



## 10. Experiment II

### 神经网络分类与 k-means 聚类

## Neural Network Classification and k-means Clustering

第二节实验课（第 12 周，3 课时，学院楼 B406）将进行对监督学习与非监督学习的实践能力考察。本小节将具体从 2 个子实验对监督学习与非监督学习的实践要求进行说明。

实验内容 1：CIFAR-10 数据集分类（全连接神经网络与卷积神经网络）

实验内容 2：MNIST 数据集聚类（k-means）

本次实验所涉及编程的模型可使用现成函数进行，即卷积层、全连接层、聚类方法过程可直接调用库函数实现。

实验报告提交（**请仔细对照**）：

- 格式：四号字体、单倍行距、中文宋体、英文新罗马字体，封面使用“实验报告封面.docx”，电子版编辑完后转为 pdf 格式发送，文件名 20AI+b+ 学号.pdf(例如：20AIb1033200101.pdf)；
- 时间节点：13 周周末（2023/05/14 18:00 前）
- 电子版发送至 raojiyong@stu.jiangnan.edu.cn，同步抄送 tianyang.xu@jiangnan.edu.cn，邮件标题:20AI+b+ 学号（例如：20AIb1033200101）

### 10.1 CIFAR-10 数据集分类（全连接神经网络与卷积神经网络）

#### 10.1.1 数据：

安装环境部分不再赘述（建议安装 GPU 版本的 pytorch 或其他框架）。数据可直接在原网址（<https://www.cs.toronto.edu/~kriz/cifar.html>）处下载；代码中可直接利用

torchvision 模块自动载入。结合下面的代码熟悉 CIFAR-10 数据集，实现对该数据集的可视化。

```
1 import torch
2 import torchvision
3 import torchvision.transforms as transforms
4
5 transform = transforms.Compose(
6     [transforms.ToTensor(),
7      transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])
8
9 batch_size = 4
10
11 trainset = torchvision.datasets.CIFAR10(root='./data', train=True,
12                                         download=True, transform=transform)
13 trainloader = torch.utils.data.DataLoader(trainset, batch_size=batch_size,
14                                           shuffle=True, num_workers=2)
15
16 testset = torchvision.datasets.CIFAR10(root='./data', train=False,
17                                         download=True, transform=transform)
18 testloader = torch.utils.data.DataLoader(testset, batch_size=batch_size,
19                                          shuffle=False, num_workers=2)
20
21 classes = ('plane', 'car', 'bird', 'cat',
22            'deer', 'dog', 'frog', 'horse', 'ship', 'truck')
23
24
25 import matplotlib.pyplot as plt
26 import numpy as np
27
28 # functions to show an image
29
30
31 def imshow(img):
32     img = img / 2 + 0.5     # unnormalize
33     npimg = img.numpy()
34     plt.imshow(np.transpose(npimg, (1, 2, 0)))
35     plt.show()
36
37
38 # get some random training images
39 dataiter = iter(trainloader)
40 images, labels = dataiter.next()
41
```

```
42 # show images
43 imshow(torchvision.utils.make_grid(images))
44 # print labels
45 print(' '.join(f'{classes[labels[j]]:5s}' for j in range(batch_size)))
```

### 10.1.2 实验课具体要求：

实现一个全连接网络的版本和一个卷积神经网络的版本；满足 CIFAR-10 分类任务；训练数据和测试数据为其自带的 50000 和 10000 条。

### 10.1.3 评分依据：

- ★★★ 调整网络参数和训练参数，经过训练，实现测试集上的高精度，并给出对应分析。
- ★★☆☆ 调整网络参数和训练参数，经过训练，实现测试集上的高精度。
- ★☆☆ 调整网络参数并成功实现训练和测试。

### 10.1.4 实验报告：

- 网络模型主代码
- 训练测试的过程分析（可选：不同学习率、不同网络规模对结果的影响）
- 实验分析（如：全连接版本和卷积版本的优劣；CIFAR-10 中哪些类难以分类正确及其原因）

## 10.2 MNIST 数据集聚类 (k-means)

### 10.2.1 数据：

可从原始网站下载<http://yann.lecun.com/exdb/mnist/>；或者从 torchvision 中直接导入，可视化方法如上述 CIFAR-10。

### 10.2.2 实验课具体要求：

调用 k-means 函数进行聚类；聚类的样本仅用 test 的 10000 条数据；分析聚类数目  $k = 10$ ,  $k = 9$ ,  $k = 3$  时的聚类结果。

### 10.2.3 评分依据：

- ★★★ 完成聚类结果并进行精度分析，通过可视化分析给出  $k = 10$ ,  $k = 9$ ,  $k = 3$  时聚类结果的原因。
- ★★☆☆ 完成聚类结果并进行精度分析，用 TSNE 可视化  $k = 10$ ,  $k = 9$ ,  $k = 3$  时的聚类结果。
- ★☆☆ 完成聚类结果并进行精度分析。

### 10.2.4 实验报告:

- $k = 10, k = 9, k = 3$  时对 MNIST 聚类结果精度分析。
- 可视化分析 (可选)。
- 聚类结果原因分析 (可选)。