关于数据集和模型的一些笔记

原创 Zerg_Wang 于 2020-04-01 16:54:25 发布 1725 收藏 3 编辑 版权

分类专栏： Machine Learning 文章标签： dataset pytorch 深度学习 人工智能 数据集

常用数据集简介

CIFAR (Canadian Institute For Advanced Research)

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

CIFAR10：分类数据集，10类，每类6000张，分辨率为32×32×3。

CIFAR100：分类数据集，20个大类，每个大类下有5小类，共100类，每类600张（其中，训练用数据500张，测试用100张）。分辨率32×32×3。

SVHN (The Street View House Numbers Dataset)

下载地址：http://ufldl.stanford.edu/housenumbers/

分类数据集，内容为自然场景中的数字（一般为通过谷歌街景获得的门牌号码图像），共10类，数字1到9对应标签1到9，数字0对应标签10。

数据集中有用于训练的73257张图像，用于测试的26032张图像，以及额外的531131张。

官方给出了两种数据格式，一种是带有识别框的图像，另一种是与MNIST数据集类似的，大小为32×32的图像。

MNIST

下载地址：http://yann.lecun.com/exdb/mnist/

是 NIST（National Institute of Standards and Technology）下的一个子数据集。内容为手写数字图像。其中，训练用样本60000张，测试用样本10000张，图像为28×28的灰度图。

ImageNet

下载地址：http://image-net.org/download

ILSVRC竞赛（The ImageNet Large Scale Visual Recognition Challenge）使用的数据集。竞赛内容为目标识别和分类。该竞赛首届于2010年举办，其中2012年夺冠的AlexNet掀起了当前深度学习热潮。随着深度学习的发展，机器在图像识别上取得的正确率已远高于人类，继续这样的比赛已意义不大，最后一届ILSVRC举办于2017年。

目前完整ImageNet数据集包含了2万多类总计1400多万张图像，而ILSVRC竞赛只会用到其中一部分（1000类，共计100万张图像左右）

用于分类的遥感影像数据集

下载地址：http://www.graphnetcloud.cn/page155

UCMerced_LandUse：分类任务的遥感数据集，共21类，绝大多数图片尺寸为256×256×3。

AID ：分类任务的遥感影像数据集，图像像素大小为 600×600，总包含 30 类场景图像，每一类大概 220-420 张，共 10000 张。该数据集于 2017 年由武汉大学和华中科技大学发布。

NWPU-RESISC45：分类任务的遥感影像数据集，图像大小256×256，一共45类，每类700张，共31500张。

RSI-CB128：分类任务的遥感影像数据集，分辨率128×128，共7大类，45小类。

RSI-CB256：分类任务的遥感影像数据集，分辨率256×256，共7大类，35小类，共24747张。

验证集与测试集的区别

参考资料：https://blog.csdn.net/kieven2008/article/details/81582591

作用上，测试集一般用来评估模最终模型的泛化能力，但不作为调参、选择特征等算法的相关依据。验证集用于调超参数，监控模型是否发生过拟合（以决定是否停止训练）。一般测试集仅使用一次，验证集可多次使用，且可以和训练集相互转化（如交叉验证）。

一个形象的比喻：

训练集——学生的课本；学生根据课本里的内容来掌握知识。

验证集——作业；通过作业可以知道学生学习情况、进步的速度快慢。

测试集——考试；考的题是平常都没有见过，考察学生对知识的实际掌握程度。

实际上，不同模型在同一数据集上的训练效果不同，精确度也不同，为了公平衡量各个模型的优劣，有些数据集会给定一个子集，作为所有模型共同的测试集。这样所有模型在同一测试集下的结果才有对比性。

Pytorch下dataset使用

torchvision.datasets

ImageFolder：加载自定义数据集

torch.utils.data.DataLoader

参数简介：

dataset：指定要加载的数据集，类型与上面torchvision.datasets的类型一致。

drop_last：默认为False，当数据集中的数据量无法被batch size整除时起效。若为True则最后一个batch的数据被丢弃，若为False则保留，该batch的数据量会比之前的小。

shuffle：默认为False，设置为True时会在每个epoch重新打乱数据。

sampler：定义从数据集中提取样本的策略。如果指定，则忽略shuffle参数。

num_workers：加载数据时所要用到的子进程数，默认为0，即使用主进程。

Pytorch下model使用

vgg16和vgg16_bn的区别：后者采用了批归一化，效果更好。

原生Alexnet因为自身架构问题，当输入图像尺寸为32×32或更小时，会因为输出过小而报错（RuntimeError: Given input size: (256x1x1). Calculated output size: (256x0x0). Output size is too small.）

预训练模型

应用模型时，参数pretrain=true的意思是：采用预训练模型。

由于pytorch所提供的预训练模型是用ImageNet训练完成的，因此在使用预训练模型训练自己的数据集时，需要更改全连接层的结构，将预训练模型的1000个输出（使用的ImageNet数据集有1000类）设置为自己数据集的类别，例如：

```
from torch import nn
from torchvision import models

model = models.__dict__[model_name](pretrained=True)
```

对于不同模型，进行不同设置：

```
model.fc = nn.Linear(2048, num_class)     #resnet50

model.classifier._modules['6'] = nn.Linear(4096, num_class)     # vgg16、alexnet

model.classifier = nn.Linear(2208, num_class)     #densenet161

```

也可以通过查看网络的架构进行修改，以resnet系列和densenet系列为例：

```
#resnet
num_in = model.fc.in_features
```

```
model.fc = nn.Linear(num_in, num_class)

#densenet
num_in = model.classifier.in_features
model.classifier = nn.Linear(num_in, num_class)
```