中国科学：信息科学

SCIENTIA SINICA Informationis

论文

PJ 实验报告

戴琪润，封启骏，王嗣超，王乐祺，吴欣怡

* 通信作者. E-mail:

1 基础修改

1.1 模型的选择

首先, 我们希望通过模型的更改以提高 performance. 鉴于本次任务训练数据极其有限, 且对模型参数也有比较严格的限制 (>10M 就施加惩罚; 且计算资源有限), 所以我们希望找到一种对图像数据的利用更加高效, 同时参数量又限制在较小范围内 (50M 及以内) 的视觉模型. 下面展示了我们实验模型的思路, 大致是: 纯卷积 -> 高效 ViT -> 卷积与 ViT 的混合模型。

1.1.1 EfficientNet：结构高效的纯卷积

对于纯卷积模型, 我们选择了 EfficientNet 这个比较强力的 baseline. EfficientNet 本身主要有两大特性: 一是通过 Neural Architecture Search 得到一个性能较好的 baseline 模型; 二是在这个 baseline 之上, 利用某种 uniform scaling 技术逐渐增大参数量, 并相应获得表示能力更强的模型。

1.1.2 DEIT：训练方式更高效的 ViT

我们主要从两个方向对 ViT 类模型做了尝试. 首先我们选择了以 DEIT 为代表的训练方式更加高效的 ViT. 它通过蒸馏技术提高 ViT 在中小型数据集上训练后的表示能力, 进而提高了 ViT 对数据的利用效率. 它的训练主要分为两步: 一是在 imagenet 上训练一个 teacher CNN, 二是在 imagenet 上训练一个 student ViT, 并要求 student ViT 的预测结果与 teacher CNN 的预测结果相匹配。

1.1.3 Swin-Transformer：模型架构更高效的 ViT

其次, 我们选择了以 Swin Transformer 为代表的模型架构更加高效的 ViT. Swin Transformer 通过 Shifted Windows 技术在 ViT 中实现了层次性的特征提取: 它既通过把 self-attention 限制在局部的 windows 之中以减少计算开销, 同时也保持了一定的 cross-window connection 以实现对全局特征的提取和关联. 这是它的高效性的根源。

1.1.4 Gc_Vit：卷积与 ViT 的混合模型

最后, 我们猜测: 或许把卷积本身在小样本上所具有的 inductive bias 优势, 和 ViT 所具有的长序建模优势结合起来, 得到的模型能够在本次任务上有更好的表现. 因此我们选择了 GC-ViT. 它既类似 swin transformer, 通过局部和全局的 self-attention module 来高效地捕捉短序和长序信息, 又额外研制了一种新的 downsampler, 从而在 ViT 中加入了 inductive bias 和 inter-channel dependencies.

1.2 数据增强

除了模型更改外, 我们认为小样本的任务上数据增强也是关键的. 鉴于 sampler 采样的数据集中, 同一张图片会重复出现三次, 所以我们对于每一张数据集中的图片, 都以 2/3 的概率做 RandAugment, 以 1/3 的概率返回原图, 以希望在三张图片平均有两张做了增强, 一张返回原图。由于单一类型的数据增强的过往实验结果一般不好, 所以我们这里就直接采用了相对成熟的 RandAugment, 并没有实验其他的增强方式。

1.3 实验细节与参数设置

1. 模型参数规模: 为更公平地比较不同类型模型在小数据集样本上的能力, 我们选择了参数规模大致相同的几种模型, 如下:
   - EfficientNet b5: 30M
   - Deit_small_distilled: 22M
   - Swin_tiny: 29M
   - GcVit_tiny: 28M

2. 学习率:
   - Cosine Scheduler(即默认设置)
   - 初始值 1e-4, 最终值 1e-5; backbone 的 lr 是线性分类头的 1/10(即默认设置)
3. weight_decay: 0
4. batch_size: 64
5. rand_augment: True (即采用 torchvision.transforms.RandAugment(num_ops: int = 2, magnitude: int = 8)
6. 其余参数均与提供的 baseline 相同. 可以看到, 上面除了模型的更改, 其余参数都是**严格控制变量**的.

1.4 实验结果与结论

1.4.1 Ablation1：数据增强

从表1可以看到, 对于其他参数完全一致的 swin transformer, 数据增强前后, 在 dataset 0,4 上 performance 有了相对明显的增长; 但是在 dataset 1,2,3 上基本没有提升. 这表明数据增强的效果可能与数据集本身的" 难易程度" 有关: dataset 1 是 class 数目最多, 分类最难的一个数据集; 而 dataset 3 是 class 数目最少, 分类最简单的一个数据集, 在这两个最难和最简单的任务上, augmentation 基本没有提升. 而在难度适中的 0,4 两个数据集上, augmentation 还是能够提高模型的学习能力的。

表 1 数据增强

	Dataset 0	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Sum of Top1 Accuracy
Swin Transformer（无数据增强）	57.100	35.498	50.099	87.027	44.388	274.1
Swin Transformer（有数据增强）	58.600	35.498	49.802	87.027	46.071	277.0

1.4.2 Ablation2：模型更改

从表2中可以看到, 纯卷积的 EfficientNet performance 比 ViT-based models 差了很多. 而在 ViT-based models 中, deit < swin < gcvit, 即随着 ViT 与卷积的 local connection / inductive bias 等特性结合得越发紧密, 模型在小样本数据集上的训练效果也不断变好。

表 2 模型更改

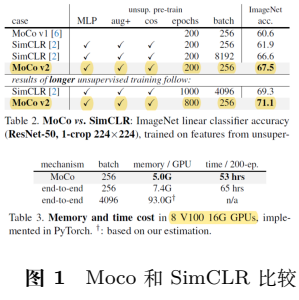
	Dataset 0	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Sum of Top1 Accuracy
EfficientNet	24.200	30.237	28.812	69.730	21.378	174.357
Data-Efficient-Vit	53.600	34.550	44.554	82.162	39.388	254.254
Swin-Transformer	58.600	35.498	49.802	87.027	46.071	276.998
Gc_Vit	61.900	35.308	49.307	87.838	46.786	281.139

1.4.3 小结

综上所述, 在 few-shot learning 这种小样本学习任务中, 在不引入蒸馏或其他复杂训练方式的情况下, 仅依靠模型自身的特性在极其有限的有标注数据上训练, 最终结果为: 纯卷积模型 < (传统)ViT 模型 < 卷积与 ViT 的混合模型. 这是符合目前对视觉领域模型性能的理解的: 卷积与 ViT 的混合模型既拥有了 ViT 的长序建模能力, 又获得了卷积的 inductive bias 优势. 这使得其既相比传统 ViT, 能够在少量数据上更高效地提取出特征, 同时又相比传统卷积, 提取出来的特征质量上限更高, 对于**长序和全局关联**建模得更好.

2 自监督学习

2.1 文献综述



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 unsuper-

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 cost** in 8 V100 16G GPUs, implemented in PyTorch. †: based on our estimation.

图 1 Moco 和 SimCLR 比较

• 鉴于提供了约 17 万张的 unlabeled data, 所以我们希望通过某种方式利用这些无标注数据. 我们主要关注了近两年比较热门的自监督学习 (self-supervised learning) 领域.

• 特别地, 考虑到小组成员的代码能力和时间限制, 我们没有选择如 CLIP 这类结合 language 与 image 的预训练方式, 而最终选择了相对传统的, 更容易实现的对比学习模型.

• 我们主要在 MoCo 和 SimCLR 两类模型中考虑: 由于计算资源的限制, 我们没有选择非常依赖大 batch-size 的 SimCLR, 而是选择了负样本量与 batch-size 解耦合的 MoCo 模型. 特别地, 出于训练稳定性的考虑, 以及训练时间有限的约束, 我们没有选择以 ViT 为 backbone 的 MoCo v3, 而是选择了仍以 resnet 为 backbone 的, 训练稳定性更好 (训练难度更低) 的 MoCo v2.

2.2 实现细节与参数设置

2.2.1 数据集

- 由于计算资源有限, MoCo backbone 的训练速度较慢, 我们就根据各数据集 classes 数目比例, sample 了 10000 + 21100 +10100 + 3310 + 6184 张图片, 共约 50000 张;
- 考虑到 10-shot 数据集不像 imagenet 各类分布均匀, 所以这样 sample, 或许能有缓解数据长尾分布的作用.

2.2.2 模型 backbone

对于模型 backbone 的实现, 我们主要参考了 moco 官方的 codebase 中 main_moco.py. 特别地, 对于 backbone 我们导入了 ResNet50 _Weights.IMAGENET1K_V1 这个预训练参数, 以提高模型初始能力, 希望能学到更好的 features.

2.2.3 模型 classifier

对于模型 classifier 的实现, 我们没有使用官方 codebase 的 main_lincls.py, 而是在本次任务提供的 codebase main.py 中简单改写. 特别地, 我们使用的是 torchvision.models 中的 resnet50 模型, 用其自身的线性分类层替代了原始的 CustomClassifier, 并在除了线性分类层以外的其它层中导入了预训练得到的 query encoder 的参数, 其文件名为 moco/checkpoint_0199.pth.tar(注意: 与 moco 官方库的实现一样, 我们没有用到 key encoder 的参数, 只使用了 query encoder 的).

2.2.4 具体超参数设置

1. backbone 训练

```
# ddp.sh
python main_moco.py \
-a resnet50 \
--lr 0.015 --epochs 200 \
--batch-size 128 --moco-k 4096 \
--dist-url 'tcp://localhost:10001' --multiprocessing-distributed --world-size 1 --rank 0 \
--resume '' \
--mlp --moco-t 0.2 --aug-plus --cos \
~
```

- Resnet Backbone 训练的主要超参数设置如上述脚本所示. 未注明的其余参数, 均与 main_moco.py 中的默认参数相同.
- 鉴于 MoCo v2 原论文的显存消耗为 8*5G, 对应 256 batch_size, lr=0.03, num_keys=65536; 故在我们有限的计算资源下, 取了 batch_size=128, num_keys=4096; lr=0.015, 该比例应当是合适的, 且也在 moco 官方库的训练说明中有所提及.

2. classifier 微调
   - 线性分类层的训练采用 end-to-end 方式 (没有 freeze backbone parameters), 超参数设置和第一部分完全相同. 具体可见 baseline.yaml 文件, 里面保存正是 moco classifier fine-tune 所使用的超参数.

2.2.5 实验结果与分析

(a) (b)

图 2 实验结果 Acc

表 3 Moco 实验结果

	Dataset 0	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Sum of Top1 Accuracy
Resnet50	18.8	30.521	14.059	66.757	8.673	138.81
EfficientNet(v1)-b5	24.200	30.237	28.812	69.730	21.378	174.357
Resnet50-moco pretrained	53.7	34.408	46.733	72.162	41.48	248.483
Data-Efficient ViT	53.600	34.550	44.554	82.162	39.388	254.254

• 首先, 使用 MoCo v2 在 unlabeled dataset 的一部分上训练 200 个 epochs 的过程中, 训练 loss 稳定减小, 训练 acc1 相对稳定地增加, 在 200 个 epochs 结束时已经达到 92% 左右. 这表明在上述参数配置下, MoCo v2 在本次任务的 unlabeled data 上确实是在有效训练的. 而且, 若是训练 epochs 数目更多, 训练 loss 和 acc1 还能更进一步提高.

• 其次, 对于 fine-tune 后做 evaluation 的结果, MoCo v2 预训练的 resnet50 比起未在 unlabeled data 上预训练、仅导入 imagenet-1k 参数的 resnet50, 在所有数据集上的表现都有显著提高. 除了最难分类的 dataset_1 和最容易分类的 dataset_3, 在剩余三个数据集上, performance 都有了至少三倍的提高, 使得最朴素的纯卷积模型 resnet50 也有和同参数 DEIT 等 ViT 模型接近的 performance. 这表明在无标注数据上的预训练确实使 backbone 模型学习到了对五个数据集建模都有效的 representation, 表明对比学习在本次任务上能够带来比较明显的提升.

• 比较遗憾的是, 本次训练的 MoCo v2 的 performance 未能超越未经预训练的同参数 ViT 类模型. 这大致有以下原因:

1. 所选取的 backbone 为最朴素的 resnet50, backbone 本身的表示能力上限较低; 而且第一部分的实验也表明, 在本次任务上, ViT 类模型比卷积类模型表现普遍要好很多.
2. 由于训练速度较慢、时间有限, 我们只选取了 5 万张左右的 unlabeled data, 没有把全部的 17 万图像都利用上. 数据量的减少也很可能损害了模型学习到的特征质量.
3. 观察训练时的 loss 和 acc1 曲线, 可以发现在 20-80 epochs 时, acc1 陷入了相对明显的停滞, 而在后续的 epochs 之中则重新稳定增长. 比较可能的原因是 lr_schedule 没有调整好, 导致在前半段训练过程中 lr 过大, 陷入局部最优点附近的震荡.

• 综上所述, 我们提出如下的改进策略:

1. 将 backbone 换成 ViT, 并相应地采用 MoCo v3 的训练策略;
2. 把所有 unlabeled data 都用来做预训练;
3. 调整更合适的 lr_schedule, 并且训练更多 epochs;

这样的改进策略应该能够进一步提升模型对于本次任务的特征学习质量, 因而可能获得超越当前未做预训练的 ViT 类模型的 performance.

3 总结

相比于试验各种提高 performance 的 tricks, 我们小组在本次任务中主要探索了最关键的三方面改进: 模型架构, 数据增强, 以及自监督学习, 并通过严谨的控制变量实验, 分别得到了三大因素各自对于整个任务的提升效果. 我们也在自监督训练的实验结果之上, 进一步提出了使用 MoCo 类模型做预训练的改进措施. 我们真诚地希望这份实验报告也能够为其他 few-shot learning 任务的解决提供一些帮助.

致谢 真诚感谢老师和助教学长在 PJ 完成过程中提供的帮助.

参考文献

1 xinlei Chen. "Improved baselines with momentum contrastive learning" . In: arXiv preprint arXiv:2003.04297 (2020).
2 Kaiming He et al. "Momentum Contrast for Unsupervised Visual Representation Learning" . In: arXiv preprint arXiv:1911.05722 (2019).
3 Zhirong Wu. "Unsupervised feature learning via non-parametric instance discrimination" . In: Proceedings of the IEEE conference on computer vision and pattern recognition (2018).
4 Mang Ye. "Unsupervised embedding learning via invariant and spreading instance feature". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019).

A 附录

用于 PJ 数据（可视化）记录的 wandb 网址: $https://wandb.ai/raidriar\_dai/aicourse?workspace=user-qirundai$