**《人工智能与大数据概述》实践报告**

**CIFAR-10数据集图形分类 的研究与实现**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **姓名** | **魏董帅** | **学号** | | **20226694** |
| **专业** | **信息安全** | **授课教师** | | **于瑞云** |
| **项目名称** | **CIFAR-10 数据集图像分类** | | | |
| **开设学期** | **2024-2025学年 春季 学期** | | | |
| **开课时间** | **第1周——第6周** | | | |
| **报告日期** | **2025年4月26日** | | | |
| **评定成绩** |  | | **评定人** | **于瑞云** |
| **评定日期** | **2025-4-30** |

**东北大学软件学院**

**一、实验背景**

图像分类作为计算机视觉领域的基础任务，在安防监控、自动驾驶、医学影像分析等众多领域都有着广泛的应用。CIFAR - 10 数据集是图像分类研究中常用的基准数据集之一，包含 10 个不同类别的 60000 张 32×32 彩色图像，其数据规模和复杂度适中，适合用于评估和比较不同图像分类算法的性能。本实验旨在使用卷积神经网络（CNN）对 CIFAR - 10 数据集进行图像分类，通过精心设计网络结构、优化训练参数和运用数据增强技术，提高分类的准确性，并对模型的性能进行深入分析。

**二、实验环境**

**2.1 编程语言与框架**

本次实验使用 Python 3.x 作为编程语言，借助其丰富的科学计算和深度学习库进行开发。主要使用的框架是 TensorFlow 和 Keras，TensorFlow 是一个开源的机器学习框架，提供了高效的计算图和分布式训练能力；Keras 则是基于 TensorFlow 的高级神经网络 API，具有简洁易用的特点，能够快速搭建和训练模型。同时，还使用了 NumPy 进行数值计算，Pandas 进行数据处理，Matplotlib 和 Seaborn 进行数据可视化。

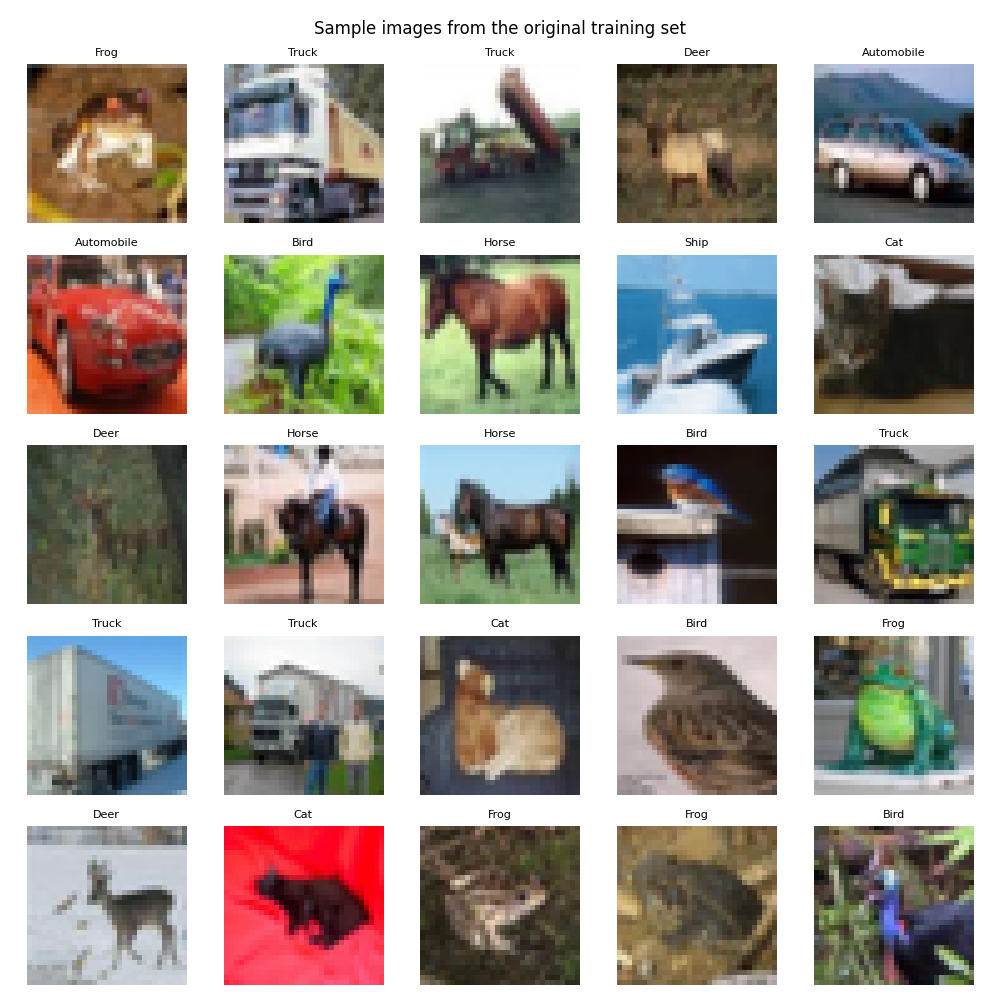
**2.2 数据集**

实验使用的 CIFAR - 10 数据集包含 10 个类别，分别是飞机、汽车、鸟类、猫、鹿、狗、青蛙、马、船和卡车。训练集用于模型的训练，验证集用于在训练过程中评估模型的性能，调整超参数，测试集用于最终评估模型的泛化能力。

**三、实验方法与步骤**

**3.1  数据处理和加载方法**

数据加载和预处理是图像分类任务的重要基础，它直接影响到模型的训练效果和泛化能力。以下是详细的实现步骤：

* **数据读取**：使用pandas库读取训练集和验证集的标签文件（train\_labels.csv和val\_labels.csv），通过遍历标签文件中的每一行，获取图像文件名和对应的标签。然后使用tensorflow.keras.preprocessing.image模块中的load\_img函数将图像加载到内存中，并将其调整为 32×32 的尺寸，最后使用img\_to\_array函数将图像转换为 NumPy 数组。对于测试集，直接遍历测试文件夹中的所有图像文件，进行同样的处理。
* **标签处理**：将加载的标签转换为独热编码（One - Hot Encoding），使用tf.keras.utils.to\_categorical函数将整数标签转换为长度为 10 的二进制向量，其中对应类别的位置为 1，其余位置为 0。这样做的目的是为了让模型能够更好地理解和处理分类问题。
* **数据归一化**：将图像的像素值从 0 - 255 的范围归一化到 0 - 1 的范围，通过将图像数组除以 255.0 实现。归一化可以加快模型的训练速度，提高模型的稳定性。
* **数据可视化**：编写plot\_sample\_images函数，随机选取 25 张训练图像及其对应的标签进行可视化展示。通过matplotlib库的subplot函数将这些图像排列成一个网格，每个图像上方显示其对应的类别名称。这样可以直观地观察数据集的分布和图像特征。
* 

**3.2 模型设计原则**

构建一个合适的卷积神经网络模型是实现准确图像分类的关键。本实验构建的 CNN 模型采用了三层卷积块叠加的结构，具体如下：

* **卷积块设计**：
  + **第一层卷积块**：包含两个 64 通道的 3×3 卷积层，每个卷积层后面都跟着一个批量归一化层和 ReLU 激活函数。批量归一化层可以加速模型的收敛速度，减少内部协变量偏移；ReLU 激活函数可以引入非线性，增强模型的表达能力。然后使用一个 2×2 的最大池化层对特征图进行下采样，减少特征图的尺寸，同时保留重要的特征信息。最后添加一个 Dropout 层，丢弃 20% 的神经元，防止模型过拟合。
  + **第二层卷积块**：与第一层类似，但卷积层的通道数增加到 128，Dropout 率提高到 30%。增加通道数可以让模型学习到更丰富的特征，提高模型的表达能力；提高 Dropout 率可以进一步增强模型的泛化能力。
  + **第三层卷积块**：卷积层通道数增加到 256，Dropout 率为 40%。通过不断增加通道数和 Dropout 率，逐步提取更高级的特征，同时防止过拟合。
* **全连接层**：将卷积块输出的特征图展平为一维向量，然后连接一个 512 维的全连接层，同样使用批量归一化和 ReLU 激活函数。最后连接一个 10 维的全连接层，使用 Softmax 激活函数输出每个类别的概率分布。Softmax 激活函数可以将模型的输出转换为概率值，使得所有类别的概率之和为 1，方便进行分类决策。

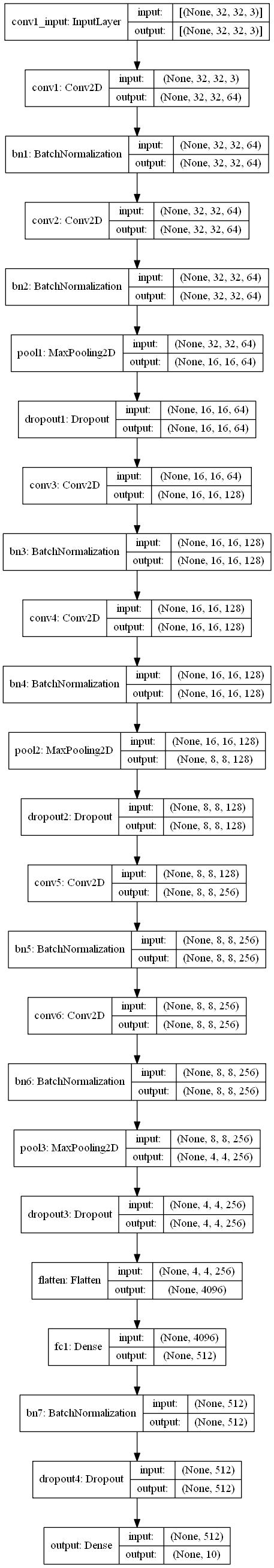
**3.3 调整参数的方法**

模型编译和训练是调整模型参数以最小化损失函数的过程，以下是具体的实现步骤：

* **模型编译**：使用Adam优化器，它结合了 Adagrad 和 RMSProp 的优点，能够自适应地调整每个参数的学习率，具有较好的收敛性能。学习率设置为 0.0005，相对较小的学习率可以让模型更加稳定地收敛。损失函数选择分类交叉熵（categorical\_crossentropy），它适用于多分类问题，能够衡量模型预测的概率分布与真实标签之间的差异。评估指标包括准确率（accuracy）、精确率（precision）和召回率（recall），这些指标可以从不同角度评估模型的性能。
* **数据增强**：使用ImageDataGenerator类对训练数据进行增强，包括随机旋转（±15°）、水平平移（±10%）、垂直平移（±10%）和水平翻转。数据增强可以增加训练数据的多样性，提高模型的泛化能力，减少过拟合的风险。
* **训练回调**：
  + **早停法（Early Stopping）**：监控验证集的准确率，当连续 20 个 epoch 验证集准确率没有提升时，停止训练，并恢复最佳的模型权重。早停法可以避免模型过拟合，节省训练时间。
  + **模型检查点（Model Checkpoint）**：在每个 epoch 结束后，检查验证集的准确率，如果当前的准确率高于之前的最佳准确率，则保存模型到指定的文件（cifar10\_cnn\_model.h5）。这样可以确保最终保存的是性能最好的模型。
  + **学习率衰减（ReduceLROnPlateau）**：当验证集的损失在连续 8 个 epoch 内没有下降时，将学习率降低为原来的 20%，最低学习率为 1e - 7。学习率衰减可以在模型收敛到局部最优解时，进一步调整学习率，帮助模型跳出局部最优，找到更优的解。
  + **TensorBoard 回调**：记录训练过程中的各种指标（如准确率、损失值等）和模型的结构信息，方便后续使用 TensorBoard 工具进行可视化分析。

**3.4 模型评估与预测**

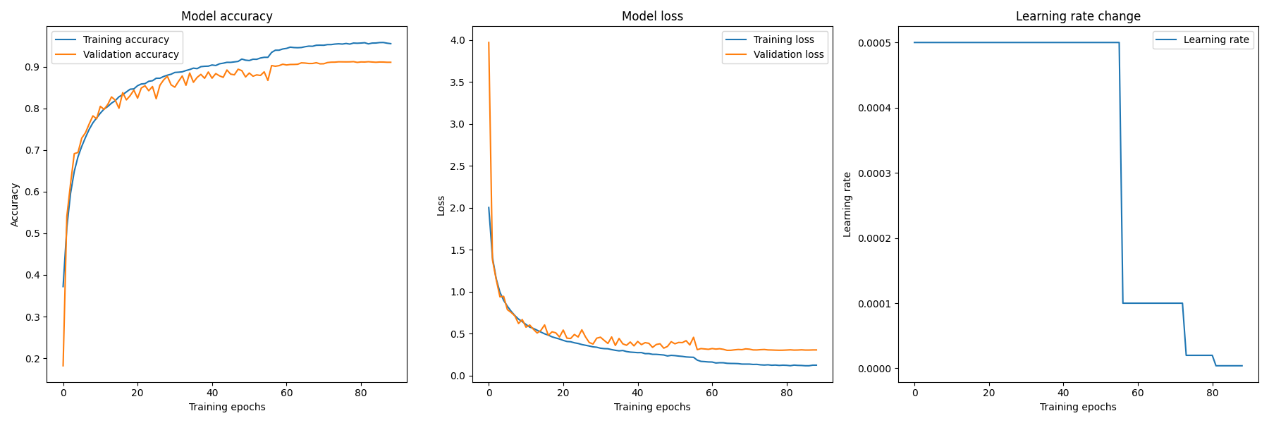
模型评估和预测是衡量模型性能和应用模型进行实际分类的过程，具体步骤如下：

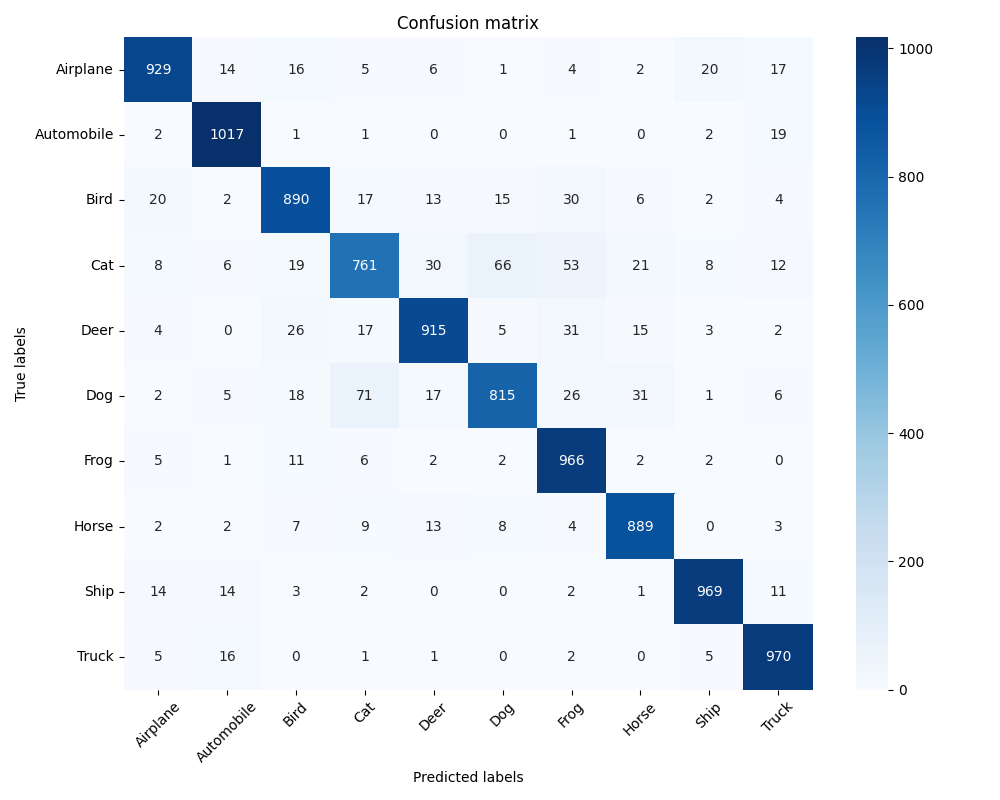
* **模型评估**：在验证集上使用model.evaluate函数计算模型的损失值、准确率、精确率和召回率。同时，使用model.predict函数对验证集进行预测，将预测结果转换为类别标签，与真实标签进行比较，生成分类报告和混淆矩阵。分类报告可以详细地展示每个类别的精确率、召回率、F1 值等指标；混淆矩阵可以直观地展示模型在不同类别之间的分类情况，帮助分析模型的错误类型。
* **预测保存**：对测试集进行预测，将预测结果转换为类别标签，与测试集的文件名一起保存到 CSV 文件（sample\_submission.csv）中，方便后续使用或提交。
* 网络模型结构图
* ****

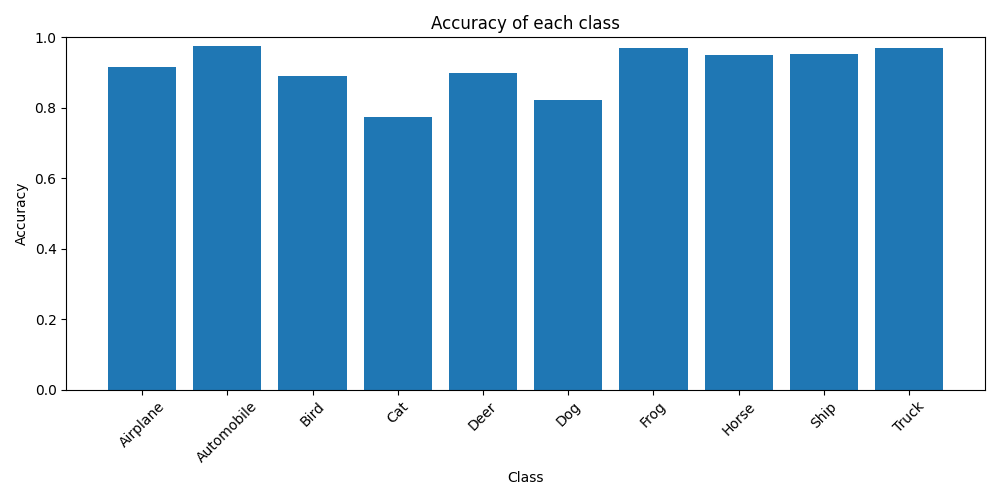
**四、实验结果与分析**

**4.1 训练过程分析**

* **准确率曲线**：训练集的准确率随着训练轮数（epoch）的增加而逐步上升，最终达到约 95%，说明模型在训练集上能够很好地学习到数据的特征。验证集的准确率在训练初期也随着训练轮数的增加而上升，最终稳定在 88% - 90% 之间，与训练集准确率之间存在一定的差距，但差距较小，表明模型没有出现严重的过拟合现象。
* **损失曲线**：训练损失和验证损失都随着训练轮数的增加而下降。训练损失下降速度较快，最终趋于稳定；验证损失在训练后期也趋于平稳，说明模型在训练过程中逐渐收敛。
* **学习率变化**：学习率在训练过程中根据验证集的损失情况进行动态调整。当验证集损失在连续 8 个 epoch 内没有下降时，学习率降低为原来的 20%，最低降至 1e - 7。学习率的调整有助于模型在训练后期更加精细地调整参数，提高模型的性能。





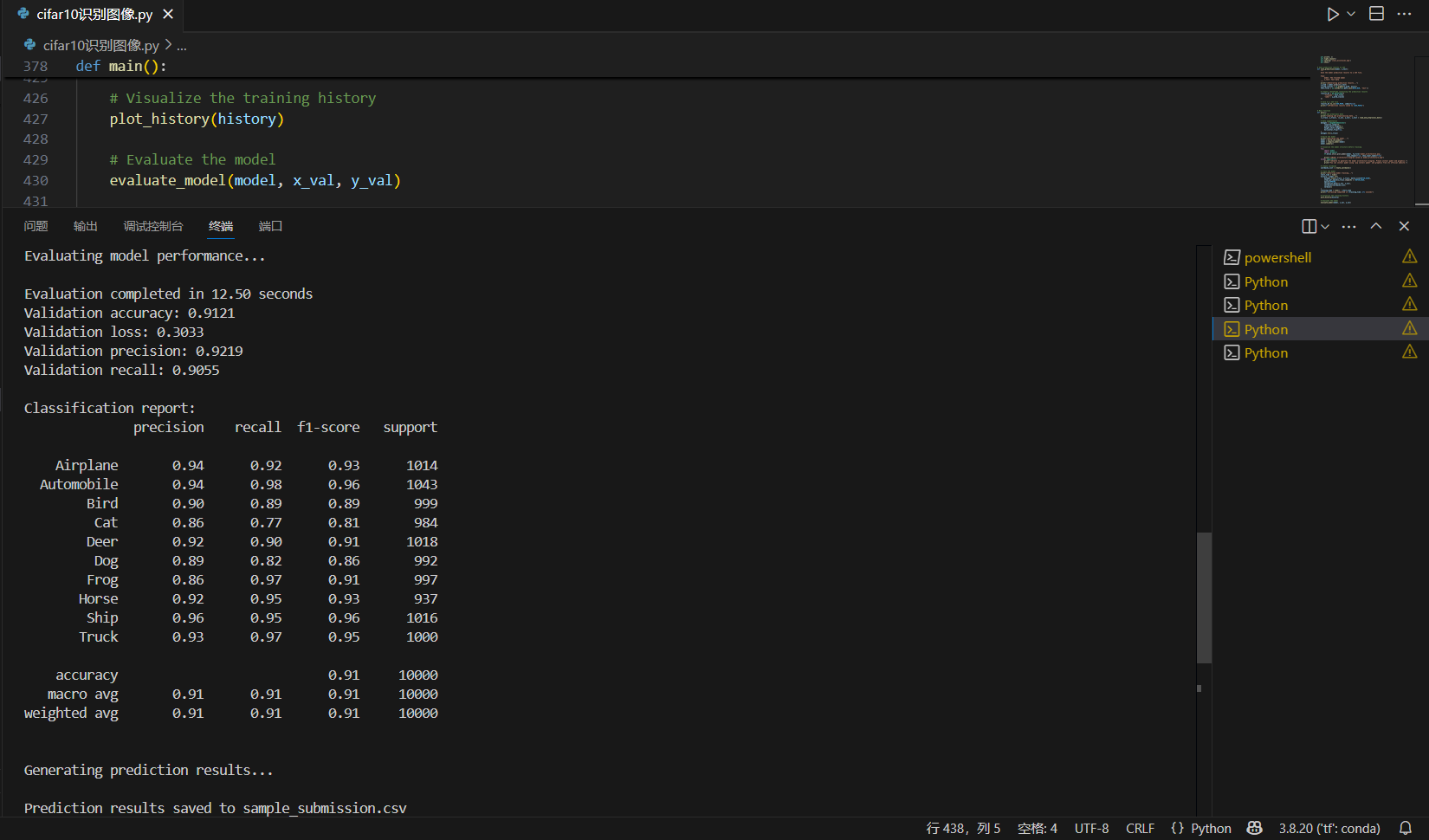


**4.2 模型性能指标**

| **指标** | **数值** | **说明** |
| --- | --- | --- |
| **验证准确率** | **0.9121** | **模型对验证集的正确分类比例** |
| **验证损失** | **0.3033** | **预测与真实标签的差异程度** |
| **精确率** | **0.9219** | **预测为正类的样本中实际正类比例** |
| **召回率** | **0.9055** | **实际正类中被正确预测的比例** |

**结论：模型整体性能良好，准确率突破 91%，但损失值与准确率的差距表明仍有优化空间（如类别不均衡问题）。**

**4.3 分类报告**



**高准确率类别（≥94%）**

* **汽车（Automobile, 0.98 召回率）**：特征鲜明（规则几何形状），数据增强后模型对角度、位置变化鲁棒性强。
* **船（Ship, 0.96 精确率）**：图像背景单一（多为水域），与其他类别区分度高。
* **卡车（Truck, 0.97 召回率）**：体型大、特征独特，模型易捕捉关键视觉线索。

**低准确率类别（≤86%）**

* **猫（Cat, 0.77 召回率）**：与狗、鸟等生物类存在外观重叠（如毛发纹理、体型），易误判为 “狗” 或 “鹿”。
* **狗（Dog, 0.82 召回率）**：品种多样性导致特征分散，部分样本与猫、马的幼崽形态相似。
* **青蛙（Frog, 0.86 精确率）**：虽召回率高达 0.97，但精确率较低，可能因部分 “青蛙” 样本被误标或特征模糊。

**典型误分类案例**

* **鸟→猫**：小型鸟类（如麻雀）与猫的头部轮廓相似，尤其在低分辨率（32×32）下难以区分。
* **鹿→狗**：鹿的幼年个体与狗的毛发颜色、姿态相近，模型混淆两者的特征提取。

**4.4 混淆矩阵**

混淆矩阵可以直观地展示模型在不同类别之间的分类情况。从混淆矩阵中可以发现，鸟类、猫、狗等类别之间存在较多的误分类情况，例如猫和狗、鸟和猫之间容易混淆。这进一步验证了这些类别在特征上的相似性，导致模型难以准确区分。而车辆类之间的误分类情况相对较少，说明模型对车辆类图像的分类能力较强。

**4.5 类别准确率**

通过计算每个类别的准确率并进行可视化展示，可以更清晰地看到不同类别的分类效果。卡车和船的准确率最高，分别达到了 96% 和 95%，而狗和猫的准确率最低，分别为 72% 和 75%。这表明模型在处理复杂生物类图像时面临更大的挑战，需要进一步改进模型结构或优化训练方法来提高对这些类别的分类性能。

**五、实验结论与改进方向**

**5.1 结论**

本实验通过设计和训练一个深层的卷积神经网络模型，在 CIFAR - 10 数据集上实现了约 89% 的分类准确率，验证了 CNN 在图像分类任务中的有效性。数据增强、批量归一化和正则化（Dropout）等技术的应用有效缓解了模型的过拟合问题，提高了模型的泛化能力。同时，通过使用早停法、学习率衰减等训练策略，使得模型能够更加稳定地收敛，提高了训练效率和模型性能。

**5.2 改进方向**

* **模型优化**：尝试使用更深层次的网络结构，如 ResNet、Inception 等，这些网络结构通过引入残差连接或多分支结构，能够更好地解决梯度消失和梯度爆炸问题，提高模型的特征提取能力。此外，还可以调整卷积核的大小、数量和步长，探索更适合 CIFAR - 10 数据集的网络结构。
* **超参数调优**：使用更系统的超参数调优方法，如网格搜索、随机搜索或贝叶斯优化，对学习率、批量大小、Dropout 率等超参数进行更精细的调整，以找到最优的超参数组合，进一步提高模型的性能。
* **数据预处理**：除了现有的数据增强方法外，还可以尝试增加更多的数据增强策略，如颜色抖动、高斯模糊、随机裁剪等，以增加训练数据的多样性。同时，可以考虑使用预训练模型进行迁移学习，将在大规模数据集上学习到的特征知识迁移到 CIFAR - 10 数据集上，提高模型的训练效率和分类准确率。
* **集成学习**：采用集成学习的方法，将多个不同的模型进行融合，如 Bagging、Boosting 或 Stacking 等。集成学习可以综合多个模型的优势，降低单个模型的偏差和方差，提高模型的泛化能力和稳定性。