

# 数据科学基础

## 期中作业



院 系： 信息与通信工程学院

班 级： 2019211112

学 号： 2021523016

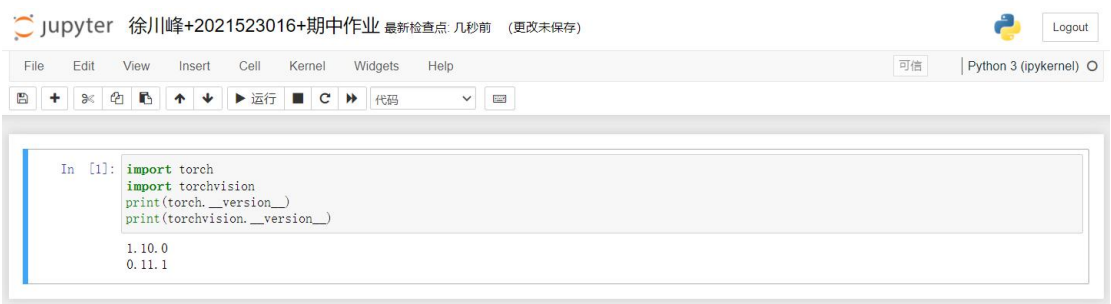
类 别： 交流生

姓 名： 徐川峰

指导老师： 刘芳

2021 年 12 月 12 日

任务一：安装配置 Pytorch 环境，检测 Pytorch 安装情况。（10 分）



The screenshot shows a Jupyter Notebook interface with the following code in a cell:

```
In [1]: import torch
import torchvision
print(torch.__version__)
print(torchvision.__version__)
```

The output of the code is:

```
1.10.0
0.11.1
```

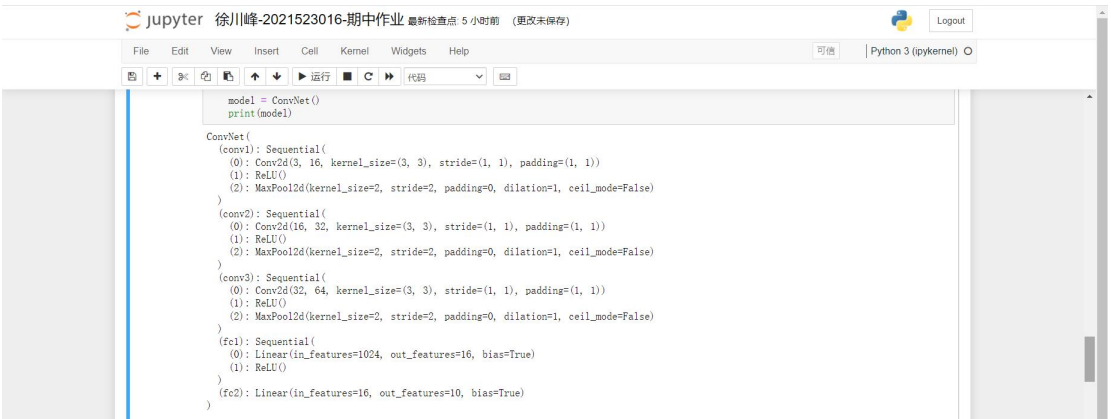
任务二：使用包含三层以上卷积层的神经网络对 CIFAR-10 数据集分类。对生成网络结构进行截图（如例 1 所示），并对训练过程的精度增长和 loss 收敛情况进行截图（如例 2 所示）。（15 分）

例1. 网络模型

```
ConvNet1: Sequential(
  (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (1): ReLU()
  (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
)
ConvNet2: Sequential(
  (0): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (1): ReLU()
  (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
)
ConvNet3: Sequential(
  (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

例2. 网络训练Loss收敛

Train Epoch: 1	[5800/50000 (12%)]	Loss: 2.260852
Train Epoch: 1	[11800/50000 (24%)]	Loss: 1.932874
Train Epoch: 1	[17800/50000 (36%)]	Loss: 1.870852
Train Epoch: 1	[23800/50000 (48%)]	Loss: 1.830403
Train Epoch: 1	[29800/50000 (60%)]	Loss: 1.617014
Train Epoch: 1	[35800/50000 (72%)]	Loss: 1.625217
Train Epoch: 1	[41800/50000 (84%)]	Loss: 1.748831
Train Epoch: 1	[47800/50000 (96%)]	Loss: 1.574987
Test set: Average loss: 1.5799, Accuracy: 4043/10000 (40%)		

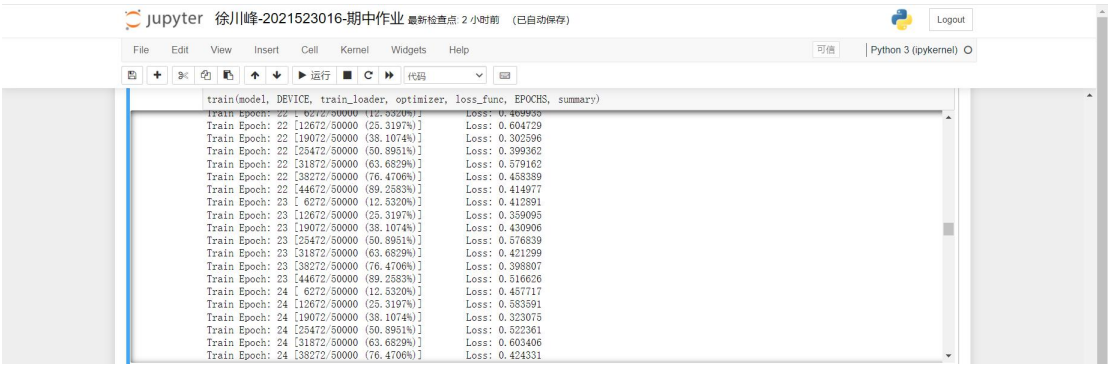


The screenshot shows a Jupyter Notebook interface with the following code in a cell:

```
model = ConvNet()
print(model)

ConvNet(
  (conv1): Sequential(
    (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (conv2): Sequential(
    (0): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (conv3): Sequential(
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fc1): Sequential(
    (0): Linear(in_features=1024, out_features=16, bias=True)
    (1): ReLU()
  )
  (fc2): Linear(in_features=16, out_features=10, bias=True)
)
```

网络模型



The screenshot shows a Jupyter Notebook interface with the following code in a cell:

```
train(model, DEVICE, train_loader, optimizer, loss_func, EPOCHS, summary)
```

The output of the code is a table showing the training process and loss convergence:

Train Epoch: 22	[ 6272/50000 (12.5320%)]	Loss: 0.409930
Train Epoch: 22	[12672/50000 (25.3197%)]	Loss: 0.604729
Train Epoch: 22	[19072/50000 (38.1074%)]	Loss: 0.302596
Train Epoch: 22	[25472/50000 (50.8951%)]	Loss: 0.999362
Train Epoch: 22	[31872/50000 (63.6829%)]	Loss: 0.579162
Train Epoch: 22	[38272/50000 (76.4706%)]	Loss: 0.458389
Train Epoch: 22	[44672/50000 (89.2583%)]	Loss: 0.414977
Train Epoch: 22	[ 6272/50000 (12.5320%)]	Loss: 0.412891
Train Epoch: 23	[12672/50000 (25.3197%)]	Loss: 0.359095
Train Epoch: 23	[19072/50000 (38.1074%)]	Loss: 0.430906
Train Epoch: 23	[25472/50000 (50.8951%)]	Loss: 0.576839
Train Epoch: 23	[31872/50000 (63.6829%)]	Loss: 0.421299
Train Epoch: 23	[38272/50000 (76.4706%)]	Loss: 0.398807
Train Epoch: 23	[44672/50000 (89.2583%)]	Loss: 0.516626
Train Epoch: 24	[ 6272/50000 (12.5320%)]	Loss: 0.457717
Train Epoch: 24	[12672/50000 (25.3197%)]	Loss: 0.583591
Train Epoch: 24	[19072/50000 (38.1074%)]	Loss: 0.323075
Train Epoch: 24	[25472/50000 (50.8951%)]	Loss: 0.522361
Train Epoch: 24	[31872/50000 (63.6829%)]	Loss: 0.603406
Train Epoch: 24	[38272/50000 (76.4706%)]	Loss: 0.424331

网络训练 Loss 收敛

任务三：对 CIFAR-10 数据进行解析和可视化展示。输出 CIFAR-10 数据集训练集、测试集大小；输出数据集包含的所有类别名称及与 label 对应情况；输出数据集中一张图片的数组 size，并将数据集测试集三张图片进行可视化展示。（15 分）

```
jupyter 徐川峰-2021523016-期中作业 最新检查点 17 小时前 (已自动保存)
File Edit View Insert Cell Kernel Widgets Help 可信 Python 3 (ipykernel) O
In [68]: DOWNLOAD_DATASET = False
transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
])

train_data = torchvision.datasets.CIFAR10(
    root='./CIFAR10',
    train=True,
    transform=transform,
    download=DOWNLOAD_DATASET
)
print('train set:', len(train_data))

test_data = torchvision.datasets.CIFAR10(
    root='./CIFAR10',
    train=False,
    transform=transform,
    download=DOWNLOAD_DATASET
)
print('test set:', len(test_data))

train set: 50000
test set: 10000
```

输出 CIFAR-10 数据集训练集、测试集大小

```
In [69]: print(train_data.classes)
print(train_data.class_to_idx)

['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck']
{'airplane': 0, 'automobile': 1, 'bird': 2, 'cat': 3, 'deer': 4, 'dog': 5, 'frog': 6, 'horse': 7, 'ship': 8, 'truck': 9}
```

输出数据集包含的所有类别名称及与 label 对应情况

```
In [70]: print(train_data.data[16].shape)

(32, 32, 3)
```

输出数据集中一张图片的数组 size

```
jupyter 徐川峰-2021523016-期中作业 最新检查点 17 小时前 (已自动保存)
File Edit View Insert Cell Kernel Widgets Help 可信 Python 3 (ipykernel) O
plt.savefig('test_3230.png')

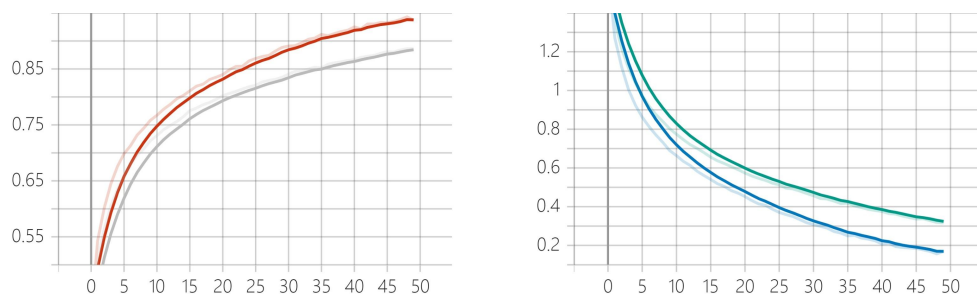
plt.imshow(test_data.data[16])
plt.show()
plt.savefig('test_16.png')
```



将数据集测试集三张图片进行可视化展示

任务四：修改网络结构（调整网络深度，使用不同的激活函数，调整神经元数量）或更改训练参数（学习率，`batch_size`），分析不同网络参数对于检测结果影响（至少分析两个变量）（20 分）

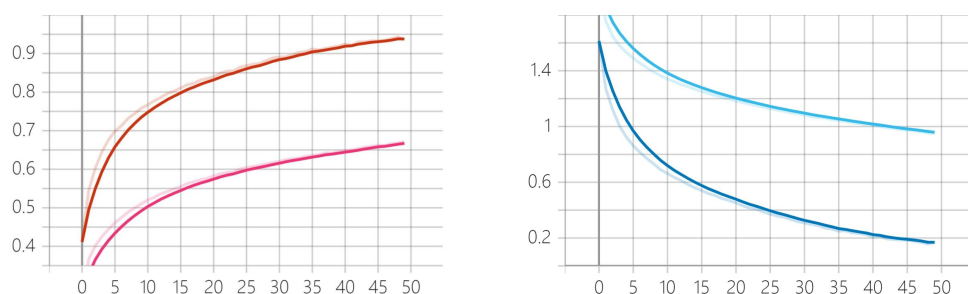
修改网络结构-调整神经元数量



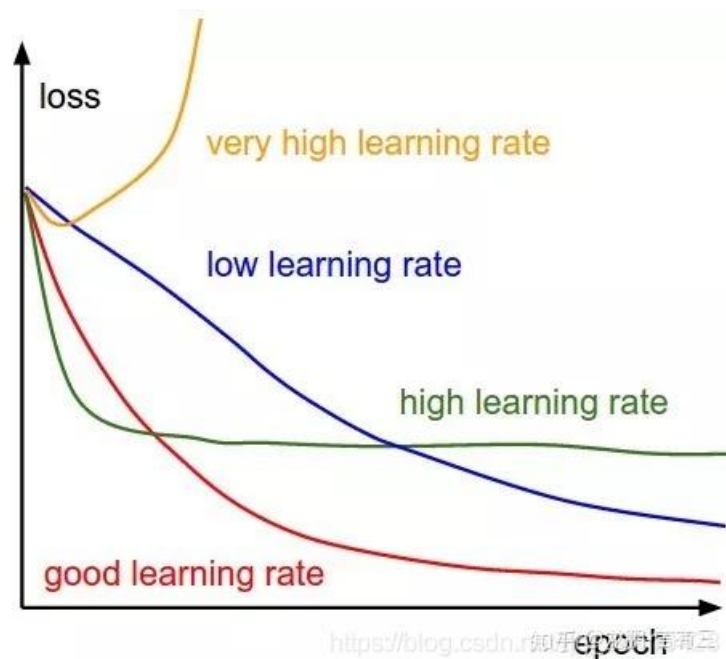
将全连接层神经元数量从 32 减少到 16

一般来说，在调参合理的情况下，层数和神经元数越多，正确率越高

更改训练参数-学习率

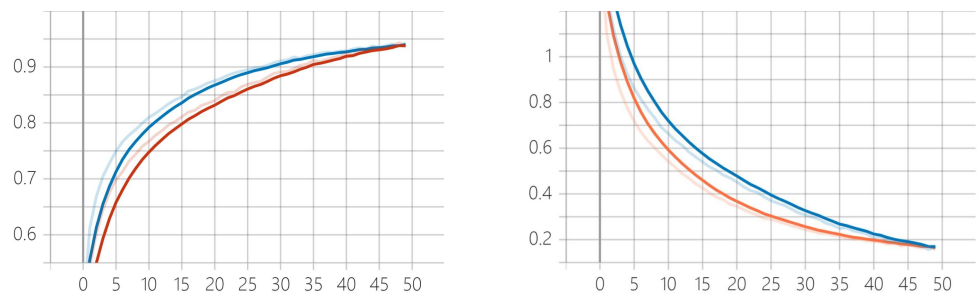


学习率为  $1e-4$

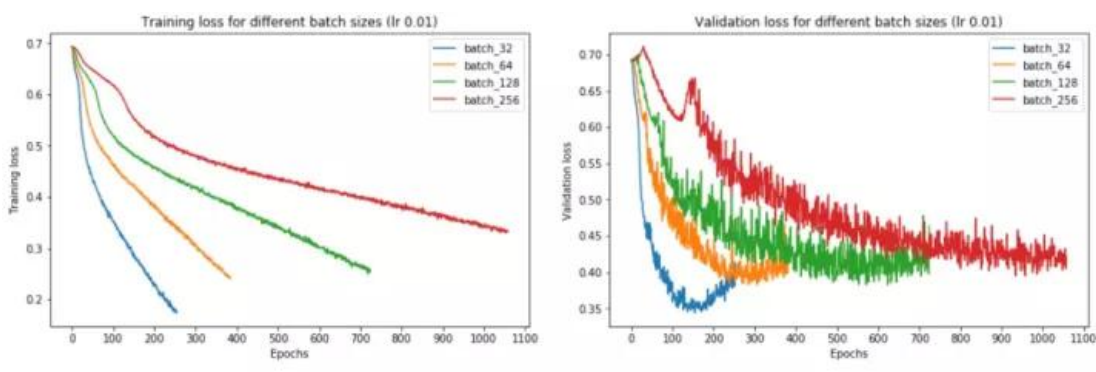


学习率过大则导致模型不收敛，过小则导致模型收敛特别慢或者无法学习

更改训练参数-batch\_size



Batch\_size 越大，训练损失减少的越慢；最小验证损失越高；每个时期训练所需的时间越少；收敛到最小验证损失所需的 epoch 越多

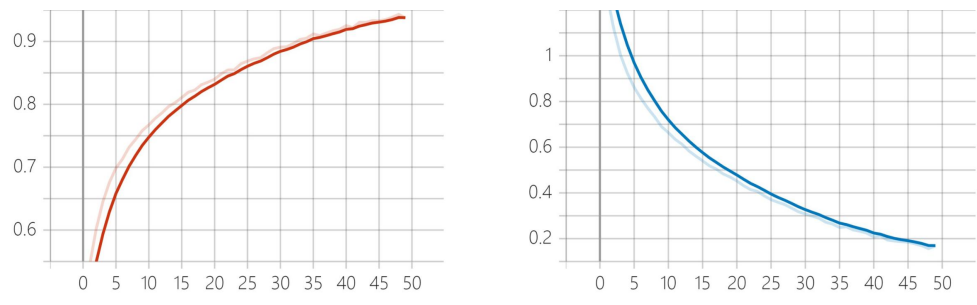


不同批次大小的训练和验证损失曲线

	Minimum training loss	Minimum validation loss
Batch size 32	0.174	0.344
Batch size 64	0.241	0.383
Batch size 128	0.250	0.383
Batch size 256	0.330	0.395

每个批次大小获得的最佳损失

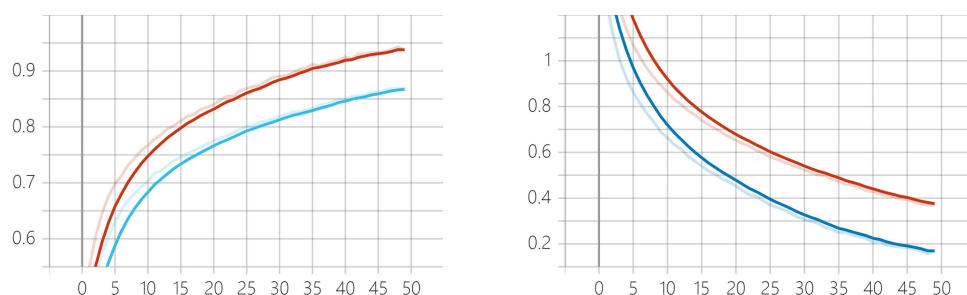
任务五：使用 tensorboard 插件对训练过程中的 loss 和精度进行观察，对 tensorboard 中 loss 曲线和 accuracy 曲线进行截图记录（10 分）



任务六：使用训练模型对于测试集中第  $i$  到  $i+10$  张图片进行预测，输出预测结果与预测概率（ $i$ =学号最后两位\*10）（10 分）

```
jupyter 徐川峰-2021523016-期中作业 最新检查点 4 小时前 (更改未保存)
File Edit View Insert Cell Kernel Widgets Help
In [77]: id_num = 16
         indices = range(id_num * 10, id_num * 10 + 10)
         test_loader = DataLoader(
             test_data,
             batch_size=BATCH_SIZE,
             sampler = indices,
             pin_memory=True)
         test(model = model, device = DEVICE, test_loader = test_loader)
>> Test accuracy: 0.6000
```

任务七：比较仅使用单通道（R 通道）作为输入和使用三通道图像作为输入训练结果的差异（选做）



仅使用 R 通道和使用 RGB 三通道的对比，仅使用 R 通道的性能没有使用三通道的

任务八：尝试使用 KNN 等机器学习算法进行分类，并将其结果与卷积神经网络结果进行对比，分析结果差异（选做）

.....

（www 来不及学习新的算法惹.....有机会一定会努力尝试一下任务 8 的！辛苦助教哥哥可以给俺一个漂亮的分数谢谢谢谢^^!! ）