人工智能A – Project2

基于CNN的图像分类

复旦大学计算机科学技术学院

陈智能

2024/4/12

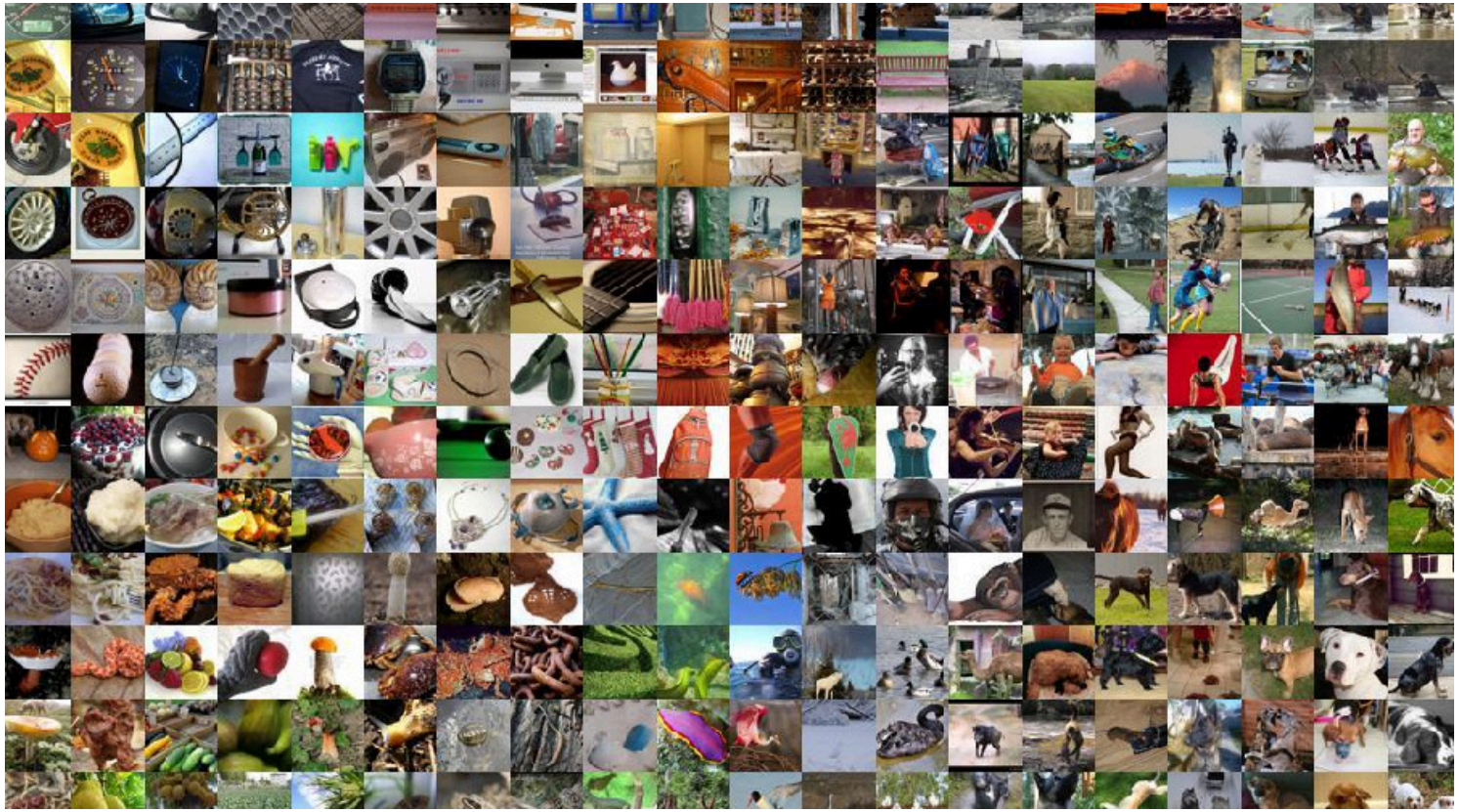
任务背景

■ 什么是图像分类?

- 图像分类是指使用**机器学习算法自动**将图像分配到预定义的类别或标签中的过程。

- 它涉及到从图像中提取特征并使用这些特征来**识别图像所属的类别。**

- 在深度学习的背景下，图像分类通常是通过构建和训练**卷积神经网络**（CNN）来完成的，这些网络能够从原始像素数据中**学习到复杂的特征表示。**



任务背景

为什么要做图像分类?

- 图像分类是计算机视觉研究领域的**基石**，对于推动AI技术的发展和应用起到了重要作用。
- 它使计算机能够**“看”和“理解”视觉世界**，从而在无需人工干预的情况下做出决策。
- 图像分类技术的进步已在多个领域引起了革命性的变化，包括**医疗、安全、娱乐、交通**等。

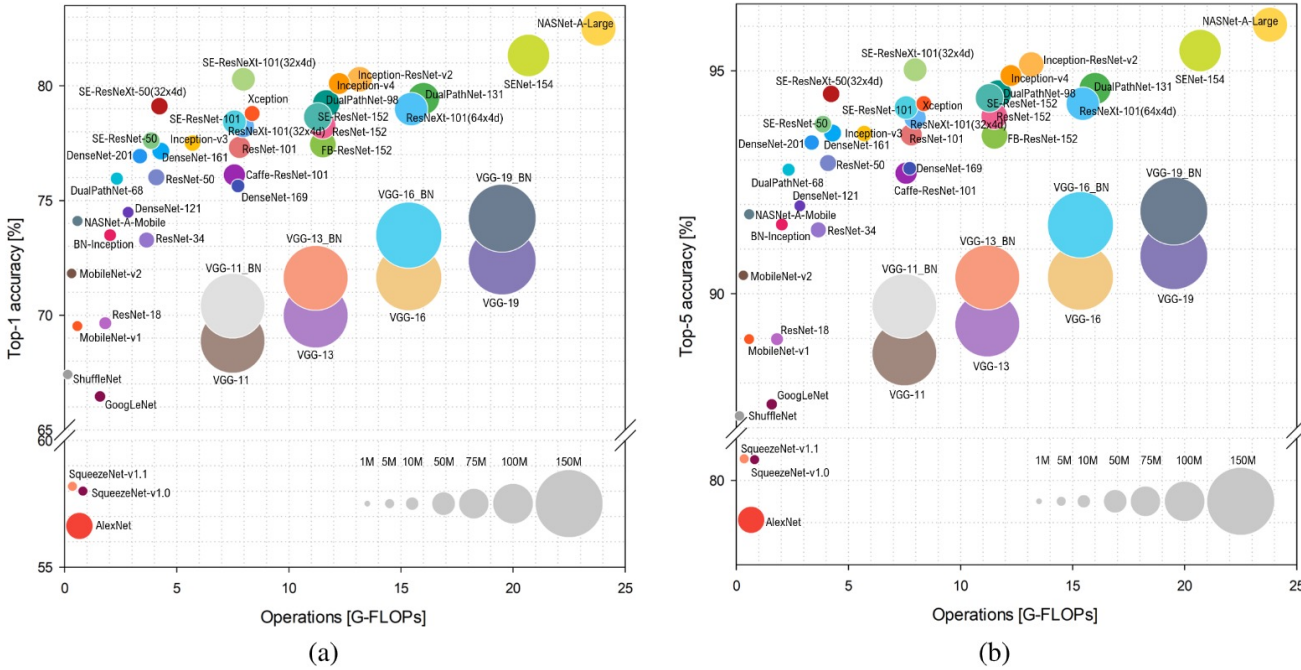


FIGURE 1. **Ball chart reporting the Top-1 and Top-5 accuracy *vs.* computational complexity. Top-1 and Top-5 accuracy using only the center crop versus floating-point operations (FLOPs) required for a single forward pass are reported. The size of each ball corresponds to the model complexity. (a) Top-1; (b) Top-5.**

图像分类是计算机视觉中最基础的一个任务，也是几乎所有的基准模型进行比较的任务。

任务背景

■ 图像分类的应用实例

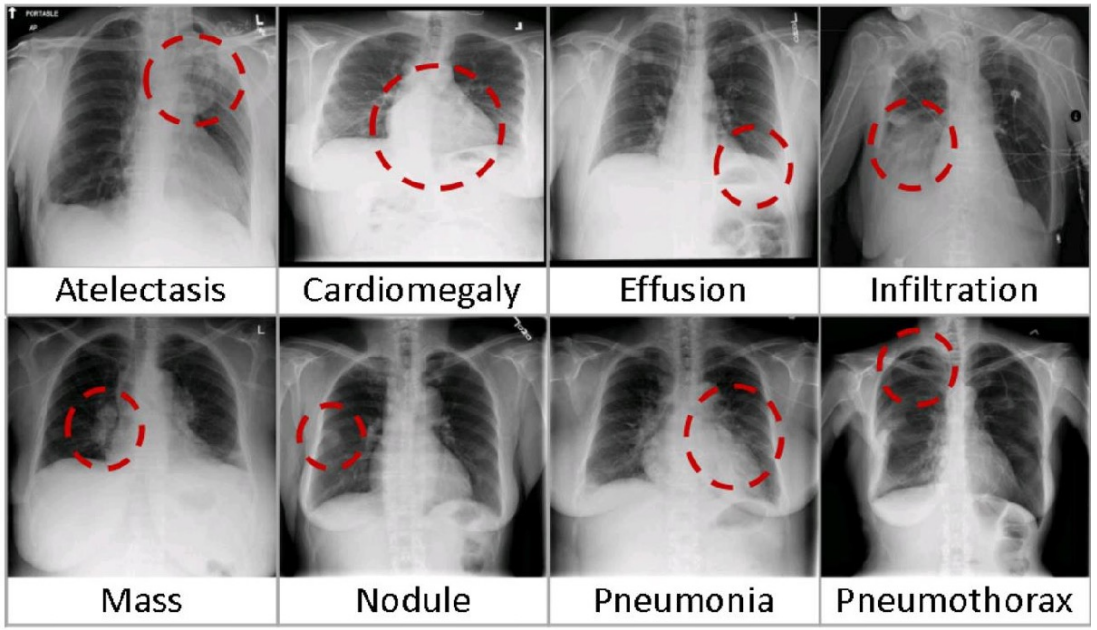
1.医疗成像：自动诊断疾病，如通过分析X光、CT扫描或MRI图像来检测癌症和其他病变。

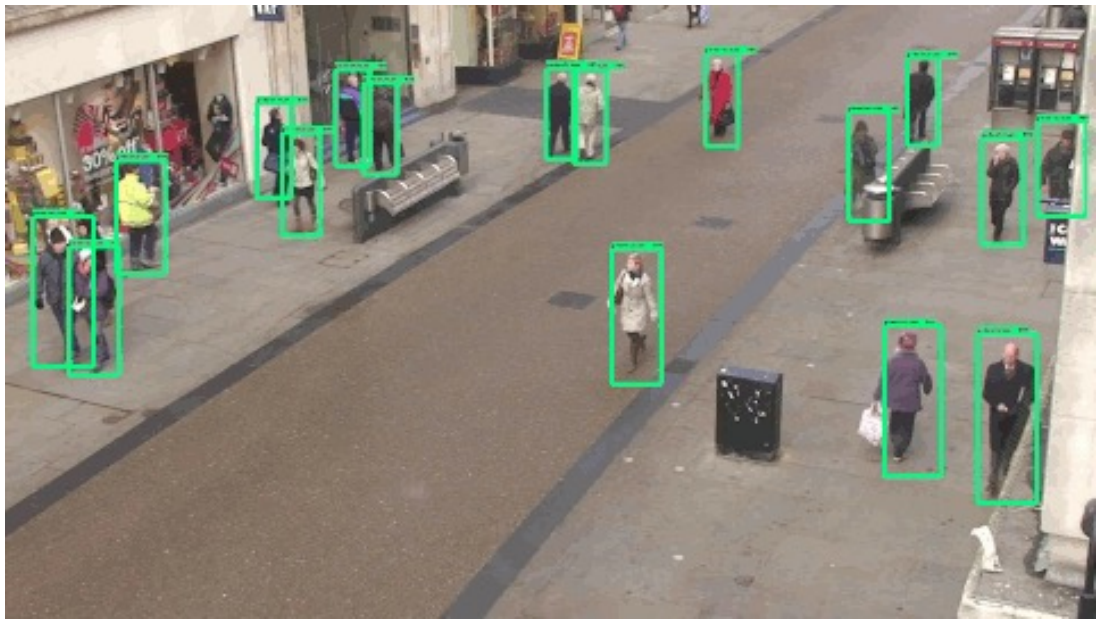
2.自动驾驶：车辆能够识别和分类道路上的物体，如其他车辆、行人、交通标志，从而做出驾驶决策。

3.安全监控：监控摄像头能够识别可疑活动或特定人物，用于防范犯罪或在紧急情况下迅速响应。

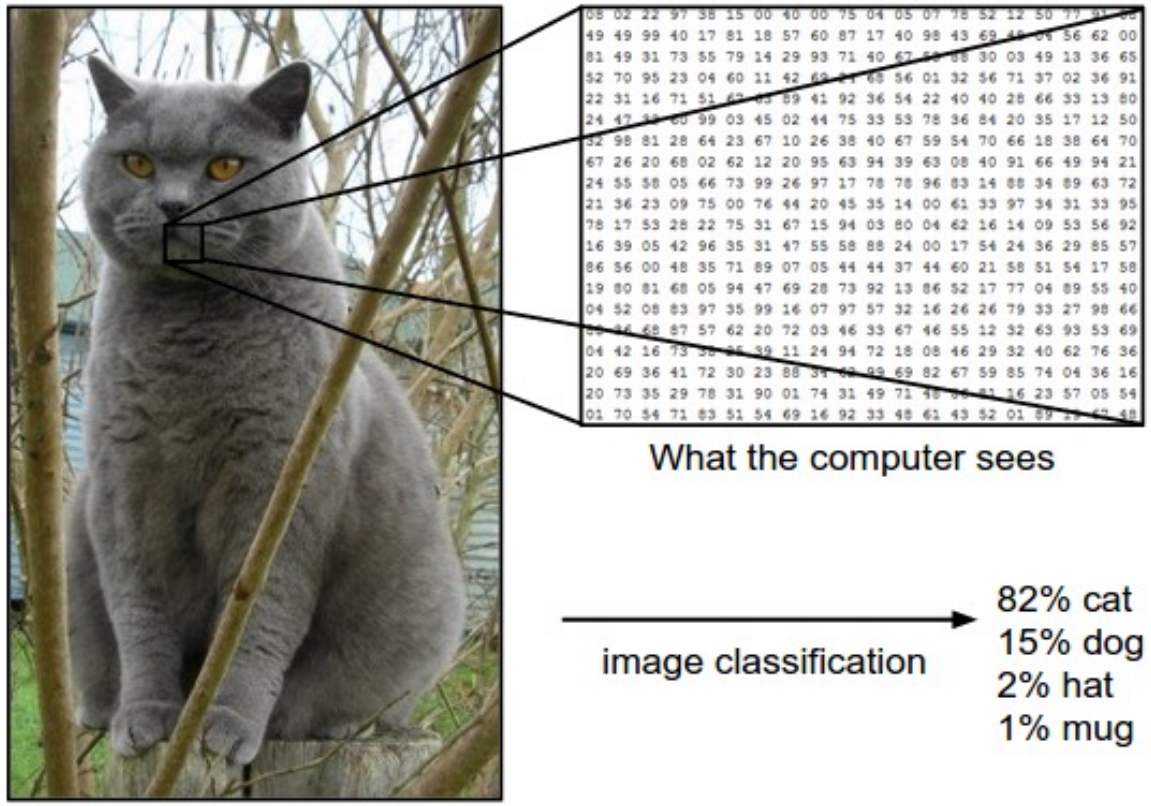
4.零售/电子商务：自动识别货架上的商品，进行库存管理和自动结账。

5.社交媒体：自动标记和分类上传的图片，提供更个性化的内容推送。





计算机如何实现图像分类?



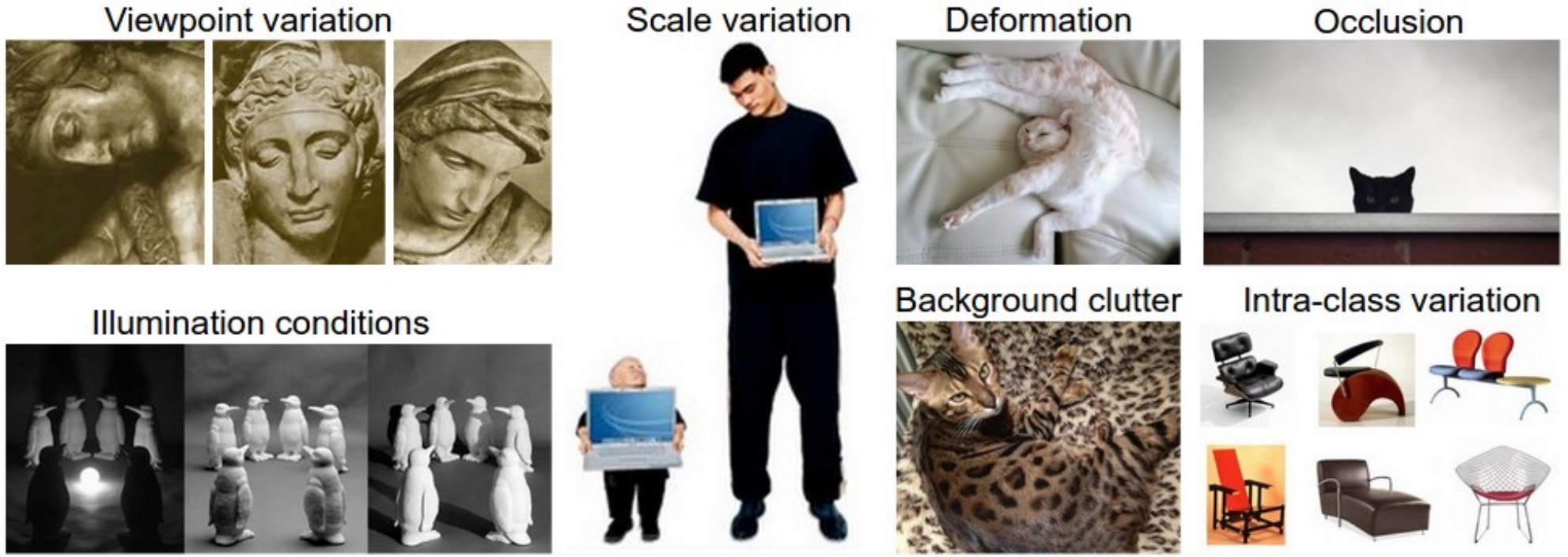
⟹ *A picture is worth a thousand words.*

猫图像宽 248 像素，高 400 像素，具有红、绿、蓝（简称 RGB）三个颜色通道。因此，该图像由 248 x 400 x 3 个数字组成，即总共 297,600 个数字。每个数字都是一个范围从 0（黑色）到 255（白色）的整数。**计算机的任务是将这接近30万个数字变成一个标签，例如*“猫”*。**

任务背景

图像分类的挑战

- 视角多变
- 尺度多变
- 物体变形
- 物体遮挡
- 光照条件
- 背景杂乱
- 类内多变

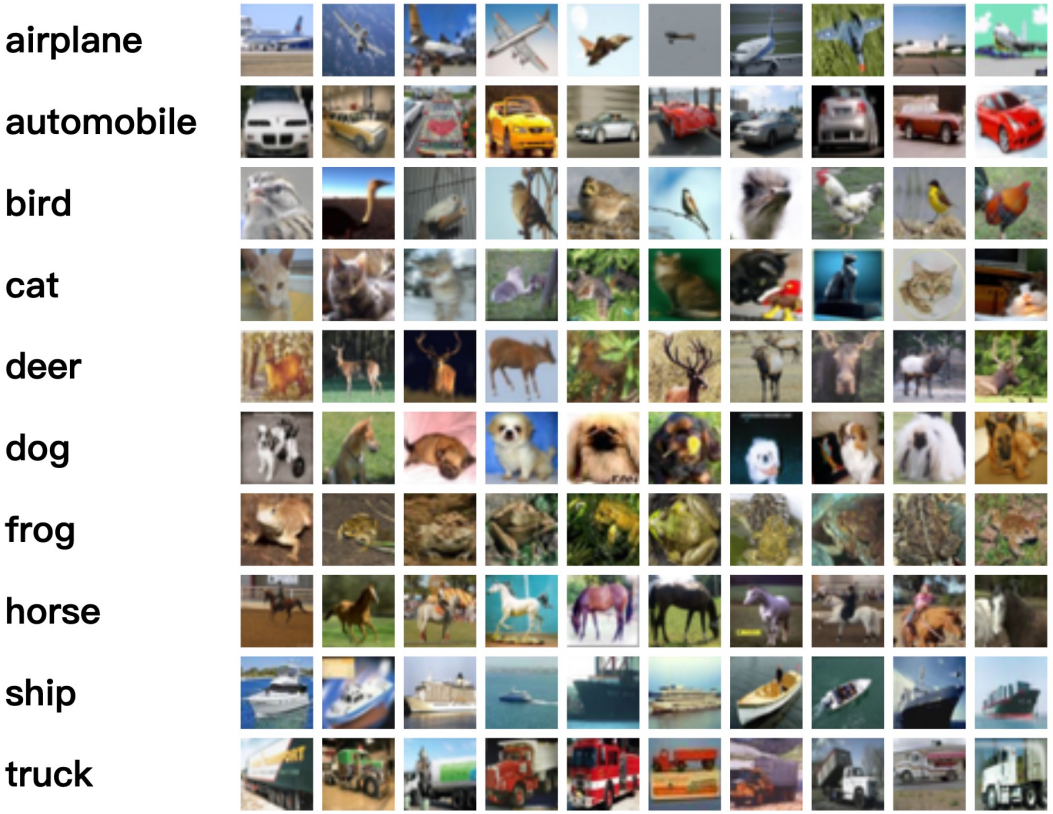


数据集介绍

CIFAR-10[1]

https://www.cs.toronto.edu/~kriz/cifar.html

CIFAR-10 数据集是用于机器视觉领域的图像分类数据集，它有飞机、汽车、鸟类、猫、鹿、狗、青蛙、马、船和卡车共计 10 个类别的 60000 张彩色图像，每一类包含6000图片，尺寸均为 32*32。其中50000张图片作为训练集，10000张图片作为测试集。该数据集由多伦多大学计算机科学系的 Alex Krizhevsky、Vinod Nair 和 Geoffrey Hinton 于 2009 年发布



[1] Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

代码结构

1. 数据下载与处理

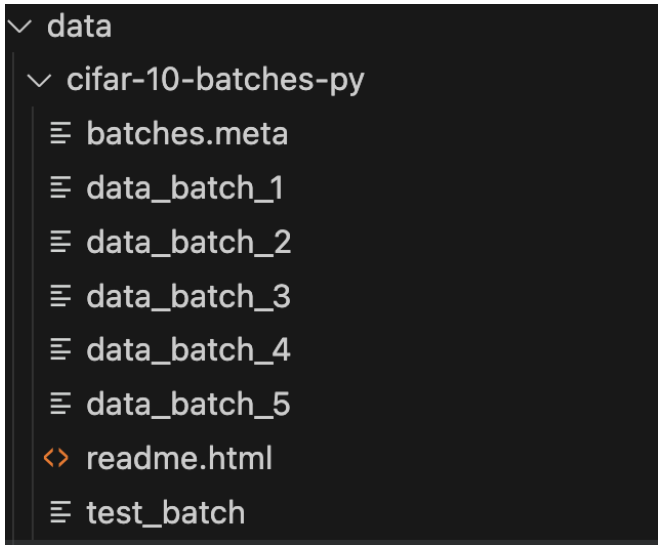
2. 模型设计

3. 模型训练、验证、测试

代码结构

1. 数据下载与处理

```
from torchvision import datasets
train_transforms = transforms.Compose([transforms.ToTensor(),
transforms.Normalize((0.491, 0.482, 0.446), (0.247, 0.243, 0.261))
]) # TODO 设计针对训练数据集的图像增强
test_transforms = transforms.Compose([
transforms.ToTensor(),
transforms.Normalize((0.491, 0.482, 0.446), (0.247, 0.243, 0.261))
])
trainset = datasets.CIFAR10(data_dir, train=True, download=True,
transform=train_transforms)
testset = datasets.CIFAR10(data_dir, train=False, download=True,
transform=test_transforms)
trainloader = torch.utils.data.DataLoader(trainset, batch_size=512,
shuffle=True, num_workers=4)
testloader = torch.utils.data.DataLoader(testset, batch_size=512,
shuffle=False, num_workers=4)
```



代码结构

2. 模型设计

```
#这里设计了ResNet-18

class ResNet(nn.Module):

# TODO
# 分析讨论其他的CNN网络设计
```

代码结构

3. 模型训练、验证、测试

```
# Training the model

optimizer = optim.SGD(model.parameters(), lr=0.01, momentum=0.9)
scheduler = ReduceLROnPlateau(optimizer, mode='min', factor=0.05, patience=2,
threshold=0.0001, threshold_mode='rel', cooldown=0, min_lr=0, eps=1e-08, verbose=True)
EPOCHS = 40
for i in range(EPOCHS):
    print(f'EPOCHS : {i}')
    model_training(model, device, trainloader, optimizer, train_acc, train_losses)
    ...
    model_testing(model, device, testloader, test_acc, test_losses)
    ...
# TODO
# 分析讨论不同的优化器optimizer与学习率lr的设计
```

代码结构

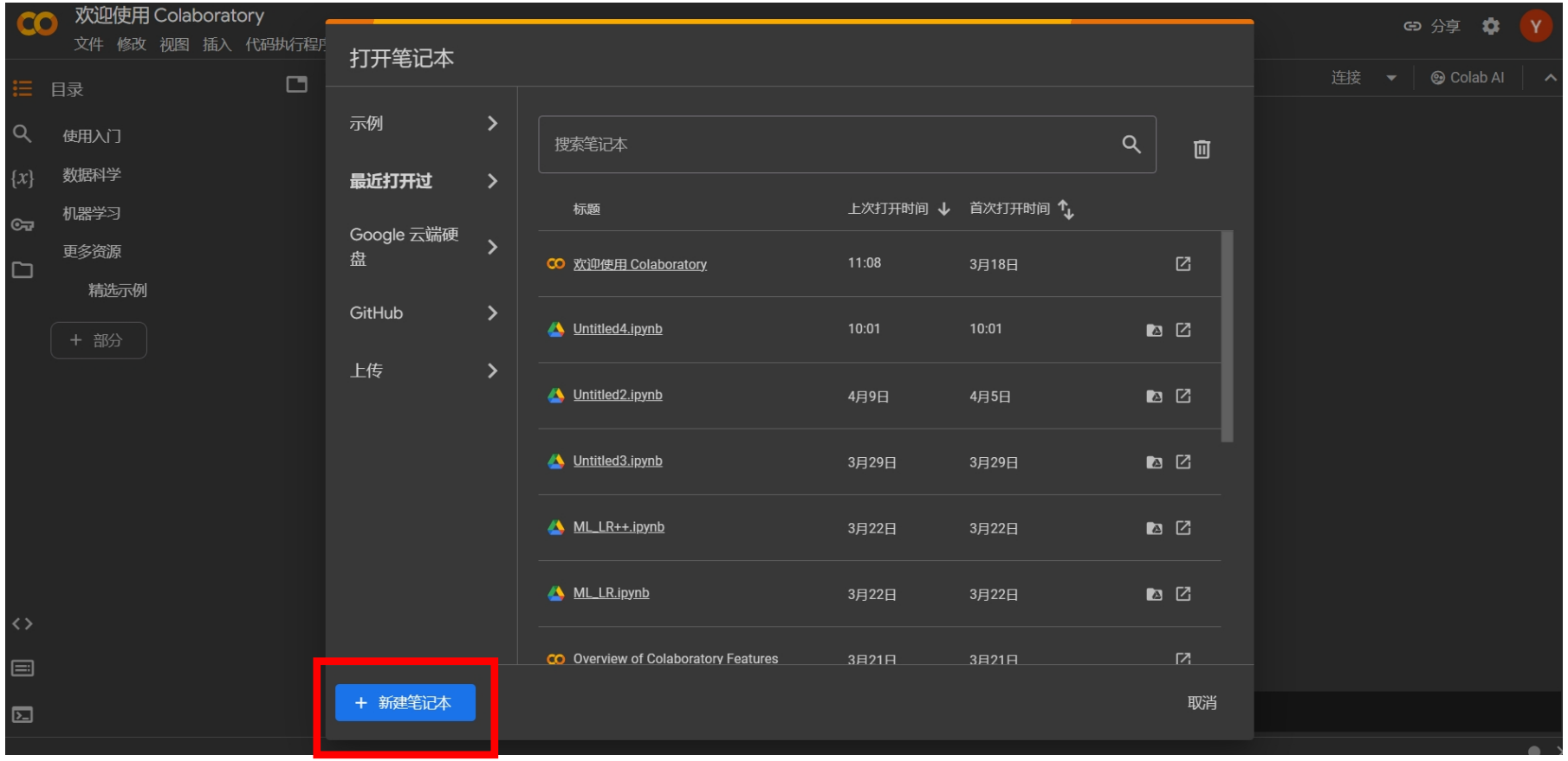
3. 模型训练/测试

```
def model_training(model, device, train_dataloader, optimizer,
train_acc, train_losses):

# TODO
# 补全模型训练代码: optimizer的操作，获取模型输出，loss设计与计算，反向传播


def model_testing(model, device, test_dataloader, test_acc,
test_losses, misclassified = []):

# TODO
# 补全模型验证代码: 获取模型输出，loss计算
```

运行环境

Google Colab

1. 新建笔记本

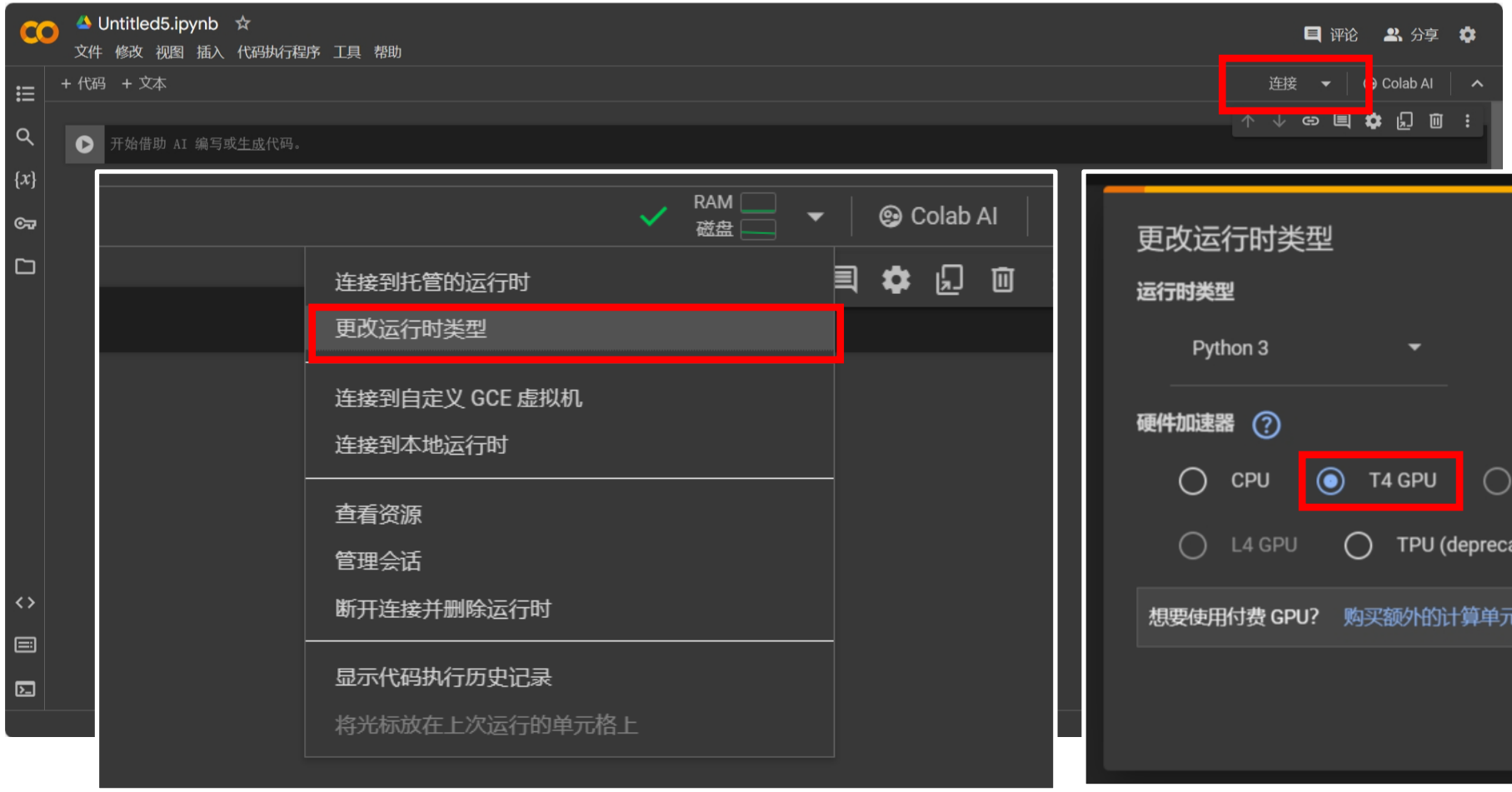


https://colab.research.google.com/

运行环境

Google Colab

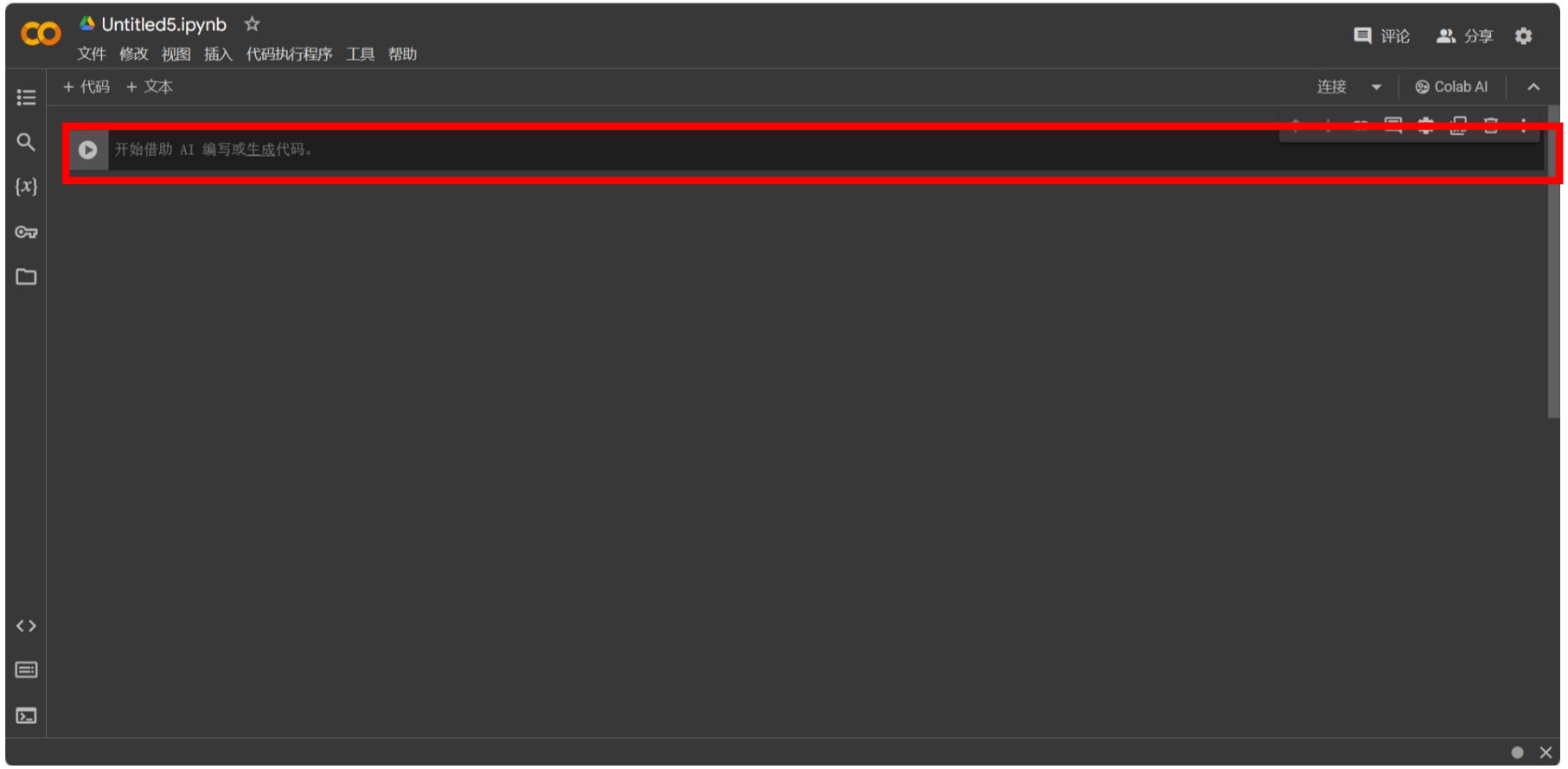
2. 连接GPU



运行环境

Google Colab

3. 单元格填入代码，点击运行（colab自带环境无需配置）



运行环境

Google Colab

4. 样例输出

```
cuda
Downloading https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz to ./data/cifar-10-python.tar.gz
100%|██████████| 170498071/170498071 [00:02<00:00, 61689240.79it/s]
Extracting ./data/cifar-10-python.tar.gz to ./data
Files already downloaded and verified
/usr/local/lib/python3.10/dist-packages/torch/utils/data/dataloader.py:558: UserWarning: This DataLoa
  warnings.warn(_create_warning_msg(
/usr/local/lib/python3.10/dist-packages/torch/optim/lr_scheduler.py:28: UserWarning: The verbose para
  warnings.warn("The verbose parameter is deprecated. Please use get_last_lr() "
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 64, 32, 32]           1,728
       BatchNorm2d-2          [-1, 64, 32, 32]             128
            Conv2d-3          [-1, 64, 32, 32]          36,864
----------------------------------------------------------------
Total params: 11,173,962
Trainable params: 11,173,962
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 15.44
Params size (MB): 42.63
Estimated Total Size (MB): 58.07
----------------------------------------------------------------
EPOCHS : 0
  0%|          | 0/98 [00:00<?, ?it/s]/usr/lib/python3.10/multiprocessing/popen_fork.py:66: RuntimeWarning: os
  self.pid = os.fork()
Loss=1.48312246799469 Batch_id=97 Accuracy=34.23: 100%|██████████| 98/98 [00:47<00:00,  2.06it/s]

Test set: Average loss: 1.4008, Accuracy: 4903/10000 (49.03%)

EPOCHS : 1
Loss=1.2152233123779297 Batch_id=97 Accuracy=52.30: 100%|██████████| 98/98 [00:49<00:00,  2.00it/s]

Test set: Average loss: 1.3217, Accuracy: 5484/10000 (54.84%)
```

一个epoch大概50s

跑完40epoch预计0.5h

实验要求

1. 完善代码实现图像分类（补全模型训练与验证代码） *- 4分*

2. 设计数据增强来提升准确率（如旋转、翻转等） *- 2分*

3. 分析讨论其他优化器和学习率的设计 *- 2分*

4. 分析讨论其他CNN网络的设计 *-2分*

PJ1 截止日期: 2024年4月26日13:30 !!
总分：10分　　提交：【代码+实验报告（上限4页）】至elearning

THANKS

陈智能

2024/4/12