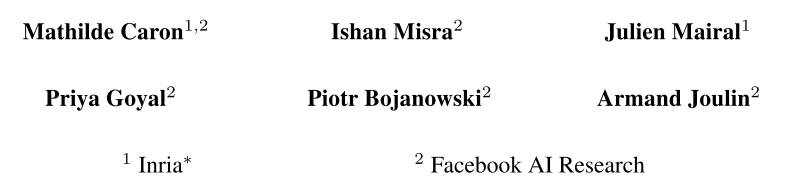
Unsupervised Learning of Visual Features by Contrasting Cluster Assignments



1. 摘要

无监督的图像表征已经大大缩小了与有监督的预训练的差距，特别是最近对比学习方法的成就。这些对比性方法通常是在作，并依赖于大量明确的成对特征比较，这在计算上具有挑战性。在本文中，我们提出了一种在线算法，SwAV，它利用了对比性方法的优势，而不需要计算成对的比较。具体来说，我们的方法对数据进行聚类，同时对同一图像的不同增强（或 "视图"）进行聚类分配，而不是像对比性学习那样直接比较特征。简单地说，我们使用了一种 "交换"预测机制，即我们从另一个视图的表述中预测一个视图的代码。我们的方法可以用大批量和小批量进行训练，可以扩展到无限量的数据。与以前的对比性方法相比，我们的方法更具有内存效率，因为它不需要一个大的内存库或特殊的动量网络。此外，我们还提出了一种新的数据增强策略-- multi-crop，它使用不同分辨率的视图组合来代替两个全分辨率的视图，而不增加内存或计算要求。我们用ResNet-50在ImageNet上取得了75.3%的最高准确率，并在所有考虑的迁移任务上超过了监督预训练，从而验证了我们的发现。

1. 问题重述

无监督的视觉表征学习，或称自我监督学习，旨在不使用人工注释而获得特征，并且正在迅速缩小与计算机视觉中有监督预训练的性能差距[10, 24, 44]。许多最近的最先进的方法建立在实例判别任务的基础上，该任务将数据集的每张图像（或 "实例"）及其转换视为一个单独的类别[16]。这项任务产生的表征能够区分不同的图像，同时实现对图像变换的一些不变性。最近使用实例判别的自我监督方法依赖于两个元素的组合。(i) 对比性损失[23]和(ii) 一组图像转换。对比性损失通过直接比较图像特征消除了实例类别的概念，而图像变换则定义了特征中编码的不变性。这两个因素对所产生的网络的质量至关重要[10, 44]，我们的工作对目标函数和变换都进行了改进。

对比性损失明确地对图像表征进行比较，以推开来自不同图像的表征，同时将来自同一图像的变换或视图的表征拉到一起。由于在一个大的数据集上计算所有的成对比较是不实际的，大多数的实现是通过在训练期间减少对随机图像子集的比较数量来近似损失[10, 24, 58]（NCE series）。另一种近似损失的方法是近似任务，即放松实例判别问题。例如，基于聚类的方法对具有相似特征的图像组而不是单个图像进行判别[7]。聚类的目标是可行的，但它不能很好地随数据集扩展，因为它需要通过整个数据集来形成图像“codes”（即聚类分配），这些代码在训练期间被用作目标。在这项工作中，我们使用了不同的范式，建议在线计算代码，同时强迫从同一图像的视图中获得的代码之间的一致性。（一致性在各个概念在对比学习中经常被提及。我们虽然不知道每个图像提取的特征是什么样子，但是我们至少知道同一张图像做different transformations后的图像之间，特征应该是相近的。）比较聚类分配允许对比不同的图像视图，而不依赖明确的成对特征比较。具体来说，我们提出了一个简单的 "交换 "预测问题，即我们从另一个视图的表述中预测一个视图的代码。我们通过同一图像的多个视图之间的交换分配（SwAV）来学习特征。特征和代码都是在线学习的，使我们的方法可以扩展到潜在的无限量的数据。此外，SwAV适用于小批量和大批量，不需要一个大的内存库[58]或动量编码器[24]。

除了我们基于在线聚类的方法，我们还提出了对图像转换的改进。尽管有证据表明在训练过程中比较更多的视图可以改善所产生的模型，但大多数对比方法都是对每幅图像进行一对变换的比较[44]。在这项工作中，我们提出了使用较小尺寸的图像来增加视图的数量，同时不增加训练期间的内存或计算要求的多重裁剪。我们还观察到，将场景的小部分映射到更多的全局视图上，可以显著提高性能。直接使用缩小的图像会在特征中引入偏见[53]，这可以通过使用不同尺寸的混合图像来避免。我们的策略很简单，但很有效，可以应用于许多自监督方法，并在性能上有一致的提高。

我们通过在几个标准的自监督基准上评估我们的方法来验证我们的贡献。特别是在ImageNet线性评估协议上，我们用标准的ResNet-50达到了75.3%的最高准确率，用更广泛的模型达到了78.5%。我们还表明，我们的多作物策略是通用的，并且提高了不同的自监督方法的性能，即SimCLR[10]、DeepCluster[7]和SeLa[2]，在ImageNet上的最高准确率为2%到4%。总的来说，我们做出了以下贡献。

- 我们提出了一种可扩展的在线聚类损失，在ImageNet上的性能提高了+2%，并且在大批量和小批量的情况下都能工作，不需要大型内存库或动量编码器。

- 我们引入了multi-crop策略，在没有计算或内存开销的情况下增加图像的视图数量。我们观察到，在ImageNet上，用这种策略对几种自监督的方法进行了2%到4%的一致改进。

- 将这两个技术贡献结合到一个模型中，我们在ImageNet上用标准的ResNet将自我监督的性能提高了+4.2%，并在多个下游任务上优于监督的ImageNet预训练。这是第一个在不对特征进行微调的情况下做到这一点的方法，也就是说，只在冻结的特征之上使用线性分类器。

1. 相关工作

**实例和对比性学习**。实例级分类认为数据集中的每张图片都是它自己的类别[5, 16, 58]。Dosovitskiy等人[16]为每张图片明确分配了一个类别，并学习了一个线性分类器，其类别与数据集中的图片一样多。由于这种方法很快就变得难以解决，Wu等人[58]通过将分类器替换为存储先前计算的表征的存储库来缓解这一问题。他们依靠噪声对比估计（NCE）[22]来比较实例，这是对比学习的一种特殊形式[29, 47]。He等人[24]通过存储来自动量编码器的表征来改进对比性方法的训练，而不是训练后的网络。最近，Chen等人[10]表明，如果批次足够大，记忆库可以完全被同一批次的元素所取代。与此不同的是，我们通过将图像特征映射到一组可训练的原型向量来避免比较每一对图像。

**深度表征学习的聚类**。我们的工作也与基于聚类的方法有关[2, 4, 7, 8, 19, 30, 59, 62, 63, 68]。Caron等人[7]表明，k-means分配可以作为伪标签来学习视觉表征。这种方法可以扩展到大型的未收集的数据集，并可用于监督网络的预训练[8]。然而，他们的表述没有原则性，最近，Asano等人[2]展示了如何将伪标签分配问题作为最优传输问题的一个实例。我们考虑了一个类似的表述，将表征映射到原型向量，但与[2]不同的是，我们保留了由Sinkhorn-Knopp算法[13]产生的软分配，而不是将其近似为一个硬分配。此外，与Caron等人[7，8]和Asano等人[2]不同的是，我们获得了在线赋值，这使得我们的方法可以优雅地扩展到任何数据集大小。

**Handcrafted pretext tasks**。许多自我监督的方法操纵输入数据，以借口任务的形式提取监督信号[1, 14, 31, 34, 36, 42, 45, 48, 49, 55, 56, 66]。我们请读者参考Jing等人[32]对这些文献的详尽评论。特别值得注意的是，Misra和vander Maaten[44]提出将拼图任务[46]编码为对比学习的不变因素。拼图瓦片是不重叠的裁剪，分辨率小，只覆盖整个图像区域的一部分（20%∼）。相比之下，我们的多作物策略包括简单地对具有两种不同尺寸的多个随机作物进行采样：一个标准尺寸和一个较小尺寸。

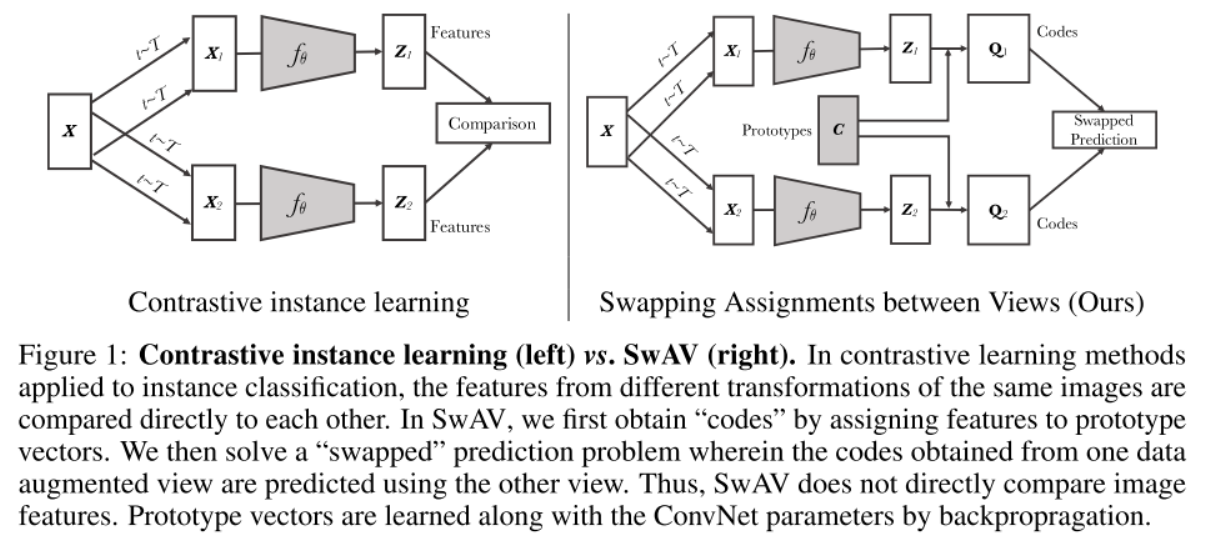
1. 详细阐述

我们的目标是以在线方式学习视觉特征，而不需要监督。为此，我们提出了一种基于聚类的在线自监督方法。典型的基于聚类的方法[2, 7]是离线的，因为它们在聚类分配步骤和训练步骤之间交替进行，前者是对整个数据集的图像特征进行聚类，后者是对不同图像视图的聚类分配即 "代码 "进行预测。不幸的是，这些方法不适合在线学习，因为它们需要多次通过数据集来计算聚类所需的图像特征。在这一节中，我们描述了一种替代方法，即在同一图像的不同增量的代码之间强制执行一致性。这个解决方案受到了对比性实例学习[58]的启发，因为我们不把代码作为目标，而只是在同一图像的不同视图之间强制执行一致性映射。我们的方法可以被解释为通过比较它们的聚类分配而不是它们的特征来对比多个图像视图的方法。

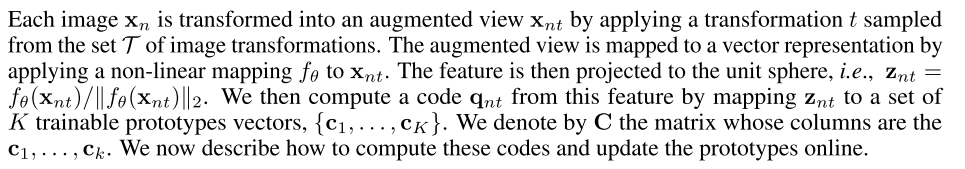
更确切地说，我们从图像的一个增强版本中计算出一个代码，并从同一图像的其他增强版本中预测这个代码。给定来自同一图像的两个不同增强版本的两个图像特征zt和zs，我们通过将这些特征与一组K原型{c1, ... , cK}相匹配来计算其代码qt和qs。, cK}。然后，我们设置一个 "交换 "预测问题，**损失函数**如下。



其中，函数l(z, q)衡量特征z和代码q之间的拟合度，详见下文。直观地说，我们的方法是用中间代码qt和qs来比较特征zt和zs。如果这两个特征捕捉了相同的信息，那么应该可以从另一个特征中预测出代码。类似的比较出现在直接比较特征的对比学习中[58]。在图1中，我们说明了对比性学习和我们的方法之间的关系。

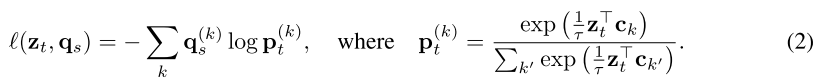


* 1. Online clustering

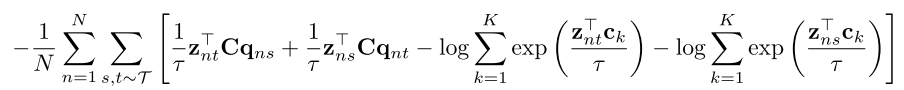


* + 1. Swapped prediction problem

公式（1）中的损失函数有两个项，设置了从特征zs预测代码qt和从zt预测qs的 "互换 "预测问题。每个项都代表了代码和通过对Zi和C中所有原型的点积取softmax的概率之间的交叉熵损失，即：



其中τ是一个温度参数[58]。在所有的图像和数据对上取这个损失扩增导致交换预测问题的损失函数如下：



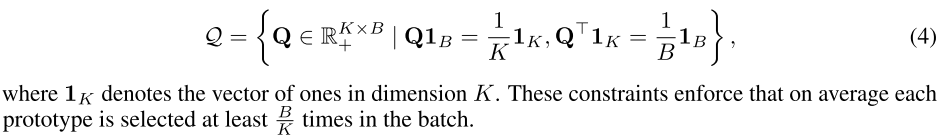
这个损失函数联合最小化：1）原型C和2）用于产生特征（z*nt*）*n,t*的图像编码器fθ的参数。

* + 1. Computing codes online

为了使我们的方法可以在线训练，我们只使用一个批次中的图像特征来计算codes。直观地说，由于prototype C是跨越不同批次使用的，SwAV将多个实例聚集到原型上。我们使用原型C来计算代码，使一个批次中的所有实例都被原型平均分割开来。这种等分约束确保了一批中不同图像的代码是不同的，从而避免了每个图像都有相同代码的琐碎解决方案。给定B个特征向量Z = [z1, ... , zB]，我们感兴趣的是将它们映射到原型C = [c1, ... , cK]。我们用Q = [q1, ... , qB]表示这种映射或代码，并优化Q，使特征和原型之间的相似度最大化，即：



其中H是熵值函数，，ε是一个用于控制映射平滑度的参数。我们观察到，强熵正则化（即使用高的ε）通常会导致一个简单的解决方案，即所有的样本都崩溃成一个独特的表示，并且是 所有的样本都被统一分配给所有的原型。因此，在实践中，我们保持较低的ε。Asano等人[2]通过限制矩阵Q属于[运输多面体](https://baike.baidu.com/item/%E8%BF%90%E8%BE%93%E5%A4%9A%E9%9D%A2%E4%BD%93/22829215)来强制执行平等的分区。他们在完整的数据集上工作，我们建议将他们的解决方案调整为在迷你批上工作，将运输多面体限制在迷你批上



一旦找到Prob.(3)的连续解Q∗，就可以通过使用舍入程序[2]得到一个离散的代码。根据经验，我们发现，当以离线方式在完整的数据集上计算代码时，离散代码的效果很好，就像Asano等人[2]那样。然而，在我们只使用迷你批的在线设置中，使用离散码的表现比使用连续码的表现差。一个解释是，获得离散代码所需的舍入是一个比梯度更新更积极的优化步骤。虽然它使模型迅速收敛，但却导致了更差的解决方案。因此，我们保留了软编码Q∗，而不是舍入。这些软码Q∗是概率(3)在集合Q上的解，采取归一化指数矩阵的形式[13]。

1. 实验
2. 评析