

基于TensorFlow的 CAPTCHA注册码识别实验

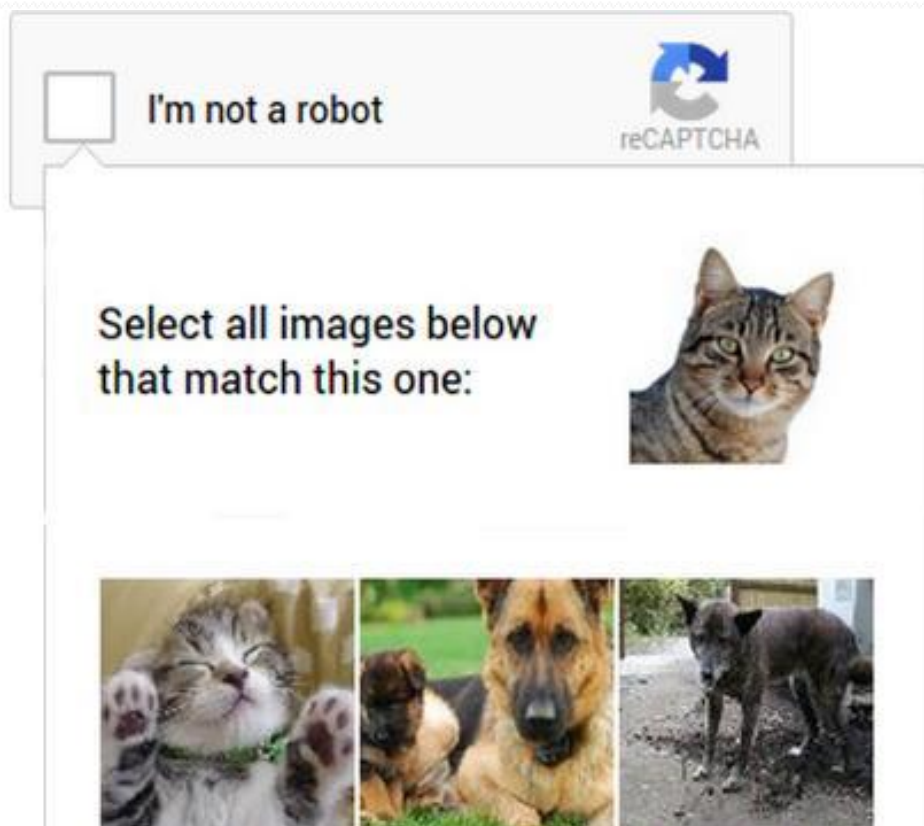
中科大软件学院 赵振刚 gavin@ustc.edu.cn

实验大纲

- CAPTCHA 库及数据集
- CAPTCHA注册码识别实践
 - 生成注册码
 - 定义卷积神经网络
 - 训练网络模型参数
 - 测试网络效果
- 调试技巧：模型参数存储与加载
- 小结

CAPTCHA 库

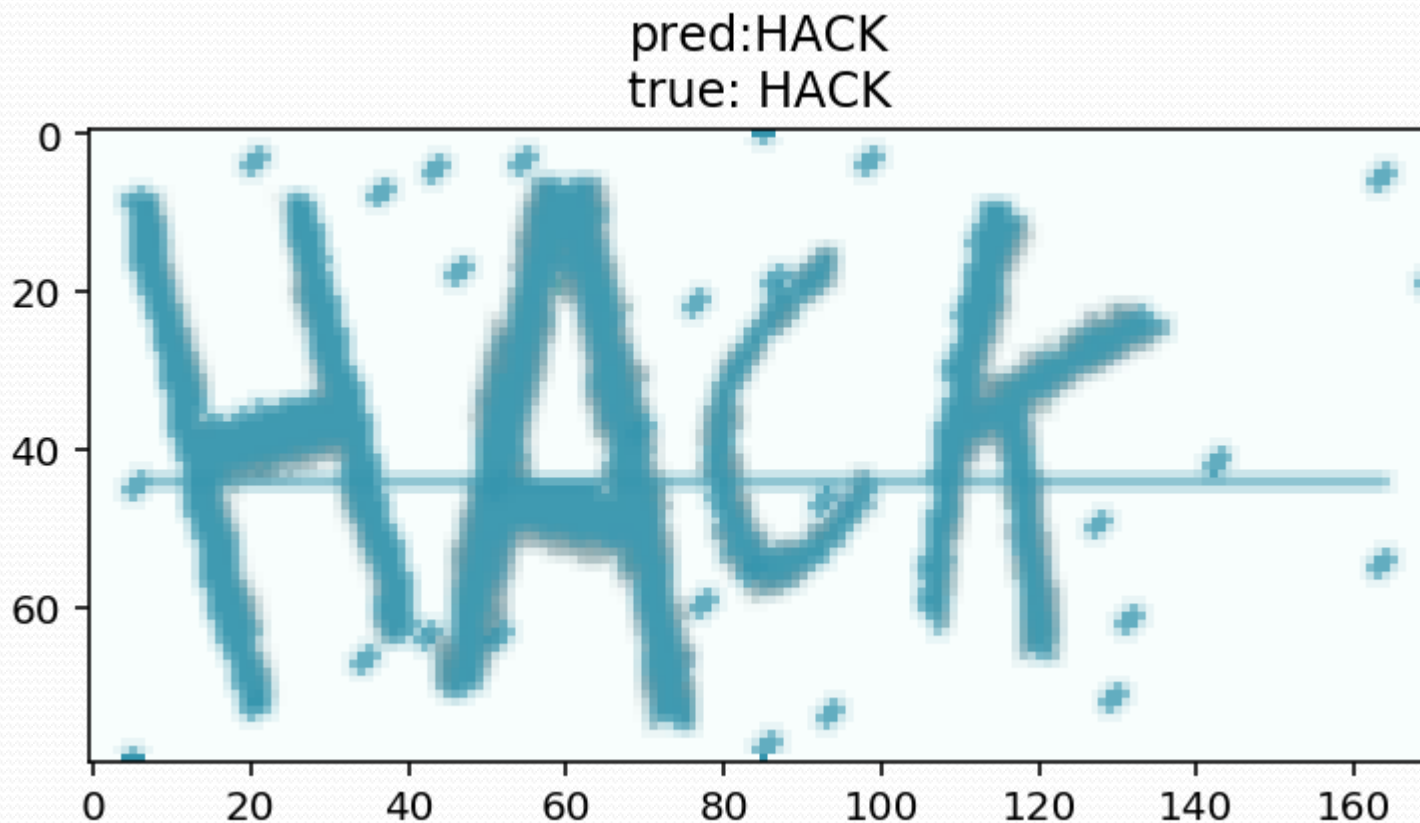
简介：Captcha（全自动区分计算机和人类的图灵测试，俗称验证码）是目前用于区分人和机器主要办法，其工作原理是通过提供模糊或是有歧义的图片，并要求用户进行回答，以此来区分人和机器
下图为Google目前采用的验证码形式



CAPTCHA 库

简介：同时，captcha 是用 python 写的生成验证码的库，它支持图片验证码和语音验证码
图片像素、字符个数均可指定

外观：



CAPTCHA 库

用例:

```
from captcha.image import ImageCaptcha
```

```
#生成一张图片
```

```
image = ImageCaptcha()
```

```
#生成一个字符
```

```
captcha_text = random_captcha_text()
```

目录

- CAPTCHA 库及数据集
- CAPTCHA注册码识别实践
 - 生成注册码
 - 定义卷积神经网络
 - 训练网络模型参数
 - 测试网络效果
- 调试技巧：模型参数存储与加载
- 小结

生成注册码

```
from captcha.image import ImageCaptcha # pip install captcha
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
#生成字符对应的验证码
def gen_captcha_text_and_image():
    image = ImageCaptcha()

    captcha_text = random_captcha_text()
    captcha_text = ''.join(captcha_text) #连接字符串

    captcha = image.generate(captcha_text)

    captcha_image = Image.open(captcha)
    captcha_image = np.array(captcha_image)
    return captcha_text, captcha_image
```

生成注册码

cnn_data.py文件可单独运行：

通过cmd命令行界面 或 Anaconda Prompt切换到工作目录，输入 `python cnn_data.py`

如果报错并提示找不到captcha，则在Prompt下安装该库：`pip install captcha`

Anaconda Prompt

```
(C:\Anaconda3) C:\Users\Gavin>activate tensorflow
```

```
(tensorflow) C:\Users\Gavin>pip install captcha
```

```
Collecting captcha
```

```
Downloading captcha-0.2.4.tar.gz (100kB)
```

```
100%
```

```
102kB 10.0kB/s
```


目录

- CAPTCHA 库及数据集
- CAPTCHA注册码识别实践
 - 生成注册码
 - 定义卷积神经网络
 - 训练网络模型参数
 - 测试网络效果
- 调试技巧：模型参数存储与加载
- 小结

实验内容

1. 定义你自己的卷积神经网络，根据CAPTCHA数据集特征确定输入层参数，根据MNIST实验内容设计卷积层、池化层参数；
2. 定义全联接层、输出层参数；
3. 仿照MNIST数据集实验，实现Python代码；
4. 调试通过；
5. 训练模型，得到收敛效果；

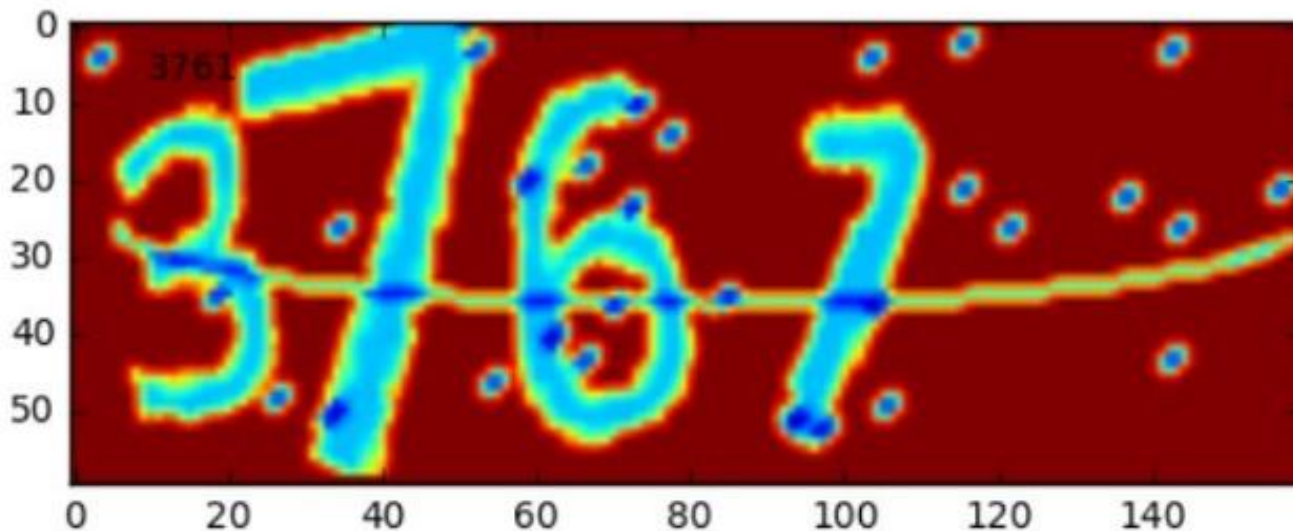
测试网络效果示例

```
plt.show()

image = image.flatten() / 255 # 将图片一维化
predict_text = crack_captcha(image)
print("正确: {} 预测: {}".format(text, predict_text))
```

验证码图像channel: (60, 160, 3)

验证码文本最长字符数 4



INFO:tensorflow:Restoring parameters from ./save/cnn_train.model-1400

正确: 3761 预测: 3761

目录

- CAPTCHA 库及数据集
- CAPTCHA注册码识别实践
 - 生成注册码
 - 定义卷积神经网络
 - 训练网络模型参数
 - 测试网络效果
- 调试技巧：模型参数存储与加载
- 小结

模型加载与存储

```
saver = tf.train.Saver()
with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())

    step = 0
    while True:
        batch_x, batch_y = get_next_batch(64)
        _, loss_ = sess.run([optimizer, loss], feed_dict={X: batch_x, Y: batch_y, keep_prob: 0.75})
        print(step, loss_)

        # 每100 step计算一次准确率
        if step % 100 == 0:
            batch_x_test, batch_y_test = get_next_batch(100)
            acc = sess.run(accuracy, feed_dict={X: batch_x_test, Y: batch_y_test, keep_prob: 1.})
            print(acc)
            saver.save(sess, "./save/cnn_train.model", global_step=step)
            if acc > 0.7:
                #saver.save(sess, "./save/cnn_train.model", global_step=step)
                break

        step += 1

#定义的测试方法
def crack_captcha(captcha_image):
    output = crack_captcha_cnn()

    saver = tf.train.Saver()
    with tf.Session() as sess:
        saver.restore(sess, "./save/cnn_train.model-1400")

        predict = tf.argmax(tf.reshape(output, [-1, MAX_CAPTCHA, CHAR_SET_LEN]), 2)
        text_list = sess.run(predict, feed_dict={X: [captcha_image], keep_prob: 1})
```

实验报告

- 给出你自己第一个完整的神经网络构建过程：
 - 生成训练数据集
 - 定义网络模型
 - 训练模型参数并保存
 - 读入参数并测试新数据
- 描述实验结果；



Thanks