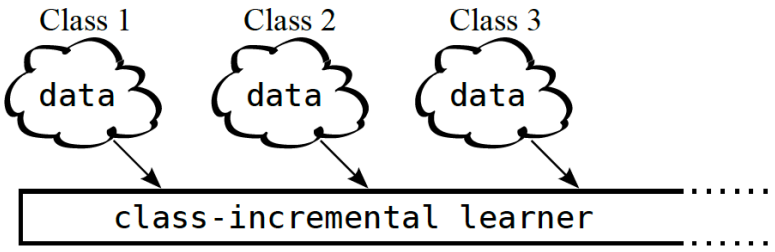


图 1-1: 类增量学习分类器的训练过程示意图[1]

在增量学习过程中，模型在存储之前学到的类别信息的同时，还要进行参数更新，来学习新的类别知识。为了得到一个满足条件 2 的分类器，科研人员探究了多种思路，从旧样本的使用，模型结构的设计，辅助数据的使用等角度来划分类增量学习算法，区分方法见表 1-1。

表 1-1: 增量学习算法分类

旧样本	辅助数据	分类器网络结构
不使用/使用/样本伪造	使用/不使用	分头网络/单头网络

以旧样本的使用情况划分有 3 类算法：完全不使用旧样本的算法，对旧样本进行采样的算法，以及使用 GAN 等生成式模型来构造假旧样本的算法；以是否使用辅助数据也可以分为两类算法；以模型中使用的分类器网络结构划分，又可以分为单头网络模型和多头网络模型。本文将重点放在对网络结构的改进，以及更高效利用辅助数据来优化模型对旧知识的记忆和对新知识的学习。

1.3 当前研究现状

在类增量学习的研究中，有一个需要解决的关键性问题——灾难性遗忘[2]。即已训练好的神经网络在学习新类别的时候，很容易出现对于旧类别的预测性能的显著降低的问题。该问题出现的主要原因在于网络向新数据拟合的过程中，很多与旧类别紧密相关的参数被调整了[3]。灾难性遗忘问题的一般解决方案是使用回放方法[4]———在网络学习新的输入数据时，可以利用之前学习类别的部分样本来进行巩固模型对旧知识的记忆。回放方法可以大大减轻灾难性遗忘现象，但却需要额外的存储空间和网络训练时间。

除了使用回放方法，近几年一些研究者们提出了其他的方案[3,5-6] 来解决灾难性遗忘问题，并且在未使用旧类样本的情况下获得不错的结果。但是，这些模型主要集中于任务增量学习。任务增量学习中不同的任务可以很容易进行划分，并且在预测阶段可以根据任务的描述符来分辨。在之前的研究中[7]，作者清晰地描述了任务增量学习与类增量学习的区别。实际上，类增量学习可以视为一种特殊的任务增量学习问题，其中学习每一批输入数据都是一项任务，但是在测试时任务描述符是未知的——这也意味着对于类增量学习，其预测机制与任务增量学习有很大不同。有研究者[8] 根据是否提供任务描述符将增量学习的评估设置分为单头设置和多头设置。实验结果还表明，任务增量学习的方法通常在多头设置中效果很好，而在单头设置中评估时往往会失败。另一方面，目前很多类增量学习的最新方法[1,9] 仍然很大程度上依赖于回放方法。

根据使用旧类数据的方法，这些方法又可以分为两类。第一类使用样本集来存储在每个训练期间采样的样本。iCaRL[1] 利用深度神经网络来学习特征表示和用于分类的 NME（Nearest-Mean-of-Exemplars）方法。End-to-End[9] 提出了一个端到端的增量学习网络来同时学习特征表示和分类器。Mu 等人[10] 使用低纬度的矩阵素描来表示之前的数据，而这些方法都需要一定的存储空间来缓存样本集。另一类方法利用生成模型来生成伪训练样本，通过深度生成重放（DGR）[11] 训练生成对抗网络[12] 模型来生成辅助训练数据，以帮助分类模型的训练。在另一项研究中[7]，作者通过利用变分自动编码器[13] 和网络蒸馏[14] 对 DGR 进行了修改。这些方法不需要空间来缓存样本集，但是在训练分类模型之前，它们必须等待生成模型生成伪数据。

总的来说，大多数现有类增量学习的方法依赖于回放方式——利用样本池来缓存旧类数据，或者使用生成式模型来构造伪样本。最近提出了深度模型固化算法 DMC[15]，该算法使用无标签数据进行知识迁移[16]。由于该算法缓解了新旧数据之间的不平衡问题，模型可以更好地整合记忆和学习。

1.4 本文研究内容

为了缓解类增量学习过程中的灾难性遗忘问题，本文分析了单头网络中存在的 softmax 抑制问题，并且提出了基于原型的类增量学习算法。本文的方法能在类增量学习的过程中较好平衡旧知识的记忆与新知识的学习，并通过在实际项目中应用，体现了算法的有效性。本文主要研究内容如下：

（1）提出了基于原型的类增量学习算法 PCRC（Prototype-based Classification and Response Consolidation)。该模型利用原型向量来作为每个类别的代表点，使得提取的特征具有类内紧凑和类间分离的特点，从而降低多个分头网络在预测阶段相互混淆的程度。并且，该模型设计可以直接利用旧数据来提高性能，且无须做任何的修改。

（2）提出了基于半监督学习的类增量学习算法 SS-PCRC（Semi-supervised PCRC)。在 PCRC 模型中，新类数据既用来进行旧知识的记忆，同时还要进行新知识的学习，这导致模型无法达到一个平衡——要么因为记忆权重较大，导致模型无法进一步学习新类；要么由于学习权重较大，导致明显的遗忘问题。为了缓解数据的不平衡问题，受半监督学习的启发，本文使用无标签数据侧重旧知识记忆，使用新类数据侧重新知识的学习。通过这种方式，进一步增强模型增量式学习的能力。实验表明，该模型只需要少量的辅助数据就能有明显的性能提升。

（3）提出了平衡化的类增量学习算法 BPCRC(Balanced PCRC)。新类数据中的不同样本在模型记忆旧类知识和学习新类知识方面起到的作用程度不同。我们把数据为关键性数据和非关键性数据，并且强化模型在关键性数据上的学习，减小模型在其他非关键性数据上的参数调整，从而缓解模型在增量学习过程中出现大幅度的参数调整问题。

（4）模型在水声目标识别场景中的应用。最后，我们在一个水声目标识别项

目中使用 PCRC 算法来解决模型增量式分类舰船噪声的子问题，在采集的实际数据中，通过与 LwF.MC 算法比较，有力地验证了 PCRC 算法在落地方面的优越性。

1.5 本文组织结构与章节安排

全文共有六章，第一章为绪论部分，主要介绍了类增量学习问题的提出背景和当前研究现状；第二章介绍类增量学习领域的背景知识和相关工作；第三章提出了类增量学习算法 PCRC；第四章使用了回放方法对 PCRC 算法进行增强，并提出了类增量学习算法 SS-PCRC 和 BPCRC；第五章使用 PCRC 算法来解决模型增量式分类水声的子问题。第六章对类增量学习研究方向进行总结与展望。

第二章　相关工作

本章首先介绍在类增量学习领域要解决的关键性难题——灾难性遗忘，并进一步介绍与之相关的 softmax 抑制问题。接下来，本文介绍基于原型的特征提取器，作为本文提出算法的背景知识。然后本文对该领域的研究做整体上的叙述，并介绍算法性能评价方法。

2.1　灾难性遗忘问题

灾难性遗忘问题指的是在顺序学习的过程中，人工神经网络在学习新信息时完全忘记先前学习的信息。灾难性遗忘问题是一个长期存在的问题，该问题不仅存在与类增量学习领域，同时也存在于任务增量学习[17]和强化连续学习[3]领域。

如图 2-1，初始阶段，模型只负责对猫、狗进行二分类，对测试集中的一张狗图片进行测试，发现 sofmax 层得到的分数向量中狗类别的分数很高；如图 2-2，模型增加鱼、鸟两个类别之后，对之前旧类狗再次作预测，发现 sofmax 层在新类别上有更高的分数。这就意味着，在增量学习过程中，模型对之前学习的猫、狗类别已经产生了遗忘现象。

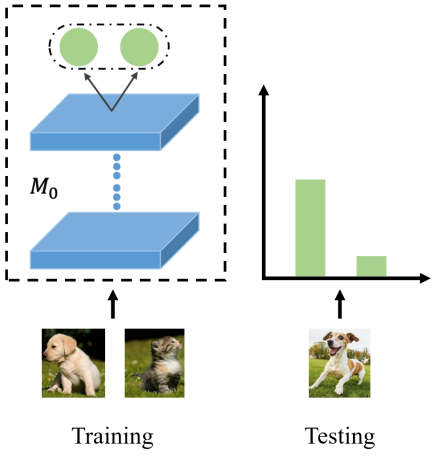


图 2-1: 初始阶段：使用猫和狗训练模型，然后对另一张狗的图片进行预测。

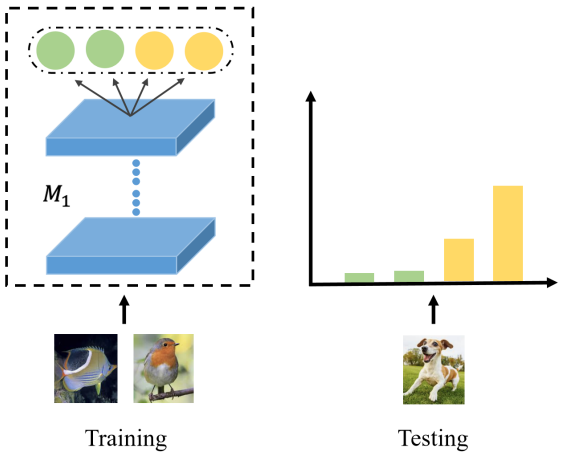


图 2-2: 增量学习阶段：模型增加了鱼、鸟两个类别之后，再对同一张狗图片进行预测。

2.2 softmax 抑制问题

为了缓解灾难性遗忘问题这一关键性难题，人们对于增量学习的评估设置进行了探索，将传统的单头设置修改为多头设置。单头设置和多头设置之间的本质区别在于任务描述符在测试时是否可用，另一个区别是网络体系结构。在以前的研究中，如果评估方式为单头设置，那么所有输出概率由同一个的 softmax 层给出，而在多头设置中，每个任务都有自己的输出头部。之前的研究表明，任务增量学习的方法通常在多头设置中效果很好。为了在类增量学习领域更深入探究两种评估方式，从而更好地解决灾难性遗忘问题，我们对单头设置进行了定量分析，发现了在单头设置中会出现 sofmax 抑制问题。

softmax 抑制问题指的是，随着网络在新数据上不断训练，网络在旧数据上进行预测得到的分数分布会出现向新类别倾斜的现象。出现这一现象的原因主要有两点（1）模型未使用回放方式进行训练，那么也就意味着模型无法再访问到旧类数据；（2）新旧类都在同一个 softmax 层进行训练。

我们使用了 MNIST 数据集进行了探索实验，以验证此问题。我们首先在前两个类（数字 0，1）中训练 CNN 模型[18]。然后，我们扩展 softmax 的输出维度，并使用后两个类（数字 2，3）继续训练网络。并且在后两个类的训练过程中，模型无法再访问前两个类。除了在最终分类层上进行简单的微调外，我们还尝试使用 EWC 和 LwF 方法来防止网络忘记旧类。训练结束后，我们在所有四个类别上测试模型，包括旧的两个类别和新的两个类别。最终 softmax 层的四个位置上的平均输出概率如图 2-3所示，无论是 Fine-tuning、EWC 还是 LwF 方法，在增量学习之后，模型在后面两个类的预测分数的平均值显著高于前面两个类。由于在新数据的训练过程中，网络不断向新类别进行拟合，所以网络会在新类别上有更高的响应；另外，由于 softmax 层得到的分数和为 1，那么由于新类别上得分的提高，必然导致旧类别上分数的降低。

进一步，如果我们不直接在原来的 softmax 增加输出单元，而是将新的输出单元组织为单独的 softmax 层，则训练网络等效于在隐藏层之上训练新的线性分类器。以此方式，原来 softmax 层对旧类别的预测能力就不会受到 softmax 抑制问题的影响。另外，网络可以根据任务描述符在多个 softmax 层的输出之间进行选择。当任务描述符不可用时，我们在测试时可以将所有的输出合并为最终

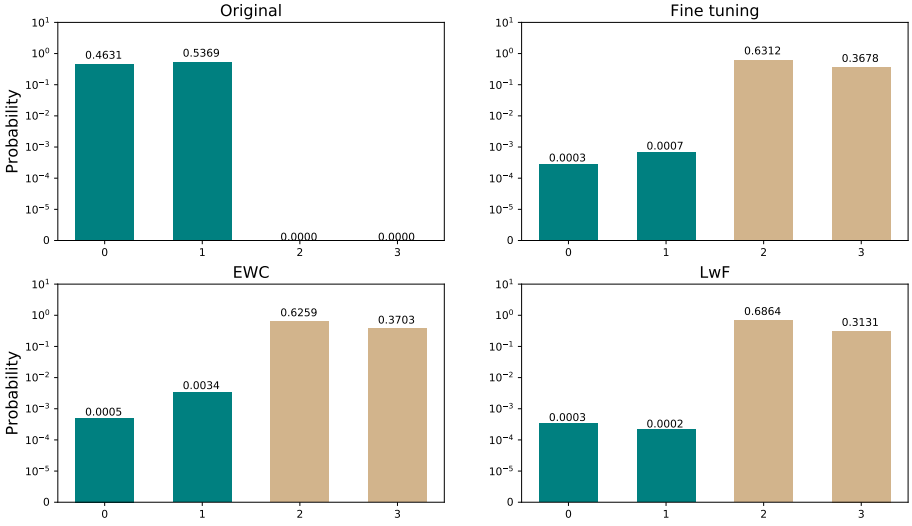


图 2-3: MNIST 前 4 个类的平均分数

预测向量，并选择具有最高输出概率的类别作为最终的预测结果。在这种情况下，如果网络经过充分的训练，则相应的输出层会在正确类别上产生更高的分数，而其他类别的得分相对较低，那么网络的预测性能依然可以得到保证。

2.3 基于原型的特征提取器

为了适应类增量学习任务，尤其是在预测阶段没有任务描述符的情况下，缓解多个分类器之间的相互混淆问题，那么就需要我们提取的特征更加具有区分性。也就是同一个类中的样本提取得到的特征分布要足够紧凑——类内聚敛性，而不同类的样本提取的特征相距要足够远——类间分离性。

而对于传统的基于 softmax 训练的特征提取器，提取的特征并不满足上面所说的任何一个。如图 2-4，基于 softmax 训练的 CNN 网络特征提取能力很差，由于分界面是超平面，那么如果样本的特征恰好落到分界超平面附近，很容易出现误分类的问题。在类增量学习场景中，特征提取器是增量式学习的，类别之间的特征就更加难以区分，导致分类错误。

目前，为了训练出区分性更强的特征提取器，研究者们做了很多工作。比如在 2016 年提出的 center loss[20]，即在训练 CNN 时，为每一个类别提供一个类别中心，最小化 min-batch 中每个样本与对应类别中心的距离，这样就可以达到缩小类内距离的目的。如图 2-5所示，图中每个白色圆点代表一个类的中心点，而

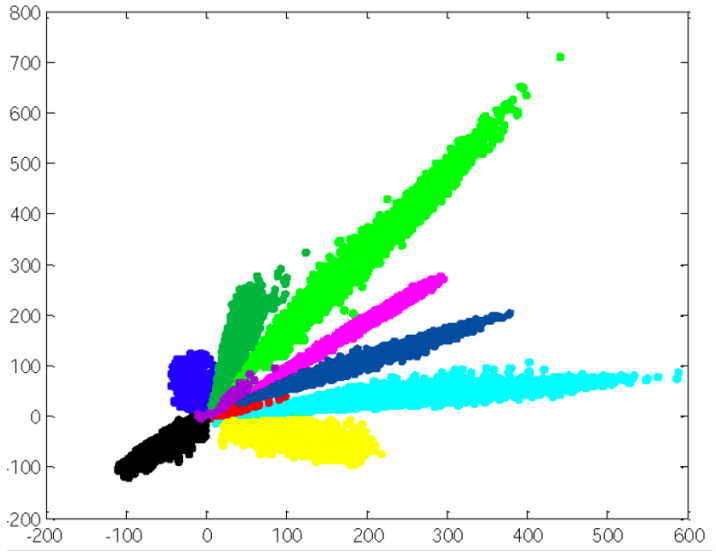


图 2-4: 使用 softmax 层的传统 CNN 网络在 MNIST 上提取特征的可视化图，不同颜色表示不同的类别[19]。

每个类的特征紧密分布在对应的中心点附近，随着 center loss 对应的权重系数 λ 的增大，特征的区分性也越来越高。

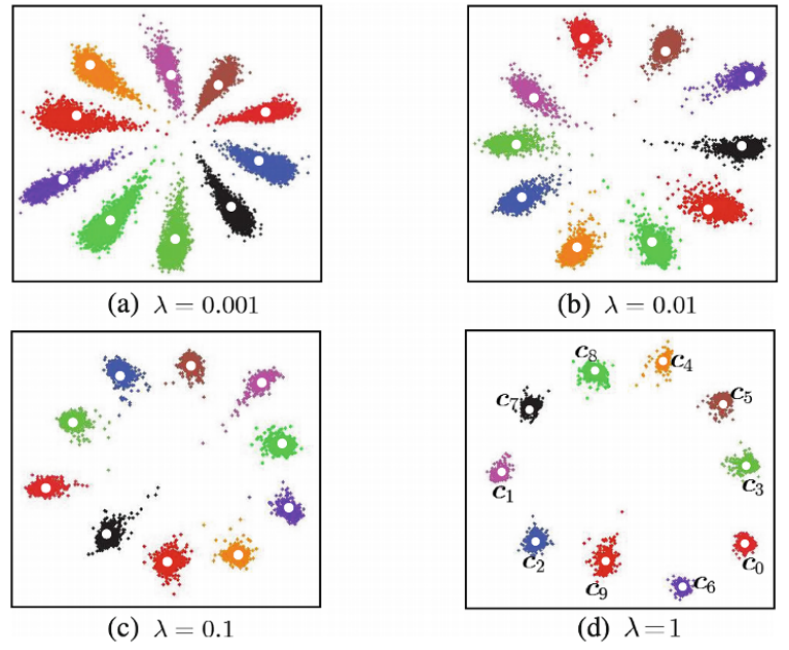


图 2-5: 在 MNIST 上使用 center loss 联合 softmax loss 训练得到的特征分布图[20]

但是，基于 center loss 进行网络训练有个很明显的缺陷是每个类的中心点无法由 CNN 直接学习得到，而是需要按照 Yandong Wen[20] 等人提出的特征学习算法预设的规则进行中心点学习。

为了实现直接从训练数据中学习到每个类的表示向量，Hong-Ming Yang[19]

等人提出了使用基于原型分类的方式，在网络通过梯度下降进行参数学习的过程中直接学习到了具有表示语义的原型向量[21]。如图 2-6，通过使用基于原型的分类器训练得到的特征分布依然满足特征的类内聚敛，类间分离的性质。

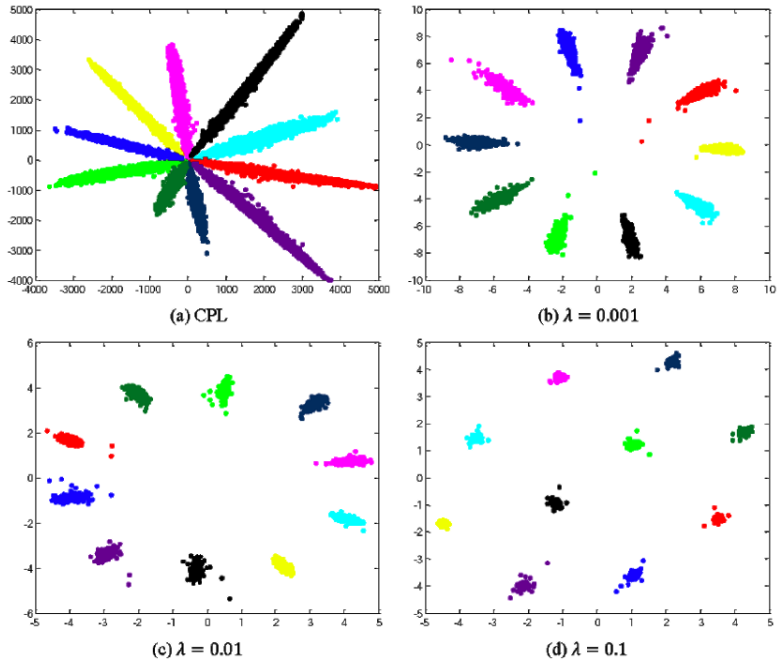


图 2-6: 在 MNIST 上使用基于原型的损失函数训练得到的特征分布图[19]

更近一步，Hong-Ming Yang 等人在 MNIST 上简单验证了基于原型训练的网络具有较好的对未知类识别[22] 的能力，而这个性质在类增量学习可以有效缓解预测阶段头部之间相互混淆的问题。

为了克服灾难性遗忘问题，研究者们在类增量学习领域提出了很多算法，这些方法主要为基于正则化的方法和基于回放的方法。接下来，本文分这两类对类增量学习研究做整体介绍，并描述其克服灾难性遗忘问题的思路。

2.4 基于正则化的方法

基于正则化的方法将正则化项与分类损失结合起来，以减轻灾难性遗忘。一些方法对网络中每个参数的权值进行正则化，并估计其重要性 [3] [23]，而另一些人则关注记忆特征表示的重要性[6] [24]。这些方法中的大多数是在任务增量学习领域提出，并被其他工作借鉴[25]。接下来，我们对基于正则化的方法分类介绍。

（1）**权重正则化**。这种方法的重点是防止与旧任务相关的权重发生漂移。在学习每个任务之后，方法会评估网络中每个参数（假设是独立的）的先验重要

性，在训练新任务时，每个参数的重要性都用来惩罚对它们的更改。也就是说，除了交叉熵分类损失外，这些方法还引入了额外的损失，其计算见公式 2-1

$$\mathcal{L}_{\text{reg}}\left(\theta^{t}\right)=\frac{1}{2}\sum_{i=1}^{\left|\theta^{t-1}\right|}\Omega_{i}\left(\theta_{i}^{t-1}-\theta_{i}^{t}\right)^{2} \tag{2-1}$$

其中 θ_i^t 指的是网络当前正在训练的权重 i，θ_i^{t-1} 指的是在任务 $t-1$ 训练结束之后权重 i 的取值，$\left|\theta^{t-1}\right|$ 为网络的权重数量，Ω_i 表示权重 i 的重要性。

Kirkpatrick 等人[3] 提出了弹性权重固化（EWC），并使用经验 Fisher 信息矩阵的对角近似来计算 Ω_i。然而，这种方法只能在学习每个任务后，最小限度地估计模型参数的重要性，而忽略了这些参数在权重空间中沿学习轨迹受到的影响。Liu 等人[26] 通过参数空间变换，来提供更好的 Fisher 信息矩阵，从而改进 EWC。在训练过程中，模型必须通过固定参数进行扩展，这不会增加网络的参数，但会产生额外的计算和存储成本。

与此相反，Zenke 等人[5] 提出了路径积分法，该方法沿着整个学习轨迹在线累积每个参数的变化。正如作者所指出的，批量更新权重可能导致高估参数的重要性，而从预训练模型开始可能导致低估参数的重要性。为了解决这个问题，记忆感知突触[27] 还提出通过累积学习函数的灵敏度（梯度的大小）在线计算 Ω_i。Riemanian-Walk（RWalk）算法[23] 融合 Fisher 信息矩阵近似和在线路径积分来计算每个参数的重要性。此外，RWalk 还使用采样样本来进一步改进结果。

（2）**数据正则化**。这类基于正则化的方法通过知识提炼来防止激活漂移，其最初设计用于从更大的教师网络中学习更紧凑的学生网络。Li 等人[6] 提出在学习新任务时，使用这种技术来防止先前的数据表示出现大幅度漂移。他们称之为 LwF 方法，其损失函数见公式 2-2

$$\mathcal{L}_{dis}\left(\mathbf{x};\theta^{t}\right)=\sum_{k=1}^{N^{t-1}}\pi_{k}^{t-1}(\mathbf{x})\log\pi_{k}^{t}(\mathbf{x}) \tag{2-2}$$

其中 $\pi_k(\mathbf{x})$ 是网络的温度标度 logits，其定义为

$$\pi_{k}(\mathbf{x})=\frac{e^{\mathbf{o}_{k}(\mathbf{x})/T}}{\sum_{l=1}^{N^{t-1}}e^{\mathbf{o}_{l}(\mathbf{x})/T}} \tag{2-3}$$

其中 $\mathbf{o}(\mathbf{x})$ 为网络在 softmax 层之前的输出值，T 为温度标度参数。我们使用 π_k^{t-1} 表示在任务 $t-1$ 训练结束后网络的预测结果。温度标度参数用来矫正网络对于正确类别的输出概率过高的问题。另外，许多方法都将温度标度参数 T 设置为 2[6][28]。需要注意的是，在使用回放方法时，蒸馏损失通常也应用于这些旧类的采样样本[28][29]。基于编码器的终身学习[30] 通过优化一个不完整的自动编码器来扩展 LwF，该自动编码器将特征投影到尺寸较小的流形上。尽管与总的模型大小相比，自动编码器较小，但该方法对于每个任务学习一个自动编码器，这使得模型参数呈线性增长。

2.5 基于回放的方法

基于回放的方法需要保留少量的样本，或生成合成图像[11]，或生成合成特征[31]。该类方法通过回放存储或生成的旧任务数据来防止遗忘旧任务。在类增量学习中，iCaRL[1] 首次提出使用回放方法。此后，该技术已应用于大多数类增量学习方法中。

（1）**样本池类型**。在模型已经适应新任务之后，必须在训练结束时扩大样本池。如果总的样本池大小都固定，则方法必须首先移除一些旧类样本，为新类样本腾出空间，以此确保样本池大小保持不变。随着学习的任务和类别增多，每个类别进行采样的样本数就越少。如果允许扩大样本池，则仅需要添加当前任务中的新样本，但这种方式以线性增大样本池为代价，可能不适用于某些应用。最后，无论样本池是否固定，每个类别的代表样本数都应该相等。

（2）**采样策略**。从训练数据中随机采样来选择样本，然后添加到样本池，这种方法已证明是非常有效的，并且不需要太多的计算成本 [1][23]。受 Welling[32] 的启发，iCaRL 建议根据类别对应的特征空间表示来选择样本。该方法为每个类迭代地选择样本，即在每个步骤中，都要选择一个样本添加到样本池，并使得所得到的样本平均值最接近真实类别的平均值。所以，添加样本的顺序很重要，并且在需要删除某些样本时算法要考虑这个因素。尽管这种迭代式选择过程通常优于随机选择，但会增加计算成本。RWalk[23] 提出了另外两种采样策略。第一种策略计算 softmax 的输出熵，并选择具有较高熵的样本。这将强制选择在所有类别中得分更高的样本。与之类似，第二种策略在特征空间和决策边界变化

不大的情况，基于样本与决策边界的距离来选择。对于给定的样本 $(\mathbf{x}_i, \mathbf{y}_i)$，通过 $f(\mathbf{x}_i;\phi)^T V_{\mathbf{y}_i}$ 计算样本到决策边界的伪距离。这意味着距离越小，样本越接近决策边界。另外，对于这些采样策略（随机抽样除外），样本的选择顺序按照重要性的降序进行记录。如果样本池大小固定，并且必须移除一些样本来为新类样本腾出空间，则重要性较低的样本先被移除。

（3）**结合回放和数据增强**。方法[1][28] 利用 LwF 中的蒸馏损失，并结合采样样本处理激活漂移问题。然而，Beloudah 和 Popescu[33] 观察发现，使用采样样本进行蒸馏会影响模型性能。

2.6 类增量学习算法评价方法

在类增量学习的研究中，一般最重要的一个增量模型评估方式就是模型在增量学习过程中分类准确率的变化，其准确率下降越缓慢，说明模型在增量学习过程中，一方面，模型能有效克服灾难性遗忘问题；另一方面，模型始终能保持比较优秀的新类学习能力。这部分一般使用准确率折线图来说明模型的准确率变化。更进一步，为了了解最终模型的预测行为，我们使用混淆矩阵，来对各个类别的分类情况进行可视化处理。另外，通过整个增量学习过程的平均准确率[9] 来描述模型的整体性能。记模型进行了 n 次类增量学习过程，Acc_i 表示第 i 批次模型对所有已知类别的预测准确率，那么平均准确率 $\overline{Acc}$ 计算过程见公式 2-4。

$$\overline{Acc} = \frac{1}{n-1}\sum_{i=2}^{n} Acc_i \tag{2-4}$$

需要指出的是，在计算平均准确率过程中，我们并不考虑模型在初始批次的预测精度，因为该数值无法反应出模型在增量学习过程中的性能。

另外，类增量学习领域一般选取的对比算法有 LwF.MC，finetuning，FR(fixed representation)，iCaRL，这些算法包括了未/已使用回放方法这两大类别中的经典算法。对于测试算法的数据集，我们一般选取 CIFAR-10/100,Tiny-ImageNet-200, 或者 iILSVRC 数据集。通过多种对比算法，以及多种实验场景设置，更能充分测试出模型的增量学习性能。

2.7 本章小结

本章首先使用图片形象化地阐述了类增量学习领域的灾难性遗忘问题，并定性分析了与之相关的 softmax 抑制问题。本章介绍的基于原型的特征提取器作为本文提出算法的背景知识。另外，本章通过对类增量学习算法的整体介绍让读者了解了该领域的发展。最后，本章介绍的算法评价方法将会在后文中的实验部分使用到。

第三章　类增量学习算法 PCRC

本章首先对类增量学习问题进行符号化描述，然后介绍 PCRC 算法的算法框架和模型结构。接下来，本章讨论 PCRC 算法的网络框架结构以及与之相关的任务分解、网络记忆固化和基于原型的分类。最后，本章通过实验来验证 PCRC 算法的有效性。

3.1　问题建立与评估设置

假设带标记数据集合 $\mathcal{D}$ 为连续输入的数据流，在第 k 个训练阶段，模型从它的子集 $\mathcal{D}_k \subsetneq \mathcal{D}$ 中进行学习，我们用 C_k 来表示 $\mathcal{D}_k$ 中出现类别，在一般情况下，我们假设不同类别的数据在不同的阶段出现，也就是 $C_i \cap C_j = \emptyset$。另外，类增量学习模型应该在第 k 个预测阶段能够分类所有它学习过的类别，也就是 $C_1 \cup C_2 \cup ... \cup C_k$。这就意味着其评估方式应该为单头设置，但是如果我们把每一个子数据集 C_k 的训练视为一个任务，即把类增量问题视为一个无法在预测阶段使用任务描述符的任务增量问题，那么就可以使用之前讨论的多头评估方式。

3.2　PCRC 算法

3.2.1　PCRC 算法框架

PCRC 算法整体框架图如图 3-1所示，我们将多分类问题拆分为多个二分类问题，然后构造对应的多头网络，整个网络由 3 部分构成：

1. 共享层作为基本的网络结构，这部分的主要作用为初步的特征提取。这部分的网络结构与一般的 DNN 网络相同。
2. 对于每一个二元分类器，里面包含的少量参数为了针对特定的类别进行特征提取。当训练数据中出现新类别 i，那么与之相关的头部将会添加在共享层的顶部，在 PCRC 的默认设置中，每个头部包含一个由全连接层构成的隐藏层 H_i，还有一个 sigmoid 输出单元 O_i。

3. 对于类别 i，存在一个与之对应的原型向量 $\boldsymbol{c}_i$。输入样本 x 在第 i 类上的输出概率通过在特征空间计算到 $\boldsymbol{c}_i$ 的距离得到。

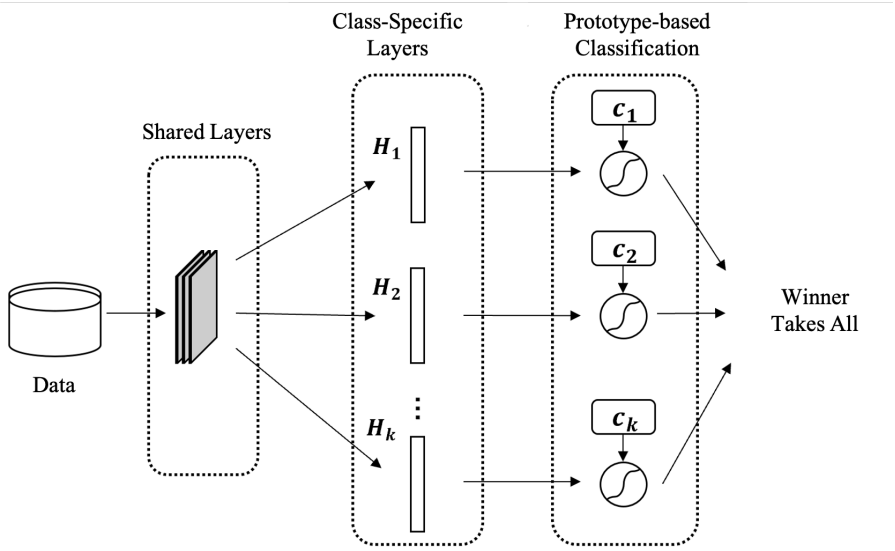


图 3-1: PCRC 算法框架图, 对于类别 i，H_i 表示它的隐藏层，$\boldsymbol{c}_i$ 表示它的原型向量。

在第一个训练阶段，PCRC 的训练过程与一般的 DNN 网络没有区别。在接下来的训练过程中，在训练开始之前，训练数据会先被送入网络，并保存网络的输出值，我们称该输出值为 response 向量。在接下来的过程中，该向量用于保持模型对旧知识的记忆；同时，新数据的标签用来指导模型对新知识的学习。其流程见图 3-2。然后，我们结合这两部分值来构造一个最终的目标向量来指导模型的学习。最后，在测试阶段，根据测试数据在各个头部的得分，选取最高分所在的类别为最终的预测结果。

3.2.2 任务分解

在章节 2.2中，我们分析了使用单头评估方式无法缓解灾难性遗忘问题。因此，我们解决该问题的出发点就是将评估方式设置为多头。另外，我们不是简单地把每一个训练阶段视为一个任务，而是将类增量学习问题分解为一个二分类任务集合。当流数据中新类数据到达时，对应的头部网络会添加在共享层的顶部。对于输入数据 $\boldsymbol{x}$，第 i 类对应的头部网络产生一个分数

$$\hat{y}_i = p(y_i = 1 \mid \boldsymbol{x}; \boldsymbol{\theta}) \tag{3-1}$$

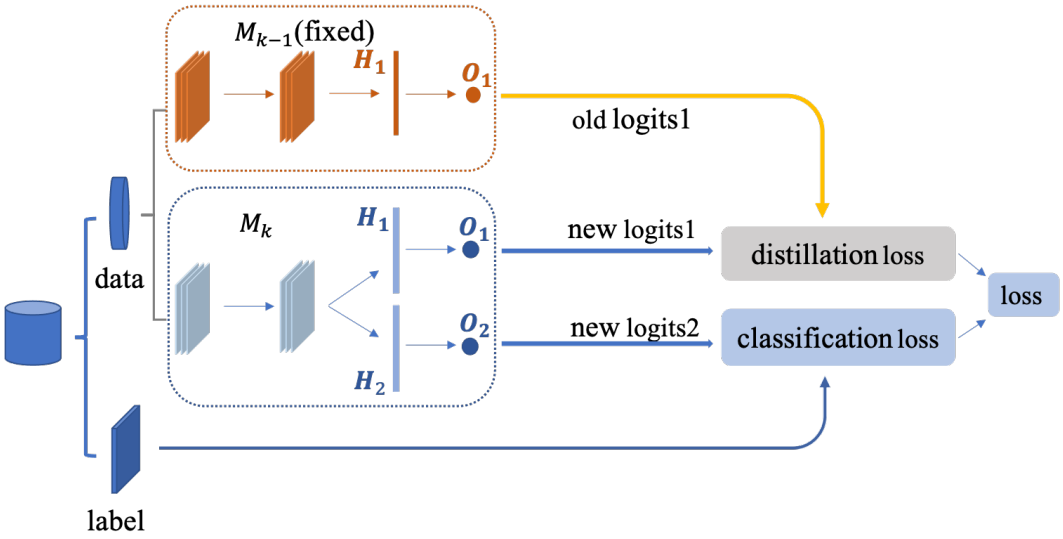


图 3-2: PCRC 算法训练过程示意图。M_{k-1} 模型表示模型副本，其参数固定，新的类别参数将会添加到模型 M_k 中进行学习。

其中 θ 表示网络学习到的参数。这种方式产生的网络输出结果构成一组伯努利分布，而不再是由单个 softmax 层输出那样服从多项式分布。对于训练数据 $(\boldsymbol{x}, \boldsymbol{y})$，在类别 i 上产生的二元交叉熵损失由如下公式计算：

$$\mathcal{L}_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = -y_i \log \hat{y}_i - (1 - y_i) \log (1 - \hat{y}_i) \tag{3-2}$$

这里，我们假设标签 $\boldsymbol{y}$ 为独热编码向量。该损失函数将类别之间的输出概率解耦，即样本在类别 i 上的得分与在其他类别上的得分无关。这也意味着，在训练过程中，一个头部产生的损失并不会更新其他头部的参数。

另外，由于在预测阶段，我们无法使用任务描述符，所以网络的所有头部都要参与到评估的过程中，然后根据最大值得到最终的评估结果。因为网络中的一个头部仅对应一个类，在最简单的情况下——每个头部网络仅包括一个输出单元，那么其参数等价于 softmax 层中为该类增加的参数。

在默认情况下，PCRC 网络的每个头部包含两大部分：（1）由全连接层构成的隐藏层，并使用 ReLU[34] 作为激活函数；（2）基于原型的分类单元。尽管对于任务分解而言，隐藏层并不是必须的，但是由于该部分参数是与特定类别相关的，所以可以进一步提高网络的分类效果。

3.2.3 网络权重固化

直接使用公式 3-2来增量式地训练网络的方式依然无法解决灾难性遗忘问题。因此，我们还需要固化网络中保存的旧类信息。对于类别 i 对应的头部网络，为了在训练数据不包括类别 i 的情况下，其输出依然稳定，我们使用了 Response Consolidation（RC）方法。

由于网络保存的信息蕴含在其输入与输出的映射关系中，那么我们通过迫使网络在每个训练期间保持原来的行为，就可以保留之前学习类别的信息。假设在第 $k-1$ 个训练阶段，网络学习到的参数为 $\boldsymbol{\theta}^{k-1}$，那么头部网络 i 的输出可以通过最小化 KL 散度来固化，即：

$$\mathcal{L}_i^{KL}(\boldsymbol{x},\boldsymbol{\theta}) = D_{KL}\left(p(y_i|\boldsymbol{x};\boldsymbol{\theta}^{k-1}) \,||\, p(y_i|\boldsymbol{x};\boldsymbol{\theta}) \right) \tag{3-3}$$

为此，在优化参数之前，我们会向网络输入所有可用的训练数据。对于数据 $\boldsymbol{x}$，其在以 $\boldsymbol{\theta}^{k-1}$ 为参数的网络的输出作为 response 向量 $\boldsymbol{y}^{k-1}$，即：

$$y_i^{k-1} = p(y_i = 1 \mid \boldsymbol{x};\boldsymbol{\theta}^{k-1}) \tag{3-4}$$

而最小化公式 3-3中的 KL 散度其实等价于最小化公式 3-2中的二元交叉熵损失，也就是我们在固化网络记忆中产生的固化损失。

通过将分类损失和固化损失结合在一起，就构成了 PCRC 算法整体的损失值。假设在第 k 个训练阶段，送入网络的类别集合为 C_k，那么在所有网络学习过的类别中，在该阶段并未出现的类别集合为：

$$S_k = (C_1 \cup C_2 \cup \ldots \cup C_{k-1}) - C_k \tag{3-5}$$

对于训练样本 $(\boldsymbol{x},\boldsymbol{y})$，产生的总的损失值为：

$$\mathcal{L}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}) = \sum_{i\in C_k} \mathcal{L}_i(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}) + \lambda \sum_{j\in S_k} \mathcal{L}_j(\boldsymbol{x},\boldsymbol{y}^{k-1},\boldsymbol{\theta}) \tag{3-6}$$

这里的 $\mathcal{L}_i(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta})$ 和 $\mathcal{L}_j(\boldsymbol{x},\boldsymbol{y}^{k-1},\boldsymbol{\theta})$ 均由公式 3-2计算得到。

在公式 3-6中，固化损失作为整体损失的正则化项，其正则化系数为 λ，公式 3-6中的第一部分致力于最小化新训练数据的分类损失，而第二部分则致力于让网络在这些未出现类别对应的头部产生的输出值与与原来保持一致，以此来保留住旧类信息。

尽管之前的讨论都是基于 PCRC 算法没有使用旧样本的情况，但是它也可以很灵活地与回放方式结合。这就意味着我们仍然可以使用来自每个已学类别的少量样本来训练网络。另外，对于样本的获取，我们既可以进行随机采样，也可以通过一些生成算法构造出来。

3.2.4 基于原型的分类

由小节 2.3所讨论的，基于原型训练的特征提取器得到的特征具有两大良好的性质：（1）类内聚敛性（2）类间分离性。由此启发，在 PCRC 中，我们也引入了基于原型的分类单元增强分类准确率。对于每个类别 i，我们在特征空间中学习一个原型向量 $\boldsymbol{c}_i$ 作为该类的特征表示。对于数据 $\boldsymbol{x}$，我们以 $f_i(\boldsymbol{x};\boldsymbol{\theta})$ 表示该样本在类别 i 对应的头部网络的隐藏层 H_i 得到的特征向量。最后，通过计算特征点到各个原型向量的距离，就可以得到基于原型的分类结果。假设 d 为在特征空间中的欧式距离，那么在 3-1中的输出概率 $\hat{y}_i$ 按公式 3-7计算得到。

$$\hat{y}_i = \frac{1}{1+\exp\left(d(\boldsymbol{c}_i, f_i(\boldsymbol{x};\boldsymbol{\theta})) - r\right)}. \tag{3-7}$$

其中，超参数 r 是预先定义的，作为特征空间中决策边界超球面的半径。通过引入半径 r，使得在增量式学习过程中，PCRC 的每个二元分类器单元进一步约束所负责类别的特征分布空间。

最后我们用公式 3-7来替换公式 3-2中的 $\hat{y}_i$，然后最小化由公式 3-6计算得到的损失值，就可以同时训练原型向量 $\boldsymbol{c}_i$ 以及参数 $\boldsymbol{\theta}$。对于属于类别 i 的样本 $\boldsymbol{x}$，通过优化 $\boldsymbol{c}_i$ 以及参数 $\boldsymbol{\theta}$，使得 $d(\boldsymbol{c}_i, f_i(\boldsymbol{x};\boldsymbol{\theta})) \leq r$，而对于不属于类别 i 的样本 $\boldsymbol{x}$，有 $d(\boldsymbol{c}_i, f_i(\boldsymbol{x};\boldsymbol{\theta})) > r$。在测试阶段，我们不用再计算输出概率，而是对于所有的类别 i，都计算对应的 $d(\boldsymbol{c}_i, f_i(\boldsymbol{x};\boldsymbol{\theta}))$，选择最短距离对应的类别作为预测结果。

由于在 PCRC 中，我们的隐藏层是与类别相关的，这就使得我们的基于原型的分类方法不同于其他类似方法。在 PCRC 中，与类别 i 相关的原型向量 $\boldsymbol{c}_i$

位于隐藏层 H_i 所在的特征空间中，而其他基于原型的分类方法将所有的原型向量置于一个共享的特征空间中。

假设我们移除所有与类相关的隐藏层，那么所有的原型向量位于同一特征空间。对于数据 $\boldsymbol{x}$，我们用 $f(\boldsymbol{x};\boldsymbol{\theta})$ 表示它在该空间中的特征向量。根据公式 3-6最小化损失函数，那么网络在最小化 $d(\boldsymbol{c}_i, f(\boldsymbol{x};\boldsymbol{\theta}))$ 的同时，还要对于其他类别 j，保持 $d(\boldsymbol{c}_j, f(\boldsymbol{x};\boldsymbol{\theta}))$ 不变。很显然，要满足所有这些约束条件去优化参数是非常困难的，而与特定类别相关的隐藏层 H_i 就可以让每个类有自己的特征空间，这样就可以降低参数优化的难度。此外，高维特征空间将会受到欧式距离的影响，从而使得超参数 r 难以设置，但我们可以通过限制隐藏层中的节点数来解决该问题。

3.3 实验与分析

3.3.1 对比实验

为了评估 PCRC 的性能，我们在 CIFAR-10，CIFAR-100 以及 Tiny-ImageNet-200 上进行实验，并与其他经典的算法进行准确率比较。正如章节 3.1所说，我们把模型的训练过程分为数个阶段。在每个阶段中，我们训练模型所使用的数据只选取整个数据集的一部分，最后在预测阶段，我们使用模型所有已经学习的类别的测试样本进行准确率测试。具体的数据集设置见表 3-1。对于每个实验，我们重复 5 次，取平均值作为最终的实验结果。

表 3-1: 数据集参数

数据集	CIFAR-10	CIFAR-100	Tiny-ImageNet-200
总类别数	10	100	200
每批次类别数	2, 5	10, 20	10, 20
每个类训练样本数	5000	500	500
每个类测试样本数	1000	100	50
评估方式	top-1	top-1	top-5

为了使对比更加公平，我们使用相同的底层网络，以及公用的超参数。在以下实验中，我们选取 ResNet-18[35] 作为所有模型的底层特征提取网络模块。在训练的过程中，我们使用 Adam 优化器[36] 调整参数。每一批次，我们送入网络

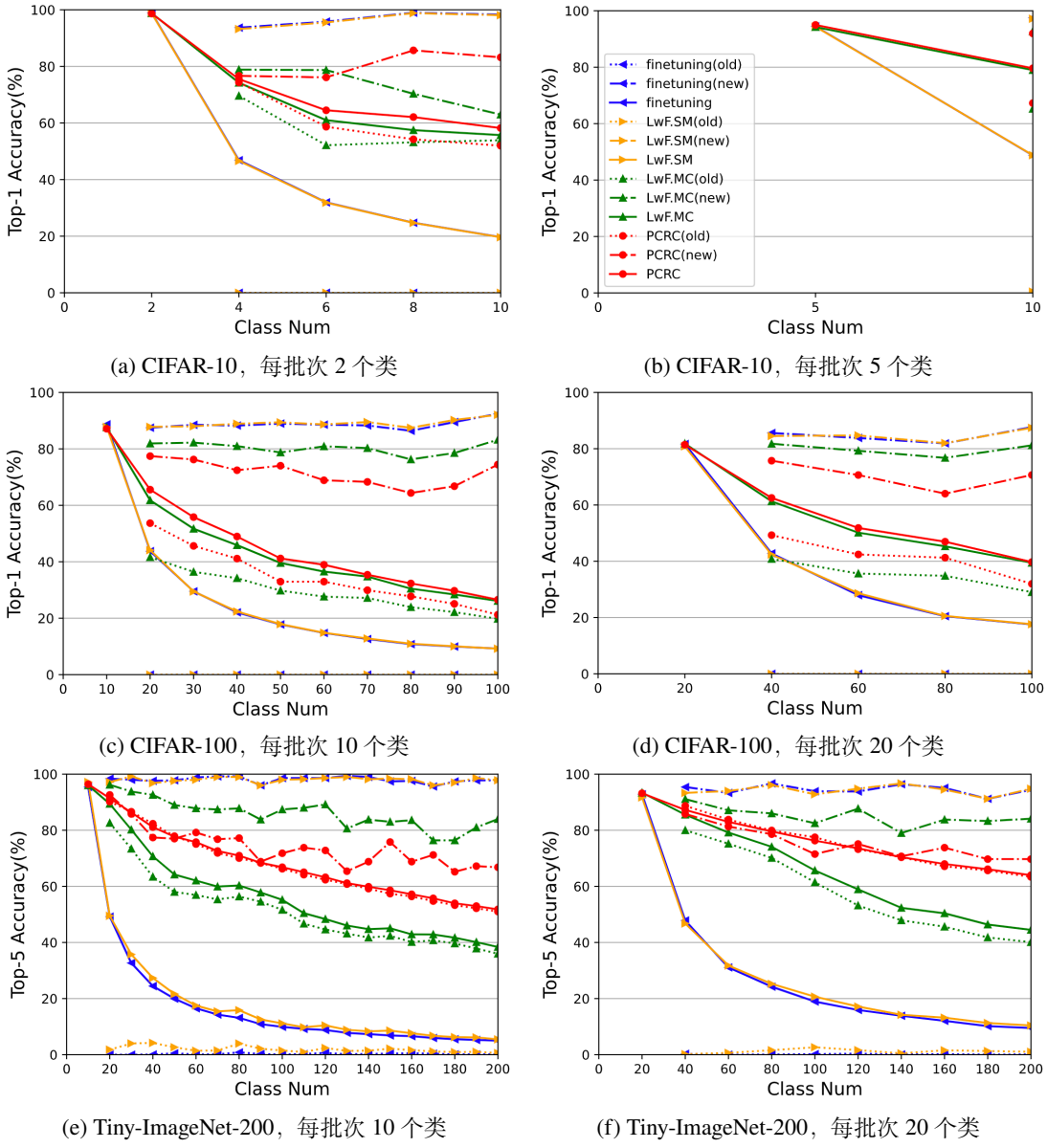
的图片为 128 张，在所有的实验中，我们设置初始学习率均为 5e-4。对于训练次数，在 CIFAR-10/100，Tiny-ImageNet-200 上，我们分别设置为 60/100/100。另外，为了提高算法的效果，我们也是使用了学习率调整策略——对于 CIFAR-10，分别在第 35/45/50 次迭代学习率减半，对于 CIFAR-100 和 Tiny-ImageNet-200，分别在 50/60/80 次迭代学习率减半。

除非特别指出，PCRC 中头部网络的隐藏神经元个数设置为 20，公式 3-6中的 λ 设置为 1，公式 3-7中的 r 设置为 10。

在这个实验中，我们把整个数据集划分成多个子集，并且任意子集之间没有交集。训练过程中，我们以类标递增的顺序进行。对于 CIFAR-10 数据集，分别以每批次 2/5 个类进行训练，每个子集两个类；对于 CIFAR-100/Tiny-ImageNet-200 数据集，分别以每批次 10/20 个类进行训练。

另外，我们选择了 3 种对比算法与 PCRC 方法进行比较，包括 finetuning，以及 LwF 单头版本 LwF.SM 算法，多头版本 LwF.MC 算法。finetuning 和 LwF.SM 这两个方法在每个训练阶段均在 softmax 层动态增加输出维度。两者区别在于，LwF.SM 采用了 LwF.MC 中的网络固化策略。与 LwF.SM 所不同的是，LwF.MC 将每批次所有新类别的多分类视为一个任务，即对一个批次所有新类别增加一个头部网络，当然，在预测阶段，LwF.MC 也无法使用任务描述符进行头部网络选择，所以在预测阶段也采用 “winner-takes-all’’ 机制得到最终的预测结果。

为了单独比较在实验过程中，不同模型的记忆能力与学习能力，我们分别统计了在实验中每个模型对于旧类、新类的预测能力，从图 3-3可以看出，，在 CIFAR-10 数据集，每批次 2 个类别的实验设置中，finetuning 方法和 LwF.SM 方法的表现极其相似。算法在新类准确率上显著高于其他算法，但是在旧类别上，其准确率为 0，即出现了彻底的遗忘现象。LwF.MC 方法和 PCRC 方法虽然在新类别准确率上相较于前两个方法低很多，但是在旧类别上的预测准确率高很多。最终，LwF.MC 方法和 PCRC 方法整体准确率在 50% 以上。虽然最后 PCRC 方法在旧类别上略微低于 LwF.MC 方法，但由于后续在新类别上有明显的优势，所以整体准确率最高。在每批次 5 个类别的实验设置中，由于只进行一次增量学习，最终结果对方法在旧知识上的记忆能力要求不高，所以 PCRC 方法相对于 LwF.MC 方法的优势有缩小。另外，finetuning 方法和 LwF.SM 方法的表现和之前一致。在 CIFAR-100 数据集，每批次 10/20 个类别的实验设置中，finetuning 方



(a) CIFAR-10，每批次 2 个类

(b) CIFAR-10，每批次 5 个类

(c) CIFAR-100，每批次 10 个类

(d) CIFAR-100，每批次 20 个类

(e) Tiny-ImageNet-200，每批次 10 个类

(f) Tiny-ImageNet-200，每批次 20 个类

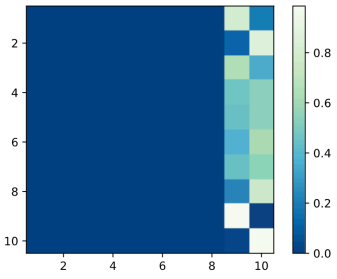
图 3-3: PCRC 算法与其他对比算法在 CIFAR-10/100,Tiny-ImageNet-200 上的准确率比较。由 old/new 描述的方法指的是该方法在旧类/新类上的准确率变化。对于旧类准确率和新类准确率，都从第二批次开始。

法和 LwF.SM 方法依然只对新类别有效。PCRC 方法相较于 LwF.MC 方法在新类别上准确率有明显劣势，但是由于在旧类别上表现最佳，所以最终还是相对 LwF.MC 方法有微弱优势。最后，在 Tiny-ImageNet-200 数据集，每批次 10/20 个类别的实验设置中，由于采用的是 top-5 准确率，所以 finetuning 方法和 LwF.SM 方法在旧类别上的准确率不再为 0，相比 CIFAR-100 数据集，算法在增量学习批次更长，或者每批次类别数更多的情况下，PCRC 方法在旧类别上分类准确率的优势更加明显。由于类别数的增多，导致旧类别在所有类别中占比增大。PCRC 方法相对于 LwF.MC 方法的明显优势完全由旧类别上的准确率优势决定。

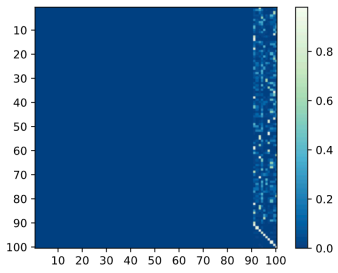
finetuning 在每个训练阶段结束之后，都能很好地区分新类别，但之前类别的信息几乎全部遗忘；LwF.SM 由于其网络固化没有生效，所以其表现与 finetuning 很相似；LwF.MC 在增量学习的过程中，通过多头网络设计能有效缓解网络严重遗忘问题，但在更长批次的增量学习过程中，其知识的遗忘现象越来越明显。最后，PCRC 方法整体上准确率虽然高于 LwF.MC，但是为了让网络对旧类知识有更好的记忆能力，导致模型在学习新知识的时候受到了束缚，不过这也是对模型记忆与学习平衡的结果。

对于 finetuning 方法，由于该方法在下一训练阶段未采取任何对网络参数的固化措施，所以仅能识别本阶段的新类信息；LwF.SM 方法由于 softmax 的抑制问题，尽管该方法采用了与 LwF.MC 一样的蒸馏方式进行旧类记忆，但实际实验结果显示，这种情况下，网络固化无法发挥其有效性；与 LwF.SM 相对应的是 LwF.MC 方法，该方法由于使用了多头网络，没有 softmax 的抑制问题，最终模型还能很好地保留以往学习到的类别信息；最后，我们的方法 PCRC 相对于 LwF.MC, 通过任务分解和基于原型的特征提取方式，并配合使用了新的损失函数，从而进一步提高了网络的性能。

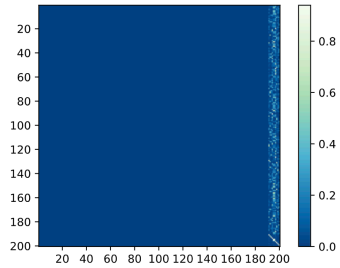
为了进一步了解各个算法在增量学习过程中，最后得到的模型在所有测试样本中的预测行为，我们绘制了各个算法最后对应的混淆矩阵。从混淆矩阵 3-4、3-5、 3-6、 3-7, 我们可以更细粒度看到不同方法得到的模型的预测行为。finetuning 和 LwF.SM 方法随着模型不停地增量学习，导致旧类别几乎全部预测为了新类别；由于 LwF.SM 方法的网络固化作用由于 softmax 的抑制问题未起作用，所以也可以看到最终 LwF.SM 方法和 finetuning 方法对应的混淆矩阵也十分相似。LwF.MC 方法产生的混淆矩阵，相比前两者，有很多旧类数据预测分布在对角线上。尤为明显的是在 CIFAR-100 数据集上，LwF.MC 方法对应混淆矩阵在对角线上呈现出了较为连续的的白色线条（矩阵主对角线上越泛白的地方，说明该类别预测结果越准确；相对应的，在矩阵其他位置，越泛白则说明对应类别出现了大量预测错误的问题），这就佐证在增量学习过程中，该方法较好地存储了旧类知识。从三个数据集训练得到的最终模型预测结果来看，尤其是在 Tiny-ImageNet-200 数据集上，PCRC 算法的混淆矩阵的对角线点最为集中，连续性最好。



(a) CIFAR-10，每批次 2 个类

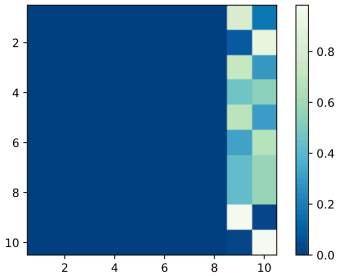


(b) CIFAR-100，每批次 10 个类

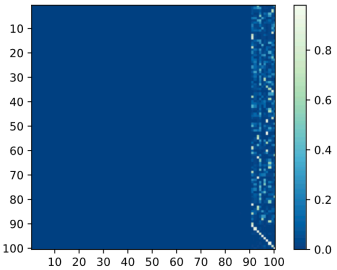


(c) Tiny-ImageNet-200，每批次 10 个类

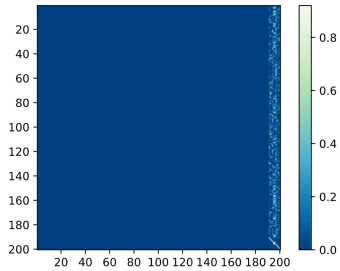
图 3-4: finetuning 算法的混淆矩阵



(a) CIFAR-10，每批次 2 个类

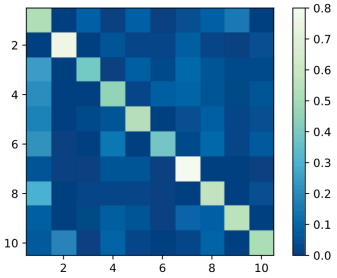


(b) CIFAR-100，每批次 10 个类

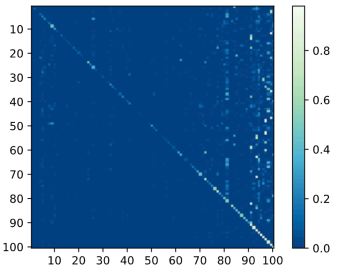


(c) Tiny-ImageNet-200，每批次 10 个类

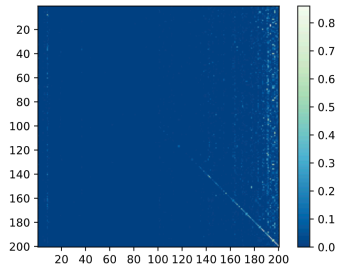
图 3-5: LwF.SM 算法的混淆矩阵



(a) CIFAR-10，每批次 2 个类

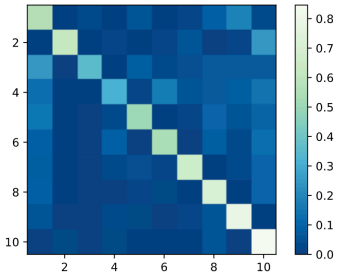


(b) CIFAR-100，每批次 10 个类

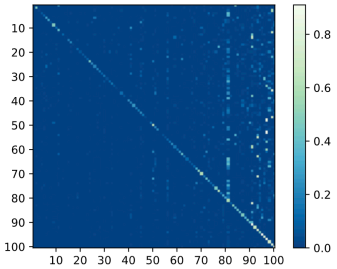


(c) Tiny-ImageNet-200，每批次 10 个类

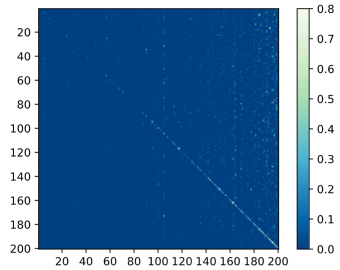
图 3-6: LwF.MC 算法的混淆矩阵



(a) CIFAR-10，每批次 2 个类



(b) CIFAR-100，每批次 10 个类



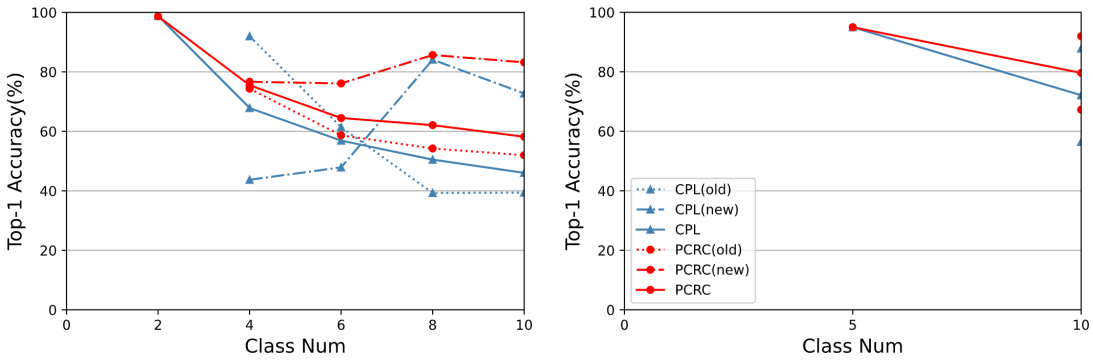
(c) Tiny-ImageNet-200，每批次 10 个类

图 3-7: PCRC 算法的混淆矩阵

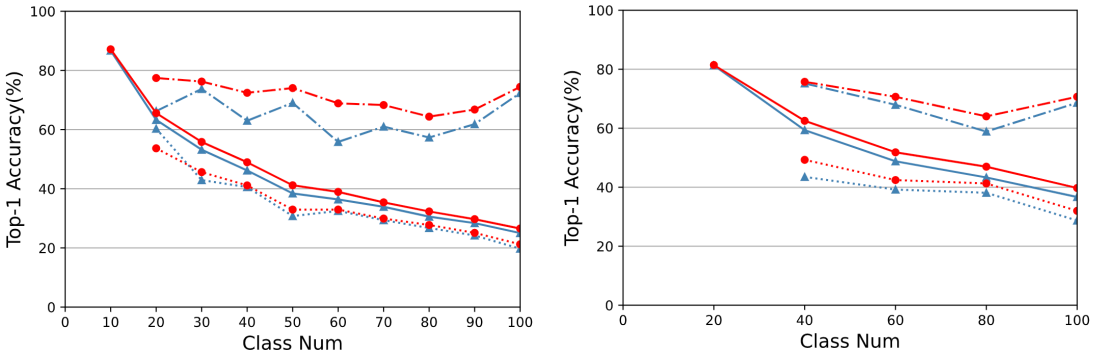
3.3.2 任务分解消融实验

在上一小节中，我们通过 LwF.MC 与其变体 LwF.SM 说明使用任务分解的必要性。但是任务分解的拆分既可以是如 PCRC 那样把每个类别作为一个二分类任务对待，也可以如 LwF.MC 那样将每一批次的所有新类当一个多分类任务来对待。为了在使用基于原型分类的情况下来比较这两种拆分方式的区别，我们增加了一个关于任务分解相关的对比实验。我们参考 Hong-Ming Yang 等人[19]的想法，实现了类增量版本算法 CPL(Convolutional Prototype Learning)。与 PCRC 不同的是，对每批次所有新类别，CPL 视为一个任务。另外，由于 CPL 每个头部是个多分类器，所以在网络固化的时候，不像 PCRC 使用二元交叉熵，而是使用多元交叉熵；在新类学习的时候，使用 CPL 提出的 DCE（Distance-based Cross Entropy Loss）损失函数，进行新类学习。该损失函数使用 $f_i(\boldsymbol{x};\boldsymbol{\theta})$ 到第 i 个头部对应的各个类别的原型向量的距离，取负之后再执行 softmax 来得到该头部各个类别的分数。在预测阶段，CPL 同样通过计算特征向量到各个类别的原型向量距离，取其最短距离对应的类别作为最终的预测类。

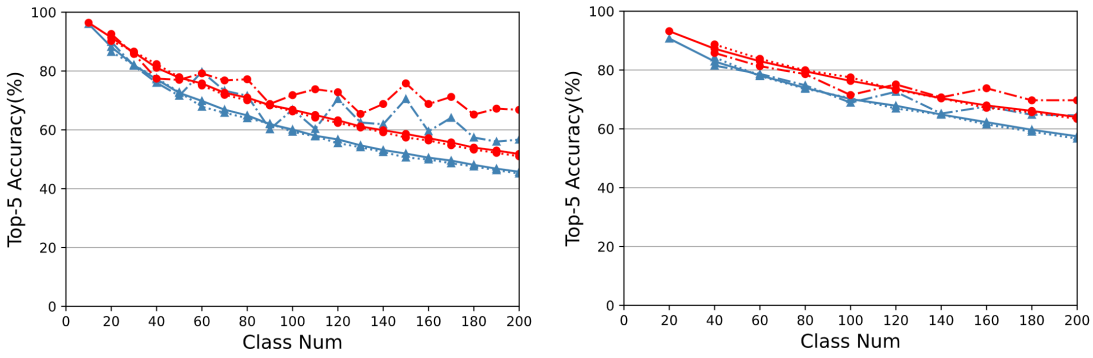
在对比实验中，我们仍然在数据集 CIFAR-10/100 以及 Tiny-ImageNet-200 上进行类似的实验。通过图 3–8可以看到，在 CIFAR-10 数据集，每批次 2 个类别的实验设置中，CPL 在增量式训练过程中，其新类准确率和旧类准确率变化都比较大。在增量学习到 6 分类的时候，CPL 的旧类别准确率高于 PCRC，其新类准确率在逐渐上升，且上升幅度比较明显。从 8 分类到最后，由于 CPL 在旧类别上表现差，再加上 PCRC 在新类上的分类准确率始终高于 CPL，所以 PCRC 的整体准确率优于 CPL，且越到后期，PCRC 与 CPL 的差距越来越大。当以每批次 5 个类别进行实验时，由于增量学习批次减少，PCRC 相对与 CPL 的优势与之前相比也变小了。在 CIFAR-100 数据集，每批次 10 个类别的实验设置中，CPL 与 PCRC 有比较接近的旧类准确率，但是 CPL 在新类别上的分类能力始终低于 PCRC，且其波动性更大。每批次 10 个类别的实验设置中，随着增量学习批次的减少，每批次类别数的增多，CPL 相对于在新类别上的准确率与 PCRC 的差距减小，但是在旧类上的差距却被拉大。最终，在 CIFAR-100 数据集上的两个实验中，CPL 方法表现相比 PCRC 都较差。在 Tiny-ImageNet-200 数据集，每批次 10/20 个类别的实验设置中，CPL 在新类别上的预测结果始终低于 PCRC，且



(a) CIFAR-10，每批次 2 个类 (b) CIFAR-10，每批次 5 个类

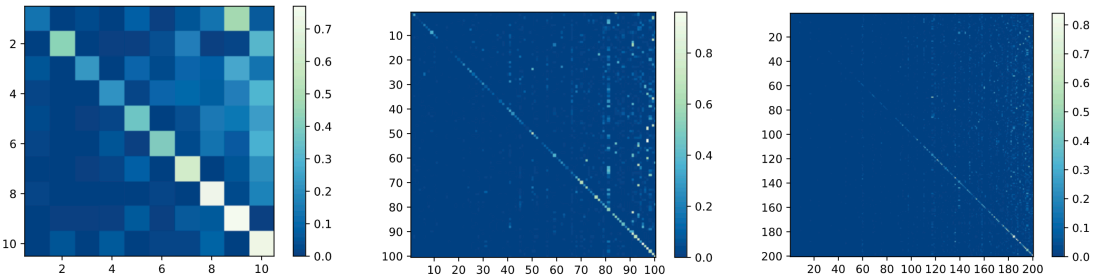


(c) CIFAR-100，每批次 10 个类 (d) CIFAR-100，每批次 20 个类



(e) Tiny-ImageNet-200，每批次 10 个类 (f) Tiny-ImageNet-200，每批次 20 个类

图 3-8: PCRC 算法与 CPL 算法对比。



(a) CIFAR-10，每批次 2 个类 (b) CIFAR-100，每批次 10 个类 (c) Tiny-ImageNet-200，每批次 10 个类

图 3-9: CPL 算法的混淆矩阵

在旧类别上的劣势更加明显，后者也是最终两者整体准确率差距的主要原因。

相比 PCRC，CPL 的整体分类性能有明显降低。从新类的准确率变化看出，CPL 算法的相比 PCRC 算法波动性更大；另外，不论在新类别，还是旧类别，CPL 的预测性能都不如 PCRC。对比还可以发现，在增量学习的时间更长，或者每批次学习的类别数更多时，两个算法之间的性能差异更大。

最后，通过图 3-9展示了 CPL 得到的混淆矩阵。可以看到，CPL 得到的混淆矩阵与 PCRC 相比，在非对角线区域有更多的白色分布，即进一步验证了上面的结论。这种结果的一个主要原因在于，CPL 算法每个头部进行多分类，预测阶段的干扰不仅是头部与头部之间的干扰，还包括了单个头部负责的类别之间的干扰。因为在预测阶段，由于每个头部网络的输出为多个类别的分数，尽管样本能在负责其类别的头部网络有正确的判断，但在其他头部网络中的某个类别上，也很容易出现得分更高的情况，最终导致预测失败。

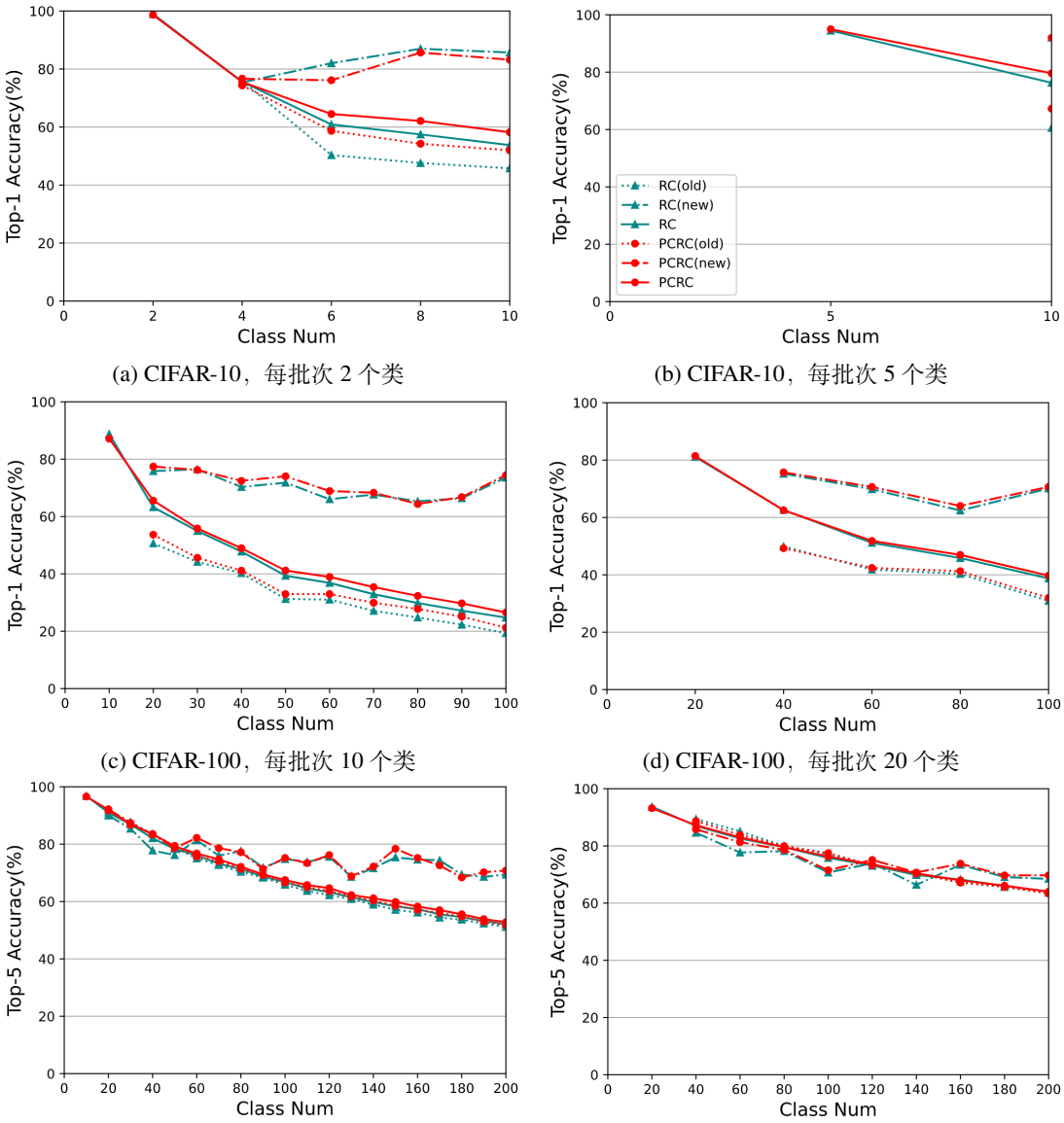
通过 CPL 与 PCRC 的对比，我们可以看到，在增量式分类过程中，我们之前提出的任务拆解思路是有效可行的。一个主要原因是，相较于二分类任务拆解，多分类任务拆解虽然头部网络数目更少，但是每个头部负责的分类数目增多，对于输入的数据，容易在不是由其负责的头部网络产生较高响应，从而加重预测阶段的相互混淆问题。

3.3.3 原型消融实验

为了进一步说明原型在整个网络训练过程中的作用，我们根据算法 PCRC 设计了其变体算法 RC。RC 与 PCRC 唯一区别是未使用基于原型的分类器进行特征提取。在 RC 使用的模型中，每个头部网络使用一个 $N \times 1$ 的全连接层（其中 N 为隐藏层神经元的个数），然后通过 sigmoid 单元计算得到类别的输出概率。RC 网络在类别 i 上的概率输出 $\hat{y}_i$ 计算过程见公式 3-8。

$$\hat{y}_i = \frac{1}{1 + \exp\left(-\mathcal{F}\left(f_i(\boldsymbol{x};\boldsymbol{\theta})\right)\right)} \tag{3-8}$$

其中 $\mathcal{F}$ 表示 RC 的分头网络中的全连接层表示的映射关系。与 PCRC 网络中的概率输出计算公式 3-7相比较，去掉了特征向量 $f_i(\boldsymbol{x};\boldsymbol{\theta})$ 到原型向量的距离计算，以及特征空间中超球面半径 r 对网络拟合的约束。

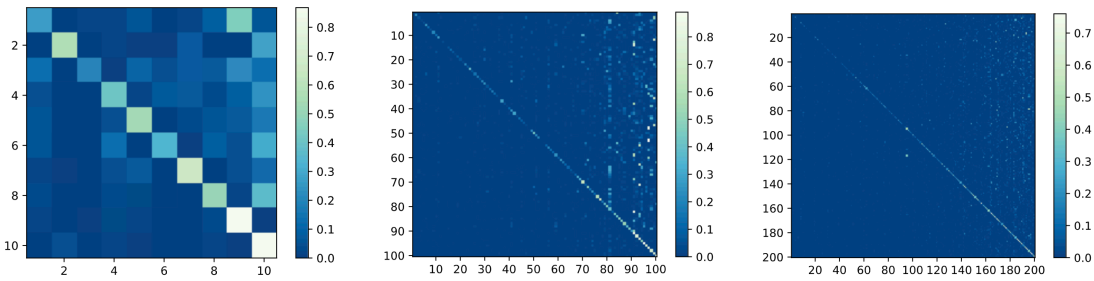


(a) CIFAR-10，每批次 2 个类 (b) CIFAR-10，每批次 5 个类

(c) CIFAR-100，每批次 10 个类 (d) CIFAR-100，每批次 20 个类

(e) Tiny-ImageNet-200，每批次 10 个类 (f) Tiny-ImageNet-200，每批次 20 个类

图 3-10: PCRC 算法与 RC 算法对比。



(a) CIFAR-10，每批次 2 个类 (b) CIFAR-100，每批次 10 个类 (c) Tiny-ImageNet-200，每批次 10 个类

图 3-11: RC 算法的混淆矩阵

我们在数据集 CIFAR-10/100 以及 Tiny-ImageNet-200 上对 RC 算法进行了测试。与 PCRC 一样，RC 的头部网络隐藏神经元个数 N 我们设置为 20。如图 3-10，在 CIFAR-10 数据集，每批次 2 个类别的实验设置中，RC 在新类别上的准确率比 PCRC 占优，但是在旧类别的准确率与 PCRC 有较明显劣势，整体准确率相差 5% 以上。在每批次 5 个类别的实验设置中，RC 与 PCRC 在新类别上准确率十分接近，而在旧类别上的准确率较 PCRC 算法低 5 个百分点以上，PCRC 的整体准确率的优势相比之前变小了。在 CIFAR-100 数据集，每批次 10 个类别的实验设置中，RC 与 PCRC 在新类别上的分类准确率十分接近，从 40 分类到 70 分类 PCRC 算法占优。在旧类别上，PCRC 的准确率相对 RC 的优势差距很稳定，所以其整体准确率变化也有类似特点。在每批次 20 个类别的实验设置中，PCRC 在新类别和旧类别上的准确率仅有微弱优势，所以在整体准确率上与 RC 差距很小。在 Tiny-ImageNet-200 数据集，每批次 10/20 个类别的实验设置中，在开始几个批次，PCRC 在新类别上相对 RC 有较明显优势，而在后期的学习中，两个算法性能表现比较接近。

PCRC 相比 RC，其整体准确率，旧/新类别准确率都有提升。可以看到，在同一个数据集上，在学习批次更多的情况下，PCRC 优势更明显；同时，随着类别数的增多，分头网络的数目也逐渐增多，在预测阶段相互混淆的情况也逐渐增多，所以通过基于原型的方式所带来的分类器性能的提升，对于最终模型性能提升的贡献程度就降低了。通过图 3-11展示的 RC 混淆矩阵验证了上述分析。需要指出的是，尽管 RC 算法未使用原型分类器，但由于使用了任务分解的思路，所以最终也有大量样本的预测结果落在对角线上。

3.3.4 实验结果整合

最后，我们把在增量学习过程中，所有算法对于旧/新类别，以及所有类别的分类准确率，统计了平均值，见表 3-2。从结果可以看出：算法 finetuning 和 LwF.SM 在 CIFAR-10/100 数据集上，在旧类别上的准确率为 0，即对于旧类知识出现了完全遗忘的现象。在 Tiny-ImageNet-200 数据集上，由于我们采用的是 top-5 准确率，所以在旧类别上的准确率接近 0，但不为 0。另一方面，正是由于网络在学习过程未受到网络固化对参数学习的束缚，所以这两个算法在新类上的学习程度最好，在这几个算法中有最高的新类学习平均准确率。LwF.MC 算法

表 3-2: 增量学习过程中各个模型在旧类别、新类别，以及所有类别上的平均准确率。

数据集		CIFAR-10		CIFAR-100		Tiny-ImageNet-200	
每批次类别数		2	5	10	20	10	20
旧类别	finetuning	0	0	0	0	0.0026	0.0015
	LwM.SM	0	0	0	0	0.0193	0.0123
	LwF.MC	0.5721	0.6526	0.2919	0.3508	0.5080	0.5726
	CPL	0.5802	0.5643	0.3412	0.3738	0.6032	0.6836
	RC	0.5489	0.6056	0.3232	0.4074	0.6657	**0.7463**
	PCRC	**0.5981**	**0.6727**	**0.3448**	**0.4124**	**0.6739**	0.7441
新类别	finetuning	**0.9670**	0.9641	0.8872	**0.8475**	**0.9804**	**0.9447**
	LwM.SM	0.9641	**0.9718**	**0.8912**	0.8467	0.9794	0.9424
	LwF.MC	0.7272	0.9272	0.8034	0.7975	0.8588	0.8496
	CPL	0.6212	0.8784	0.6448	0.6766	0.6792	0.7097
	RC	0.8252	0.9205	0.7037	0.6941	0.7539	0.7361
	PCRC	0.8041	0.9196	0.7143	0.7027	0.7644	0.7513
所有类别	finetuning	0.3080	0.4875	0.1894	0.2719	0.1365	0.2041
	LwF.SM	0.3070	0.4881	0.1907	0.2729	0.1500	0.2119
	LwM.MC	0.6210	0.7899	0.3947	0.4906	0.5475	0.6189
	CPL	0.5533	0.7213	0.3948	0.4703	0.6099	0.6859
	RC	0.6193	0.7631	0.3965	0.4959	0.6708	0.7403
	PCRC	**0.6508**	**0.7962**	**0.4160**	**0.5026**	**0.6808**	**0.7424**

在旧类知识记忆方面的表现弱于 PCRC，CPL，与 RC 相比较，仅在 CIFAR-10 上具有优势。与之相对应的是，LwF.MC 算法在新类平均准确率上高于算法 CPL，RC，PCRC，不过由于 LwF.MC 算法进行了网络参数固化，所以与 finetuning 算法比较，其新类准确率仍然有明显的降低。但是从网络的综合性能考虑，LwF.MC 算法的整体平均准确率除了在 CIFAR-10 上高于 CPL，RC 算法，在其他实验设置上，算法 CPL，RC 的整体性能基本上都高于 LwF.MC 算法。尤其是对于 PCRC 算法，虽然模型在新类准确率上并不占优，但是最终模型在所有类别上的平均准确率相比较其他算法更高，尤其是在 Tiny-ImageNet-200 数据集上，该算法的最终结果与其他一些对比算法的差距比较明显。这些数据更加充分说明了 PCRC 算法使用任务分解思想以及原型分类的有效性。通过这两种改进，PCRC 算法能

够在缓解灾难性遗忘问题的同时，能够保持良好的对新类别学习的能力。

3.4 本章总结

本章提出了新的类增量学习算法 PCRC，并从算法框架、任务分解、网络权重固化和基于原型的分类函数设计四个方面对算法进行介绍。算法框架部分描述了整体工作原理，包括模型的训练和预测过程；任务分解部分将每个新类别的分类视为一个二分类任务，并使用二元交叉熵损失函数进行网络训练；网络权重固化部分通过知识蒸馏来保存模型在之前学习到的旧知识；最后，本章设计了基于原型计算预测概率的分类函数，该函数的输出值是训练阶段中二元交叉熵损失函数的输入。在实验与分析一节，本章交代了实验的相关参数值设置，学习率调整策略等信息。对比实验验证了 PCRC 算法的有效性。另外，为了充分说明二分类任务分解的作用以及基于原型分类的效果，本章分别增加了任务分解消融实验和原型消融实验。最后，本章用表格汇总本章所有的实验数据，并做了更加细节性的分析。

第四章　对类增量学习算法 PCRC 的增强与优化

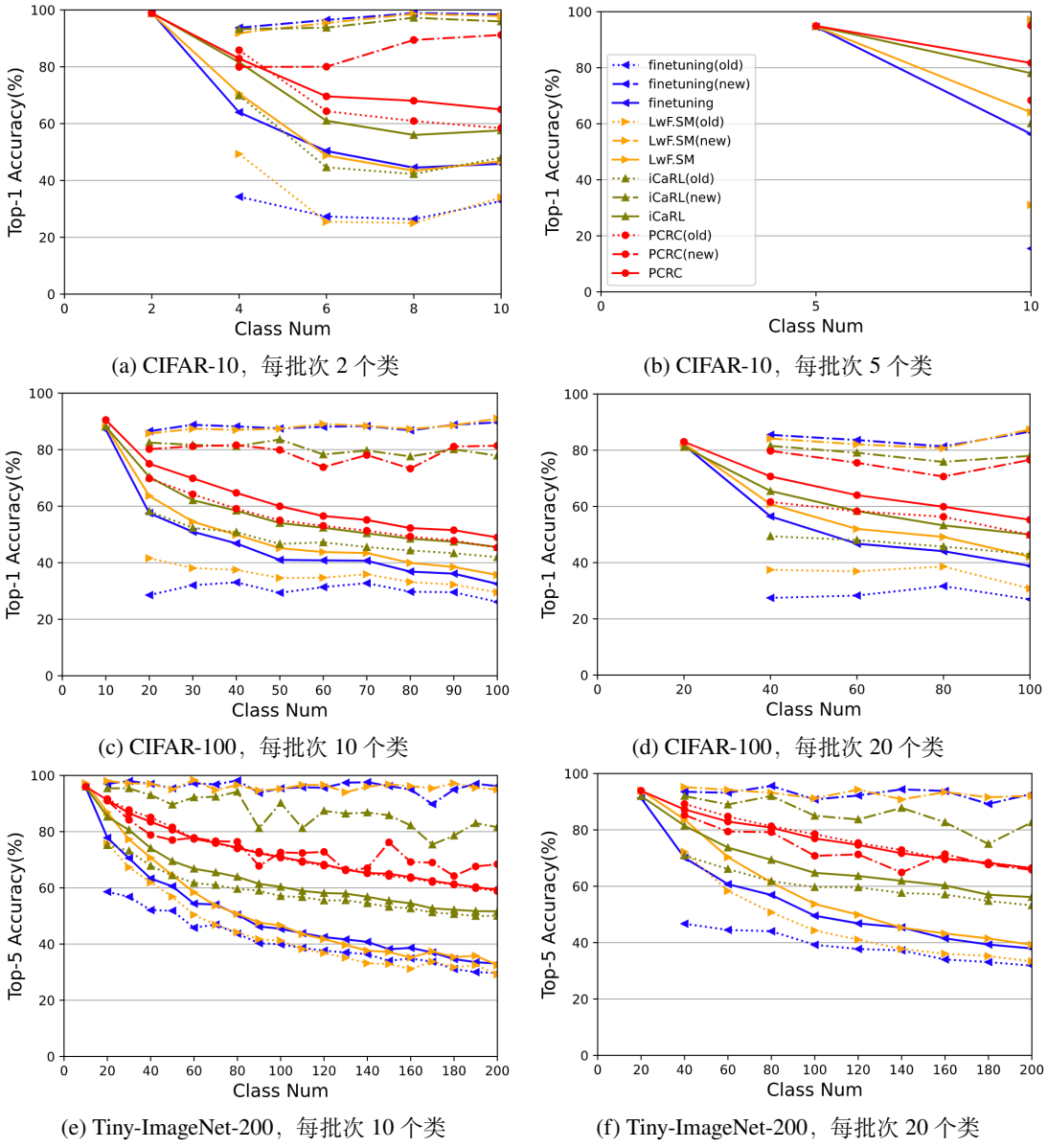
本章首先使用回放方法来对 PCRC 算法进行增强，并通过实验来验证其有效性。接下来，本章分别提出两个对 PCRC 进行改进的算法 SS-PCRC 和 BPCRC，并对每个算法进行实验分析。SS-PCRC 算法引入额外的辅助数据来缓解类增量学习过程中的数据不平衡问题，BPCRC 算法通过将新类数据划分为关键性数据和关键性数据，有针对性地更新网络参数。

4.1　使用回放方法来增强 PCRC 算法

通过对旧类进行少量样本采样，模型在接下来的类增量学习过程中，利用这些旧类采样数据可以有效缓解灾难性遗忘问题。为了量化这种策略的实际效果，对于我们在上一章提到的 PCRC 模型，在不改变任何网络结构以及训练方式的情况下，仅需要修改加载的数据源即可利用回放方法。我们依然在 CIFAR-10/100,Tiny-ImageNet-200 这 3 个数据集上进行实验，与 3.3.1唯一的区别是，我们在每批次训练结束之后，还需要对当前批次训练样本进行随机采样，然后把采样样本放到采样池中。在接下来的训练过程中，总的训练样本不仅包括本批次的新类数据，还包括采样池中的旧类样本。为了进一步分析采样数量对实验结果的影响，我们将每个类别的采样数目分别设置为 20/50。另外，由于 LwF.MC 无法直接支持回放方法，所以在本次实验中不考虑该对比算法，而是增加 iCaRL 算法进行比较。需要指出的是，iCaRL 算法以 herding 方法进行采样。

最终,我们把各个算法在多个实验场景中得到的实验数据绘制折线图 4-1 4-2，展示模型在整个训练阶段中旧类别，新类别，以及所有类别准确率的变化过程。

从图 4-1中可以看到，在 CIFAR-10 数据集，每批次 2 个类别的实验设置中，finetuning 方法依然取得最高的新类别分类准确率。另外，由于有旧类别数据的参与，finetuning 方法和 LwF.SM 方法在旧类别上分类准确率不再为 0。且当每个类别采样 50 张图片时，这两个方法在旧类别上的准确率从 20% 多上升到 40%



(a) CIFAR-10，每批次 2 个类

(b) CIFAR-10，每批次 5 个类

(c) CIFAR-100，每批次 10 个类

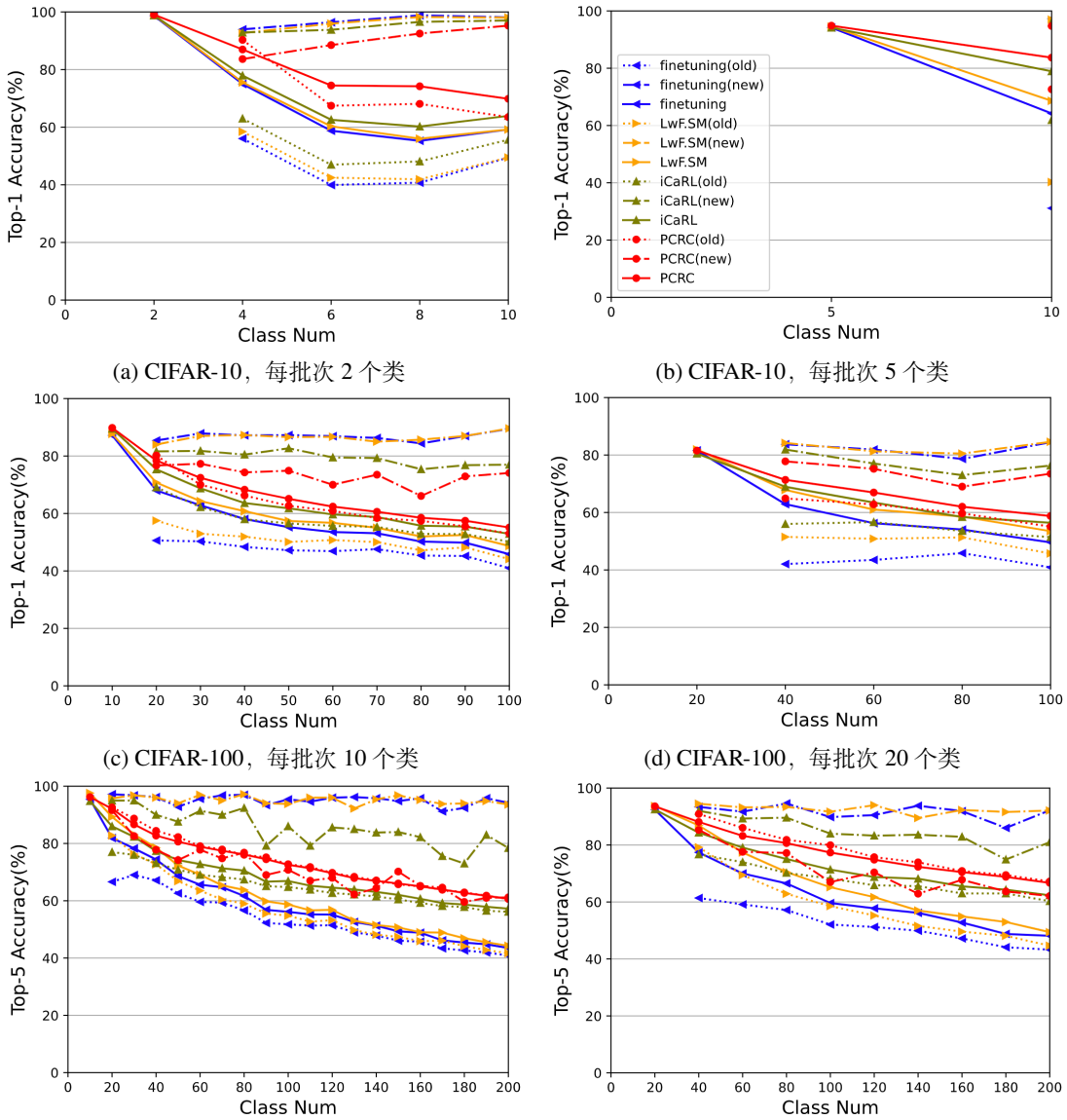
(d) CIFAR-100，每批次 20 个类

(e) Tiny-ImageNet-200，每批次 10 个类

(f) Tiny-ImageNet-200，每批次 20 个类

图 4-1: 每个类别采样 20 张图片，PCRC 算法与其他对比算法的准确率比较。

多。iCaRL 算法在新类别上的准确率与前两个方法接近，但在旧类别上的准确率却能高出接近 20%。PCRC 算法在新类别上的准确率比其他算法都低，但是旧类别的准确率在 10 分类的时候接近 60%，远远高于其他方法，所以最终的整体准确率依然能比 iCaRL 方法高 7%。在每批次 5 个类别的实验设置中，三个对比算法在新类别上的准确率接近。有个细微差别在于，LwF.SM 方法在旧类别上的准确率明显高于 finetuning 方法。在 CIFAR-100 数据集，每批次 10 个类别的实验设置中，iCaRL 相对于 PCRC 在新类别上的优势变小，两者在所有类别的准确率稳定在 4% 左右。在每批次 20 个类别的实验设置中，iCaRL 相对于 PCRC 在新类别上的优势变大。在旧类别上的准确率，PCRC 有 6%-10% 的优势。在 CIFAR-100



(a) CIFAR-10，每批次 2 个类

(b) CIFAR-10，每批次 5 个类

(c) CIFAR-100，每批次 10 个类

(d) CIFAR-100，每批次 20 个类

(e) Tiny-ImageNet-200，每批次 10 个类

(f) Tiny-ImageNet-200，每批次 20 个类

图 4-2: 每个类别采样 50 张图片，PCRC 算法与其他对比算法的准确率比较。

数据集的两种实验设置中，finetuning 方法在旧类别上的准确率均弱于 LwF.SM，且在每批次 20 个类别的实验设置中相差更多。在 Tiny-ImageNet-200 数据集，每批次 10 个类别的实验设置中，iCaRL 相对与 finetuning 和 LwF.SM 方法在新类别上的准确率相差 15% 左右。PCRC 方法又比 iCaRL 算法在新类别上的准确率低 10% 左右，但在旧类别上，PCRC 方法却有 10% 左右的优势。在每批次 20 个类别的实验设置中，PCRC 方法的整体准确率比 iCaRL 方法优势更大，在 10% 左右。

再将图 4-1和图 4-2对照起来，我们可以看出：（1）对于同一个算法，随着模型在训练过程中对每个类别采样数量的增加，模型的性能也随着提高，不过

随着采样数目的增加，模型性能提高的幅度变小。（2）PCRC 模型相比较其他模型而言，在使用相同采样数目的条件下，其模型性能最优，随着采样数目的增加，PCRC 模型性能与其他算法模型性能差距明显减小。（3）与章节 3.3.1 相比较，PCRC 模型使用少量的旧类数据就有明显的模型性能提升，这也意味着在实际模型部署的时候，PCRC 落地效果能有一定的保障。（4）与经典的 iCaRL 算法比较，尽管 iCaRL 算法使用了精巧的采样算法（如果采样相同数量的样本，iCaRL 算法需要更多的计算时间），但 PCRC 算法在旧类别上的记忆程度上依然优于 iCaRL 算法。从最终的模型分类效果来看，PCRC 算法在所有的实验结果中均获得了最佳值。

另外，值得注意的是，在使用回放方法来对 finetuning 和 LwF.SM 算法进行增强的过程中，我们发现：（1）这两个算法对旧类出现完全性的遗忘现象消失了。这主要是通过在训练中增添旧类别样本，使得 finetuning 算法的训练与普通的分类模型训练类似，区别仅在于当前训练数据中新类数据总是明显多于旧类。（2）对于 LwF.SM 算法，相比较 finetuning 算法，其使用的网络固化方法在有旧类参与的情况下，模型对旧类的记忆效果有所提高，而在上一章的实验结果中，LwF.SM 算法的网络固化方法几乎没有起到任何作用。产生这一变化的一个重要原因在于，旧类数据从一定程度上提高了 LwF.SM 在网络固化中所需要软标签的精度。也就是说，尽管 LwF.SM 算法使用的是单头网络，但旧类数据的参与，对旧知识的记忆（网络固化）和新知识的学习（分类器训练）做了一定程度上的解耦。这种解耦的思路，其实不仅可以用于采样旧类样本，也可以通过采用辅助数据来使用。在下一节中，我们在这一方面也做了一些工作。

4.2 基于半监督的类增量学习算法 SS-PCRC

4.2.1 算法优化背景

在上一节，我们使用回放方法对 PCRC 算法进行增强。然而，在有些情况下，比如出于安全考虑，之前用于训练模型的数据无法获得，那么此时就无法使用回放方法。但是，回放方法蕴含的解耦思想，并不只能使用旧类数据才能实现，利用辅助数据，依然可以达到优化模型的效果。本节，我们便尝试使用

辅助数据来优化类增量学习过程中的新旧类别的数据不平衡问题。受半监督学习的启发，考虑到目前网络上存在许多公开图片数据集，那么在类增量学习的过程中，我们不仅可以利用有标签的新数据，同时也可以使用这种辅助数据集获取的数据作为无标签数据来提高模型的性能。具体而言，针对 PCRC 算法中的耦合性问题——同时使用新数据进行旧知识记忆和新知识学习，通过引入的辅助数据集，我们将这两个工作进行了拆解。对于旧知识的记忆，我们倾向于使用辅助数据集来完成；对于新知识的学习，我们倾向于使用新类数据来完成。通过这种方式，新类数据可以让模型获得更好的新类别区分能力，同时辅助数据又可以用来缓解灾难性遗忘问题。通过这种解耦思想，PCRC 的性能得到了进一步提高。我们把这个改进的算法称为 Semi-Supervised PCRC（SS-PCRC）。

4.2.2 SS-PCRC 算法

SS-PCRC 算法相对于 PCRC 算法，主要有两个地方发生了变化：（1）在模型进行增量分类过程中，我们不仅需要之前的旧模型——该模型为当前模型的副本，其参数被冻结，另外，还需要一个用于指导学习新类别的模型，我们称为教师模型——该模型为仅使用新类别数据训练得到的分类模型。通过结合旧模型和教师模型，我们可以对辅助数据集中的图片生成软标签；（2）将新数据和辅助数据进行混合，再进行模型的增量训练，以这样的方式兼顾模型的记忆和学习。针对上面所提到的软标签，我们使用目标向量生成算法 4.1来生成。参与目标向量生成的数据不仅包括辅助数据，也同样包括了新类数据。对于这两类数据，均以旧模型的输出结果作为目标向量在旧类别上的分量。两者在目标向量生成过程中一个很重要的不同点就是：对于新类数据，目标向量在新类别上的分量以独热编码方式进行，而对于辅助数据，该分量为教师模型的输出结果。将新类别对应的分量分情况处理，从而让新类数据侧重新知识学习，辅助数据侧重旧知识的记忆，同时也对新知识起到一定的帮助作用。图 4-3a，4-3b更加直观描述新类数据和辅助数据的目标向量的生成过程。另外，为了使模型训练中能相对均匀接触到新类数据和辅助数据，在将最终的训练数据送入新模型之前，我们有必要对所有的数据进行一次洗牌操作。

在目标向量生成工作结束之后，整个网络的增量训练过程与 PCRC 相同，参见图 3-2。在损失函数值的计算过程中，使用的也是二元交叉熵损失函数 3-2。

算法 4.1 目标向量生成算法

输入:

$f(x, \theta_{old})$: 在旧类别数据 $\{X^1, X^2, ..., X^s\}$ 上训练得到的旧模型

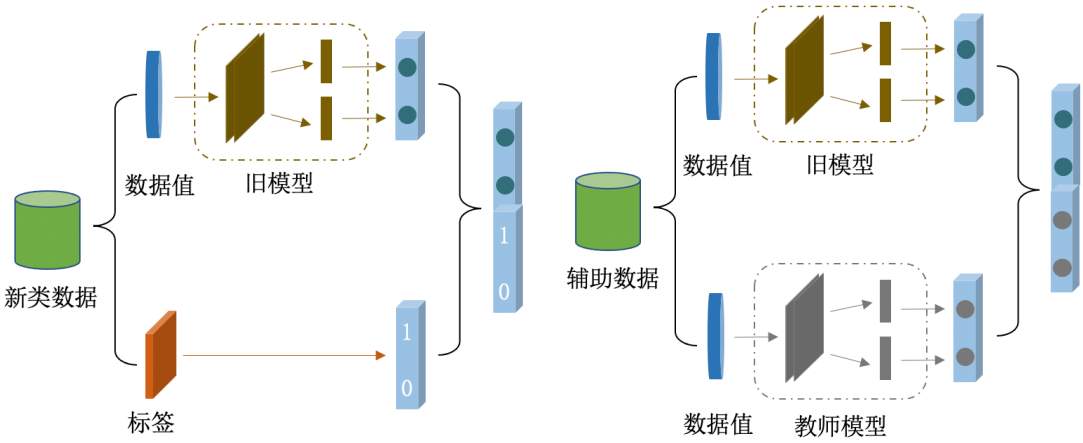
$f(x, \theta_{curr})$: 在新类数据 $\{X^{s+1}, ..., X^{s+t}\}$ 上训练得到的教师模型

$T = \{x_1, x_2, ..., x_n\}$: 将新类数据和辅助数据进行洗牌操作，作为最终的训练数据

输出:

目标向量矩阵 $Y_{(n,s+t)} = [y_1, y_2, ..., y_n]^T$

for $i = 1$ to n **do**
  $\hat{y}_{old} = f(x_i, \theta_{old})$
  $\hat{y}_{curr} = f(x_i, \theta_{curr})$
  $y_i = \vec{0}$
  for $j = 1$ to s **do**
    $y_i[j] = \hat{y}_{old}[j]$
  end for
  if x_i is unlabeled data **then**
    for $j = 1$ to t **do**
      $y_i[s + j] = \hat{y}_{curr}[s + j]$
    end for
  else[The label of x_i is class c_i]
    $y_i[c_i] = 1$
  end if
end for



(a) 新类数据的目标向量生成过程　　(b) 辅助数据的目标向量生成过程

图 4-3: 新类数据和辅助数据的目标向量生成示意图。

不过与 PCRC 算法不同的是：由于两类数据侧重点不同，所以在计算损失值的时候 SS-PCRC 算法需要引入相应的损失权重。为了更直观描述这一计算过程，我们将公式 3-2改写为如下形式：

$$\text{loss} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}\left[-y_i^j \log \hat{y}_i^j - \left(1 - y_i^j\right)\log\left(1 - \hat{y}_i^j\right)\right] \tag{4-1}$$

其中，N 表示所有的样本数，C 表示总的类别数，$\hat{y}_i^j$ 表示第 i 个样本在第 j 个类别上的得分，y_i^j 表示 i 个样本在第 j 个类别上的目标值，也就是上文第 i 个样本对应目标向量的第 j 分量；

进一步，分别考虑辅助数据在旧知识固化方面产生的损失值 loss_1，在新知识学习方面产生的损失值 loss_2，其计算公式分别见公式 4-2， 4-3。

$$\text{loss}_1 = \sum_{i=1}^{N_u} \sum_{j=1}^{S} \left[-y_i^j \log \hat{y}_i^j - \left(1 - y_i^j\right) \log \left(1 - \hat{y}_i^j\right) \right] \tag{4-2}$$

$$\text{loss}_2 = \sum_{i=1}^{N_u} \sum_{j=S+1}^{S+T} \left[-y_i^j \log \hat{y}_i^j - \left(1 - y_i^j\right) \log \left(1 - \hat{y}_i^j\right) \right] \tag{4-3}$$

新类数据在旧知识固化方面产生的损失值 loss_3，在新知识学习方面产生的损失值 loss_4，其计算公式分别见公式 4-4， 4-5。

$$\text{loss}_3 = \sum_{i=1}^{N_l} \sum_{j=1}^{S} \left[-y_i^j \log \hat{y}_i^j - \left(1 - y_i^j\right) \log \left(1 - \hat{y}_i^j\right) \right] \tag{4-4}$$

$$\text{loss}_4 = \sum_{i=1}^{N_l} \sum_{j=S+1}^{S+T} \left[-y_i^j \log \hat{y}_i^j - \left(1 - y_i^j\right) \log \left(1 - \hat{y}_i^j\right) \right] \tag{4-5}$$

最终的损失函数值由上面四项损失值进行权重求和，见公式 4-6

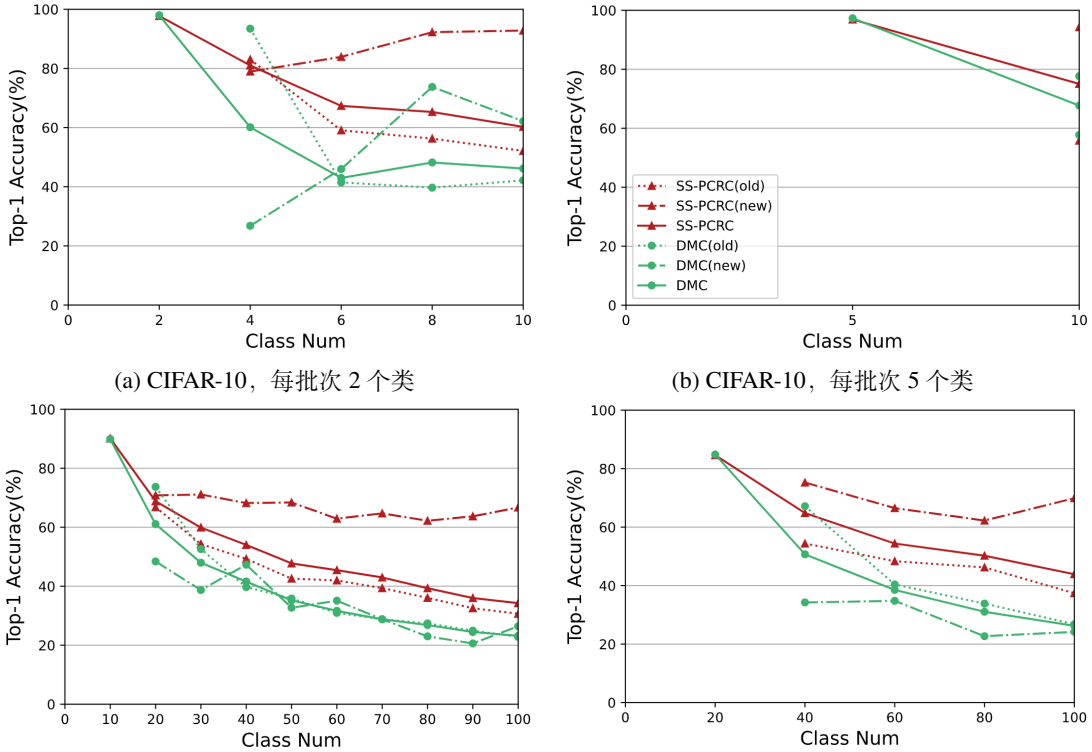
$$\text{loss} = \frac{1}{N_u + N_l} \left[w_1 * \text{loss}_1 + w_2 * \text{loss}_2 + w_3 * \text{loss}_3 + w_4 * \text{loss}_4 \right] \tag{4-6}$$

其中 N_u，N_l 分别表示无标签数据的样本数和新类数据的样本数，S 表示旧类别数目，T 表示新类别数目，w_1、w_2、w_3、w_4 分别表示这 4 部分损失值 loss_1、loss_2、loss_3、loss_4 占比关系，由于辅助数据更侧重旧知识的记忆，那么有 $w_1 > w_2$，而新数据侧重于新知识的学习，那么有 $w_3 < w_4$。

4.2.3 实验与分析

类似于 PCRC 的实验设计，对于 SS-PCRC，我们在 CIFAR-10/100 上进行测试。另外，我们使用了 Tiny-ImageNet-200 作为辅助数据集，在使用前对数据进行了降采样处理[37]，让辅助数据中的图片也调整为 32×32。为了更好地对比 SS-PCRC 算法的实验效果，我们选取了 DMC 算法作为本次实验的对比算法。

在我们的实验中，取 $w_1 : w_2 : w_3 : w_4 = 10 : 1 : 1 : 10$。另外，我们随取抽取 Tiny-ImageNet-200 中的 10 个类别的训练集样本作为辅助数据，其他的参数设置参考 PCRC 算法。



(a) CIFAR-10，每批次 2 个类

(b) CIFAR-10，每批次 5 个类

(c) CIFAR-100，每批次 10 个类

(d) CIFAR-100，每批次 20 个类

图 4–4: SS-PCRC 算法与 DMC 算法在 CIFAR-10/100 上的准确率比较。

为了更加清楚辅助数据集规模对算法的影响，我们又增加了一组对照实验：对于辅助数据集 Tiny-ImageNet-200，我们分别随机抽取 5，10，20 个类别作为辅助数据集，实验结果见表 4–1。

如图 4–4，在 SS-PCRC 算法和 DMC 算法的训练过程中，我们依然将模型在旧类别，新类别，以及所有类别的测试集上做了准确度测试。在 CIFAR-10 数据集，每批次 2 个类别的实验设置中，在 4 分类时，DMC 算法在旧类别上的准确率高于 SS-PCRC 算法，但在后续的增量学习过程中，SS-PCRC 在旧类别上的准确率均超过 10%。对于新类别，两者的准确率差距十分明显。在每批次 5 个类别的实验设置中，两者在旧类别上的表现比较接近，准确率在 56% 左右。但在新类别上，SS-PCRC 的准确率超过 DMC 算法 20% 以上。另外，SS-PCRC 相对 DMC 的整体准确率优势缩小到 7% 左右。在 CIFAR-100 数据集，每批次 10 个类别的实验设置中，尽管 DMC 算法在旧类别上已经与 SS-PCRC 有超过 8% 的

表 4–1: 使用不同类别数目的辅助数据，DMC 与 SS-PCRC 算法在旧类别、新类别，以及所有类别上的平均准确率。

算法	DMC				SS-PCRC			
数据集	CIFAR-10		CIFAR-100		CIFAR-10		CIFAR-100	
每批次类别数	2	5	10	20	2	5	10	20
旧类别	0.5144	0.8476	0.3367	0.4245	0.5944	0.4878	0.4227	0.4567
新类别	0.5003	0.4148	0.2934	0.2845	0.8851	0.9498	0.6866	0.6880
所有类别	0.4680	0.6312	0.3128	0.3634	0.6710	0.7188	0.4696	0.5278

(a) 抽取 Tiny-ImageNet-200 中的 5 个类别构成辅助数据集。

算法	DMC				SS-PCRC			
数据集	CIFAR-10		CIFAR-100		CIFAR-10		CIFAR-100	
每批次类别数	2	5	10	20	2	5	10	20
旧类别	0.5419	0.7766	0.3742	0.4205	0.6263	0.5580	0.4372	0.4660
新类别	0.5218	0.5774	0.3344	0.2899	0.8698	0.9428	0.6652	0.6848
所有类别	0.4936	0.6770	0.3564	0.3664	0.6847	0.7504	0.4761	0.5334

(b) 抽取 Tiny-ImageNet-200 中的 10 个类别构成辅助数据集。

算法	DMC				SS-PCRC			
数据集	CIFAR-10		CIFAR-100		CIFAR-10		CIFAR-100	
每批次类别数	2	5	10	20	2	5	10	20
旧类别	0.5554	0.8518	0.4210	0.4582	0.6566	0.5690	0.4605	0.4845
新类别	0.5620	0.6246	0.4123	0.3676	0.8392	0.9414	0.6462	0.6821
所有类别	0.5169	0.7382	0.4106	0.4187	0.6918	0.7552	0.4903	0.5445

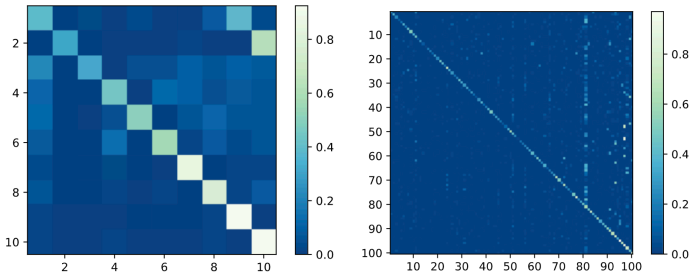
(c) 抽取 Tiny-ImageNet-200 中的 20 个类别构成辅助数据集。

明显差距，但是在新类别上的表现更差。所以在整个学习过程中，两者整体准确率的差距超过 12%。在每批次 20 个类别的实验设置中，DMC 算法在新类别上的准确率差距再次拉大，导致最终整体准确率与 SS-PCRC 相差 15% 以上。

通过对比可以发现，在每批次类别数相同的情况下，SS-PCRC 模型相比较 DMC 模型的精度有较明显提升。主要原因在于 DMC 算法在进行新类别训练的过程中，完全使用教师模型，通过知识蒸馏手段来进行参数调整，而相对于 SS-PCRC 算法，我们在进行新知识学习的时候直接使用训练数据，从而使得学习过程更加高效，这也就导致如果 DMC 想获得与 SS-PCRC 算法比较接近的效果，那么要使用的辅助数据集规模就要大很多。

通过表 4-1所展示的两个算法对应的平均准确率，可以看出：（1）辅助数据集的规模越小，SS-PCRC 在新类别准确率上相对于 DMC 的优势更加明显。（2）随着我们增大辅助数据集规模，模型的记忆能力越来越强。DMC 算法对新类别，旧类别的预测准确率均有提高，但是对于 SS-PCRC 算法，在旧类别准确率提高的同时，新类别的准确率却稍有降低。这一现象的主要原因在于：对于 DMC 算法而言，其所有数据在后续的增量学习阶段，仅辅助数据作为蒸馏学习过程中用到的数据，共同用于新类学习与旧类固化，所以数据的使用对于这两个学习目标而言并没有倾向性。但对于 SS-PCRC 算法，辅助数据集的增加，直接导致在学习过程中，有更多的数据用于模型的记忆。即算法对旧知识记忆的侧重程度加大，虽然辅助数据也同时用于新类别学习，但是其权重很低。同时新类数据样本数不变，所以网络参数最终更偏向于记忆旧知识，模型在新类别上的学习程度被削弱。

最后，在抽取 Tiny-ImageNet-200 数据集中的 10 个类别作为辅助数据集时，SS-PCRC 算法的混淆矩阵如图 4-5。与第三章的 PCRC 算法的混淆矩阵对比可以看出，SS-PCRC 算法对应的混淆矩阵，其对角线上的白线更加连续，更加明显，这说明了我们使用辅助数据对 PCRC 算法进行解耦是有效的。



(a) CIFAR-10，每批次 2 个类　(b) CIFAR-100，每批次 10 个类

图 4-5: SS-PCRC 算法的混淆矩阵

4.2.4 SS-PCRC 算法小结

本节我们从数据的角度分析了原有 PCRC 算法可以提升的地方，并提出了基于半监督学习的算法 SS-PCRC。在 SS-PCRC 算法中，我们精心设计了目标向量生成算法，来分别对新类数据和辅助数据进行处理。另外，由于辅助数据的引入，我们对损失函数的计算过程进行了修改，通过引入权重系数来平衡两类数据在模型参数更新过程中的使用。最后，我们将同样使用辅助数据集的 DMC 算

法作为对比算法，实验结果表明 SS-PCRC 算法对辅助数据的利用效果更好。需要指出的是，SS-PCRC 算法以牺牲时间来提升模型精度，所以其适用范围不适合实时场景。

4.3 平衡化的类增量学习算法 BPCRC

如上文所说的，通过让模型在增量式学习的过程中能接触部分旧类样本，或者使用辅助数据，从而来有效地提高最终模型的整体分类准确率。当然，这是从数据角度来缓解数据不平衡问题，也是一种相对直观的方式。但是这种方式在持续性的流数据场景中，必然会导致一个问题，即随着总类别数的增多，如果在总的存储空间一定的情况下，那么每个类别的采样数量会不断减少，或者每个类别的采样数量不变，那么会导致总的存储成本不断提高。如果使用辅助数据，那么模型的更新速度就会变慢。本节提出的算法是在 PCRC 算法的基础进行改进，得到了算法 BPCRC（Balanced PCRC）。该算法在不对旧类别数据进行缓存的情况下，通过对新类数据的另一种利用方式来缓解数据不平衡问题。

4.3.1 算法优化背景

在实际的新类学习过程中，并不是所有的新类数据对网络参数的学习贡献程度是一样的。如图 4-6，对于绿色分界线里面的正样本，距离分界线越近的为关键正样本；对于绿色分界线外面的负样本，距离分界线越近的为关键负样本。其他的正负样本对网络参数的训练影响就远不如这些关键样本。

对于关键性的数据，我们提高网络对其敏感度，即增大网络参数在这些数据上的调整；而对于非关键性数据，减小网络参数在这些数据上的调整。通过这种方式，一方面，网络已经存储了旧类信息，通过减小整体参数调整程度，从而来提高网络对旧类知识的记忆能力；另一方面，由于网络依然会根据新类别中的关键数据进行参数学习，所以依然可以继续有效学习新类知识。

4.3.2 BPCRC 算法

BPCRC 整个网络结构是与 PCRC 是相同，不过由于考虑到新类数据的关键程度，我们不再直接使用二元交叉熵损失函数来训练分类器单元。受 focal loss[38]

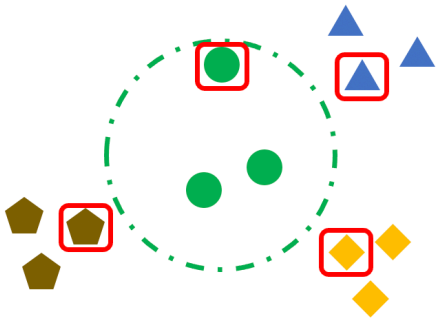


图 4-6: 关键性正负样本示意图，由红色方框圈出的为关键性样本。

启发，我们使用下面的损失函数来训练每个二元分类器。

$$\mathcal{L}_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \begin{cases} -(1-\hat{y}_i)^2 \log \hat{y}_i & y_i = 1 \\ -\hat{y}_i^2 \log(1-\hat{y}_i) & y_i = 0 \end{cases}. \tag{4-7}$$

为便于对比讨论，可改写为如下形式

$$\mathcal{L}_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = -(y_i - \hat{y}_i)^2 \left[y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i) \right] \tag{4-8}$$

与公式 3-2比较，公式 4-8前面的系数 $(y_i - \hat{y}_i)^2$ 用来细化数据的关键程度。对于容易区分的正样本，$\hat{y}_i$ 就会接近 1，或者对于容易区分的副样本，$\hat{y}_i$ 就会接近 0，总之都会减少参数的调整幅度。在我们的实验中，我们把 focusing parameter 设置为 2。

由于 BPCRC 学习损失函数发生了调整，为了更好地固化之前网络保存的旧类信息，我们采用了基于欧式距离来对旧类知识进行蒸馏，即：

$$\mathcal{L}_i(\boldsymbol{x}, \boldsymbol{\theta}^{k-1}, \boldsymbol{\theta}) = (d(\boldsymbol{c}_i, f_i(\boldsymbol{x}; \boldsymbol{\theta})) - d(\boldsymbol{c}_i, f_i(\boldsymbol{x}; \boldsymbol{\theta}^{k-1})))^2. \tag{4-9}$$

在BPCRC的实验中，由于使用了新的损失函数，与公式3-3对比，使用公式4-9能取得更好的实验结果。记

$$\boldsymbol{y}^{k-1} = [d(\boldsymbol{c}_0, f_0(\boldsymbol{x}; \boldsymbol{\theta})), \ldots, d(\boldsymbol{c}_n, f_n(\boldsymbol{x}; \boldsymbol{\theta}))]^T \tag{4-10}$$

其中 $n = |C_k| + |S_k|$ 即为总的类别数，C_k，S_k 定义分别参考章节 3.1以及公式 3-5。最终 BPCRC 的损失函数见公式 3-6。其中的 $\mathcal{L}_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$ 和 $\mathcal{L}_j(\boldsymbol{x}, \boldsymbol{y}^{k-1}, \boldsymbol{\theta})$ 分别由

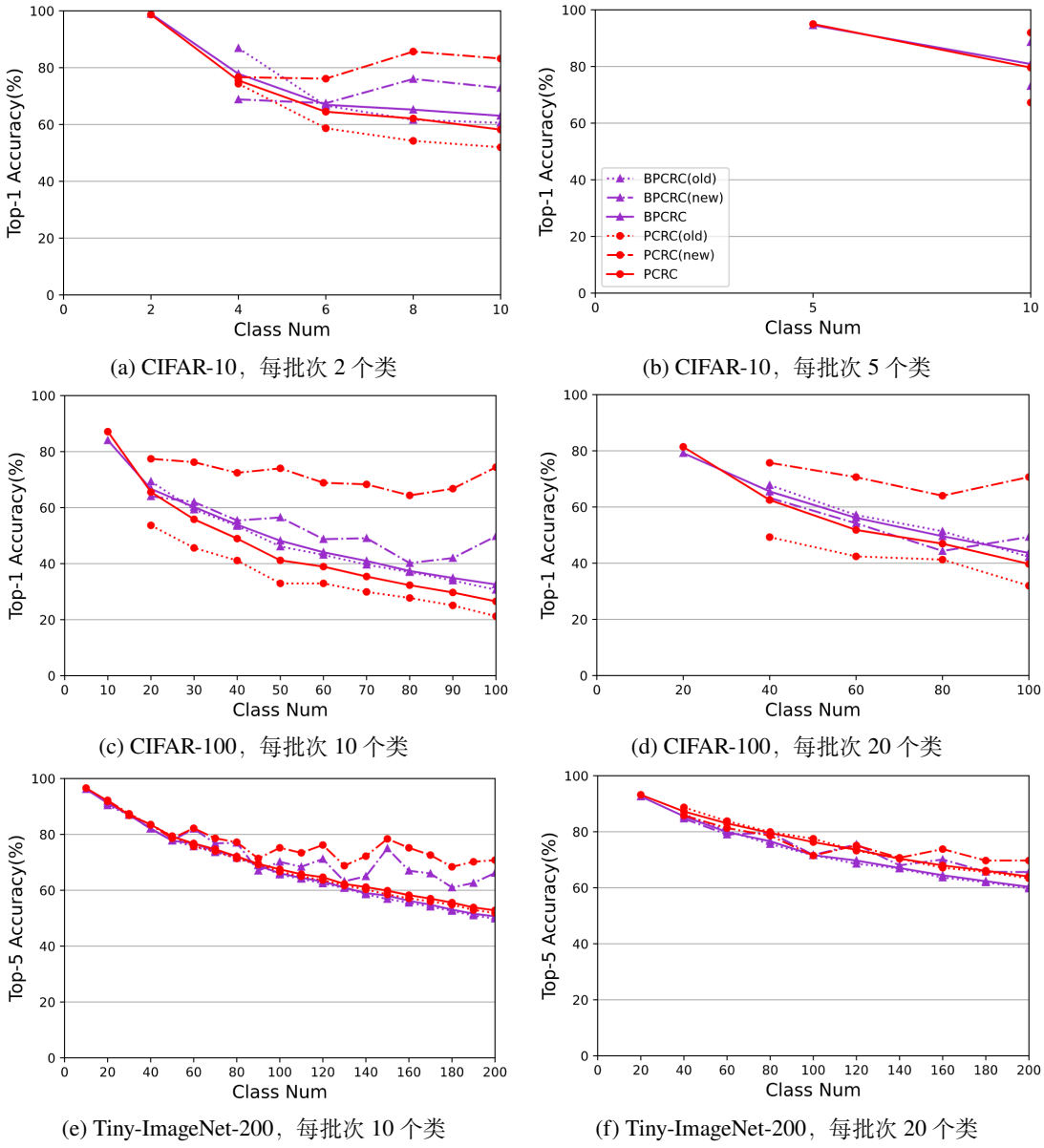
公式 4-7，4-9得到。

4.3.3 实验与分析

类似于 PCRC 的实验设计，对于 BPCRC，我们依然在 CIFAR-10/100,Tiny-ImageNet-200 这三个数据集进行测试。同样的，数据集依然按照 3-1进行设置。对于 BPCRC 的参数设置，公式 4-7中的参数 α 在 CIFAR-10/100 上均设置为 1，在 Tiny-ImageNet-200 上设置为 0.5；公式 3-6正则化系数 λ 设置为 0.01。另外 BPCRC 训练的学习率调整策略与 PCRC 相同。对于每个训练批次，模型无法访问到之前的旧类数据。在模型训练结束之后，我们会分别计算 BPCRC 模型在旧类数据，新类数据，以及全体数据上的预测精度，并报告平均准确率。

如图 4-7，在 CIFAR-10 数据集，每批次 2 个类别的实验设置中，BPCRC 算法在旧类别上的准确率相比 PCRC 算法有 8% 左右的优势，但是在新类别上，其分类准确率始终比 PCRC 低 9%，但由于旧类别占比越来越大，所以 BPCRC 的整体准确率与 PCRC 越拉越大。每批次 5 个类别的实验设置中，BPCRC 与 PCRC 两者不论在新类别上的准确率，还是在旧类别上的准确率，之间的差距都有所减小。最终整体准确率 BPCRC 仅高出 1%。在 CIFAR-100 数据集，每批次 10 个类别的实验设置中，两个算法在新旧类别上的分类表现对比十分明显。具体而言，在新类别上，BPCRC 算法的分类准确率低 PCRC 算法 20% 以上，但是在旧类别上，又高于 PCRC 算法 10%。在每批次 20 个类别的实验设置中，两个算法在新旧类别上的分类准确率差距也同样有所减小，最终的整体分类准确率差距未超过 4%。在 Tiny-ImageNet-200 数据集，每批次 10 个类别的实验设置中，PCRC 算法在新类别上的准确率始终高于 BPCRC 算法。另外，在旧类别上，PCRC 算法也有 2% 的优势。在每批次 10 个类别的实验设置中，PCRC 相对于 BPCRC 的优势被拉大到了 3% 左右。也就是说，在 Tiny-ImageNet-200 数据集上，BPCRC 算法性能表现弱于 PCRC 算法。

与 PCRC 算法相比较，在每批次类别数较少时，如在 CIFAR-10/100 数据集，BPCRC 相比 PCRC 的优势更大。而在更长时间的增量学习过程中，BPCRC 所存在的劣势就凸显出来了，尤其是每批次类别数设置较多的时候更是如此。主要原因在于新类的学习主要依赖于关键数据，导致在新类参数调整的时候，相比 PCRC 而言，BPCRC 中每个类的特征点在超平面内部的分布相对来说比较分散，



(a) CIFAR-10，每批次 2 个类

(b) CIFAR-10，每批次 5 个类

(c) CIFAR-100，每批次 10 个类

(d) CIFAR-100，每批次 20 个类

(e) Tiny-ImageNet-200，每批次 10 个类

(f) Tiny-ImageNet-200，每批次 20 个类

图 4-7: BPCRC 算法与 PCRC 算法在 CIFAR-10/100,Tiny-ImageNet-200 上的准确率比较。

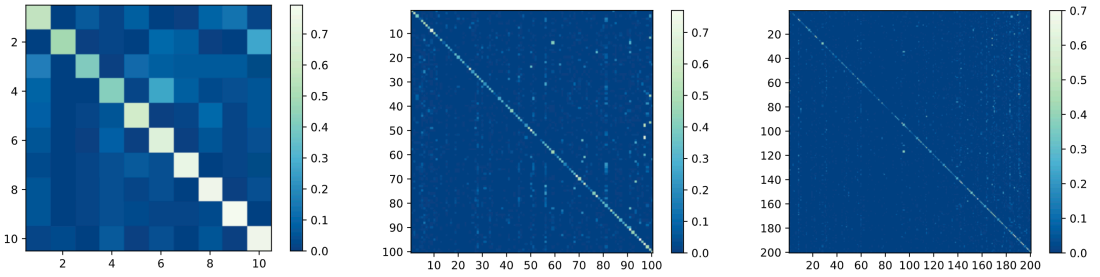
从而导致每个类别学习到的原型较差。所以随着类别数的增多，后续批次中新类别的学习程度被削弱，影响了最终的模型性能。

我们统计了 BPCRC 算法在旧类别，新类别，所有类别上的平均准确率，见表 4-2。与图 4-7结合起来，不难看出，BPCRC 算法在减少网络参数调整的同时，也对新类的学习能力产生了影响。也就是说，BPCRC 会提高网络对旧知识的记忆能力，但由于降低了整体参数的调整，导致网络向新类数据拟合不足。

最后，我们也绘制了 BPCRC 算法对应在各个实验设置下的混淆矩阵，如图 4-8。可以看到，BPCRC 最终大多数预测值分布在了对角线上。与 PCRC 的混淆矩阵 3-7对比还可以发现，BPCRC 在非对角线区域，泛白点分布得更加均

表 4-2: 增量学习过程中 BPCRC 模型在旧类别、新类别，以及所有类别上的平均准确率。

数据集	CIFAR-10		CIFAR-100		Tiny-ImageNet-200	
每批次类别数	2	5	10	20	10	20
旧类别	0.6894	0.7312	0.4585	0.5464	0.6625	0.7051
新类别	0.7130	0.8862	0.5198	0.5279	0.7249	0.7333
所有类别	0.6826	0.8087	0.4655	0.5374	0.6674	0.7084



(a) CIFAR-10，每批次 2 个类　(b) CIFAR-100，每批次 10 个类　(c) Tiny-ImageNet-200，每批次 10 个类

图 4-8: BPCRC 算法的混淆矩阵

匀。这也就意味着，对于 BPCRC 模型，既有较多的新类别预测为旧类别，也有较多的旧类别预测为新类别。前者说明网络对新类的学习能力降低，后者就是灾难性遗忘问题。在 Tiny-ImageNet-200 数据集上，BPCRC 混淆矩阵的对角线连续性更好，但是相比 PCRC 算法，网络对更多新类别数据预测失败（对角线左下方有大量白点分布），所以最终整体的模型性能反而低一些。

4.3.4 BPCRC 算法小结

本节首先分析了增量学习过程中，新类数据中关键性数据和非关键性数据对模型参数影响的差异性。为了让模型能在继续学习新类知识，调整底层共享模块参数的时候，尽量减小对旧知识的遗忘程度，我们通过增强占比少的关键数据的损失权重，降低占比多的非关键数据的损失权重，来降低整体网络参数的调整程度。通过实验验证，我们看到相比 PCRC 模型，BPCRC 模型对旧类知识的记忆能力更强，不过也可以看到，在更长时间的类增量学习过程中，由于大量的非关键性数据对网络参数影响的减弱，BPCRC 模型的整体表现稍微弱于 PCRC 模型。

4.4 本章总结

本章首先通过回放方法来对 PCRC 算法进行增强，实验结果充分说明了回放方法在 PCRC 算法上的有效性。接下来，我们分别从两个角度对第三章提出的 PCRC 算法做进一步改进。(1) PCRC 算法在模型训练过程中，每个阶段模型仅能接触到新类别数据，而这些数据既要用于旧知识的记忆，又要用于新知识的学习。这导致了模型在记忆旧类知识和学习新类知识上的冲突，从而出现灾难性遗忘问题。这种学习与记忆之间的耦合性制约了最终模型的整体性能。本章提出使用辅助数据来对 PCRC 算法进行解耦，得到了改进算法 SS-PCRC。(2) 虽然训练数据中关键性数据仅占小部分，但对模型的参数学习更加重要。针对这一问题，本章提出了改进算法 BPCRC，通过提高网络参数在关键性数据上的调整程度，同时弱化网络在其他非关键性数据的学习来优化 PCRC 算法。

与 DMC 算法完全通过辅助数据进行知识蒸馏的方式不同，SS-PCRC 算法更有效利用了辅助数据和新类数据，即让辅助数据侧重模型的记忆，让新类数据侧重模型的学习。对于非关键性数据，BPCRC 算法降低了网络参数的调整力度。随着类别数的增多，非关键性数据量越来越大，模型对新类别的拟合能力相对被削弱，导致相对于 PCRC 算法的优势被缩小。

在时间方面，SS-PCRC 算法由于需要额外训练教师模型，所以导致模型的更新时间变长；如果不考虑损失函数的计算差异，BPCRC 算法并没有增加复杂度。在实时性不高的场景中，SS-PCRC 便可以通过牺牲模型更新的及时性来取得更高的准确率；而 BPCRC 算法更适合非长期的增量式分类学习场景。

第五章　水声目标识别场景下的类增量学习

本章将算法应用到一个实际的水声目标识别项目，考虑到实际舰船噪声样本数极少，且对模型的实时性要求较高，本章选择更适合该场景的 PCRC 算法来解决模型增量式分类舰船噪声的问题，并使用回放方法来增强模型性能。最后，本章通过实验说明 PCRC 算法在落地方面的优势。

5.1　水声目标识别

随着人类在海洋中的活动日益频繁，水声通信技术得到越来越多的应用，如海洋搜救、渔业生产作业等。在各种应用场景中，存在不同的水声信号，例如主动声纳，舰船辐射噪声和水声通信信号等。水声接受传感器往往能接受各种水声信号，如何有效地区分出不同的水声信号是一个至关重要的问题。当初步解决水声信号分类问题后，如何利用这些水声信号是目前研究的热点问题。其中，一个重要的研究方向是利用水声信号识别不同物体。不同的舰船类型所发出的水声信号具有特异性，如舰船辐射噪声，使得提取水声信号特征识别目标成为可能。

海洋目标识别是海洋装备发展的关键技术之一。目前，海洋目标识别大多基于时频探测特征，海洋环境的复杂性会使信号受到环境噪声、信道衰落的影响，造成低信噪比下的局部数据缺失，严重影响海洋目标信号的特征提取和识别的性能。本项目通过借助深度神经网络强大的特征提取能力，来设计模型实现水声目标信号的特征自动提取以及分类识别，希望借此来为海洋目标识别提供新思路和新方法。其研究将使目标识别技术更适应动态的环境，促进装备更加信息化、自动化与智能化。

在海洋目标识别任务中，我们通过深度神经网络的方法，对采集到的海洋环境下的舰船辐射噪声信号进行识别。考虑到随着时间的发展，会遇到新型的舰船，这也就意味着原有的识别模块需要增量式对这些新的辐射噪声信号进行

分类学习，而 PCRC 算法正适合解决这个子问题。一方面，由于公开的舰船噪声极少，所以我们无法通过 SS-PCRC 算法来优化模型性能。另一方面，考虑到我们实际采集的数据量很少，如果使用 BPCRC 算法训练模型，反而会削弱模型对新类的学习效果。最终，我们考虑使用回放方法来对 PCRC 算法进行增强。

5.2 水声信号的产生机理和特点

船舶辐射噪声主要由机械噪声、螺旋桨噪声和水动力噪声三部分组成，其中机械噪声和螺旋桨噪声是辐射噪声最主要的两个成分。由船上机械产生的噪声是机械噪声；螺旋桨噪声是一种混合型的噪声，它与机械噪声和水动力噪声有共同的特征和共同的源；水动力噪声是由不规则的水流流过在海中航行的舰船产生的辐射噪声和由水动力过程的变化引起的噪声。对于柴油机-电机推进的舰船，机械噪声的来源有主机和辅机。螺旋桨噪声则主要由螺旋桨引起的船壳共振，以及螺旋桨上或其附近的空化产生。水动力噪声包括水流辐射噪声、空腔、板和附件的共振，以及在支柱和附件的空化。在正常的情况下，水动力噪声不重要，容易被机械噪声和螺旋桨噪声所掩盖。但在特殊的情况下，如在结构部件或空腔被激励成线谱的共振源时，水动力噪声成为线谱的主要噪声源。

由于各种噪声成分产生的机理不尽相同，且各类水中目标的自身动力系统、机械装置结构和所处工作环境不同，导致辐射噪声谱线形状比较复杂。但同一类型的舰船辐射噪声具有相似性，不同类型舰船辐射噪声具有差异性，正是利用这种差异才使得水下目标的识别成为可能。船舶辐射噪声携带大量能反映舰船本质的特征信息，是舰船目标识别的重要研究对象。

5.3 将 PCRC 算法运用到水声目标识别场景

5.3.1 数据预处理

在这个课题中，我们发现实测的舰船噪声数据存在下面的特点：（1）原始数据数据点数多，直接将原始数据输入神经网络模型，会导致模型的输入维度过大，增加模型训练的难度。这里我们调研了声音信号识别的常用预处理手段，选择了如下的处理方案：先通过窗口切片法，将原始数据切片成若干较小的时

间片段，然后为了进一步利用时域信号的时频特征，我们进一步提取了数据的梅尔倒谱系数特征，将其作为模型的输入特征；（2）采集的实际数据存在数据量少的问题。我们的实验数据和目前常用的语音信号识别常用的 VoxCeleb 数据集[39] 相比，标注数据的规模小了好几个数量级。为了缓解这个问题，我们采用了时间序列问题常用的数据增强手段——窗口切片法来进行数据增强。（3）采集的实际数据存在类间数据不平衡的问题，有的船只类型噪声有几十条，有的却只有几条，直接进行训练会导致模型倾向于将样本预测为多数类别。为了解决这个问题，我们试验了多种数据过采样手段，最后采取了直接过采样方法来缓解类间数据不平衡的问题。

具体而言，首先我们对原始信号首先进行梅尔频率倒谱系数（MFCC）变换，得到对应的 MFCC 特征；然后在获得了两维的 MFCC 特征之后，将其按时间维度进行切片；最后，将上一步通过窗口切片得到的若干样本进行数据划分，将 80% 切片数据划分到训练集中，后 20% 的切片数据划分到测试集中。经过数据增强后，我们在原来小样本数据的基础上，获得了更多的实验样本来进行网络模型的参数训练。

5.3.2 网络设计

根据 PCRC 算法框架 3-1，可以看到，针对不同类别的数据，我们需要选择有针对性的底层网络作为共享层模块。对于声音信号，相比较 ResNet 模块构建的网络，使用 TDNN 层[40]，即时延网络层来构建的网络，在我们的测试中有更好的实验结果。

TDNN 网络的结构设计受到了神经网络音素识别任务的启发。对于使用一般的神经网络做音素识别，其网络结构设计如图 5-1，输入层的数据为每一帧的特征向量（如 MFCC 特征），但是这种传统网络未考虑连续帧之间的相关关系，所以效果不佳。

不同于传统神经网络，TDNN 网络考虑到了多帧数据，这样就可以利用到多帧的特征。如图 5-2所示，输入层根据延时设置，该图中为 2，那么最终的输入层特征向量是这相邻 3 帧向量的加权特征。

通过上面的分析可以看到，时延网络的作用和卷积层的作用类似，时延网络通过时间延迟步参数，来控制网络所提取的时间片特征的长短，这里我们使

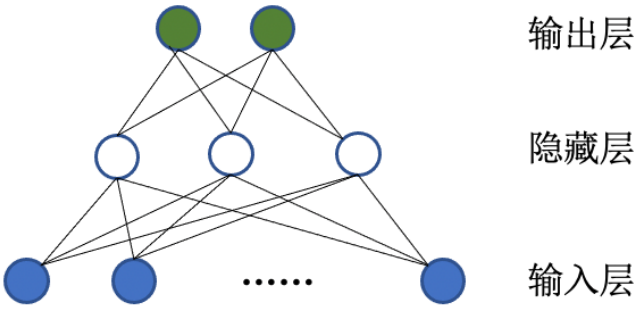


图 5-1: 传统神经网络识别音素

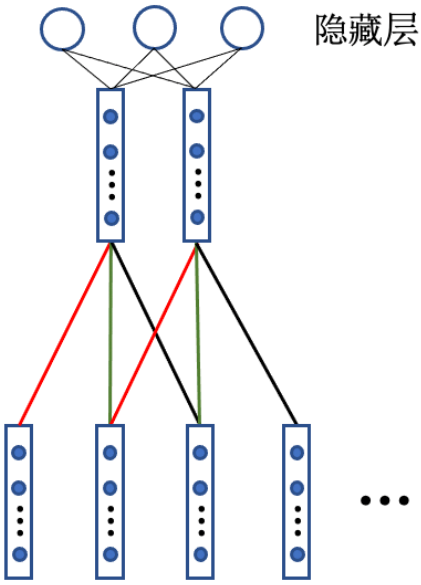


图 5-2: TDNN 神经网络识别音素

用的时间延迟步参数较小，都在 5 以下，因此可以认为前 5 层 TDNN 层实际上提取了信号的多级短时特征，因为在一段音频信号中舰船辐射噪声信号的分布是稀疏且不连续的，所以时延网络可以很好地从音频中提取辐射噪声的特征。

我们在获得了时间序列的多级短时特征之后，需要一个汇总的阶段，来获得整个时间信号的深度特征。一种常用的方法就是通过平均池化[41] 操作，即把前面提取到的短时特征在时间维度上进行平均，得到的平均向量作为整个信号的深度特征。但是这种方法存在一个弊端，就是不同的时间片对最终的预测结果的贡献是不同的，直接平均相当于是淡化了有效特征，这对我们的分类任务是不利的。因此我们设计了一种基于注意力机制[42] 的池化层，首先给每个时间片对应的短时特征一个可学习权重，然后计算这些短时特征的加权平均。这种方法会在训练过程中逐渐提高有效片段的权重，逐渐降低无用片段的权重，相当于自动地滤除了信号中的无效片段。

表 5-1: 基于注意力机制的多级时延网络结构

层数	层类别	特征维度
1-4	TDNN	512
5	TDNN	1500
6	Pooling	3000
7	FC	512
8	FC	64

我们最终根据 TDNN 层搭建了 PCRC 中的共享层，其网络结构见表 5-1。时延网络模型的输入为窗口切片得到的噪声信号的 MFCC 特征，其输出作为 PCRC 算法中头部网络的输入。

最后对于 PCRC 的头部网络，我们使用由一个隐藏层和一个输出层的构建网络结构。在我们的实验设置中，头部网络的隐藏层神经元数量设置为 100。

5.3.3 实验与分析

对于我们需要分类的所有舰船噪声数据，包含了 6 个大类，每个大类又包含了若干小类。我们以每批次 2 个类，分 3 批次设置增量式分类实验。模型训练参数设置如下：迭代次数设置为 50，批次样本数设置为 32，学习率设置为 3e-4，同时使用 Adam 优化器调整参数。最后，需要指出的是，由于采样获得的样本具有显著的不平衡问题，为了获得更好的预测效果，我们对同一个舰船噪声的多个测试切片采用投票方式，把预测次数最多的类别作为作为最终的预测结果。另外，LwF.MC 算法使用相同的底层网络结构来训练模型，与 PCRC 算法进行对比，最终结果见图 5-3。

如图 5-3所示，在 4 分类的时候，LwF.MC 算法在新类别上有更强的分类能力，但是与此同时，其网络产生了明显的遗忘问题。PCRC 算法在 4 分类的时候，为了保留模型对初始的两类语音信号的分辨能力，所以参数向后两类语音信号拟合幅度降低，其分辨能力便弱于 LwF.MC 算法。也正是 PCRC 算法更有效地缓解灾难性遗忘问题，所以在 6 分类学习的时候，PCRC 算法训练得到的模型性能与 LwF.MC 算法进一步拉大。实验结果表明，在这种数据极不平衡，训练样本数偏少的情况下，PCRC 算法的效果明显优于经典算法 LwF.MC。

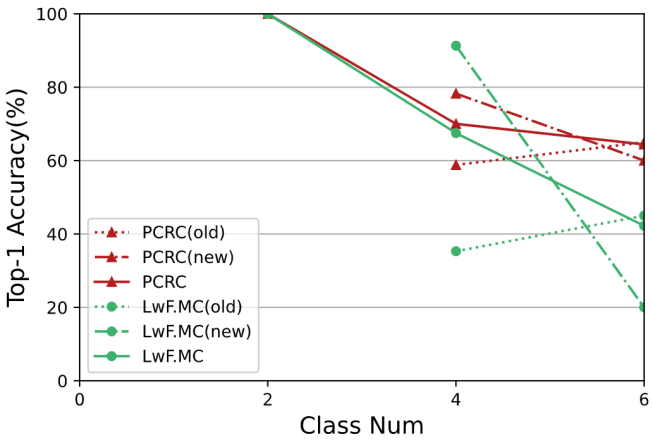


图 5-3: 使用舰船噪声数据，PCRC 算法与 LwF.MC 算法在新类别，旧类别以及所有类别的分类准确率。

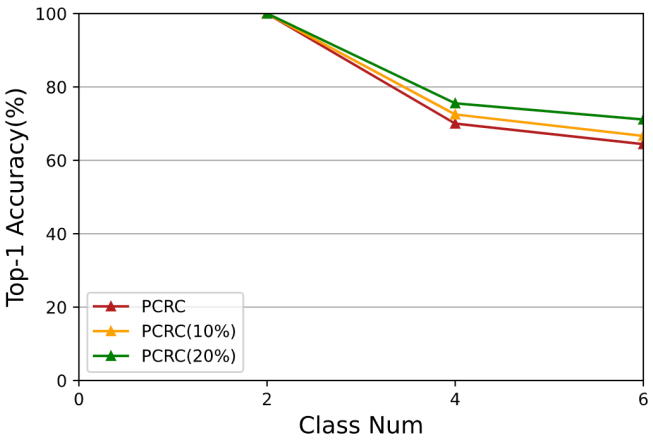


图 5-4: 以回放方式增强 PCRC 算法，网络在所有类别上的分类准确率。括号中的百分数表示采样比例。

另外，我们也构造了 PCRC 算法在使用回放方法情况下的实验，分别采样 10%，20% 的旧类别训练数据来提升模型的性能，最终网络在所有类别上的分类准确率如图 5-4所示。将算法最终训练的结果与不使用回放方法的情况进行对比，可以看到，如在第三章讨论的那样，在实际算法使用中，如果条件允许，即模型在训练可以接触到旧类样本，那么通过存储部分旧类数据的方式对后面模型最终性能的提升还是十分有效的。在这里，通过采样 20% 的旧类样本，网络精度提升了 7%。另外，需要指出的是，因为在该项目中每一类语音的训练数据较少，所以采样比例从 10% 提升到 20% 对模型性能的提升作用依然比较明显。

5.4 本章总结

本章在一个水声目标识别项目中使用 PCRC 算法来解决模型增量式分类舰船噪声的子问题。根据实际的语音数据，我们重新设计了 PCRC 算法需要使用的底层网络。从最后与 LwF.MC 算法的结果对比可以看出，PCRC 算法能有效缓解灾难性遗忘问题，且具有较好的落地效果。

第六章　总结与展望

本章首先对本文的研究工作进行总结，然后对类增量学习在未来的研究方向进行展望。

6.1　本文工作总结

本文首先对类增量学习领域一直存在的灾难性遗忘问题进行了分析，指出了 softmax 抑制问题，进而说明在类增量学习过程中，模型使用单头网络设计并不合适。而对于多头网络，由于在类增量学习领域无法像任务增量学习那样，模型可以使用任务描述符来确定具体的头部输出作为最终的预测结果，就导致最终的网络预测需要综合所有头部的输出来产生预测值，在这个过程中，便会出现头部网络之间的相互混淆问题。为了缓解该问题，我们需要优化网络的特征提取器。具体而言，我们需要网络提取的特征具有两大优良特性：（1）类内聚敛性（2）类间分离性。经过分析，我们发现使用基于原型的分类器能更好地满足要求。另外，对于多头网络的使用，我们又提出了基于任务分解的思路，并比较了两种方案：（1）一个头部网络负责一个类别（2）一个头部网络负责一批类别。通过实验的方式，发现第一种方案更优。

为了模型的使用范围更广，性能更优良。我们设计的 PCRC 算法在不修改模型结构的情况下，可以通过回放方法来对模型训练效果进行增强。通过实验，我们发现少量的旧类样本对缓解灾难性遗忘问题起到有效帮助。这也说明，在能考虑通过牺牲空间来换取模型性能的情况下，回放方法是十分可取的。进一步，我们又从训练数据方面对 PCRC 算法进行分析与优化。

（1）考虑到一些边缘智能体在更新旧模型的时候，可能无法接触到之前的旧类样本。那么在这种情况下，我们可以尝试使用辅助数据集，通过让辅助数据侧重旧知识的记忆，新类数据侧重新知识的学习，来解耦 PCRC 算法中的瓶颈问题——新类数据同时用于模型的记忆与学习。根据这个思想，我们提出了算法 SS-PCRC。

（2）在每个增量学习阶段，我们根据训练数据中的每个样本对模型参数更新

的重要性，将数据划分为关键性数据和非关键性数据。通过强化模型在关键性数据上的参数调整，同时弱化其他非关键性数据的作用，从而缓解模型在增量学习过程中大幅度的参数调整问题。根据这个思想，我们提出了算法 BPCRC。

最后，我们在一个水声目标识别项目中使用 PCRC 算法来解决模型增量式分类舰船噪声的子问题，在采集的实际数据中，通过与 LwF.MC 算法比较，有力地验证了 PCRC 算法在落地方面的优越性。

6.2 未来研究展望

在未来的研究中，本文还希望从以下两个方面来继续优化本文提出的类增量学习算法：

（1）对模型中的不同参数，进行有权重的调整。目前本文提出算法，在模型参数学习过程中，并不考虑参数对旧类别的重要性，而模型在旧类别上的性能对这些重要参数的调整十分敏感。因此，未来我们希望能够在算法中考虑模型参数的重要性，进行精细化的调整。

（2）在特征空间进行网络权重固化。目前本文提出的算法，使用每个分头网络的得分进行网络固化，而通过这种方式来记忆旧知识只是满足了必要性要求。实际上，如果模型能够在特征空间上保持旧类特征，便能够更加有效地克服灾难性遗忘问题。因此，未来我们希望能够改进算法的网络权重固化模块。

参考文献

[1] REBUFFI S, KOLESNIKOV A, SPERL G, et al. icarl: Incremental classifier and representation learning[J]. computer vision and pattern recognition, 2017: 5533-5542.

[2] MCCLOSKEY M, COHEN N J. Catastrophic interference in connectionist networks: The sequential learning problem[M]//Psychology of learning and motivation: volume 24. S.l.: Elsevier, 1989: 109-165.

[3] KIRKPATRICK J, PASCANU R, RABINOWITZ N, et al. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the national academy of sciences, 2017, 114(13):3521-3526.

[4] ROBINS A, MCCALLUM S. Catastrophic forgetting and the pseudorehearsal solution in hopfield-type networks[J]. Connection Science, 1998, 10(2):121-135.

[5] ZENKE F, POOLE B, GANGULI S. Continual learning through synaptic intelligence[J]. international conference on machine learning, 2017, 8:3987-3995.

[6] LI Z, HOIEM D. Learning without forgetting[J]. european conference on computer vision, 2017:614-629.

[7] VAN DE VEN G M, TOLIAS A S. Generative replay with feedback connections as a general strategy for continual learning[J]. arXiv preprint arXiv:1809.10635, 2018.

[8] CHAUDHRY A, DOKANIA P K, AJANTHAN T, et al. Riemannian walk for incremental learning: Understanding forgetting and intransigence[J]. arXiv:1801.10112, 2018.

[9] CASTRO F M, MARINJIMENEZ M J, GUIL N, et al. End-to-end incremental learning[J]. european conference on computer vision, 2018:241-257.

[10] MU X, ZHU F, DU J, et al. Streaming classification with emerging new class by class matrix sketching.[C]//AAAI. S.l.: s.n., 2017: 2373-2379.

[11] SHIN H, LEE J K, KIM J, et al. Continual learning with deep generative replay [J]. arXiv preprint arXiv:1705.08690, 2017.

[12] GOODFELLOW I J, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets[C]//International Conference on Neural Information Processing Systems. S.l.: s.n., 2014.

[13] KINGMA D P, WELLING M. Auto-encoding variational bayes[J]. international conference on learning representations, 2014.

[14] HINTON G E, VINYALS O, DEAN J. Distilling the knowledge in a neural network[J]. arXiv: Machine Learning, 2015.

[15] ZHANG J, ZHANG J, GHOSH S, et al. Class-incremental learning via deep model consolidation[J]. arXiv:1903.07864, 2019.

[16] TORREY L, SHAVLIK J. Transfer learning[M]//Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. S.l.: IGI global, 2010: 242-264.

[17] GEPPERTH A, HAMMER B. Incremental learning algorithms and applications [C]//European symposium on artificial neural networks (ESANN). S.l.: s.n., 2016.

[18] O'SHEA K, NASH R. An introduction to convolutional neural networks[J]. arXiv preprint arXiv:1511.08458, 2015.

[19] YANG H M, ZHANG X Y, YIN F, et al. Robust classification with convolutional prototype learning[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. S.l.: s.n., 2018: 3474-3482.

[20] WEN Y, ZHANG K, LI Z, et al. A discriminative feature learning approach for deep face recognition[C]//European conference on computer vision. S.l.: Springer, 2016: 499-515.

[21] ROSCH E. Prototype classification and logical classification: The two systems[J]. New trends in conceptual representation: Challenges to Piaget' s theory, 1983: 73-86.

[22] SCHEIRER W J, DE REZENDE ROCHA A, SAPKOTA A, et al. Toward open set recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 35(7):1757-1772.

[23] CHAUDHRY A, DOKANIA P K, AJANTHAN T, et al. Riemannian walk for incremental learning: Understanding forgetting and intransigence[C]//Proceedings of the European Conference on Computer Vision (ECCV). S.l.: s.n., 2018: 532-547.

[24] JUNG H, JU J, JUNG M, et al. Less-forgetting learning in deep neural networks [J]. arXiv preprint arXiv:1607.00122, 2016.

[25] DELANGE M, ALJUNDI R, MASANA M, et al. A continual learning survey: Defying forgetting in classification tasks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.

[26] LIU X, MASANA M, HERRANZ L, et al. Rotate your networks: Better weight consolidation and less catastrophic forgetting[C]//2018 24th International Conference on Pattern Recognition (ICPR). S.l.: IEEE, 2018: 2262-2268.

[27] ALJUNDI R, BABILONI F, ELHOSEINY M, et al. Memory aware synapses: Learning what (not) to forget[C]//Proceedings of the European Conference on Computer Vision (ECCV). S.l.: s.n., 2018: 139-154.

[28] HOU S, PAN X, LOY C C, et al. Learning a unified classifier incrementally via rebalancing[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. S.l.: s.n., 2019: 831-839.

[29] WU Y, CHEN Y, WANG L, et al. Large scale incremental learning[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. S.l.: s.n., 2019: 374-382.

[30] RANNEN A, ALJUNDI R, BLASCHKO M B, et al. Encoder based lifelong learning[C]//Proceedings of the IEEE International Conference on Computer Vision. S.l.: s.n., 2017: 1320-1328.

[31] KEMKER R, KANAN C. Fearnet: Brain-inspired model for incremental learning [J]. arXiv preprint arXiv:1711.10563, 2017.

[32] WELLING M. Herding dynamical weights to learn[C]//Proceedings of the 26th Annual International Conference on Machine Learning. S.l.: s.n., 2009: 1121-1128.

[33] BELOUADAH E, POPESCU A. Il2m: Class incremental learning with dual memory[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. S.l.: s.n., 2019: 583-592.

[34] NAIR V, HINTON G E. Rectified linear units improve restricted boltzmann machines[C]//Icml. S.l.: s.n., 2010.

[35] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. S.l.: s.n., 2016: 770-778.

[36] KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[37] CHRABASZCZ P, LOSHCHILOV I, HUTTER F. A downsampled variant of imagenet as an alternative to the cifar datasets[J]. arXiv preprint arXiv:1707.08819, 2017.

[38] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection [C]//Proceedings of the IEEE international conference on computer vision. S.l.: s.n., 2017: 2980-2988.

[39] NAGRANI A, CHUNG J S, ZISSERMAN A. Voxceleb: a large-scale speaker identification dataset[J]. arXiv preprint arXiv:1706.08612, 2017.

[40] WAIBEL A, HANAZAWA T, HINTON G, et al. Phoneme recognition using time-delay neural networks[J]. IEEE transactions on acoustics, speech, and signal processing, 1989, 37(3):328-339.

[41] LIN M, CHEN Q, YAN S. Network in network[J]. arXiv preprint arXiv:1312.4400, 2013.

[42] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[43] DENG J, DONG W, SOCHER R, et al. Imagenet: A large-scale hierarchical image database[C]//2009 IEEE conference on computer vision and pattern recognition. S.l.: Ieee, 2009: 248-255.

[44] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. S.l.: s.n., 2012: 1097-1105.

[45] GOODFELLOW I J, MIRZA M, XIAO D, et al. An empirical investigation of catastrophic forgetting in gradient-based neural networks[J]. arXiv preprint arXiv:1312.6211, 2013.

[46] ZHU X J. Semi-supervised learning literature survey[J]. 2005.

[47] ZHU X, GOLDBERG A B. Introduction to semi-supervised learning[J]. Synthesis lectures on artificial intelligence and machine learning, 2009, 3(1):1-130.

[48] LONGADGE R, DONGRE S. Class imbalance problem in data mining review [J]. arXiv preprint arXiv:1305.1707, 2013.

[49] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[50] MENSINK T, VERBEEK J, PERRONNIN F, et al. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost[C]//European Conference on Computer Vision. S.l.: Springer, 2012: 488-501.

[51] KIM Y, RUSH A M. Sequence-level knowledge distillation[J]. arXiv preprint arXiv:1606.07947, 2016.

[52] MASANA M, TUYTELAARS T, VAN DE WEIJER J. Ternary feature masks: continual learning without any forgetting[J]. arXiv preprint arXiv:2001.08714, 2020.

[53] MALLYA A, DAVIS D, LAZEBNIK S. Piggyback: Adapting a single network to multiple tasks by learning to mask weights[C]//Proceedings of the European Conference on Computer Vision (ECCV). S.l.: s.n., 2018: 67-82.

[54] MALLYA A, LAZEBNIK S. Packnet: Adding multiple tasks to a single network by iterative pruning[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. S.l.: s.n., 2018: 7765-7773.

[55] SERRA J, SURIS D, MIRON M, et al. Overcoming catastrophic forgetting with hard attention to the task[C]//International Conference on Machine Learning. S.l.: PMLR, 2018: 4548-4557.

[56] ROSENFELD A, TSOTSOS J K. Incremental learning through deep adaptation [J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 42(3): 651-663.

[57] FERNANDO C, BANARSE D, BLUNDELL C, et al. Pathnet: Evolution channels gradient descent in super neural networks[J]. arXiv preprint arXiv:1701.08734, 2017.

[58] ALJUNDI R, CHAKRAVARTY P, TUYTELAARS T. Expert gate: Lifelong learning with a network of experts[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. S.l.: s.n., 2017: 3366-3375.

[59] RUSU A A, RABINOWITZ N C, DESJARDINS G, et al. Progressive neural networks[J]. arXiv preprint arXiv:1606.04671, 2016.

[60] SCHWARZ J, CZARNECKI W, LUKETINA J, et al. Progress & compress: A scalable framework for continual learning[C]//International Conference on Machine Learning. S.l.: PMLR, 2018: 4528-4537.

[61] XU J, ZHU Z. Reinforced continual learning[J]. arXiv preprint arXiv:1805.12369, 2018.

[62] LI X, ZHOU Y, WU T, et al. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting[C]//International Conference on Machine Learning. S.l.: PMLR, 2019: 3925-3934.

[63] PARISI G I, LOMONACO V. Online continual learning on sequences[M]//Recent Trends in Learning From Data. S.l.: Springer, 2020: 197-221.

[64] LOPEZ-PAZ D, RANZATO M. Gradient episodic memory for continual learning [J]. arXiv preprint arXiv:1706.08840, 2017.

[65] CHAUDHRY A, RANZATO M, ROHRBACH M, et al. Efficient lifelong learning with a-gem[J]. arXiv preprint arXiv:1812.00420, 2018.

简历与科研成果

基本信息

毛乐坤，男，汉族，1996 年 4 月出生，陕西省安康市人。

教育背景

2018 年 9 月 — 2021 年 6 月 南京大学计算机科学与技术系 硕士

2014 年 9 月 — 2018 年 6 月 西安电子科技大学物理与光电工程学院 本科

攻读硕士学位期间完成的学术成果

1. Zhang Xu, Yao Yang, Xu Baile, Mao Lekun,Shen Furao,Zhao Jian,Lin Qingwei, “Label mapping neural networks with response consolidation for class incremental learning”[J]. arXiv preprint arXiv:1905.07835, 2019.

攻读硕士学位期间完成的专利

1. 申富饶, 毛乐坤, 徐百乐. “一种基于半监督学习的增量式图片分类方法” (2020113965751)。
2. 葛轶洲, 徐百乐, 毛乐坤, 张旭, 韩峰, 周青, 赵健, 申富饶. “一种基于 prototype 的增量式信息分类方法” (2020105395807)。

致　　谢

时光荏苒，岁月如梭，三年的研究生生涯就要结束了，心中充满了喜悦，亦充满了感激。这三年，我在科研上慢慢培养了独立思考，发现问题，解决问题的能力，对我个人而言更重要的是，我重新认识自己，思考生活。

在这里，我要感谢申富饶教授。在我遇到迷茫的时候，耐心开导我，在科研方面，给予了我长期的指导，以及充分选择的自由；感谢吴楠副教授，给予我帮助和求学的机会；感谢徐百乐师兄，在科研方面经常与我探讨问题，锻炼了我的科研能力；感谢实验室里面的同学，和大家一起快乐生活，快乐学习；感谢一些我在南京遇到的朋友，遇到你们我是幸运的。

另外，我要感谢我的爸爸妈妈、杨红阿姨、我的妹妹、我自己。感谢爸爸妈妈辛勤的付出为我提供了求学的机会；感谢杨红阿姨亦师亦友，她的无私让我感受到了最淳朴的善良，也让我明白生活是美好的，爱自己，才能更好地爱他人；感谢我的妹妹让我再次体会到了童年的天真烂漫，给我的生活增添了许多欢乐与微笑；感谢我的努力与进取，终于成为了更好的自己。

学位论文出版授权书

本人完全同意《中国优秀博硕士学位论文全文数据库出版章程》（以下简称“章程”），愿意将本人的学位论文提交“中国学术期刊（光盘版）电子杂志社”在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版，并同意编入《中国知识资源总库》，在《中国博硕士学位论文评价数据库》中使用和在互联网上传播，同意按“章程”规定享受相关权益。

作者签名：_____

_____年_____月_____日

论文题名	基于原型的类增量学习算法研究				
研究生学号	MF1833046	所在院系	计算机科学与技术系	学位年度	2021
论文级别	☐ 硕士　☑ 硕士专业学位 ☐ 博士　☐ 博士专业学位　（请在方框内画勾）				
作者 Email	lekmao@smail.nju.edu.cn				
导师姓名	申富饶教授，吴楠副教授				

论文涉密情况：

☐ 不保密

☑ 保密，保密期：_____年_____月_____日　至　_____年_____月_____日

注：请将该授权书填写后装订在学位论文最后一页（南大封面）。