

题目 2: 基于 PyTorch 的图像任务 (此题 100 分)

一. 题目概述

CIFAR10 是一个常用的图像分类数据集, 是 $3 \times 32 \times 32$ 彩色图像, 其中 3 为颜色通道数, 32×32 为图片尺寸。我们选取了 5000 张训练图像, 1000 张测试图像, 这 6000 张图像为拍照得到的图像, 定义为“真实”图像。其中, “真实”图像的训练集存储路径 `cifar_uvit/train/cifar`, “真实”图像的测试集路径为 `cifar_uvit/test/cifar`。

扩散模型是一个非常强大的图像生成模型, 可以用于生成“虚假”图像。我们使用了一个 CIFAR10 上的扩散模型, 生成了 6000 张“虚假”图像, 其中 5000 张“虚假”图像的训练集存储路径 `cifar_uvit/train/uvit`, 1000 张“虚假”图像的测试集存储路径为 `cifar_uvit/test/uvit`。

二. 提交要求

请基于数据集完成以下 5 道小题, 按照要求把答案复制到答题卡的指定位置, 并提交整个作答过程中的 Notebook 文档 (会影响评分)。一共需要提交以下两个文件, 请按照命名规则命名。

1. **jupyter notebook 运行文件 (包括运行结果的运行文件), 命名规则为 NOAI2_学生编号.ipynb**
 - (1) 不提交 jupyter notebook 文件的, 此题计 0 分。
 - (2) 在作答过程中如使用了多个 jupyter notebook, 请提供清晰的命名, 例如用来做哪道题的。
2. **NOAI2 答题卡的.docx 文档, 命名规则为 NOAI2_学生编号.docx**

三. 参考用库

1. 可以将下表用库直接复制到 notebook 中

```
import os
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms
from torch.utils.data import DataLoader, Dataset
from PIL import Image
import numpy as np
import random
```

2. 在整个答题过程中, 为了避免卡在代码细节, 你可以参考组委会提供的 pandas、numpy、matplotlib、PyTorch 参考代码, 可以根据自己需要进行复制和修改。

四. 题目主体

一共有 5 道小题，每道 20 分。

1. 代码阅读与任务解读（20 分）

以下是数据读取部分的代码，请阅读代码并执行后，回答以下问题：

```
class CustomDataset(Dataset):
    def __init__(self, root_dir, transform=None):
        self.root_dir = root_dir
        self.transform = transform
        self.image_paths = []
        self.labels = []
        for label, sub_dir in enumerate(os.listdir(root_dir)):
            full_dir = os.path.join(root_dir, sub_dir)
            for img_name in os.listdir(full_dir):
                img_path = os.path.join(full_dir, img_name)
                self.image_paths.append(img_path)
                self.labels.append(label)
    def __len__(self):
        return len(self.image_paths)
    def __getitem__(self, idx):
        img_path = self.image_paths[idx]
        image = Image.open(img_path).convert('RGB')
        label = self.labels[idx]
        if self.transform:
            image = self.transform(image)
        return image, label

# 数据预处理
transform = transforms.Compose([
    transforms.Resize((32, 32)),
    transforms.ToTensor(),
    transforms.Normalize((0.5, ), (0.5, ))
])

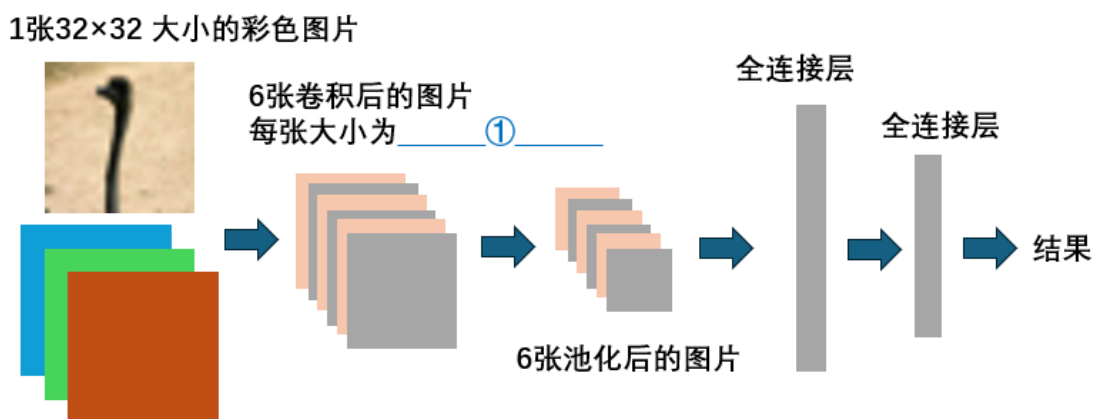
# 创建数据加载器
train_dir = 'cifar_uvit/train'
test_dir = 'cifar_uvit/test'
train_dataset = CustomDataset(train_dir, transform=transform)
test_dataset = CustomDataset(test_dir, transform=transform)
train_loader = DataLoader(train_dataset, batch_size=64, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=64, shuffle=True)
```

- (1) 请问 train_loader 和 test_loader 里面的数据样本数分别为多少？
- (2) 读取后的数据有几类标签 (Label)，取值分别是什么，分别代表什么数据？
- (3) 如果将 train_loader 里面的数据作为训练集 (Training Set)，test_loader 里面的数据作为测试集 (Testing Set)，设计神经网络模型进行训练和测试，可以实现一个什么任务？（用不超过 4 行的文字作答，不要只填写“图像分类”，应该结合题目概述中的内容，简述实现了一个什么样的分类）

请将以上问题的答案填到答题卡对应位置。

2. 请将小爱同学的代码补全（20 分）

为了实现上述分类过程，小爱同学设计了一个神经网络 FixNet，他（她）绘制了整个网络的结构，并提供了部分的代码，具体如下：



网络结构示意图

小爱同学模型定义的部分代码

```
class FixNet(nn.Module):
    def __init__(self):
        super(FixNet, self).__init__()
        self.conv1 = nn.Conv2d(3, ②, kernel_size=3)
        self.pool = nn.MaxPool2d(2, 2)
        self.fc1 = nn.Linear(③, 84)
        self.fc2 = nn.Linear(84, __根据上一问填写__) #如果是二分类，填写 1；如果是多分类，填写分类数，后面激活函数相应选择 nn.Softmax
        self.activation = __根据上一问填写__ #这是一个激活函数，如果是二分类，使用 nn.Sigmoid()；如果是多分类，选择 nn.Softmax(dim=1) #dim=1 一般表示输入的第二个维度是用来存储 label 的维度，一般多分类问题都这么使用

    def forward(self, x):
        x = self.pool(torch.relu(self.conv1(x)))
        x = x.view(-1, ④)
        x = torch.relu(self.fc1(x))
        x = self.fc2(x)
        x = self.activation(x)
        return x
```

为了让模型能够顺利运行，请补全图中的①②③④，并将图中的①和代码中的②③④填写到答题卡对应的位置四个位置的具体要求如下：

- ① 填写最终的计算结果（为一个整数）；
- ② 填写最终的计算结果（为一个整数）；
- ③ 填写最终的计算结果（为一个整数），并简单填写计算过程（例如 $a*b*c$ ）；
- ④ 填写最终的计算结果（为一个整数），并简单填写计算过程（例如 $a*b*c$ ）；

3. 请使用小爱同学的模型，实现训练和测试过程（20 分）

具体要求如下：

- ① **model**: 小爱同学的模型
- ② **optimizer**: 优化器选择 **Adam**, **learning_rate** 设定为 **0.001**
- ③ **criterion**: 损失函数选择 **BCELoss**
- ④ **epoch**: 5 轮

每轮训练过后，都将测试集（**Testing Set**）导入进行测试，并将每轮的训练集（**Training Set**）和测试集（**Testing Set**）的准确率打印出来，样例如下（具体准确率数值没有参考价值）。你在打印时，请将打印日志中“**Xiao Ai**”替换成你的名字的全拼，例如张三，请替换成“**Zhang San**”。

备注：请不要假装执行了过程，直接修改文本，如发现直接修改文本提交的，整个题目计 0 分！

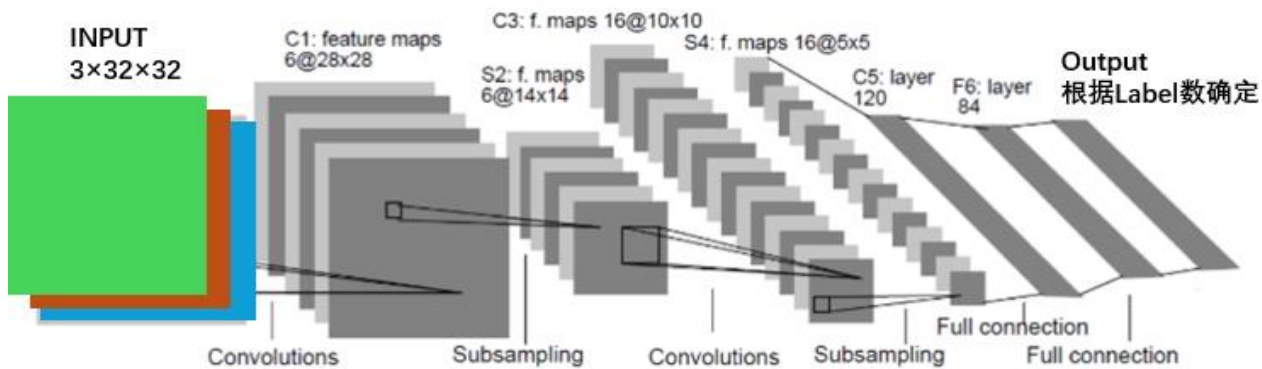
```
Epoch 1/5, Xiao Ai Train Accuracy: 0.5000,Xiao Ai Test Accuracy: 0.5001  
  
Epoch 2/5, Xiao Ai Train Accuracy: 0.5001,Xiao Ai Test Accuracy: 0.5002  
  
Epoch 3/5, Xiao Ai Train Accuracy: 0.5002,Xiao Ai Test Accuracy: 0.5003  
  
Epoch 4/5, Xiao Ai Train Accuracy: 0.5003,Xiao Ai Test Accuracy: 0.5004  
  
Epoch 5/5, Xiao Ai Train Accuracy: 0.5004,Xiao Ai Test Accuracy: 0.5005
```

请将你撰写的小爱同学的 **FixNet** 模型 **class**（能体现全部网络结构的代码）和打印出来的每轮 **epoch** 准确率的信息填写到答题卡相应位置。

```
class FixNet(nn.Module):  
    def __init__(self):
```

注：本题的损失（**Loss**）和提升准确率（**Accuracy**）的数值只作为是否按照题目结构要求搭建网络的评分依据，其高低不作为评分依据，不需要纠结于降低损失（**Loss**）和提升准确率（**Accuracy**）。

4. 请搭建如下图所示的 LeNet-5 模型，实现训练和测试过程（20 分）



LeNet-5 结构图

具体要求如下：

① model: LeNet-5，简单增加一些补充说明，以保证 LeNet-5 正常运行

本题如果直接调用 PyTorch 自带的 LeNet-5 模型，只能得到一半的分！

<i> 其中第一层输入层由于是彩色图片，所以第一个卷积层的输入可以相应调整为 3；

<ii> 最后输出层的激活函数，如果是二分类，使用 nn.Sigmoid()；如果是多分类，选择 nn.Softmax(dim=xx)

<iii> 中间层中的每一个卷积层之后需要在池化之前加入 relu 激活函数，调用命令 torch.relu()

除了输出层，中间层中的每一个全连接层后都需要加一个 relu 激活函数。

② optimizer: 优化器选择 Adam，learning_rate 设定为 0.001

③ criterion: 损失函数选择 BCELoss

④ epoch: 5 轮

每轮训练过后，都将测试集（Testing Set）导入进行测试，并将每轮的训练集（Training Set）和测试集（Testing Set）的准确率打印出来，样例如下（具体准确率数值没有参考价值）。你在打印时，请将打印日志中“Xiao Ai”替换成你的名字的全拼，例如张三，请替换成“Zhang San”。

备注：请不要假装执行了过程，直接修改文本，如发现直接修改文本提交的，整个题目计 0 分！

```
Epoch 1/5, Xiao Ai Train Accuracy: 0.5000,Xiao Ai Test Accuracy: 0.5001  
Epoch 2/5, Xiao Ai Train Accuracy: 0.5001,Xiao Ai Test Accuracy: 0.5002  
Epoch 3/5, Xiao Ai Train Accuracy: 0.5002,Xiao Ai Test Accuracy: 0.5003  
Epoch 4/5, Xiao Ai Train Accuracy: 0.5003,Xiao Ai Test Accuracy: 0.5004  
Epoch 5/5, Xiao Ai Train Accuracy: 0.5004,Xiao Ai Test Accuracy: 0.5005
```

请将你撰写的 LeNet 模型 class（能体现全部网络结构的代码）和打印出来的每轮 epoch 准确率的信息填写到答题卡相应位置。

```
class LeNet(nn.Module):  
    def __init__(self):  
        super(LeNet, self).__init__()
```

注：本题的损失（Loss）和提升准确率（Accuracy）的数值只作为是否按照题目结构要求搭建网络的评分依据，其高低不作为评分依据，不需要纠结于降低损失（Loss）和提升准确率（Accuracy）。

5. 请设计一个与 LeNet-5 结构不同的卷积神经网络，实现训练和测试过程（20 分）

具体要求如下：

① **model**: 取名为 MyNet()

包含

<i> 至少 2 个卷积层和 2 个 max pooling 层；

<ii> **最多**只能用 2 个全连接层（线性层）；

<iii> 如果卷积层和 LeNet-5 层数一样，则需要保证至少一层结构和 LeNet-5 不同；

② **optimizer**: 优化器选择 Adam，learning_rate 设定为 0.001

③ **criterion**: 损失函数选择 BCELoss

④ **epoch**: 5 轮

5 轮之内的训练集和测试集准确率至少有一次达到在 70%以上。

每轮训练过后，都将测试集（Testing Set）导入进行测试，并将每轮的训练集（Training Set）和测试集（Testing Set）的准确率打印出来，样例如下（具体准确率数值没有参考价值）。你在打印时，请将打印日志中“Xiao Ai”替换成你的名字的全拼，例如张三，请替换成“Zhang San”。

备注：请不要假装执行了过程，直接修改文本，如发现直接修改文本提交的，整个题目计 0 分！

```
Epoch 1/5, Xiao Ai Train Accuracy: 0.5000,Xiao Ai Test Accuracy: 0.5001  
  
Epoch 2/5, Xiao Ai Train Accuracy: 0.5001,Xiao Ai Test Accuracy: 0.5002  
  
Epoch 3/5, Xiao Ai Train Accuracy: 0.5002,Xiao Ai Test Accuracy: 0.5003  
  
Epoch 4/5, Xiao Ai Train Accuracy: 0.5003,Xiao Ai Test Accuracy: 0.5004  
  
Epoch 5/5, Xiao Ai Train Accuracy: 0.5004,Xiao Ai Test Accuracy: 0.5005
```

请你撰写的 MyNet 模型 class（能体现全部网络结构的代码）和打印出来的每轮 epoch 准确率的信息填写到答题卡相应位置。

```
class MyNet(nn.Module):  
    def __init__(self):  
        super(MyNet, self).__init__()
```

注：本题的损失（Loss）和提升准确率（Accuracy）的数值只作为是否按照题目结构要求搭建网络的评分依据，其高低不作为评分依据，不需要纠结于降低损失（Loss）和提升准确率（Accuracy）。