实验报告（赛道一：开放域少样本分类任务）

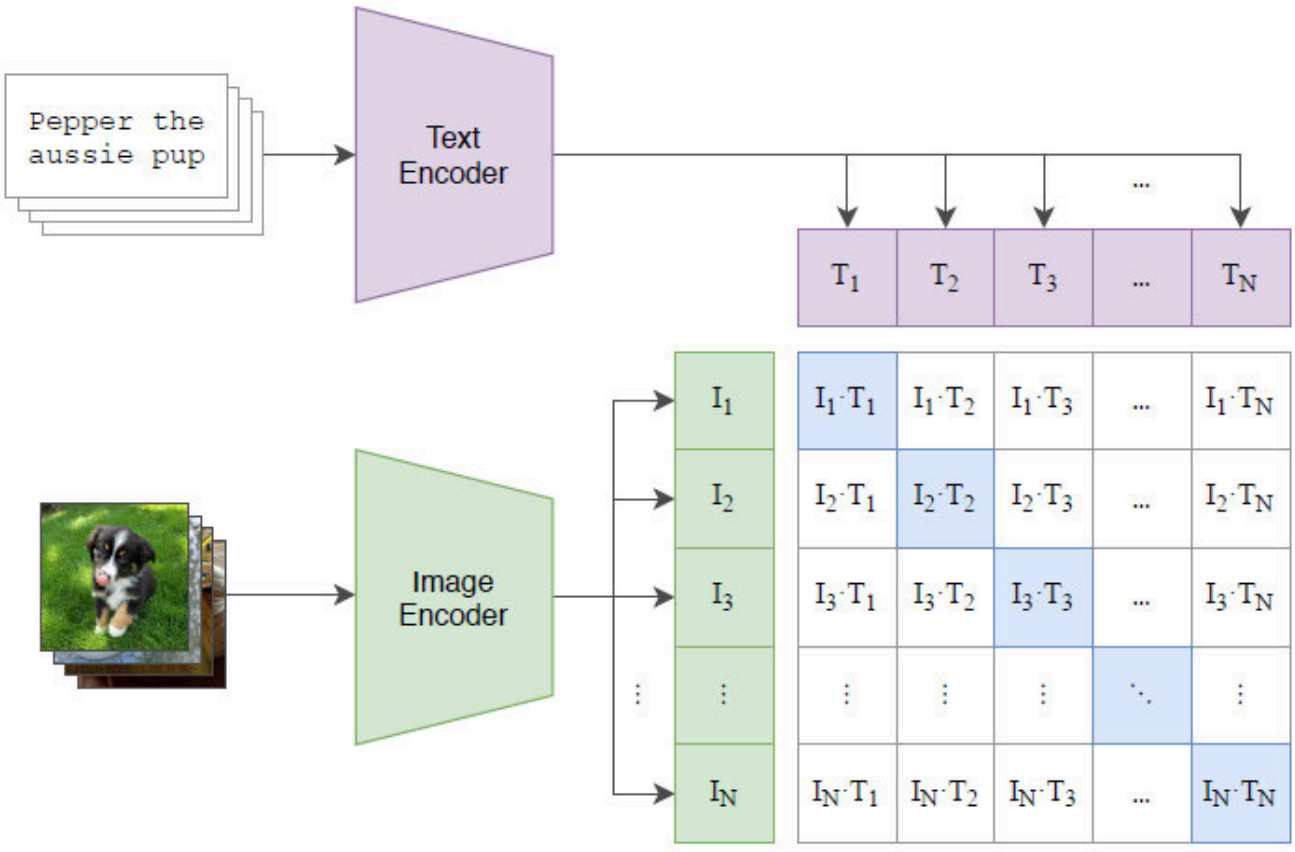
[redacted]

一、实验目的

在CLIP预训练模型的基础上，搭建利用少样本(每种类别仅**4**张图像)训练模型，从而在**374**类别分类任务下取得比 `baseline.py` 和 `baseline_ft.py` 更高的**top1**和**top5**准确率。

二、基础模型

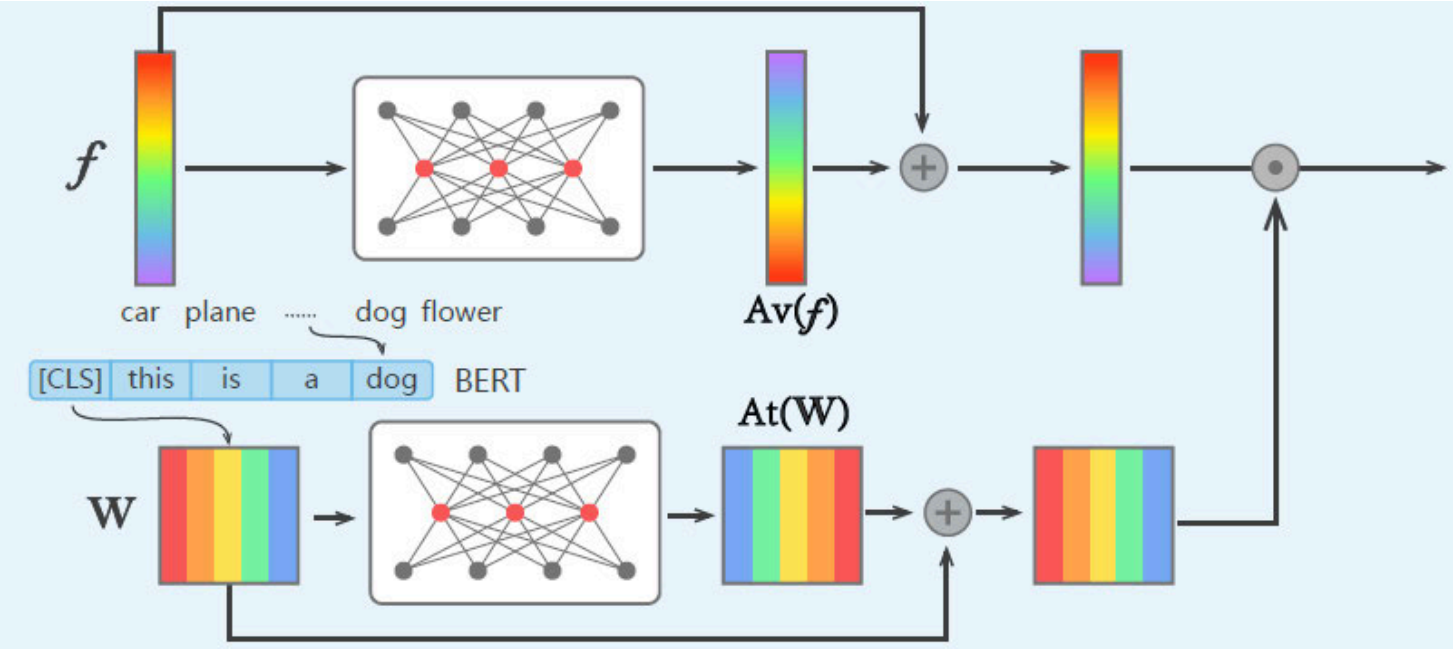
1. CLIP模型:



用自然语言监督学习图像表示，利用4亿个**文本-图像pair**的训练样本。如论文中图1所示，将N个图像和其对应的自然语言文本分别经过编码器后转化为**512-D**向量嵌入到具有潜在语义信息的空间中，做内积运算得到相似度矩阵$M \in \mathbb{R}^{N \quad N}$。

矩阵M对角线上的样本是正样本，不在对角线上的样本是负样本。 因此共有N个正样本和N^2 N个负样本。论文作者指出通过对比学习过程训练得到的Text Encoder和Image Encoder构成的CLIP模型在多个数据集上进行分类任务上取得了较好的zero-shot性能，即不进行微调，在未经过训练的样本上进行分类任务获得了较好的成绩。

2. CLIP-Adapter



CLIP-Adapter是CLIP模型的微调方法，旨在提升模型在少样本学习场景下的性能。它通过在CLIP模型的视觉或语言分支上引入轻量级的特征适配器来实现这一目标。这些适配器是瓶颈层，通过线性变换学习新的特征表示，同时利用残差连接技术与原始预训练特征相结合，以防止过拟合并保留有价值的原始知识。残差比例α和β作为调节因子，控制着新旧特征的融合程度，它们可以是手动设定的超参数，也可以通过超网络动态学习得到最优值。不妨设经过编码器的图像和文本特征向量为$\mathbf{f}$和$\mathbf{w}$,则上述过程即为

$$\mathbf{f}^* = \alpha\mathbf{f} + (1 \quad \alpha)A_v(\mathbf{f})$$

$$\mathbf{w}^* = \beta\mathbf{w} + (1-\beta)A_v(\mathbf{w})$$

在微调过程中，CLIP-Adapter仅更新适配器的参数，而保持原始CLIP模型参数不变，这样做大幅减少了训练过程中需要调整的参数数量，有效降低了过拟合风险。CLIP-Adapter在生成分类权重时，采用了与传统分类任务相似的框架，但通过适配器对特征表示进行了优化，提高了分类的准确性。此外，CLIP-Adapter设计了三种结构变体，分别针对图像分支、文本分支以及同时对两者进行微调，以适应不同的应用场景。

CLIP-Adapter在训练时使用交叉熵损失函数来优化适配器参数，确保模型能够准确预测类别标签。特别地，它针对少样本学习进行了优化，即使在标注样本数量有限的情况下也能快速学习并泛化到新数据上。通过这种方式，CLIP-Adapter能够显著提升模型在视觉分类任务中的性能，尤其是在数据稀缺的环境中。

三、方法改进

- **训练数据选择**：相比于 `baseline` 按照顺序取样可能容易取到较为相似的图片，改进后采用随机采样方法，每个类别随机采样4个图片-标签对作为训练集数据。
- **模型调整**： `baseline.py` 没有进行训练，是基于image-encoder和text-encoder进行的zeroshot分类方法，以而 `baseline_ft.py` 采用的方法是只基于image-encoder，并利用训练数据额外训练一个逻辑回归分类器头。本实验采用CLIP-Adapter方法，仅在CLIP模型的图像分支上进行改进：引入全连接层，并以残差形式将原来的图像特征输出以一定的比例加到全连接层输出上。

```
class Adapter(Model):
    def __init__(self,c_in=512,reduction=4):
        super(Adapter,self).__init__()
        self.fc = nn.Sequential(
            nn.Linear(c_in,c_in//reduction,bias = False),
            nn.ReLU(),
            nn.Linear(c_in//reduction,c_in,bias = False),
            nn.ReLU()
        )
    def execute(self, x):
        x = nn.dropout(x,0.1)
        x = self.fc(x)
        return x


class CustomJCLIP(Model):
    def __init__(self):
        super().__init__()
        self.text_encoder = model.encode_text
        self.dtype = model.dtype
        self.adapter = Adapter()
    def execute(self, image_feature,text_features):
        x = self.adapter(image_feature)
        global ratio
        image_feature = ratio * x + (1 - ratio) * image_feature
        image_feature = image_feature / image_feature.norm(dim=-1, keepdim=True)
        predict_vector = (100.0 * image_feature @ text_features.transpose(0, 1)).softmax(dim=-1)
        _ , top_label_predicted = predict_vector.topk(1)
        predict_vector = jt.float32(predict_vector)
        return predict_vector, top_label_predicted
```

- **训练过程**：

损失函数：交叉熵损失函数

$$\mathcal{L}(\mathbf{\Theta}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk}\log(p_{nk})$$

优化器：Adam优化器

$$\mathbf{m} \leftarrow \beta_1 \cdot \mathbf{m} + (1-\beta_1)\cdot \mathbf{g}$$

$$\mathbf{v} \leftarrow \beta_2 \cdot \mathbf{v} + (1-\beta_2)\cdot \mathbf{g}^2$$

$$\hat{\mathbf{m}} \leftarrow \frac{\mathbf{m}}{1-\beta_1^t}$$

$$\hat{\mathbf{v}} \leftarrow \frac{\mathbf{v}}{1-\beta_2^t}$$

$$\mathbf{x} \leftarrow \hat{\mathbf{m}}/(\sqrt{\hat{\mathbf{v}}}+\epsilon)$$

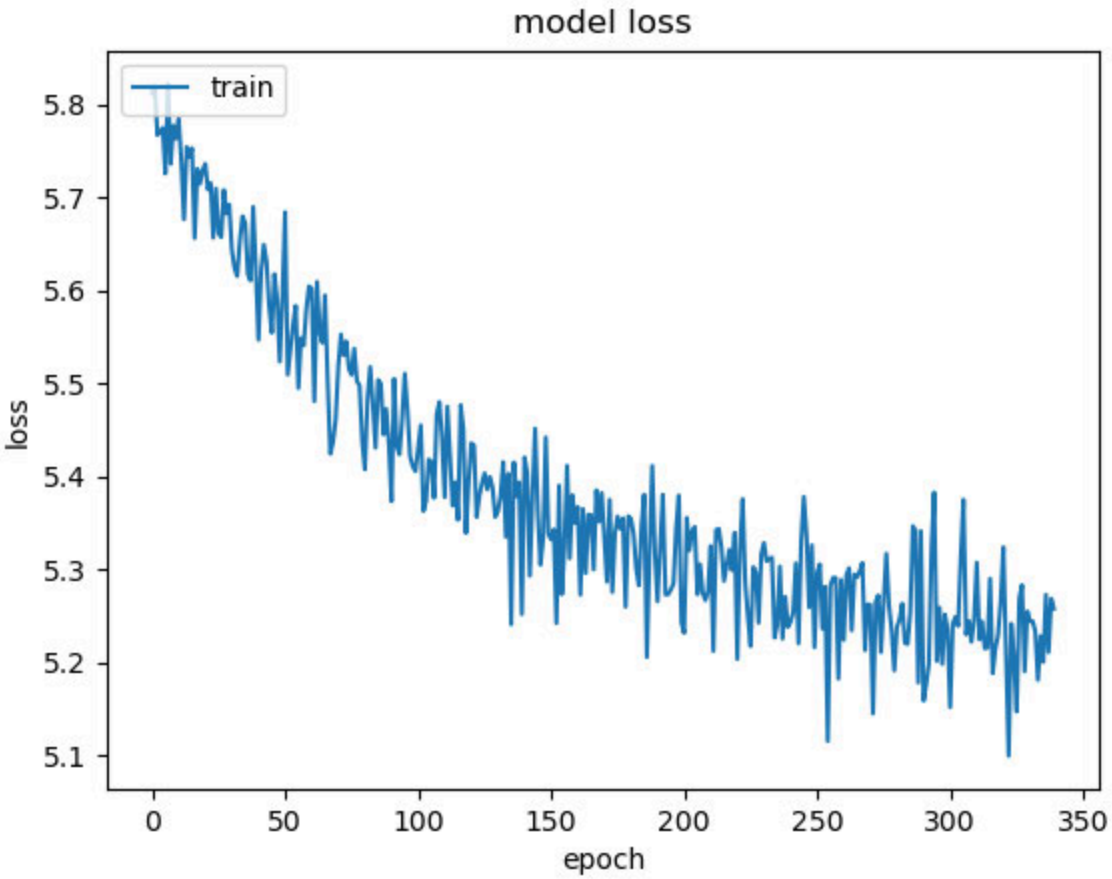
调整类别的语言描述

三、实验结果

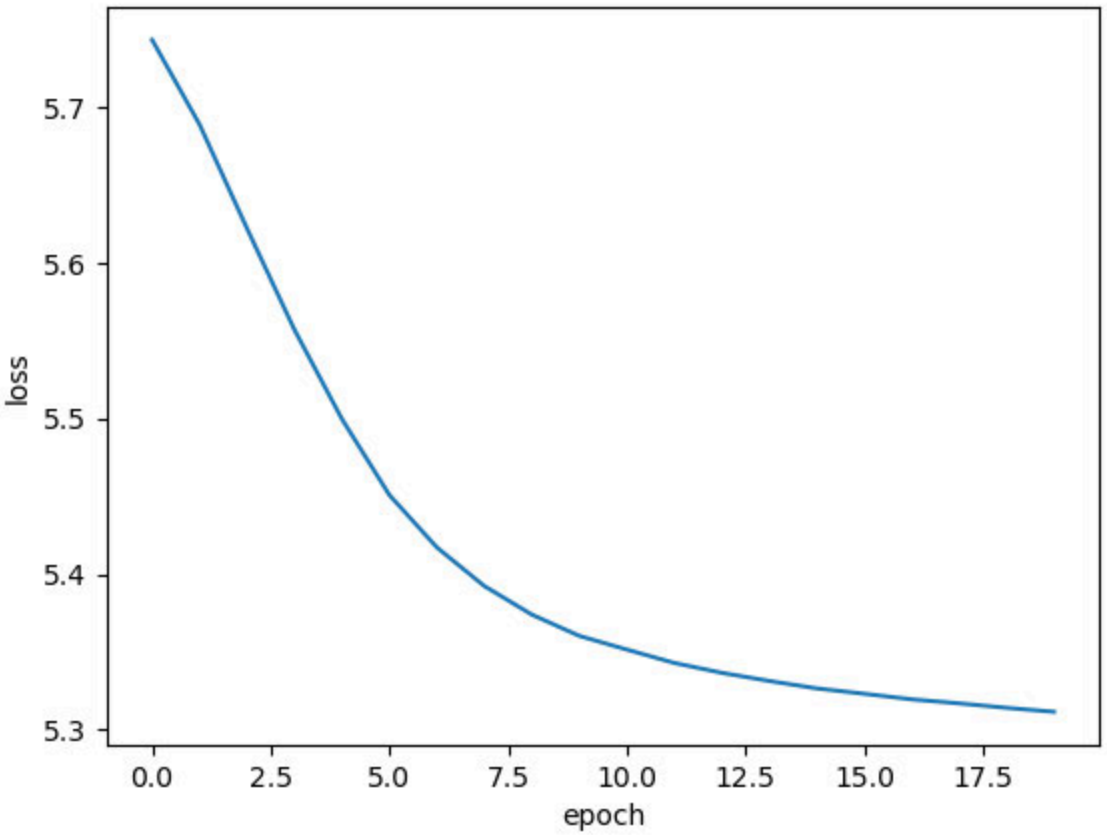
将1400多个训练样本随机按照4：1的比例分为训练集和验证集。下面展现的是按照如下的超参数配置得到的实验结果

- epoch = 20
- batch_size = 64
- ratio = 0.9
- learning_rate = 0.0001

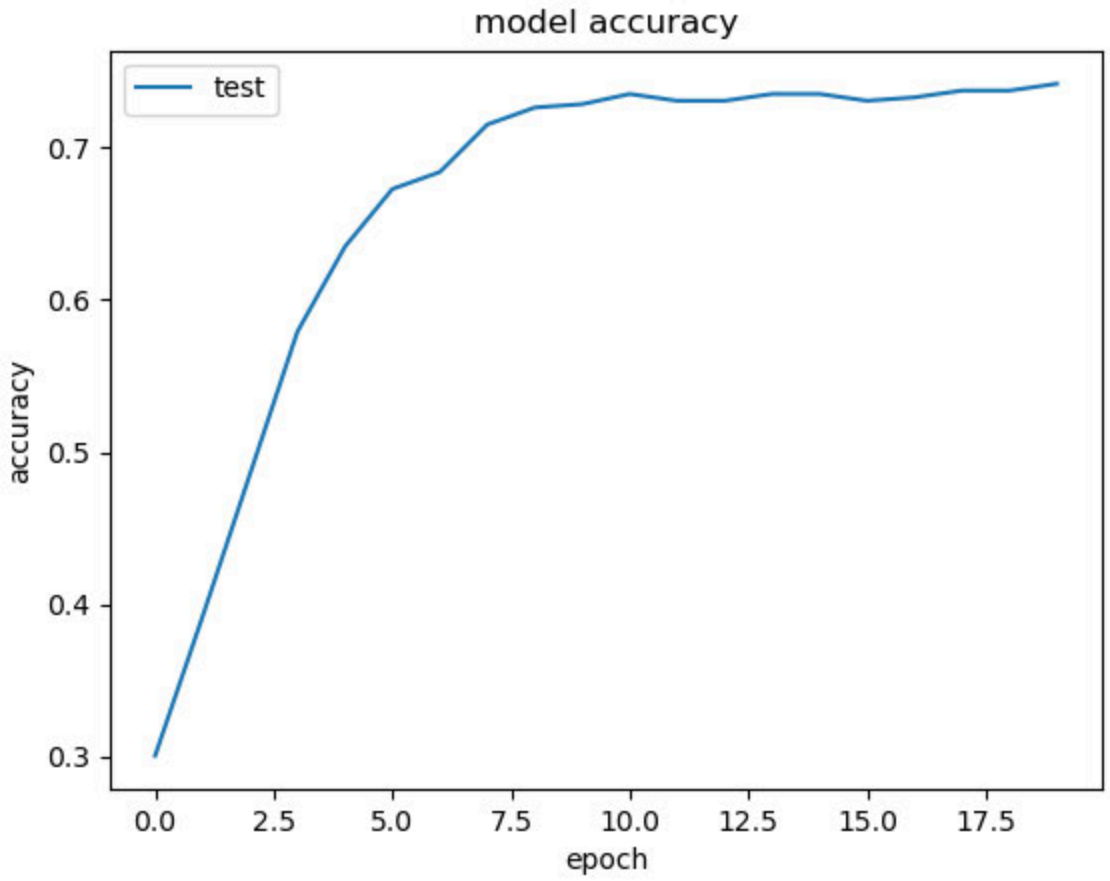
如图分别展示了在训练的过程中，在训练集的loss下降曲线（注：每一个batch记录一次数据，因此会有强烈震荡,然后这个图的横轴标错了，标度应该除以20）



在验证集上的loss下降曲线（注：每个epoch记录一次数据，因此比较平滑）



在验证集上的top1accuracy上升曲线（注：每个epoch记录一次数据，因此比较平滑）



然而，在头歌测评平台上的准确率并不高，top1-accuracy仅仅为63%左右，top5-accuracy也仅为82%左右。

四、心得体会

展示实验的主要结果

- 不同模型的性能比较

以下为不同模型的准确率比较：

Invalid Equation

2024072512134079904245	result (5)...	王子轩	2024-07-25 12:13:40	完成	0.6267	0.8207	下载

- 实验结果分析
  调超参数对平台测试的准确率提高哦似乎没有什么效果，不知道是不是平台上的测评数据和提供的traindataset有比较大的gap
  盲猜是评测数据用的大多是狗细颗粒度分类的数据。。。。。

- 一点点心得体会

实验过程还是比较顺利的，通过对比学习的思想，引入适配器，在图像分支上进行微调，在少样本学习场景下，取得了不错的效果。但是实验结果并不理想，平台上的准确率并不高，可能是因为数据集的不匹配，或者是模型的过拟合。
但是由于对计图框架不是很熟、、、而且是第一次用Python写代码（）、、调试用了比较久的时间，经常报错张量维度不匹配等等问题（（（（
不过学到了很多东西嘿嘿
谢谢穆老师和助教们的指导帮助

参考文献

arXiv:2103.00020v1 [cs.CV] 26 Feb 2021
arXiv:2110.04544v1 [cs.CV] 9 Oct 2021