机器学习课程实验报告

实验者：何峙

学号：21215122

专业：大数据与人工智能

实验目标

1. 掌握Pytorch等深度学习框架的环境搭建

2. 掌握fine-grained图像分类任务的训练和测试流程

实验步骤

1. 配置实验环境如下（详细步骤略）

* Python v3.19.12
* Pytorch v1.11.0
* Visual Studio Code

2. fine-grained图像分类

本实验以JU HE等提出的transFG【hoho】模型为依据，进行细粒度图像分类实验。

* 使用Vision Transformer作为基础图片特征提取器

Vision Transformer（以下简称ViT）是Google【hoho】提出的一个基于transformer的图片分类模型，可从https://console.cloud.google.com/storage/vit\_models/下载其预训练模型，然后进行模型加载（本实验使用ViT-Base-16的模型配置）：

config = CONFIGS['ViT-B\_16']

model = VisionTransformer(config, num\_classes=1000, zero\_head=False, img\_size=224, vis=True)

model.load\_from(np.load('model\_checkpoints/ViT-B\_16-224.npz'))

本层会输出所有patch\_embedding（包含CLS\_embedding）。

* 搭建Part Selection Module层

本层主要用于自注意力的加强，以选取CLS\_token关注度高的图片patch\_token：

class PartLayer(nn.Module):

def \_\_init\_\_(self, vit\_config):

super(PartLayer, self).\_\_init\_\_()

self.part\_transformer = Block(vit\_config, vis=True)

self.part\_norm = LayerNorm(vit\_config.hidden\_size, eps=1e-6)

def forward(self, vit\_features, att\_weight\_list):

att\_part\_index = fetch\_part\_attention(vit\_features, att\_weight\_list)

part\_feature = fetch\_part\_features(vit\_features, att\_part\_index)

part\_states, part\_attention\_weights = self.part\_transformer(part\_feature)

part\_states = self.part\_norm(part\_states)

return part\_states, part\_attention\_weights

本层最后输出注意力高的K个图片patch\_embedding（K为注意力头的数量）,以及CLS\_embedding。

* 搭建最终图片分类层

经过Part Selection Module层的输出，将其中CLS\_embedding输入到最终的全连接层以获得最终的图片分类分布：

viT\_embed\_dim = 768

n\_classes = len(class2idx)

classifier = nn.Linear(viT\_embed\_dim, n\_classes)

综上，通过对ViT预训练模型进行fine-tune，以及对新增的Part Selection Module层和最终图片分类层进行训练，可完成图片细粒度分类任务。模型架构图1所示。

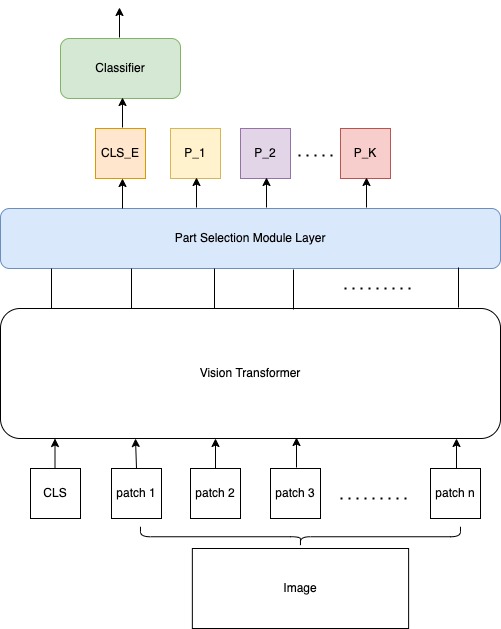


图1 实验模型架构

实验结果

本实验使用CUB-200-2011数据集，它是一个鸟类品种的数据集，一共118080张图片，其中含200个鸟类品种。本实验将训练集大小设置为80%的数据集，10%为验证集大小，剩下10%为测试集大小。训练的学习率设置为3.x 10-6， 训练轮数为20轮，硬件环境Geforce GTX 2070。

训练结果如图2所示。可见训练集误差与验证集误差随训练过程不断减少，对测试集的准确度不断上升。训练结束时，模型在测试集的分类准确度大约为88%。

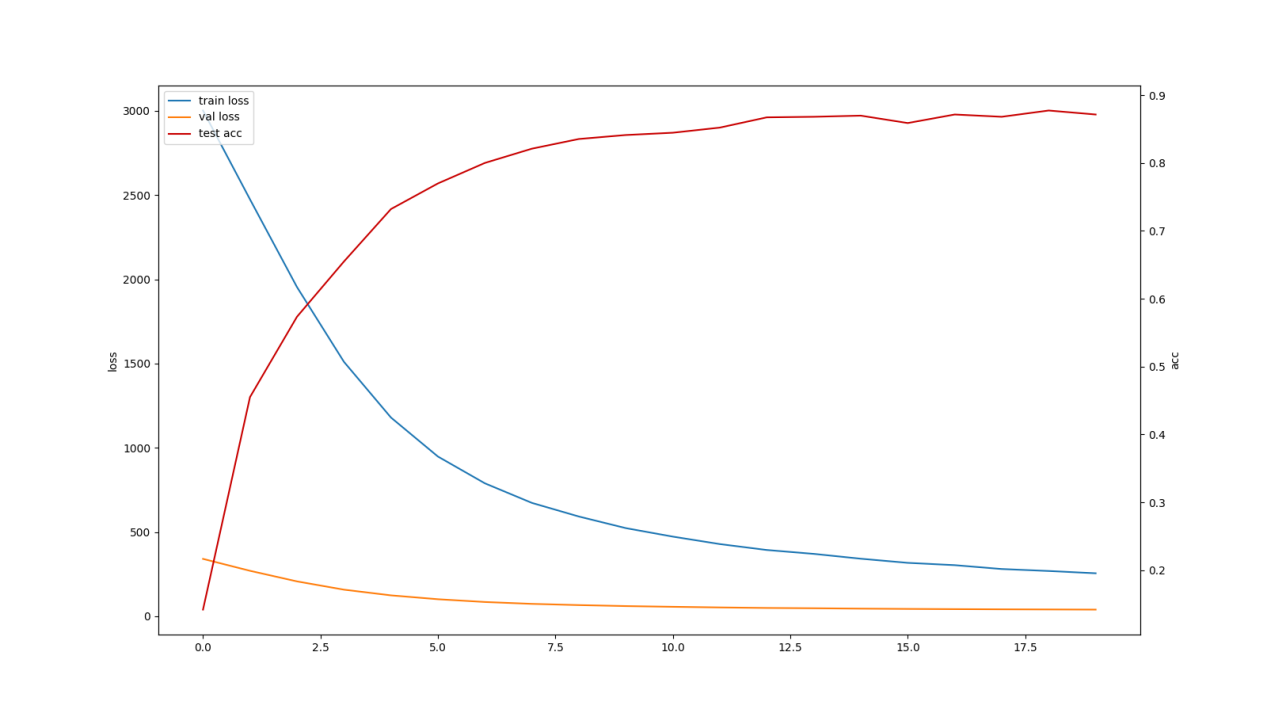
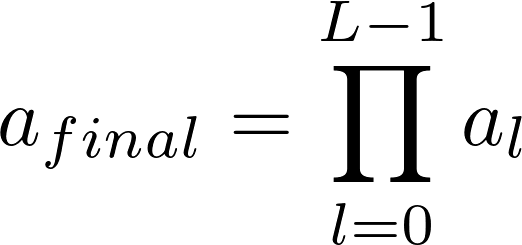


图2 模型训练效果度量

实验分析

ViT利用transformer的优势，可以学习到图片每个patch\_token的相互注意力，所以可以得到最后用于分类的CLS\_token所关注的图片区域。而TransFG【hoho】对这些注意力进一步加强，它把ViT每层transformer学习到的注意力权重进行连乘：



然后对afinal中的每个注意力头，找到CLS\_token对其他图片patch\_token的最大注意力值的序号，即表示CLS\_token最关注的图片的某个patch\_token，以此形成对图片某些区域更加精确的关注。图3为利用模型最终输出的注意力权重对原始图片进行遮罩的效果叠加，形象表示了模型确实可以关注到图片对于类别的真实区域。

图3 模型对图片的关注区域（左图为原图，右图为遮罩效果图）

参考文献