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[1]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[2]（以下简称 ViT）是 Google 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。

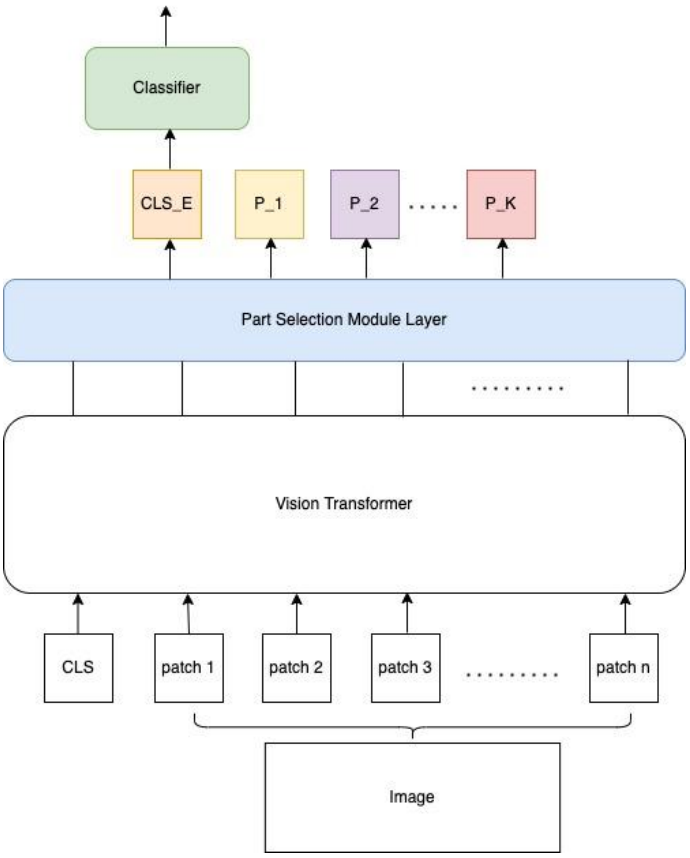

图 1 实验模型架构

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，其划分为 200 个鸟类品种。本实验将数据集大小的 80%设置为训练集，10%设为验证集，剩下 10%为测试集。训练的学习率设置为 $3.x\ 10^{-6}$，使用 Adam 优化方法，训练轮数为 20 轮，硬件环境 Geforce GTX 2070。

训练效果如图 2 所示。可见训练集误差与验证集误差随训练进行而不断减少，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%。

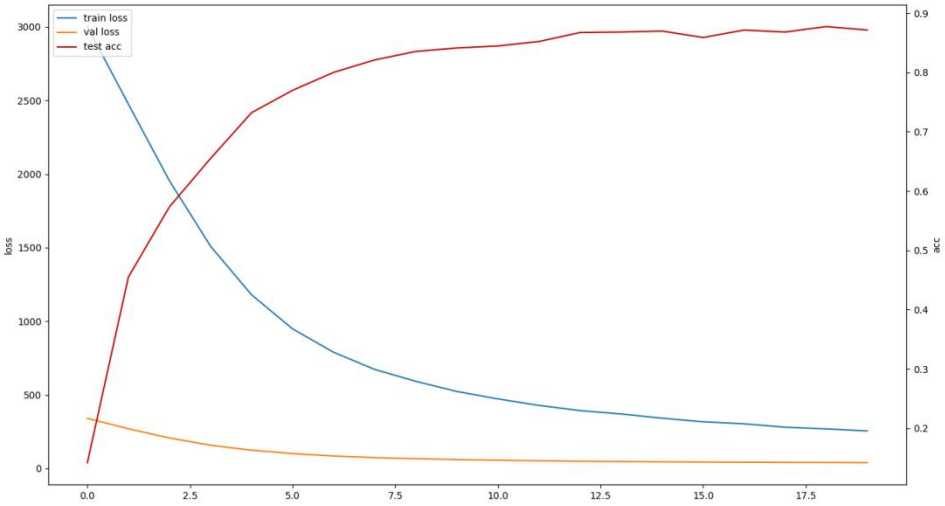

图 2 模型训练效果度量

实验分析

ViT 利用 transformer 的特点，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[1]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：

$$a_{final} = \prod_{l=0}^{L-1} a_l$$

然后对于 a_{final} 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 显示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）

代码说明

所有代码运行前需在 https://console.cloud.google.com/storage/vit_models/下载 ViT 预训练模型文件，并将其放至 model_checkpoints 文件夹，然后可运行如下代码文件：

- transfg_run.py 为 TransFG 模型训练与验证；
- object_recognize.py 为将图片加上注意力遮罩效果；

参考文献

[1] J. He *et al.*, “TransFG: A Transformer Architecture for Fine-grained Recognition.” arXiv, Dec. 01, 2021. Accessed: Jun. 09, 2022. [Online]. Available: http://arxiv.org/abs/2103.07976

[2] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.” arXiv, Jun. 03, 2021. Accessed: Jun. 09, 2022. [Online]. Available: http://arxiv.org/abs/2010.11929