Few Shot Learning

2018011359 计84 乐阳

组员：杜晨熙，李芷冰，肖雅迪

任务简介

少样本学习是在给定样本数很少的情况下高效的学习出数据的特征，完成分类等任务。

在本项目中，我们将在一个预训练模型（AlexNet在ImageNet-1K）的基础之上，对Caltech256数据集上的50个新类别进行小样本学习。

我们采用的主要方法基于Prototypical Network及其变种。我们尝试了使用模型的不同**隐层输出**作为图片特征，使用不同的**测试方法**给出查询图片的标签，**是否需要**对网络进行训练，以及在训练时使用的**损失函数**种类等。

方法

问题定义

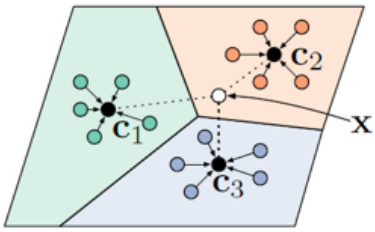
在一个少样本学习任务中，共有 K 个类别，每个类别有 N 个训练样本，这些带有标签的样本称为支撑集。

$$S = \cup_{k=1}^{K} S_k, |S_k| = N$$

我们希望得到一个特征提取网络 $z = f(x)$，利用支撑集中图片的特征和标签$\{(z_i, y_i)\}_{i=1}^{KN}$，判断测试集图片的标签。在特征空间中，衡量两个向量相似程度的距离函数为 $d(z_i, z_j)$，常见的有欧式距离、余弦距离等。

$$d_e(z_i, z_j) = \|z_i - z_j\|_2^2$$
$$d_c(z_i, z_j) = \frac{\langle z_i, z_j \rangle}{\|z_i\|\|z_j\|}$$

Prototypical Network



根据Prototypical Network的思想，一个类别的Prototype其实就是这个类别所有图像的特征的平均

$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f(x_i)$$

给定一个新的样本 x，其所处的类别的概率分布由其特征与所有类的Prototype的距离决定

$$p(y=k|x)=\frac{\exp(-d(f(x),c_k))}{\sum_{k'}\exp(-d(f(x),c_{k'}))}$$

在网络 f 的训练过程中，将整个训练集随机分为支撑集 S 和查询集 Q 两部分（遵循原论文的训练设计），损失函数为查询集图片属于正确类别的概率值的倒数。训练的过程就是让查询集与其对应类别的Prototype的距离尽量小。

$$L_{\text{proto}}(S,Q)=-\frac{1}{|Q|}\sum_{(x_i,y_i)\in Q}\log(\frac{\exp(-d(f(x_i),c_{y_i}))}{\sum_k\exp(-d(f(x_i),c_k))})$$

基于距离的小样本学习：更多尝试

Prototypical Network 是一类基于距离的小样本学习的代表，我们还尝试了它的几类变种。

首先，给定支撑集图片的特征，查询集中图片的标签如何确定其实有各种不同的方法。除了计算Prototype，还有K-近邻等方法。这里我们举出三种

- Prototype方法：$y(x)=\arg\min_k d(f(x),c_k)$
- K-近邻方法：$y(x)=\arg\max_k\sum_{(x_i,y_i)\in N}1_{y_i=k}$，其中 N 为距离查询图片 x 最近的K张支撑集图片
- 软分配方法（Soft Assignment）：$y(x)=\arg\max_k\sum_{(x_i,k)\in S}\exp(-d(f(x),f(x_i)))$

其次，我们也可以用其他损失函数来替代Prototypical损失，其中一种就是近邻成分分析(Neighborhood Component Analysis, NCA)。NCA损失对一个批次的数据定义，且不需要把训练集分为支撑集和查询集两部分。

$$L_{\text{NCA}}(B)=-\frac{1}{|B|}\sum_{(x_i,y_i)\in Q}\log(\frac{\sum_{j\neq i,y_i=y_j}\exp(-d(z_i,z_j))}{\sum_{k\neq i}\exp(-d(z_i,z_k))})$$

直观理解，NCA损失函数同时完成两件事：最大化不同类别样本的距离，最小化同类别样本的距离。

实验及结果

实验设定

经过一些预实验，我们发现如果不冻结预训练网络的参数，少量的样本将不足以驱动大网络学习到泛化性较好的特征。因此我们将冻结预训练的Backbone，利用其某一层的隐层的输出作为图片的特征，在此基础上再进行下一步操作。AlexNet由前卷积网络和MLP分类器组成，因此我们可以提取出MLP的不同层作为图像的特征。具体而言，我们选择了三个位置的隐层用作特征：

```
# classifier of AlexNet
self.classifier = nn.Sequential(
    # layer 3
    nn.Dropout(),
    nn.Linear(256 * 6 * 6, 4096),
    nn.ReLU(inplace=True),
    # layer 2
    nn.Dropout(),
    nn.Linear(4096, 4096),
    nn.ReLU(inplace=True),
    # layer 1
    nn.Linear(4096, num_classes),
)
```

选择多个位置的原因是：预训练模型的最后输出可能过度拟合了预训练任务，不利于迁移到新任务中。

以下所有实验中，我们的数据集类别个数为50，每类有10个样本。

非训练直接测试

首先，我们考虑不做任何训练，直接将预训练网络的输出作为特征进行测试。这样做的考量是：我们相信预训练网络已经有了足够的表达能力。事实证明我们的猜想基本正确，**直接使用预训练网络的测试效果已经很好了**。按照上文的介绍，我们有三种不同的测试方法：Prototypical，K-近邻，软分配；同时我们尝试了欧氏和余弦两种距离。另外我们还用rbf核的SVM作为基线。实验结果如下：

	SVM	Proto(cos)	KNN(cos)	Soft(cos)	Proto(L2)	KNN(L2)	Soft(L2)
Layer 1	58.0	67.27	57.73	64.2	64.87	43.53	2.07
Layer 2	58.9	**68.47**	59.47	62.47	64.93	37.67	2.07
Layer 3	52.6	65.07	55.4	58.73	58.8	15.33	2.07

可见使用Prototypical和软分配方法的性能高于K-近邻，使用余弦距离的性能大大超出欧式距离。我们还能发现在预训练模型的某一隐层（Layer2）的效果高于其他位置。在不训练的情况下，网络已经能达到68.47%的测试正确率。

此外，我们还考虑了对训练集做**数据增强**，部分降低小样本带量的影响。事实证明一些增强（并非所有）能够使直接测试的性能进一步提高。

具体而言，我们尝试了将图片做水平翻转、转换为灰度这两种操作，收到了一定的成效。

	Proto(cos)	KNN(cos)	Soft(cos)
No Aug	68.47	59.47	62.47
+hFlip	68.53	62.27	62.73
+hFlip + rgb2gray	**69.0**	62.8	62.47

训练网络

既然非训练方法已经有一定的基础性能，我们相信如果我们引入可以学习的参数将有更好的效果。具体而言，我们让预训练模型的输出经过一个可学习的线性层，映射到维数为 H 的空间中。在这个空间中我们计算损失，训练该线性层。

根据前文的介绍，我们有Prototypical损失和NCA损失两种损失函数可供使用；此外我们还可以直接将特征维度设为50（类别数），使用交叉熵损失做监督训练。这几种训练方法的性能比非训练的测试结果略高一筹。

Eval Directly	Eval with Aug	Fine Tune	Proto Loss	NCA Loss
68.47	69.0	68.8	**69.87**	69.07

以上测试都使用了余弦距离Prototypical方法（前一部分得到性能最好的测试方法）。

我们尝试了不同的特征维维数 H 的大小，最终在 $H = 1024$ 时得到了最好的结果。维数过小则不足以将多个类别区分，而维数过大则引入更多参数，更容易造成过拟合。

我们还尝试了对训练数据做数据增强，但在训练的情况下并没能取得更好的效果。

我们还尝试了在训练数据中加入基类的特征（base feature），但最终发现额外的数据会造成性能的损失，推测是额外的类别降低了训练目标的50类的敏感性。

结论

本项目中，我们利用Prototypical Network的思想完成了Caltech256上的小样本学习，实验结果表明：

- 直接使用预训练模型测试的性能已经很高，再添加一个可学习的线性映射可以进一步提高性能。
- 使用基于Prototype的测试方法和余弦距离效果最好。
- 使用 Prototypical 损失的效果略好于 NCA 损失。

最终我们达到的测试正确率为 69.87。

参考文献

1. Snell J, Swersky K, Zemel R S. Prototypical networks for few-shot learning[J]. arXiv preprint arXiv:1703.05175, 2017.
2. Laenen S, Bertinetto L. On Episodes, Prototypical Networks, and Few-shot Learning[J]. arXiv preprint arXiv:2012.09831, 2020.
3. Wu Z, Efros A A, Yu S X. Improving generalization via scalable neighborhood component analysis[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 685-701.
4. Wang, Y. X., & Hebert, M. (2016, October). Learning to learn: Model regression networks for easy small sample learning. In *European Conference on Computer Vision* (pp. 616-634). Springer, Cham.

个人贡献说明

- 文献调研由全组成员共同完成
- 本人完成了大部分Prototypical Network的代码编写和实验
- 其他组员还尝试了Model Regression方法（本报告中未列出）