识别验证码

一：主要函数

1. def convert2gray(img):

图片转为黑白，3维转1维

1. def text2vec(text, captcha\_len=CAPTCHA\_LEN， captcha\_list=CAPTCHA\_LIST):

验证码文本转为向量

参数text为验证码的五位数值，转换为向量形式表达

vector = np.zeros(captcha\_len \* len(captcha\_list))语句先建立一个零向量矩阵

在通过循环在特定的五个位置将0标成1

1. def next\_batch(path,num,width=CAPTCHA\_WIDTH,

height=CAPTCHA\_HEIGHT):

获得训练图片组

由path参数地址，遍历得到此文件夹下所有训练图片

图片的标题前五位为类别，调用text2vec（）函数变为矩阵表示形式，存入batch\_y

读入每张图片，首先调用np方法将图片用矩阵表示，再调用convert2gray（）函数变为一维向量，存入batch\_x

返回batch\_x, batch\_y

1. def test\_img(path, num, width=CAPTCHA\_WIDTH, height=CAPTCHA\_HEIGHT):

获得测试图片组

针对测试集写的函数，因为此文件夹中验证码图片标题没有指出类别，

所以只用处理图片返回batch\_x

1. def weight\_variable(shape, w\_alpha=0.01):

增加噪音，随机生成权重

1. def bias\_variable(shape, b\_alpha=0.1):

增加噪音，随机生成偏置项

1. def conv2d(x, w):

局部变量线性组合，步长为1，模式‘SAME’代表卷积后图片尺寸不变即，零边距

1. def max\_pool\_2x2(x):

max pooling,取出区域内最大值为代表特征， 2x2pool，图片尺寸变为1/2

9， def cnn\_graph(x, keep\_prob, size, captcha\_list=CAPTCHA\_LIST, captcha\_len=CAPTCHA\_LEN):

三层卷积神经网络计算图

1. def optimize\_graph(y, y\_conv):

优化计算图

传入预测值和真实值

计算损失值

最小loss优化

1. def accuracy\_graph(y, y\_conv, width=len(CAPTCHA\_LIST), height=CAPTCHA\_LEN):

偏差计算图

传入预测值和真实值，计算正确率

1. def train(height=CAPTCHA\_HEIGHT, width=CAPTCHA\_WIDTH, y\_size=len(CAPTCHA\_LIST)\*CAPTCHA\_LEN, train\_path=TRAIN\_PATH, test\_path=TEST\_PATH,train\_num=TRAIN\_NUM, test\_num=TEST\_NUM):

cnn模型训练

1. def captcha2text(path=TEST\_PATH, height=CAPTCHA\_HEIGHT, width=CAPTCHA\_WIDTH):

获得预测值

二：代码流程（主要介绍train函数）

基于tensorflow库

按照图片大小申请占位符，调用了tf.placeholder（）函数，定义了三个变量：x，图片数据；y，图片类别；keep\_prob ， 防止过拟合 训练时启用 测试时不启用

调用cnn\_graph()函数，得预测值：y\_conv = cnn\_graph(x, keep\_prob, (height, width))

最优化：optimizer = optimize\_graph(y, y\_conv)

正确率：accuracy = accuracy\_graph(y, y\_conv)

写好各个值的计算过程，启动会话.开始训练

saver = tf.train.Saver()

sess = tf.Session()

sess.run(tf.global\_variables\_initializer())

设置循环次数step变量=0

开始循环

首先调用next\_batch()函数处理训练数据得到标准格式的数据集合和类别及合batch\_x, batch\_y

调用sess.run（）函数，参数为batch\_x, batch\_y，训练模型并优化

每训练两次计算一次正确率，若正确率大于acc\_rate的值结束循环

训练好模型后调用captcha2text（）函数，对测试集进行预测

将得出的预测值按照指定格式保存

三：难点

本次作业较复杂，是参考网络上的已完成的代码。

大概参考了三四个已完成的代码，改成了现在这一个。

然后我本机的Python配置·有一点问题，无法使用tensorflow库，就去谷歌使用Colaboratory,是一个免费的Jupyter 笔记本环境

然后需要上传数据集，很难上传，上传完成解压也出了一些错误，

然后我就将训练集删减到7000张左右，训练，正确率是百分之2.7

还有一个小难点是处理数据，选择图片类别的表达形式

后来多上传了几次，完整的数据集上传解压都成功了，但是运行时都会系统崩溃，数据量过大，所以我还是使用七千多张图训练出来的模型

模型的选择，网络版本基本都是三层卷积网络，所以我使用的这一种，但是看其他同学的卷积层很多，出来的效果也更好，但是我对神经网络的掌握还不是很全面，所以没有进行尝试