用 Kaggle 经典案例教你用 CNN 做图像分类！

本文作者：AI研习社 编辑：贾智龙 2017-07-25 15:12

导语：本文讲的是Kaggle经典的CIFAR图像分类问题。

雷锋网按：本文原作者天雨粟，原文载于作者的知乎专栏——机器不学习，雷锋网(公众号：雷锋网)经授权发布。

前言

在上一篇专栏中，我们利用卷积自编码器对 MNIST 数据进行了实验，这周我们来看一个 Kaggle 上比较经典的一个图像分类的比赛 CIFAR(CIFAR-10 - Object Recognition in Images)，这个比赛现在已经关闭了，但不妨碍我们来去通过它学习一下卷积神经网络做图像识别的代码结构。相信很多学过深度学习的同学都尝试过这个比赛，如果对此比较熟悉的可以跳过本篇，如果没有尝试过的同学可以来学习一下哈。

整个代码已经放在了我的 GitHub 上，建议可以把代码 pull 下来，边看文章边看代码。

GitHub 地址：NELSONZHAO/zhihu

如果觉得有帮助，麻烦点个 star 啦~

介绍

文章主要分为两个部分，第一部分我们将通过一个简单的 KNN 来实现图像的分类，第二部分我们通过卷积神经网络提升整个图像分类的性能。

第一部分

提到图像分类，我们可能会想到传统机器学习中 KNN 算法，通过找到当前待分类图像的 K 个近邻，以近邻的类别判断当前图像的类别。

由于我们的图像实际上是由一个一个像素组成的，因此每一个图像可以看做是一个向量，那么我们此时就可以来计算向量（图片）之间的距离。比如，我们的图片如果是 32x32 像素的，那么可以展开成一个 1x1024 的向量，就可以计算这些向量间的 L1 或者 L2 距离，找到它们的近邻，从而根据近邻的类别来判断图像的类别。

以下例子中 K=5。

AI研习社
编辑

聚焦数据科学，连接AI开发者。

发私信

当月热门文章

今日头条成功的核心技术秘诀是什么？深度解密个性化资讯推荐技术

提高驾驶技术：用GAN去除(爱情)动作片中的马赛克和衣服

从零开始教你用 Python 做词云

贷还是不贷：如何用 Python 和机器学习帮你决策？

一文读懂矩阵的秩和 意义

1

最新文章

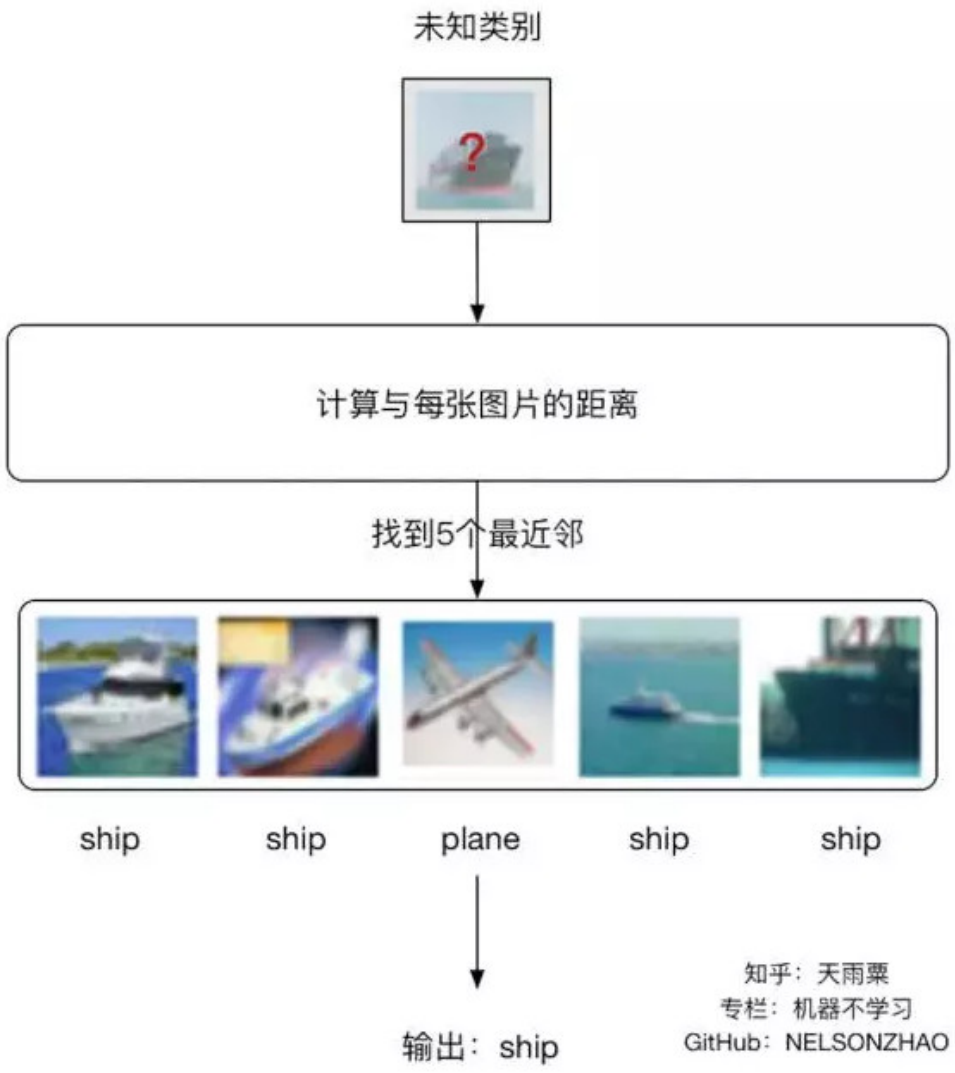
看完立刻理解GAN！初学者也没关系

聊聊数据挖掘竞赛中的套路与深度学习的局限

CVPR最有趣的5篇论文，不容错过！内含最佳学生论文！ | CVPR2017

从概念到应用，全面了解强化学习

通过一个 kaggle 实例学习解决机



下面我们就来用 scikit-learn 实现以下 KNN 对图像的分类。

首先我们需要下载数据文件，网址为 https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz 。我们数据包含了 60000 万图片，每张图片的维度为 32 x 32 x 3，这些图片都有各自的标注，一共分为了以下十类：

- airplane
- automobile
- bird
- cat
- deer
- dog
- frog
- horse
- ship
- truck

数据是被序列化以后存储的，因此我们需要使用 Python 中的 pickle 包将它们读进来。整个压缩包解压以后，会有 5 个 data_batch 和 1 个 test_batch。我们首先把数据加载进来：

器学习问题

一文详解如何用 python 做中文分词

热门搜索

滴滴 Uber 比特币 OPPO TechCrunch Disrupt eBay 电子商务 小米手环 健康 AT&T Indiegogo

1

```python
# 加载数据
def load_cfar10_batch(cifar10_dataset_folder_path, batch_id):
    """
    加载单批量的数据

    参数:
    cifar10_dataset_folder_path: 数据存储目录
    batch_id: 指定batch的编号
    """
    with open(cifar10_dataset_folder_path + '/data_batch_' + str(batch_id), mode='rb') as file:
        batch = pickle.load(file, encoding='latin1')

    # features and labels
    features = batch['data'].reshape((len(batch['data']), 3, 32, 32)).transpose(0, 2, 3, 1)
    labels = batch['labels']

    return features, labels
```

```python
# 加载所有训练数据
cifar10_path = '/Users/Nelson/Desktop/Computer/courses/cifar-10-batches-py'
# 共有5个batch的训练数据
x_train, y_train = load_cfar10_batch(cifar10_path, 1)
for i in range(2, 6):
    features, labels = load_cfar10_batch(cifar10_path, i)
    x_train, y_train = np.concatenate([x_train, features]), np.concatenate([y_train, labels])
```

```python
# 加载测试数据
with open(cifar10_path + '/test_batch', mode='rb') as file:
    batch = pickle.load(file, encoding='latin1')
    x_test = batch['data'].reshape((len(batch['data']), 3, 32, 32)).transpose(0, 2, 3, 1)
    y_test = batch['labels']
```

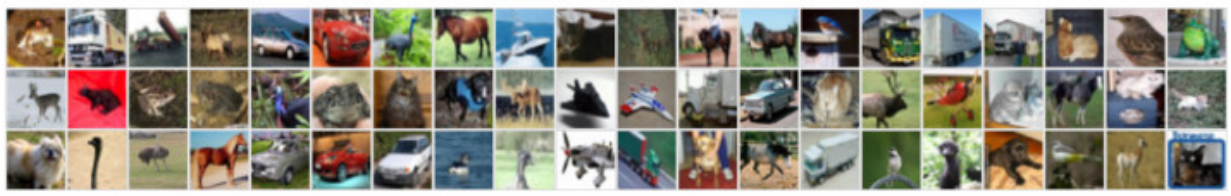
我们定义了一个函数来获取 batch 中的 features 和 labels，通过上面的步骤，我们就可以获得 train 数据与 test 数据。

我们的每个图片的维度是 32 x 32 x 3，其中 3 代表 RGB。我们先来看一些这些图片长什么样子。

```python
# 显示图片
fig, axes = plt.subplots(nrows=3, ncols=20, sharex=True, sharey=True, figsize=(80,12))
imgs = x_train[:60]

for image, row in zip([imgs[:20], imgs[20:40], imgs[40:60]], axes):
    for img, ax in zip(image, row):
        ax.imshow(img)
        ax.get_xaxis().set_visible(False)
        ax.get_yaxis().set_visible(False)

fig.tight_layout(pad=0.1)
```



每张图片的像素其实很低，缩小以后我们可以看到图片中有汽车，马，飞机等。

构造好了我们的 x_train, y_train, x_test 以及 y_test 以后，我们就可以开始建模过程。在将图片扔进模型之前，我们首先要对数据进行预处理，包括重塑和归一化两步，首先将 32 x 32 x 3 转化为一个 3072 维的向量，再对数据进行归一化，归一化的目的在于计算距离时保证各个维度的量纲一致。

用 Kaggle 经典案例教你用 CNN 做图像分类！

到此为止，我们已经对数据进行了预处理，下面就可以调用 KNN 来进行训练，我分别采用了 K=1,3,5 来看模型的效果。

用 Kaggle 经典案例教你用 CNN 做图像分类！

从 KNN 的分类准确率来看，是要比我们随机猜测类别提高了不少。我们随机猜测图片类别时，准确率大概是 10%，KNN 方式的图片分类可以将准确率提高到 35% 左右。当然有兴趣的小伙伴还可以去测试一下其他的 K 值，同时在上面的算法中，默认距离衡量方式是欧式距离，还可以尝试其他度量距离来进行建模。

虽然 KNN 在 test 数据集上表现有所提升，但是这个准确率还是太低了。除此之外，KNN 有一个缺点，就是所有的计算时间都在 predict 阶段，当一个新的图来的时候，涉及到大量的距离计算，这就意味着一旦我们要拿它来进行图像识别，那可能要等非常久才能拿到结果，而且还不是那么的准。

第二部分

在上一部分，我们用了非常简单的 KNN 思想实现了图像分类。在这个部分，我们将通过卷积神经网络来实现一个更加准确、高效的模型。

加载数据的过程与上一部分相同，不再赘述。当我们将数据加载完毕后，首先要做以下三件事：

- 对输入数据归一化
- 对标签进行 one-hot 编码
- 构造训练集，验证集和测试集

对输入数据归一化

在这里我们使用 sklearn 中的 minmax 归一化。

用 Kaggle 经典案例教你用 CNN 做图像分类！

首先将训练数据集重塑为 [50000, 3072] 的形状，利用 minmax 来进行归一化。最后再将图像重塑回原来的形状。

对标签进行 one-hot 编码

同样我们在这里使用 sklearn 中的 LabelBinarizer 来进行 one-hot 编码。

用 Kaggle 经典案例教你用 CNN 做图像分类！

构造 train 和 val

目前我们已经有了 train 和 test 数据集，接下来我们要将加载进来的 train 分成训练集和验证集。从而在训练过程中观察验证集的结果。

用 Kaggle 经典案例教你用 CNN 做图像分类！

我们将训练数据集按照 8：2 分为 train 和 validation。

卷积网络

完成了数据的预处理，我们接下来就要开始进行建模。

首先我们把一些重要的参数设置好，并且将输入和标签 tensor 构造好。

用 Kaggle 经典案例教你用 CNN 做图像分类！

img_shape 是整个训练集的形状，为 [40000, 32, 32, 3]，同时我们的输入形状是 [batch_size, 32, 32, 3]，由于前面我们已经对标签进行了 one-hot 编码，因此标签是一个 [batch_size, 10] 的 tensor。

接下来我们先来看一下整个卷积网络的结构：

用 Kaggle 经典案例教你用 CNN 做图像分类！

在这里我设置了两层卷积 + 两层全连接层的结构，大家也可以尝试其他不同的结构和参数。

用 Kaggle 经典案例教你用 CNN 做图像分类！

> conv2d 中我自己定义了初始化权重为 truncated_normal，事实证明权重初始化对于卷积结果有一定的影响。

在这里，我们来说一下 conv2d 的参数：

- 输入 tensor：inputs_
- 滤波器的数量：64
- 滤波器的 size：height=2, width=2, depth 默认与 inputs_的 depth 相同
- strides：strides 默认为 1x1，因此在这里我没有重新设置 strides
- padding：padding 我选了 same，在 strides 是 1 的情况下，经过卷积以后 height 和 width 与原图保持一致

- kernel_initializer：滤波器的初始化权重

> 在这里讲一下卷积函数中的两种常见 padding 方式，分别是 valid，same。假设我们输入图片长和宽均为 h，filter 的 size 为 k x k，strides 为 s x s，padding 大小 = p。当 padding=valid 时，经过卷积以后的图片新的长（或宽）为 $h_{new} = \frac{h-k}{s} + 1$；当 padding=same 时，经过卷积以后 $h_{new} = \frac{h-k+2p}{s} + 1$。但在 TensorFlow 中的实现与这里有所区别，在 TensorFlow 中，当 padding=valid 时，$h_{new} = ceil(\frac{float(h-k+1)}{float(s)})$；当 padding=same 时，$h_{new} = ceil(\frac{float(h)}{float(s)})$。

其余参数类似，这里不再赘述，如果还不是很清楚的小伙伴可以去查看官方文档。

在第一个全连接层中我加入了 dropout 正则化防止过拟合，同时加快训练速度。

训练模型

完成了模型的构建，下面我们就来开始训练整个模型。

用 Kaggle 经典案例教你用 CNN 做图像分类！

在训练过程中，每 100 轮打印一次日志，显示出当前 train loss 和 validation 上的准确率。

我们来看一下最终的训练结果：

用 Kaggle 经典案例教你用 CNN 做图像分类！

> 上图是我之前跑的一次结果，这次跑出来可能有所出入，但准确率大概会在 65%-70% 之间。

最后在 validation 上的准确率大约稳定在了 70% 左右，我们接下来看一下在 test 数据上的准确率。下面的代码是在 test 测试准确率的代码。

用 Kaggle 经典案例教你用 CNN 做图像分类！

我们把训练结果加载进来，设置 test 的 batchs_size 为 100，来测试我们的训练结果。最终我们的测试准确率也基本在 70% 左右。

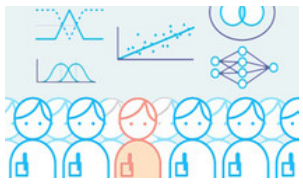
总结

至此，我们实现了两种图像分类的算法。第一种是 KNN，它的思想非常好理解，但缺点在于计算量都集中在测试阶段，训练阶段的计算量几乎为 0，另外，它的准确性也非常差。第二种我们利用 CNN 实现了分类，最终的测试结果大约在 70% 左右，相比 KNN 的 30% 准确率，它的分类效果表现的相当好。当然，如果想要继续提升模型的准确率，就需要采用其他的一些手段，如果感兴趣的小伙伴可以去看一下相关链接（http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html#43494641522d3130）里的技巧，Kaggle 上的第一名准确率已经超过了 95%。

如果觉得有用，请记得给 GitHub 打一个 Star，非常感谢！

用 Kaggle 经典案例教你用 CNN 做图像分类！

聊聊数据挖掘竞赛中的套路与深度学习的局限

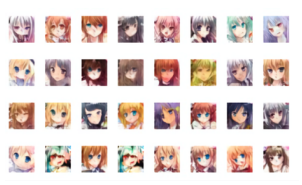


通过一个 kaggle 实例学习解决机器学习问题

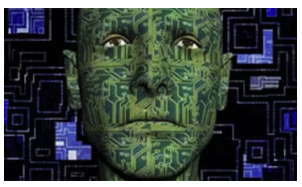
继收购DeepMind、Kaggle之后，谷歌又专门成立了一

开发者自述：我是如何从 0 到 1 走进 Kaggle 的

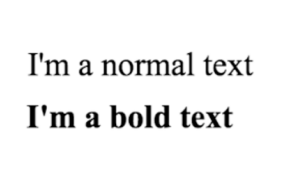
知乎的「野心与终局」



GAN学习指南：从原理入门到制作生成Demo，总共



机器学习零基础？手把手教你用TensorFlow搭建图像识



手把手教你如何用 TensorFlow 实现基于 DNN

文章点评：

我有话要说.....

同步到新浪微博

提交

最新评论

狄克•梅斯菲 3小时前

1000M光纤网络-，SS高速VPN番羽土啬限时免费领取，官网: fq1000.net?l

(0) 回复

热门关键字

热门标签 微信小程序平台 微信小程序在哪 CES 2017 CES 2016年最值得购买的智能硬件 2016 互联网 小程序 微信朋友圈 抢票软件 智能手机 智能家居 智能手环 智能机器人 智能电视 360智能硬件 智能摄像机 智能硬件产品 智能硬件发展 智能硬件创业 黑客 白帽子 大数据 云计算 新能源汽车 无人驾驶 无人机 大疆 小米无人机 特斯拉 VR游戏 VR电影 VR视频 VR眼镜 VR购物 AR 直播 扫地机器人 医疗机器人 工业机器人 类人机器人 聊天机器人 微信机器人 微信小程序 移动支付 支付宝 P2P 区块链 比特币 风控 高盛 人脸识别 指纹识别 黑科技 谷歌地图 谷歌 IBM 微软 乐视 百度 三星s8 腾讯 三星Note8 小米MIX 小米Note 华为 小米 阿里巴巴 苹果 MacBook Pro iPhone Facebook GAIR IROS 双创周 云栖大会 智能硬件公司 智能硬件 QQ红包 支付宝红包 敬业福 支付宝敬业福 支付宝集五福 Waymo 虚拟现实 深度学习 人工智能 中国银联 蚂蚁金服 WRC CNCC 5g天线 hdfs 互联网医疗 华为p8 工控安全 vr h片 京东自动下单 小米soundbar 2016款macbookpro评测 乐视最近出什么事了 魅蓝note2拆机图解 deepdream iccv2017 sony z5 compact iphone6壁纸高清 更多