上机试题——现代神经网络：深度卷积神经网络进行花分类的实现与理论验证

本试题要求掌握深度学习模型构建、训练、与预测的基本方法，理解将深度模型用于实际应用相关的重要概念，并进行代码实现。

注意：在实验过程中请保存训练过程中 **loss 和精度随 epoch 变化**以及使用训练好的模型进行**预测所得结果**的**截图**，便于考核。

基本任务——模型的构建、训练、预测：

1. 通过**编辑 model.py**，定义模型的各个子模块和前向过程：
   1) 使用 nn.Sequential 在**__init()__**中定义 AlexNet 的 2 个子模块：
   a） 特征提取器 self.features：
   self.features 中包含 5 个卷积层和 3 个最大池化层，并在每个卷积层后使用 ReLU 激活函数来为模型带来非线性。具体地，其中依次堆叠了如下结构：

layer_name	kernel_num	kernel_size	stride	padding
Conv1	48	11	4	2
Maxpool1	-	3	2	0
Conv2	128	5	1	2
Maxpool2	-	3	2	0
Conv3	192	3	1	1
Conv4	192	3	1	1
Conv5	128	3	1	1
Maxpool3	-	3	2	0

   提示：
   - 特征提取器的输入是 3 通道 224*224 的 RGB 图像（input：[3, 224, 224]），输出的形状则为[128, 6, 6]；
   - 使用 MaxPool2d 不会改变通道数量。

   b） 分类器 self.classifier：
   self.classifier 的输入为展平后的图像特征，输出为图像属于各个类别的概率，并包含 3 个线性层。3 个线性层的参数情况如下：

layer_name	in_features	out_features
FN1	128 * 6 * 6	2048
FN2	2048	2048
FN3	2048	num_classes

   要求：
   - 对于前两个线性层，在调用前使用概率为 0.5 的 dropout 操作来缓解过拟合，在调用后使用 ReLu 来增加非线性。
   - 最后一个线性层则负责将提取到的特征映射到与类别数数量相同个输出。

   2) 在 **forward()**中补充模型前向过程的定义：
   a) 通过 self.features 提取输入图像的特征；
   b) 使用 torch.flatten()将特征展平为适配分类器输入的形状；
   c) 将特征输入 self.classifier 得到图像属于各个类别的概率。
2. 通过**编辑 train.py**，实现以下训练设置并补充训练过程：
   1) 训练设置：

a） 定义损失函数 loss_function：采用二元交叉熵损失
b） 定义优化器对象 optimizer：AdamW，学习率为 0.0002
2) 在每个 epoch 中，实现以下操作：
a） 清除模型中所有参数（tensor）的梯度；
b） 将 images 输入我们的模型 net，得到输出 outputs;
c） 根据模型的输出 outputs 和真实标签 labels 计算损失 loss;
d） 利用 loss 计算梯度;
e） 更新模型参数。

3. 使用 **predict.py** 对验证集中的指定鲜花图像（daisy/15207766_fc2f1d692c_n.jpg）以及从网上获取的 daisy 鲜花图像进行预测，分别记录它们属于各个类别的概率。

进阶任务——理论验证与分析：

1. **模型的收敛速度：**从作业中使用的鲜花分类数据集中**选择 4 类花朵（去除 tulips）作为训练数据**进行尝试，与使用 5 类数据训练得到的模型比较训练误差和在验证集上的精度的变化趋势。在这个过程中，你可能会发现使用 4 类训练数据的情况下训练更少的 epoch 时就已经达到了更好的精度。

2. **过拟合：**作业中使用的鲜花分类数据集中**选择 4 类花朵（去除 tulips）作为训练数据的同时减少训练样本（使用 reduced-c4 中的训练数据，每类仅保留 30 个训练样本）**进行尝试。在这个过程中，你可能会发现使用少量的训练数据的情况下模型的训练误差能够减小但验证集上的精度的提升减少了，这时可能出现了过拟合的现象。