# **CIFAR10 图像分类**

10215101475王兴隆 10215101484聂之扉

## **摘要**

这个项目是我（王兴隆）和聂之扉同学课余时间自学深度学习后决定做的，是一个基于pytorch官方提供的CIFAR10数据集进行10种图像分类的项目。包含：飞机,汽车,鸟,猫,鹿,狗,青蛙,马,船,卡车。

项目中，我和聂之扉同学分别负责RestNet18残差神经网络和VGG16神经网络的学习和编写，然后由我来完成训练神经网络模型的代码部分以及最终使用已经训练好的神经网络模型来实现图像分类的代码部分。

## **正文**

#### **项目环境：**

python 3.6.13

torch 1.10.2

cuda 11.3

torchvision 0.11.3

pillow 8.4.0

tensorboard 2.10.1

#### **思路：**

##### **学习过程：**

我们首先学习了一本叫做《白话机器学习的数学》，对ai方面有了初步的了解，随后学习了一位网络博主小土堆的深度学习入门教学：<https://www.bilibili.com/video/BV1hE411t7RN?vd_source=ae43d9e9ab78266a397ed9841816fcf4>

在了解了训练一个分类ai的详细步骤后，决定使用CIFAR10数据集，并采取领域内针对此训练集分类效果较为优越的RestNet18和VGG16神经网络进行图像分类，两种神经网络的结构和搭建在互联网上不论是图片还是视频，都很常见，这里我就不再列举详细的资源链接。

##### **训练框架：**

首先是获取数据集，因为是pytorch官方提供的数据集，所以直接在官方文档里面找到相关的函数用法，然后在训练代码里面调用，就能自动下载数据集，下载后用pytorch的dataloader以batch\_size=64载入。

然后为了提高训练的效率，要定义运行的硬件设备是英伟达显卡的cuda核心（如果没有的话，就是cpu）。

接下来就是提供一个可选择模型的输入提示，让用户能选择想要训练的模型。两种模型相关的训练参数如学习率以及自调整学习率，优化器，训练轮数等在完成模型后调整。

选择好模型后创建模型实例，创建损失函数，实例以及函数都调整到cuda核心设备上。

训练还提供tensorboard的可视化训练数据记录。

神经网络其实就像一个函数，里面有更复杂的结构和函数，也就是各种处理的层，卷积层，池化层等等，每一层有各自的作用，我的理解是，卷积层负责特征提取，池化层压缩数据量等等，然后对每轮训练的每份data（batch\_size=64）（包含图片以及对应的标签）调整到cuda设备上，输入到神经网络中并得到输出，把输出传入损失函数得到损失值，将梯度置0后用损失值对模型实例的参数进行优化，并在100次训练后输出一次损失率。

每轮训练后将进行一次测试，将测试集的内容输入到模型实例中，计算正确率并输出。

一轮训练和一轮测试结束后，将模型自动保存在model目录下。

##### **RestNet18：**

ResNet18采用的是基本残差块，CIFAR10图像尺寸是3232，经过每一个基本残差块图像尺寸减半，最后生成深度为512的22大小的5feature map，采用池化降采样为1\*1，最后一层通过全连接生成10分类的结果。

学习率是0.001，优化器是Adam，权重衰减是5e-3，训练50轮

##### **VGG16：**

VGG16的话，相关的结构和图片网络上都有，大小限制这里不再插入。

首先构建网络的卷积层和池化层，命名features，因为通常认为经过这些操作的输出为包含图像空间信息的特征层。

接着构建卷积层之后的全连接层以及分类器。

前向传播的时候先经过卷积层和池化层， 再将features（得到网络输出的特征层）的结果拼接到分类器上

最终输出是10类

学习率是0.01，优化器是SGD，权重衰减是5e-3，等间隔调整学习率 StepLR， 将学习率调整为 lr\*gamma，训练10轮

##### **模型使用：**

demo.py提供使用已训练好模型的功能，运行后首先选择model目录下的预训练模型文件，选择后，选择待测试图片，最后将在img目录下生成一个处理后的图片，图片左上角会被标注出其被神经网络检测出的物品类别。

##### **学习心得：**

经过这次项目实践，我们觉得图像处理相关的内容十分丰富而且有趣，难度其实也没有想象中的那么大，万事开头难，只要找到合适的学习路线，就能一步一步摸索出成果，项目中的难点在于神经网络的学习使用，感觉一般人是很难做到自己设计神经网络的，一个好的神经网络决定了ai的效果上限，不仅要选取合适的神经网络模型，还要学习调整参数，有的时候单单是学习率的调整就能有很大的影响，只有经过一步步的调整参数，才能得到一个相对较高的正确率，因为数据集的大小并不是很大，每张图片也就32\*32，也是供新手使用的数据集，所以最终的正确率肯定不会太高，因为数据集的质量也决定了最终的效果。我们的两个模型最终都能达到80%左右甚至更高的成功率，感觉已经取得了不错的成绩，尽管相对于一些能够达到99%的大佬，还是差了不少。pytorch还提供了CIFAR100数据集，是专门提供100分类训练的数据集，但是难度也更大，所以没有尝试，如果以后有机会的话会尝试100分类的ai，毕竟，10分类用途很局限，能够100分类自然是好的。

## **参考文献**

白话机器学习的数学：<https://baike.baidu.com/item/%E7%99%BD%E8%AF%9D%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0%E7%9A%84%E6%95%B0%E5%AD%A6/56692833?fr=ge_ala>

pytorch的使用：<https://www.bilibili.com/video/BV1hE411t7RN?vd_source=ae43d9e9ab78266a397ed9841816fcf4>