人工智能中的编程 Task 1

模型训练代码在Task_1/LeNet.py 中,LeNet_test.py 用于加载训练的权重并查看测试集上整体以及各个类别的acc,代码与hw1中相同，模型参考[https://docs.pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html]进行细微调整以进行和task3的对比

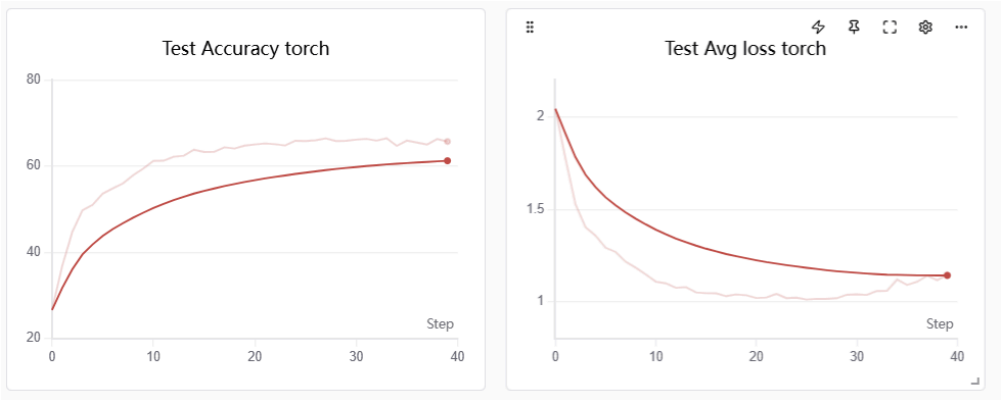
batch_size = 50
epochs= 40
lr=0.001

```
class LeNet(torch.nn.Module):
        def __init__(self):
                super().__init__()
                self.conv1 = nn.Conv2d(3, 6, stride=1,padding=1, kernel_size=3)
                self.pool = nn.MaxPool2d(2, 2)
                self.conv2 = nn.Conv2d(6, 16, stride=1, padding=1, kernel_size=3)
                self.fc1 = nn.Linear(1024, 120)
                self.fc2 = nn.Linear(120, 84)
                self.fc3 = nn.Linear(84, 10)
                self.relu = nn.ReLU()

        def forward(self ,x):
            x = self.pool(self.relu(self.conv1(x)))
            x = self.pool(self.relu(self.conv2(x)))
            x = torch.flatten(x, 1)
            x = self.relu(self.fc1(x))
            x = self.relu(self.fc2(x))
            x = self.fc3(x)
            return x

model = LeNet().to(device)

criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(model.parameters(), lr=lr, momentum=0.9)
scheduler = torch.optim.lr_scheduler.StepLR(optimizer, step_size=5, gamma=0.9)
```



最终训练集上acc 为 65.7%