机器学习基石学位

猫狗大战项目

开题报告

1.项目背景：

猫狗识别项目内容为训练一个卷积神经网络模型，该模型训练后可识别图片内动物是猫还是狗，该项目属于计算机视觉领域，计算机视觉领域内任务主要为识别、定位等，如识别图片内的一个特定物体，或物体的运动状态等。在整体系统中，计算机视觉可以提取内容中的图像信息，为后面机器的决策提供信息。所以如何更有效、更准确认地提取图像信息成为了目标。

2.问题描述：

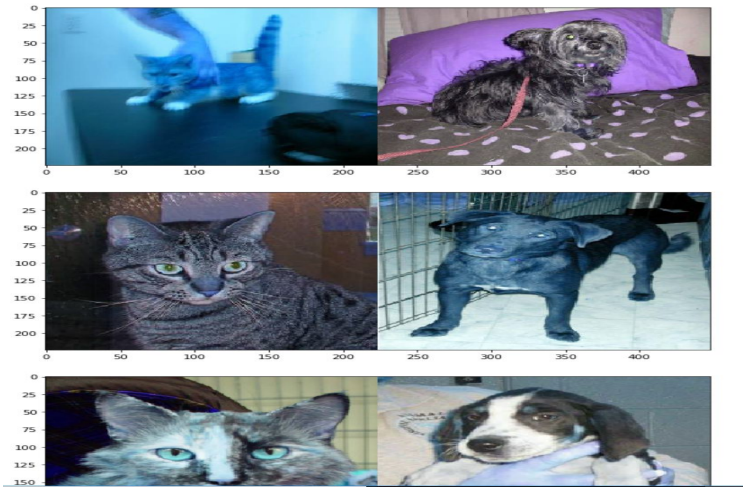
该项目提供一个有猫和狗图片的训练集，要求建立一个卷积网络模型，经过训练集训练后，该模型可输出测试图片内含有狗的概率，若输出为1则视为图片内一定有狗，输出为0则视为图片内一定有猫。

3.输入数据：

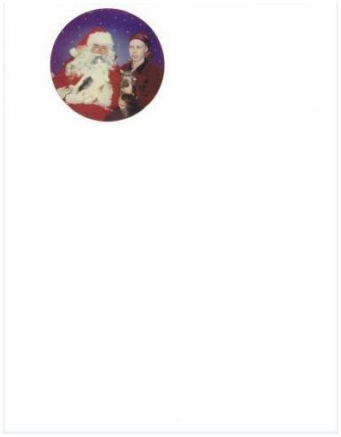
猫狗识别项目提供一个训练集以及一个测试集，训练集为25000张图片，格式为jpg，尺寸大小不固定，其中12500张图片内容为猫，12500张图片内容为狗。图片名称为typename（cat or dog）.number（0---12500）.jpg，测试集为12500张图片，jpg格式，没有标定是猫还是狗。

opencv通道顺序为BGR，matplotlib默认顺序为RGB，我们可以使用opencv的方法，cv2.cvtColor(img,cv2.BGR2RGB)转换通道。

以下是部分训练集的例子：



需注意的是训练集中有一些异常图片，如cat10059图片：



4.解决办法：

该项目的图片预处理需要使用opencv中的resize方法对图片进行大小的统一，同时考虑到训练集中有异常图片，我们可以直接使用imagenet中的模型例如inceptionV3、resnet 加载ImageNet权值，直接识别训练集内的猫狗图片，我们就可以筛选出非猫狗图片。清洗掉异常数据。

在对模型进行训练之前，需要对训练集划分两个部分，第一部分是训练集，第二部分是验证集，检验训练中的模型的表现。Val_loss指标可以看出模型是否欠拟合、过拟合以及学习率是否设置过大等问题。

该项目可使用卷积神经网络模型，通过训练的卷积核可以高度概括出出图片所具有特征，后接全连接层作为分类器。同时我们可以参考已发表论文的模型，如VGG、RESNET、INCEPTIOPN等，可以减少摸索时间，另外这些模型都参加过imagenet，imagenet内就有猫与狗的分类图片，运用迁移学习，将已经通过imagenet训练过的模型权值加载进来，可以更准确地提取特征。模型的表现水平可以使用logloss作为量化标准，

5.基准模型：

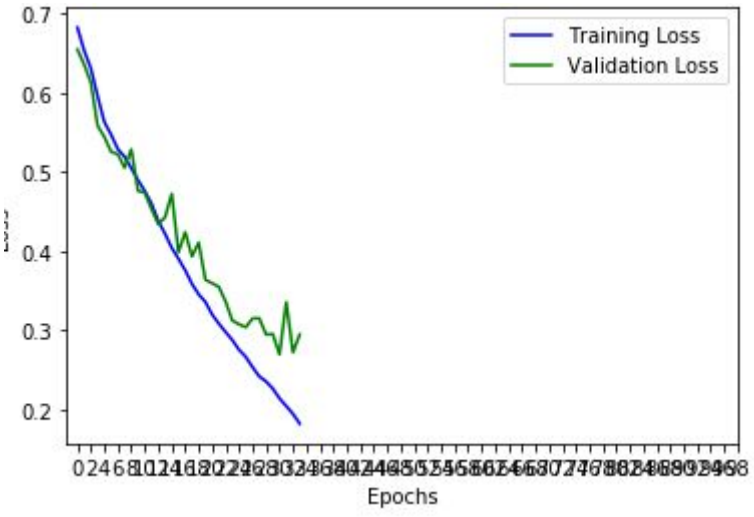
参考VGG论文，建立一个VGG19模型，结构如下：

```
model = Sequential()

#model.add(Convolution2D(32, 3, 3, border_mode='same', input_shape=(3, ROWS, COLS), activation='relu'))
#基于TensorFlow的input——shape（a,b,3）格式，我自己改了注释代码
model.add(Conv2D(64,(3, 3), border_mode='same', input_shape=( ROWS, COLS,3), activation='relu'))
model.add(Conv2D(64,(3, 3), border_mode='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128,(3, 3), border_mode='same', activation='relu'))
model.add(Conv2D(128,(3, 3), border_mode='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, (3, 3), border_mode='same', activation='relu'))
model.add(Conv2D(256, (3, 3), border_mode='same', activation='relu'))
model.add(Conv2D(256, (3, 3), border_mode='same', activation='relu'))
model.add(Conv2D(256, (3, 3), border_mode='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(512, (3, 3), border_mode='same', activation='relu'))
model.add(Conv2D(512, (3, 3), border_mode='same', activation='relu'))
model.add(Conv2D(512, (3, 3), border_mode='same', activation='relu'))
model.add(Conv2D(512, (3, 3), border_mode='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(512, (3, 3), border_mode='same', activation='relu'))
model.add(Conv2D(512, (3, 3), border_mode='same', activation='relu'))
model.add(Conv2D(512, (3, 3), border_mode='same', activation='relu'))
model.add(Conv2D(512, (3, 3), border_mode='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(4096, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(4096, activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss=objective, optimizer=optimizer, metrics=['accuracy'])
```

优化器采用Addetla，学习率为0.01，训练34轮，val_loss为0.2947， val_acc为0.8886，如图所示：

```
Epoch 34/100
20000/20000 [==============================] - 2454s 123ms/step - loss: 0.1819 - acc: 0.9281 - val_loss: 0.2947
- val_acc: 0.8886
```

整体变化趋势如图所示：



将结果提交至kaggle，得分为 0.50126，排名仅为938名。

要求最后的模型表现应在kaggle排名 10%以内，即得分需小于0.06127分

6.评估指标：

该项目为二分类问题，可以使用logloss作为量化标准，logloss定义如下图：

$$logloss = -\frac{1}{n}\sum_{i=1}^{n} y_i log(p_i) + (1-y_i) log(1-p_i)$$

logloss为对数损失函数，y_i为该样本的真实标签，p_i为该样本为阳性（即为1）的概率。

并提交至kaggle平台，获取得分并且得知模型表现在全球的排名水平。

7.设计大纲：

重新训练一个模型不仅需要耗费较多的时间，效果也不是最好。可以选择多个模型，加载“imagenet”权值并冻住卷积层，分别将训练集图片按照论文要求进行预处理，并输入进模型中提取特征，将各个模型的特征融合至一起，并输入至一个分类器中进行训练。
简单流程如图所示：

```
hight = 299
x = Input(shape=(hight,hight,3))
x = Lambda(inception_v3.preprocess_input)(x)
base_model = InceptionV3(weights = 'imagenet', include_top = False ,
input_tensor = x,pooling = 'avg')

train_gap = base_model.predict(data_train,batch_size=128 , verbose =
1 )
test_gap  = base_model.predict(data_test , batch_size=128 , verbose=
1 )
print(train_gap.shape)
```

上图为采用预训练模型InceptionV3，并加载”imagenet“权值，对图片提取特征
后可以搭建可训练的分类器进行训练，最后对测试集图片进行预测，输出为一个CSV文件，第一列为文件序号，第二列为 图片内容为狗的概率。如图所示：

```
x= Input(shape=(X_train.shape[1],))
y = Dropout(0.5)(x)
y = Dense(1 ,activation='sigmoid')(y)
model = Model(inputs = x , outputs = y)
sgd = SGD(lr=1e-3, decay=1e-6, momentum=0.9, nesterov=True)
ada = optimizers.Adadelta(lr=0.1)
model.compile(loss='binary_crossentropy', optimizer=ada, metrics=['ac
curacy'])
#print('Trainable: %d, Non-Trainable: %d' % get_params_count(model))

model.fit(x = X_train, y = y_train, batch_size=16, epochs=5, validati
on_data=(X_val, y_val))
```

```
import pandas as pd
a=[]
b=[]
for i in range(12500):
    b.append(0)
for i in range(12500):
    a.append(i+1)
    b[int(test_sorted[i])-1] = prediction[i,0]

c = pd.DataFrame({"id":a , "label":b})
c.to_csv("test28.csv",index=None)
os.getcwd()
d=pd.read_csv('C:/Users/DY/Downloads/sample_submission.csv')
print(c)
```

```
          id     label
0          1  0.995000
1          2  0.995000
2          3  0.995000
3          4  0.995000
4          5  0.005000
5          6  0.005000
6          7  0.005000
7          8  0.005000
8          9  0.005000
9         10  0.005000
10        11  0.005000
```

引用：
[1].Simonyan, Karen and Zisserman, Andrew. Very Deep Convolutional Networks for Large-Scale Image Recognition.eprint arXiv:1409.1556
[2].Szegedy, Christian; Vanhoucke, Vincent; Ioffe, Sergey; Shlens, Jonathon; Wojna, and Zbigniew. Rethinking the Inception Architecture for Computer Vision.eprint arXiv:1512.00567