Improved Baselines with Momentum Contrastive Learning

Xinlei Chen Haoqi Fan Ross Girshick Kaiming He

Facebook AI Research (FAIR)

Abstract

Contrastive unsupervised learning has recently shown encouraging progress, e.g., in Momentum Contrast (MoCo) and SimCLR. In this note, we verify the effectiveness of two of SimCLR's design improvements by implementing them in the MoCo framework. With simple modifications to MoCo—namely, using an MLP projection head and more data augmentation—we establish stronger baselines that outperform SimCLR and do not require large training batches. We hope this will make state-of-the-art unsupervised learning research more accessible. Code will be made public.

【摘要】

对比无监督学习最近取得了令人鼓舞的进展，例如，在动量对比（MoCo）和 SimCLR 中。在本文中，我们通过在 MoCo 框架中实现 SimCLR 的两个设计改进来验证它们的有效性。通过对 MoCo 的简单修改，即使用 MLP 投影头和更多的数据扩充（增强），我们建立了性能优于 SimCLR 的更强大的基线，并且不需要更大的训练 batchsize。我们希望这将使最先进的无监督学习研究更容易获得。代码将被公开。

1. Introduction

Recent studies on unsupervised representation learning from images [16, 13, 8, 17, 1, 9, 15, 6, 12, 2] are converging on a central concept known as contrastive learning [5]. The results are promising: e.g., Momentum Contrast (MoCo)[6] shows that unsupervised pre-training can surpass its ImageNet-supervised counterpart in multiple detection and segmentation tasks, and SimCLR [2] further reduces the gap in linear classifier performance between unsupervised andsupervised pre-training representations.

This note establishes stronger and more feasible baselines built in the MoCo framework. We report that two design improvements used in SimCLR, namely, an MLP projection head and stronger data augmentation, are orthogonal to the frameworks of MoCo and SimCLR, and when used with MoCo they lead to better image classification and object detection transfer learning results. Moreover, the MoCo framework can process a large set of negative samples without requiring large training batches (Fig. 1). In contrast to SimCLR's large 4k~ 8k batches, which require TPU support, our "MoCo v2" baselines can run on a typical 8-GPU machine and achieve better results than SimCLR. We hope these improved baselines will provide a reference for future research in unsupervised learning.

【1.介绍】

最近关于从图像中进行无监督表征学习的研究[16,13,8,17,1,9,15,6,12,2]正在集中于一个被称为对比学习的中心概念[5]。结果是有希望的：例如，动量对比（MoCo）[6]表明，无监督预训练可以在多个检测和分割任务中优于其ImageNet监督的同类，SimCLR[2]进一步缩小了无监督和有监督预训练表示之间在线性分类器性能上的差距。

本说明建立了在MoCo框架上的更强大和更可行的基线。我们报告了在SimCLR中使用的两个设计改进，即MLP投影头和更强的数据增强，它们与MoCo和SimCLR框架是正交的，与MoCo一起使用可以获得更好的图像分类和目标检测迁移学习结果。此外，MoCo框架可以处理大量的负样本，而不需要大量的训练批大小（图1）。与SimCLR需要TPU支持的4k~ 8k大型批处理不同，我们的“MoCo v2”基线可以在典型的8-GPU机器上运行，并获得比SimCLR更好的结果。希望这些改进后的基线能为今后无监督学习的研究提供参考。

2. Background

Contrastive learning. Contrastive learning [5] is a framework that learns similar/dissimilar representations from data that are organized into similar/dissimilar pairs. This can be formulated as a dictionary look-up problem. An effective contrastive loss function, called InfoNCE [13], is:

$$\mathcal{L}_{q,k^+,\{k^-\}} = -\log \frac{\exp(q\cdot k^+/\tau)}{\exp(q\cdot k^+/\tau) + \sum_{k^-} \exp(q\cdot k^-/\tau)}. \quad (1)$$

Here q is a query representation, k+ is a representation of the positive (similar) key sample, and {k-} are representations of the negative (dissimilar) key samples. τ is a temperature hyper-parameter. In the instance discrimination pretext task [16] (used by MoCo and SimCLR), a query and a key form a positive pair if they are data-augmented versions of the same image, and otherwise form a negative pair.

The contrastive loss (1) can be minimized by various mechanisms that differ in how the keys are maintained [6]. In an end-to-end mechanism (Fig. 1a) [13, 8, 17, 1, 9, 2], the negative keys are from the same batch and updated end-to-end by back-propagation. SimCLR [2] is based on this mechanism and requires a large batch to provide a large set of negatives. In the MoCo mechanism (Fig. 1b) [6], the negative keys are maintained in a queue, and only the queries and positive keys are encoded in each training batch. A momentum encoder is adopted to improve the representation consistency between the current and earlier keys. MoCo decouples the batch size from the number of negatives.

【2.背景】

对比学习。对比学习[5]是一种从组织成相似/不相似对的数据中学习相似/不相似表示的框架。这可以表述为字典查找问题。

一个有效对比损失函数，称为InfoNCE[13]，为：

$$\mathcal{L}_{q,k^+,\{k^-\}} = -\log \frac{\exp(q\cdot k^+/\tau)}{\exp(q\cdot k^+/\tau) + \sum_{k^-} \exp(q\cdot k^-/\tau)}. \quad (1)$$

这里 q 是一个查询表示，k+是正（相似）键样本的表示，而{k-}是负（不相似）键样本的表示。τ 是温度超参数。在实例识别代理任务[16]（由 MoCo 和 SimCLR 使用）中，如果查询和密钥是同一图像的数据增强版本，则它们形成一个正对，否则形成负对。

对比损失（1）可以通过不同的键维护机制来最小化[6]。在端到端机制（图 1a）[13、8、17、1、9、2]中，负键来自同一批并通过反向传播进行端到端更新。SimCLR[2]基于这种机制，需要更大的批处理来提供大量的负样本集。在 MoCo 机制（图 1b）[6]中，负键保持在队列中，并且在每个训练批中仅对查询和正键进行编码。采用动量编码器来提高当前键与先前键的表示一致性。MoCo 将批大小与负样本数量解耦。

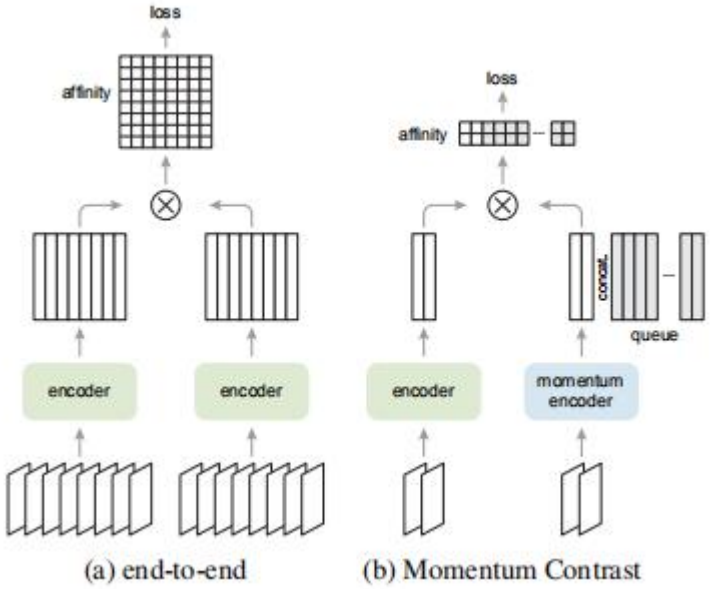

Figure 1. A batching perspective of two optimization mechanisms for contrastive learning. Images are encoded into a representation space, in which pairwise affifinities are computed.

图 1。两种对比学习优化机制的批处理视角。图像被编码到一个表示空间中，在这个空间中计算成对的仿射。

Improved designs. SimCLR [2] improves the end-to-end variant of instance discrimination in three aspects: (i) a substantially larger batch (4k or 8k) that can provide more negative samples; (ii) replacing the output fc projection head[16] with an MLP head; (iii) stronger data augmentation.

In the MoCo framework, a large number of negative samples are readily available; the MLP head and data augmentation are orthogonal to how contrastive learning is instantiated. Next we study these improvements in MoCo.

改进设计。 SimCLR[2]从三个方面改进了实例识别的端到端变体：（i）大批量（4k 或 8k），可以提供更多的负样本；（ii）用 MLP 头替换输出 fc 映射头[16]；（iii）更强的数据扩充（增强）。

在 MoCo 框架中，大量的负样本是现成的；MLP 头和数据扩充（增强）与对比学习的实例化方式是正交的。接下来我们研究 MoCo 中的这些改进。

3. Experiments

Settings. Unsupervised learning is conducted on the 1.28M ImageNet [3] training set. We follow two common protocols for evaluation. (i) ImageNet linear classification: features are frozen and a supervised linear classifier is trained; we report 1-crop (224×224), top-1 validation accuracy. (ii) Transferring to VOC object detection [4]: a Faster R-CNN detector [14] (C4-backbone) is fine-tuned end-to-end on the VOC 07+12 trainval set and evaluated on the VOC 07 test set using the COCO suite of metrics [10]. We use the same hyper-parameters (except when noted) and codebase as MoCo [6]. All results use a standard-size ResNet-50 [7].

3. 实验

设置 在 1.28M ImageNet[3]训练集上进行无监督学习。我们遵循两种常见的评估协议。（i） ImageNet 线性分类：特征被冻结并训练一个有监督的线性分类器；我们报告了 1-crop（224×224），top-1 验证精度。（ii）迁移到 VOC 目标检测[4]：Faster R-CNN 检测器[14]（C4 主干）在 VOC 07+12 训练集上进行端到端微调，并在 VOC 07 测试集上使用 COCO 度量套件进行评估[10]。我们使用与 MoCo 相同的超参数（除非另有说明）和代码库[6]。所有结果都使用标准尺寸的 ResNet-50[7]。

MLP head. Following [2], we replace the fc head in MoCo with a 2-layer MLP head (hidden layer 2048-d, with ReLU). Note this only influences the unsupervised training stage; the linear classification or transferring stage does not use this MLP head. Also, following [2], we search for an optimal τ w.r.t. ImageNet linear classification accuracy:

τ	0.07	0.1	0.2	0.3	0.4	0.5
w/o MLP	60.6	**60.7**	59.0	58.2	57.2	56.4
w/ MLP	62.9	64.9	**66.2**	65.7	65.0	64.3

Using the default τ = 0.07 [16, 6], pre-training with the MLP head improves from 60.6% to 62.9%; switching to the optimal value for MLP (0.2), the accuracy increases to 66.2%. Table 1(a) shows its detection results: in contrast to the big leap on ImageNet, the detection gains are smaller.

MLP 头 在[2]之后，我们将 MoCo 中的 fc 头替换为 2 层 MLP 头（隐藏层 2048-d，使用 ReLU）。注：这只影响无监督训练阶段；线性分类或迁移阶段不使用此 MLP 头。此外，在[2]中，我们寻找最佳的 τ w.r.t. ImageNet 线性分类精度：

τ	0.07	0.1	0.2	0.3	0.4	0.5
w/o MLP	60.6	**60.7**	59.0	58.2	57.2	56.4
w/ MLP	62.9	64.9	**66.2**	65.7	65.0	64.3

使用默认值 τ = 0.07[16，6]，使用 MLP 头部进行预训练将从 60.6%提高到 62.9%；切换到 MLP 的最佳值（0.2），准确率提高到 66.2%。表 1（a）显示了它的检测结果：与 ImageNet 上的巨大飞跃相比，检测增益更小。

Augmentation. We extend the original augmentation in [6] by including the blur augmentation in [2] (we find the stronger color distortion in [2] has diminishing gains in our higher baselines). The extra augmentation alone (i.e., no MLP) improves the MoCo baseline on ImageNet by 2.8% to 63.4%, Table 1(b). Interestingly, its detection accuracy is higher than that of using the MLP alone, Table 1(b) vs. (a), despite much lower linear classification accuracy (63.4% vs. 66.2%). This indicates that linear classification accuracy is not monotonically related to transfer performance in detection. With the MLP, the extra augmentation boosts ImageNet accuracy to 67.3%, Table 1(c).

增强 我们]拓展了原来的[6]数据增强，如[2]中加入的模糊增强（我们发现在[2]中，更强的颜色失真会降低高基线的增益）。单独的额外增强（即没有 MLP）使 ImageNet 上的 MoCo 基线提高了 2.8%到 63.4%，如表 1（b）所示。有趣的是，只使用数据增强的检测精度比只使用 MLP 层的检测精度高。而 63.4% 与 66.2%相比，线性分类准确率要低得多（也就是数据增强的线性分类比 MLP 的低，而检测精度比 MLP 高）。这表明线性分类精度与检测中的迁移性能不是单调相关的。使用 MLP，额外的增强将 ImageNet 的精确度提高到 67.3%，表 1（c）。

	unsup. pre-train				ImageNet	VOC detection		
case	MLP	aug+	cos	epochs	acc.	AP_{50}	AP	AP_{75}
supervised					76.5	81.3	53.5	58.8
MoCo v1				200	60.6	81.5	55.9	62.6
(a)	✓			200	66.2	82.0	56.4	62.6
(b)		✓		200	63.4	82.2	56.8	63.2
(c)	✓	✓		200	67.3	**82.5**	57.2	63.9
(d)	✓	✓	✓	200	67.5	82.4	57.0	63.6
(e)	✓	✓	✓	**800**	**71.1**	**82.5**	**57.4**	**64.0**

Table 1. **Ablation of MoCo baselines,** evaluated by ResNet-50 for (i) ImageNet linear classification, and (ii) fine-tuning VOC object detection (mean of 5 trials). "MLP": with an MLP head; "aug+": with extra blur augmentation; "cos": cosine learning rate schedule.

表 1 **MoCo 基线的消融实验，**用 ResNet-50 评估（i）ImageNet 线性分类，（ii）微调 VOC 目标检测（5 次试验的平均值）。"MLP"：带 MLP 头；"aug+"：带额外模糊增强；"cos"：余弦学习速率计划。

Comparison with SimCLR. Table 2 compares SimCLR[2] with our results, referred to as MoCo v2. For fair comparisons, we also study a cosine (half-period) learning rate schedule [11] which SimCLR adopts. See Table 1(d, e). Using pre-training with 200 epochs and a batch size of 256, MoCo v2 achieves 67.5% accuracy on ImageNet: this is 5.6% higher than SimCLR under the same epochs and batch size, and better than SimCLR's large-batch result 66.6%. With 800-epoch pre-training, MoCo v2 achieves 71.1%, outperforming SimCLR's 69.3% with 1000 epochs.

与 SimCLR 进行比较 表 2 将 SimCLR[2]与我们的结果（称为 MoCo v2）进行了比较。为了公平比较，我们还研究了 SimCLR 采用的余弦（半周期）学习速率调度[11]，见表 1（d、e）。使用 200 个时期（epoch）和 256 个批次的预训练，MoCo v2 在 ImageNet 上达到 67.5%的准确率：在相同的时间段和批次大小下，这比 SimCLR 高 5.6%，优于 SimCLR 的大批量结果 66.6%。经过 800 个时期(epoch)的预训练，MoCo v2 达到了 71.1%，超过了 SimCLR 在 1000 个 epoch 的 69.3%。

case	unsup. pre-train MLP	aug+	cos	epochs	batch	ImageNet acc.
MoCo v1 [6]				200	256	60.6
SimCLR [2]	✓	✓	✓	200	256	61.9
SimCLR [2]	✓	✓	✓	200	8192	66.6
MoCo v2	✓	✓	✓	200	256	**67.5**
*results of **longer** unsupervised training follow:*						
SimCLR [2]	✓	✓	✓	1000	4096	69.3
MoCo v2	✓	✓	✓	800	256	**71.1**

Table 2. **MoCo vs. SimCLR:** ImageNet linear classifier accuracy (ResNet-50, 1-crop 224×224), trained on features from unsupervised pre-training. "aug+" in SimCLR includes blur and stronger color distortion. SimCLR ablations are from Fig. 9 in [2] (we thank the authors for providing the numerical results).
表 2 **MoCo 与 SimCLR:**ImageNet 线性分类器精度（ResNet-50，1-crop 224×224），根据无监督预训练的特征进行训练。SimCLR 中的"aug+"包括模糊和更强的颜色失真。SimCLR 的消融实验来自[2]中的图 9（我们感谢作者提供数值结果）。

Computational cost. In Table 3 we report the memory and time cost of our implementation. The end-to-end case reflects the SimCLR cost in GPUs (instead of TPUs in [2]). The 4k batch size is intractable even in a high-end 8-GPU machine. Also, under the same batch size of 256, the end-to-end variant is still more costly in memory and time, because it back-propagates to both q and k encoders, while MoCo back-propagates to the q encoder only.

Table 2 and 3 suggest that large batches are not necessary for good accuracy, and state-of-the-art results can be made more accessible. The improvements we investigate require only a few lines of code changes to MoCo v1, and we will make the code public to facilitate future research.

mechanism	batch	memory / GPU	time / 200-ep.
MoCo	256	**5.0G**	**53 hrs**
end-to-end	256	7.4G	65 hrs
end-to-end	4096	93.0G†	n/a

Table 3. **Memory and time cos**t in 8 V100 16G GPUs, implemented in PyTorch. †: based on our estimation.
表 3 **内存和时间开销**在 8 v100 16g GPU 中，在 PyTorch 中实现。†：根据我们的估计。

计算成本 在表 3 中，我们报告了实验的内存和时间开销。端到端案例反映了 GPU 中的 SimCLR 成本（而不是[2]中的 TPU）。4k 的批处理大小即使在高端的 8-GPU 机器中也是难以处理的。同样，在 256 的批处理大小下，端到端变体在内存和时间上仍然更昂贵，因为它向后传播到 q 和 k 编码器，而 MoCo 仅仅只反向传播到 q 编码器。

表 2 和表 3 表明，为了获得良好的准确度，无需大的批量，并且可以使最新的结果更容易获得。我们研究的这些改进只需要对 MoCo v1 进行几行代码更改，我们将公开这些代码以方便今后的研究。