

人工智能中的编程 Task 3

训练过程中训练曲线，参数，时间的记录是基于swanlab库的 [<https://swanlab.cn/>] ,若运行环境没有啊、swanlab库，可以注释掉代码中记录的部分

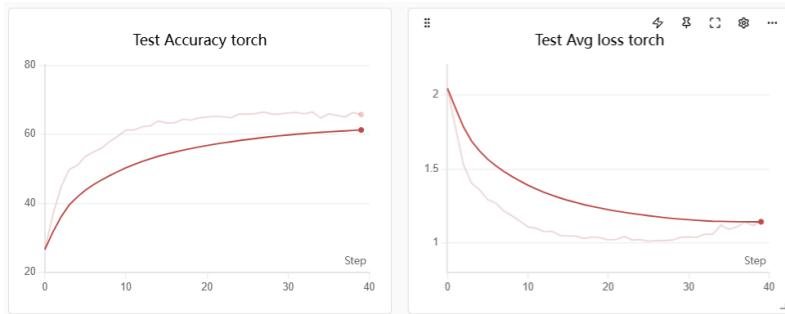
自主实现的基于cuda和pybind的模型网络基类在source文件夹下，Task_3/utils 中定义了用于训练的优化器等

实验一：对比Task1的数据：

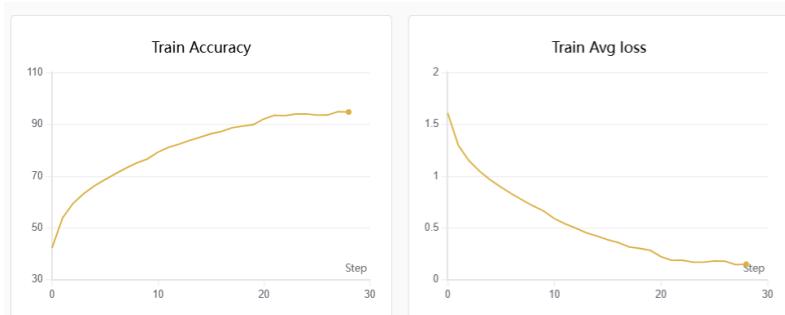
在Task_3/train_1.py 中基于自主实现的mytensor库，定义了和Task1中相同的网络结构，

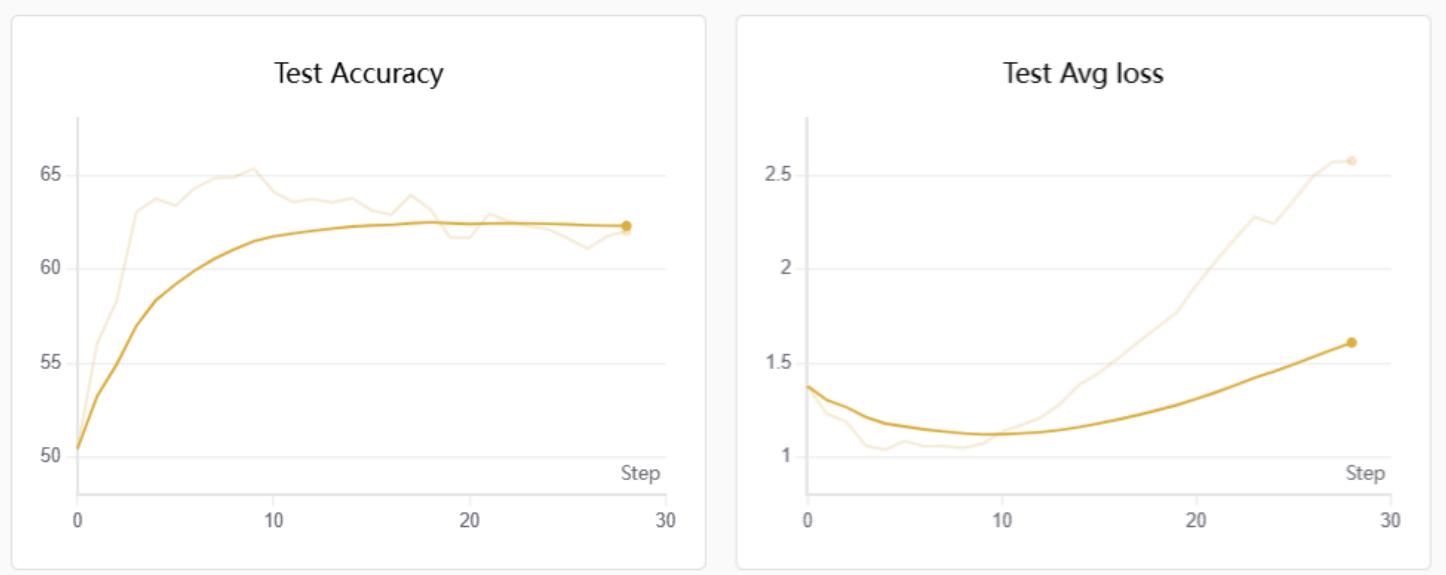
```
model = Model(  
    mt.Conv(3, 6),  
    mt.Relu(), mt.Pool(),  
    mt.Conv(6, 16),  
    mt.Relu(), mt.Pool(),  
    mt.Linear(1024, 120, mt.Device.GPU), mt.Relu(),  
    mt.Linear(120, 84, mt.Device.GPU), mt.Relu(),  
    mt.Linear(84, 10, mt.Device.GPU))
```

并训练了CFIR10数据集，结果如下：



这是 Task1的结果，最终test_acc 为65.7%





这是 Task3中和Task完全相同的模型的结果，最终30个epoch 的train_acc 94.7% , test_acc 62.03%

两次训练中Task1 epoch数为40，采用了SGD的动量模式，momentum=0.9，而Task3中epoch数设置为30,SGD未采用动量模式

其余参数lr = 0.001, batch_size = 50,scheduler中 step_size=5, gamma = 0.9相同

可以看到，在训练集上最终的acc是相近的，这说明自主实现的模型和Task1的模型在acc上的表现相近，而Task3中模型出现了一定的训练不稳定的情况（过拟合+test_acc 中后期小幅下降）可能是因为自主实现的模型SGD优化没有实现动量模式

根据记录的运行时间，Task1中训练时间为15min32s，Task3中训练时间为30min31s,这说明自主实现的模型的训练速度较pytorch原生慢，耗时为两倍多一点

这说明自主实现的cuda底层还可优化

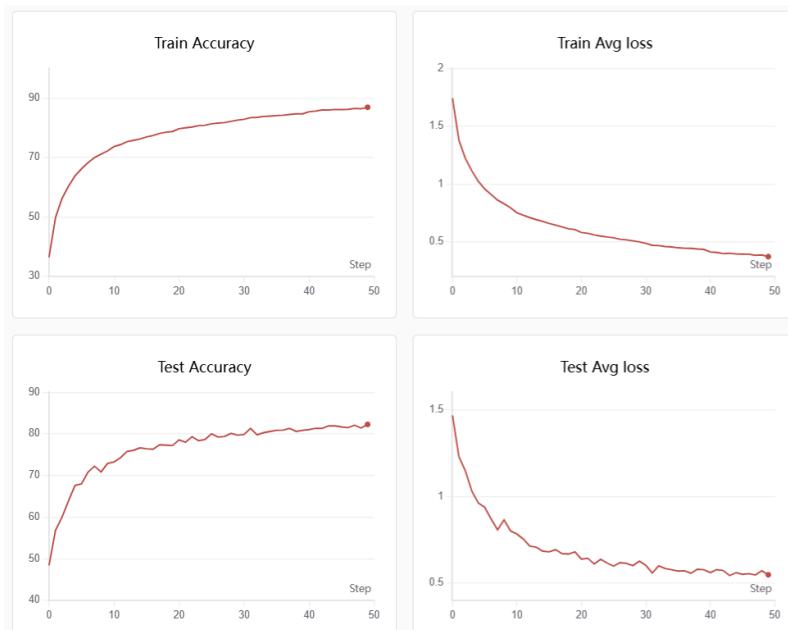
实验二：为了避免实验一中的过拟合现象，同时为了验证自主实现的框架在更复杂模型上的表现，在 Task_3/train_2.py 中定义了更复杂模型，

```

model = Model(
    mt.Conv(3, 32),
    mt.Relu(), mt.Pool(),
    mt.Conv(32, 64),
    mt.Relu(), mt.Pool(),
    mt.Conv(64, 100),
    mt.Relu(), mt.Pool(),
    mt.Conv(100, 256),
    mt.Relu(), mt.Pool(),
    mt.Linear(1024, 256, mt.Device.GPU), mt.Relu(),
    mt.Linear(256, 128, mt.Device.GPU), mt.Relu(),
    mt.Linear(128, 10, mt.Device.GPU))

```

并训练了CFIR10数据集，结果如下：



最终50个epoch后 train_acc 86.9%, test_acc 82.3%

实验二中的模型结构和一并提交的报告CV_HW_5.pdf(在计算机视觉某次课程作业中的实验结果) 中自定义的cnn结构基本一致，根据报告中的内容，其在50个epoch后
train_acc 98.12%, test_acc 76.3%
这再次说明自主实现的模型在复杂模型上的表现和pytorch原生的模型表现相近

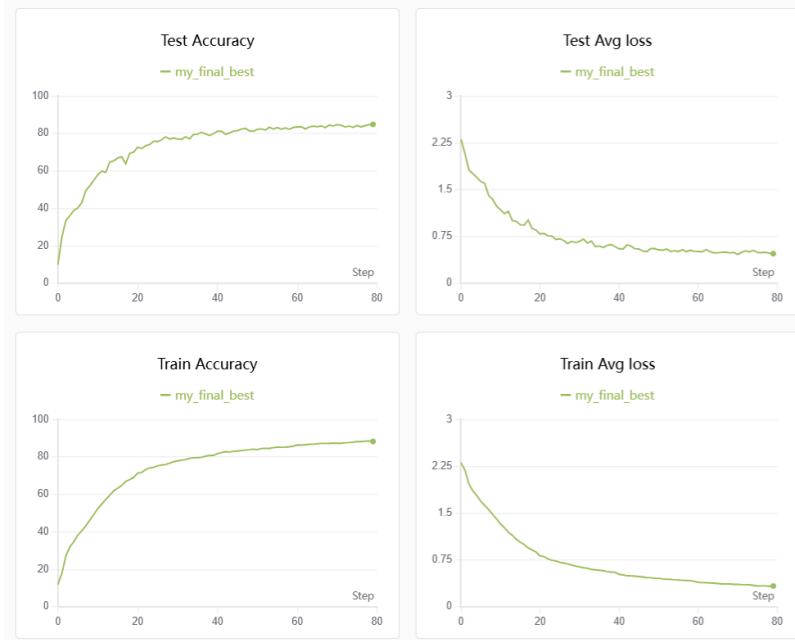
附：Task3/Task_3_2_log.txt为实验二的运行日志

实验三为自主实现的框架在较复杂模型（较实验二进一步提升参数）上最高的acc训练结果

```

model = Model(
    mt.Conv(3,32),mt.Relu(),mt.Conv(32,32),mt.Relu(),mt.Conv(32,32),mt.Relu(),mt.Pool(),
    mt.Conv(32,64),mt.Relu(),mt.Conv(64,64),mt.Relu(),mt.Conv(64,64),mt.Relu(),mt.Pool(),
    mt.Conv(64,100),mt.Relu(),mt.Conv(100,100),mt.Relu(),mt.Conv(100,100),mt.Relu(),mt.Pool(),
    mt.Conv(100,256),mt.Relu(),mt.Pool(),
    mt.Linear(1024,256,mt.Device.GPU),mt.Relu(),
    mt.Linear(256,128,mt.Device.GPU),mt.Relu(),
    mt.Linear(128,10,mt.Device.GPU))

```



80个epoch后train_acc 88.3% , test_acc 84.8%

附：Task3/Task_3_3_log.txt为实验三的运行日志