

《模式识别与机器学习》课程大作业实验报告

基于传统方法与深度神经网络的鸟类图像分类

自动化系 2023011015 胡延

1 作业背景与任务分析

本次大作业旨在解决 CUB-200 鸟类数据集的细粒度图像分类问题。任务分为两部分：

- 基于传统方法的分类：利用官方提供的预提取特征向量（384维），重点考察对SVM、逻辑回归等经典机器学习算法的理解与实现，以及在特征空间上的分类性能对比。
- 基于深度神经网络的分类：直接以原始 RGB 图像作为输入，要求从头训练深度神经网络，禁止使用预训练权重。重点在于深度网络结构的设计以及训练策略的综合调优。

评价指标为测试集上的分类准确率

$$Acc = \frac{\text{正确分类样本数}}{\text{样本总数}} \times 100$$

2 任务一：传统方法

2.1 分析与代码实现

任务一的输入数据为 384 维的特征向量。特征经过高层抽象后通常使数据在特征空间中表现出较好的可分性，故可以使用。根据前半学期所学内容，本次作业实现了四种分类器并进行结果比较分析。

2.1.1 逻辑回归

基于 NumPy 实现多分类逻辑回归，采用 Softmax 函数将线性输出映射为概率分布，使用交叉熵损失函数衡量预测偏差；使用梯度下降来迭代更新权重矩阵 **W** 和偏置 **b**。为防止过拟合，在损失函数中引入 $\lambda ||\mathbf{W}||^2$ 进行 L_2 正则化。

```
class LogisticRegression:
    def fit(self, X, y):
        self.weights = np.random.randn(n_features, n_classes) * 0.01
        for i in range(self.n_iterations): # 迭代进行梯度下降
            z = np.dot(X, self.weights) + self.bias
            y_pred = self._softmax(z)
            loss = -np.mean(np.sum(y_one_hot * np.log(y_pred + 1e-8),
axis=1))
            dz = (y_pred - y_one_hot) / n_samples
            self.weights -= self.learning_rate * dw
```

2.1.2 支持向量机 (SVM)

基于 One-vs-Rest 策略实现多分类 SVM；对 K 个类别训练 K 个二分类器，其中第 i 个分类器将类别 i 视为正例，其余所有类别视为负例。参考学期中的编程小作业，实践对比线性核、RBF 径向基核和多项式核进行横向

比较。使用 **sklearn** 机器学习库辅助，在 **SVM** 类中实现多分类的投票与决策逻辑并封装如下：

```
class SVM:
    def fit(self, X, y):
        self.classes = np.unique(y)
        for cls in self.classes: # 为每个类别训练一个二分类器
            y_binary = np.where(y == cls, 1, -1)
            binary_clf = self._train_binary_svm(X, y_binary)
            self.classifiers.append(binary_clf)

    def predict(self, X):
        '''汇总决策分数'''
        decision_values = np.zeros((n_samples, n_classes))
        for i, clf in enumerate(self.classifiers):
            decision_values[:, i] = clf.decision_function(X)
        return self.classes[np.argmax(decision_values, axis=1)]
```

2.1.3 贝叶斯和 KNN

高斯朴素贝叶斯分类器手动计算了每个类别的先验概率以及特征的均值和方差，并基于高斯分布假设计算后验概率；第四组实验基于欧氏距离实现 KNN 分类器，通过统计最近 K 个邻居的类别进行多数投票。考虑到数据量较大，适当提高 K 值到 5 以平滑噪声影响。

2.2 实验结果与详细分析

在选取的 10 个类别上进行训练和检验，得到各模型在测试集上表现对比如下实验表格：

模型方法	参数设置	训练集准确率	测试集准确率
Logistic Regression	lr=0.1, reg=0.01	100.00%	100.00%
SVM	Kernel='linear'	100.00%	98.25%
SVM	Kernel='rbf'	100.00%	98.25%
SVM	Kernel='poly'	100.00%	96.49%
Naive Bayes	Gaussian	99.18%	98.25%
KNN	K=5	98.56%	96.49%

几种分类模型都表现出 98% 以上的高准确率，尤其是逻辑斯谛回归的准确率达到到了 100%，也即类别间的区分度在特征空间中较为明显。SVM 的线性核和 RBF 核均有一个分类错误，RBF 在更高维的空间中分类并未提升性能；而多项式核准确率稍低，分析认为在高维空间中再引入高阶函数会导致一定的过拟合现象。

朴素贝叶斯模型的测试准确率也达到了 98.25%，可以认为特征之间基本是独立的，符合朴素贝叶斯假设；与 SVM 的 poly 核相同的是，KNN 的分类准确率也较低。分析认为系高维空间的维度灾难，使得距离度量部分失效导致性能略低于线性模型。

综上所述可以认为在特征提取质量较高，且分类项数较低的前提下，逻辑回归和线性 SVM 等线性分类器往往能以最小的计算开销获得较为优秀的性能。此时相比计算开销的省略，较为复杂的非线性模型并不能带来更好

的表现，反而有可能出现过拟合现象。

3 任务二：深度神经网络

对图片的细粒度分类依赖于捕捉细微的纹理差异，使用 CNN 类方法较为有效。CUB-200 数据集每类约 60 张图片，此时在全集尺度上数据量十分稀缺，而且图片大小仅有 224x224，故需要对数据进行处理和增强来使训练模型更快收敛且不易过拟合。

3.1 基础 CNN 模型

最初构建了一个类 VGG 的 5 层卷积网络，为 **Task2_CNN.py** 文件附于作业压缩包中。网络包含 5 个卷积块，每个块经 **3x3** 卷积后通过 **MaxPool** 下采样；通道数从 64 逐层增加至 512 后接入全连接层进行分类。使用 **ReLU** 激活函数和 **BatchNorm** 进行归一化。

实验结果十分不理想，在训练集正确率达到 50% 左右时，验证集准确率仅为 19.98%；分析认为网络层数较浅使感受野有限；同时普通的堆叠卷积结构在从头训练时难以优化，且容易发生过拟合。

3.2 残差连接

为了加深网络和解决退化问题，复习交流并查找资料之后选用 ResNet-18 结构进行网络实现，引入利用跳跃连接 $F(x) + x$ 的残差块优化反向传播。同时，针对 CUB200 的小样本场景，如下代码实现 Mixup 策略，随机打乱索引并混合图像来增加数据多样性，防止过拟合。

```
def mixup_data(x, y, alpha=0.4):  
    lam = np.random.beta(alpha, alpha)  
    index = torch.randperm(batch_size) # 依混合系数随机打乱索引  
    mixed_x = lam * x + (1 - lam) * x[index, :] # 图像混合  
    return mixed_x, y, y[index], lam
```

运行该版代码进行实验得到结果；准确率有了明显的提高，上升到 70.03%，说明残差结构在特征提取上具有更加强大的能力。最终方案在 Resnet 基础上采用更深的 **ResNet-34**，采用 **[3, 4, 6, 3]** 的 Block 堆叠方式，相比 ResNet-18 拥有更强的非线性表达能力，理论上能更好的捕捉如羽毛纹理、鸟喙形状等更深层的语义特征。同时，在该版方案中如下实现 Kaiming Normal 方法来初始化卷积层权重，使得训练方差相对稳定，从而加快收敛速度。

```
nn.init.kaiming_normal_(m.weight, mode='fan_out', nonlinearity='relu')
```

3.3 数据增强

在三次训练中，模型将独热编码软化，防止模型对训练样本的 confidence 过高导致学习能力降低，增强泛化能力；同时使用了较多的数据增强方法对数据进行多样化处理：

- **RandomRotation(15)** & **RandomAffine**: 随机对图片进行裁切、旋转和镜像以扩展单类鸟类的不同姿态；
- 使用 **ColorJitter** 来随机调整亮度、对比度和饱和度；

- 随机遮挡图像区域，减小整体轮廓和部分局部特征的影响，是模型有机会关注并学习更加多样化的细节特征。

```
def get_cosine_schedule_with_warmup(optimizer, num_warmup_epochs,
num_training_epochs):
    def lr_lambda(current_epoch):
        if current_epoch < num_warmup_epochs:
            return float(current_epoch + 1) / float(max(1,
num_warmup_epochs))
```

同时，在结构的实现过程中，利用 AdamW 优化器将权重衰减解耦，相比于 Adam 提供更好的正则化效果；并采用 Warmup 策略动态地调整学习率，在训练初期前 10 个 Epoch 线性增大学习率，随后按余弦曲线衰减学习率，在训练后期进行更精细的微调，防止训练过程中的震荡。

3.4 实验结果

对三个模型进行实验，得到实验结果如下：

模型结构	优化和增强	Loss	验证集最佳准确率	
Simple CNN	基础增强	3.4147	19.98%	-
ResNet-18	残差连接、Mixup、AdamW、Label	1.3540	70.03%	+50.05%
ResNet-34	Warmup	75.92%	1.0210	+5.89%

在从头训练细粒度分类任务时，增加网络深度是有效的，但是在 18-34 的过程中增加不甚明显，此时一味地增加深度是不那么必要的。要达到更好的表现，需要应用一些机器学习 Trick 来增强数据和学习策略，使学习过程更加高效。最终，基于 ResNet-34 的模型在测试集上实现了约 76% 的准确率，可认为在该分类任务是有效的。

4 实验总结

本次大作业围绕 CUB-200 细粒度图像分类任务，成功探究了基于预提取特征的传统机器学习方法和基于原始图像的深度学习方法。在拥有高质量特征向量的前提下，简单的逻辑回归即可实现较高的准确率，而非线性核 SVM 优势不大；即当特征空间具备良好的线性可分性时，模型复杂度反而可能引发过拟合。

同时，单纯堆叠卷积层会导致严重的退化问题，而残差连接有效解决了梯度消失问题，搭配正则化、优化调度和数据增强可以进一步推高模型表现。在实际工程应用中，针对小样本等数据特点设计数据增强与训练策略也是十分必要的。

A 附录

实验生结果附于 ExperimentResults.md 文件中；实验所需环境见 requirements.txt 文件。