

Cnn的应用和避坑

----- 聂杨



目录

Contents

01

cnn的简单介绍

结构，卷积，池化

02

代码分享

Tensorflow版本分享。

03

训练中可能出现的问题

过拟合和欠拟合



01

cnn的简单介绍



问题1:人是如何区分不同的图片

卷积----cnn中的特征寻找器

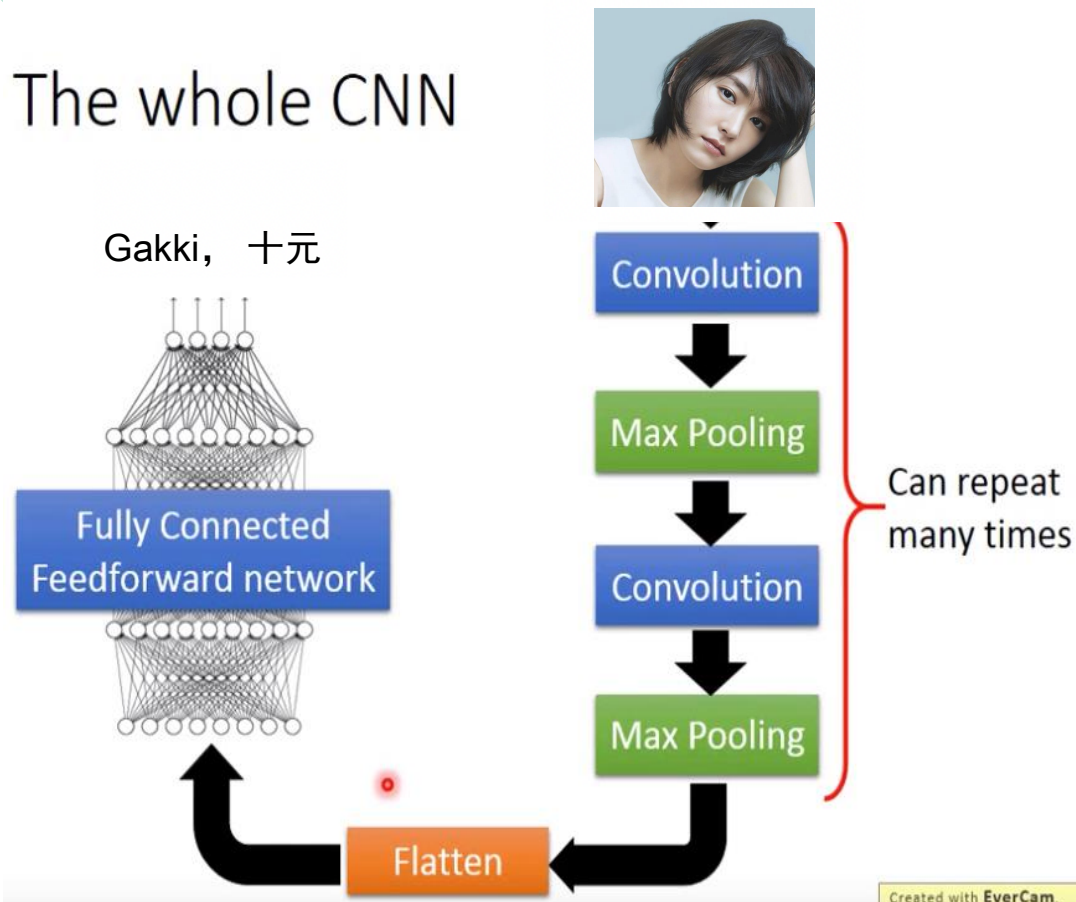


问题2:图片的大小对我们识别目标影响有多大

池化---cnn中的降维方式



The whole CNN



三个部分：input hidden output

池化层是接在卷积层之后的

CNN – Convolution

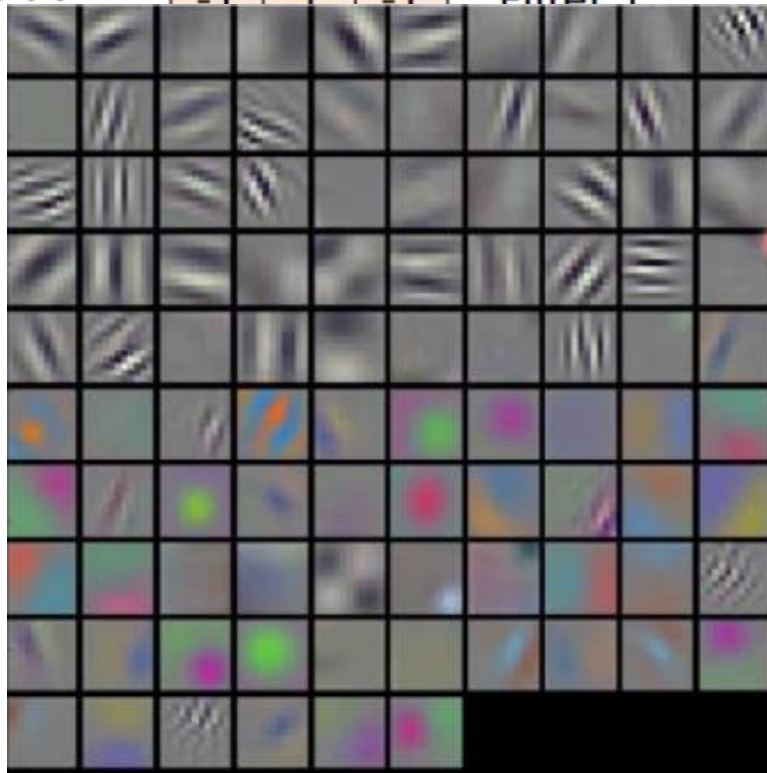
stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1

Filter 1



02

代码实现

- ◆ 场景还原和输入输出
- ◆ 卷积和池化
- ◆ 存储和其他

真实图片

2 c z s

x z Y n

h q M L

input

```
[[255 255 255 255]
 [255 255 255 255]
 [255 255 255 255]
 ...
 [255 255 255 255]
 [255 255 255 255]
 [255 255 255 255]]

[[255 255 255 255]
 [255 255 255 255]
 [255 255 255 255]
 ...
 [255 255 255 255]
 [255 255 255 255]
 [255 255 255 255]]

[[255 255 255 255]
```

Output

```
[0,0,1,0,0,0,0,0,0,0...]
```

图片存储方式

- ▶ test
- ▶ training

1:4


```
number = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u',
            'v', 'w', 'x', 'y', 'z']
IMAGE_HEIGHT = 80
IMAGE_WIDTH = 160
MAX_CAPTCHA = 4
CHAR_SET_LEN = len(number + alphabet)
```

```
INPUT = tf.placeholder(tf.float32, [None, IMAGE_HEIGHT * IMAGE_WIDTH])
OUTPUT = tf.placeholder(tf.float32, [None, MAX_CAPTCHA * CHAR_SET_LEN])
keep_prob = tf.placeholder(tf.float32)
```

```

def crack_captcha_cnn(w_alpha=0.01, b_alpha=0.1):
    # 3 conv layer batchsize大小-1 代表有n个自动计算匹配
    x = tf.reshape(INPUT, shape=[-1, IMAGE_HEIGHT, IMAGE_WIDTH, 1])
    w_c1 = tf.Variable(w_alpha * tf.random_normal([3, 3, 1, 32]))
    b_c1 = tf.Variable(b_alpha * tf.random_normal([32]))
    conv1 = tf.nn.elu([batch, height, weight, channels]
                      tf.nn.bias_add(tf.nn.conv2d(x, w_c1, strides=[1, 1, 1, 1], padding='SAME'), b_c1)) # out_size = 80*160*32
    conv1 = tf.nn.max_pool(conv1, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME') # out_size = 40*80*32
    # dropout的作用是防止过拟合按照一定的概率将一部分神经网络单元暂时从网络中丢弃
    conv1 = tf.nn.dropout(conv1, keep_prob)
    w_c2 = tf.Variable(w_alpha * tf.random_normal([3, 3, 32, 64]))#大小input_output
    b_c2 = tf.Variable(b_alpha * tf.random_normal([64]))
    conv2 = tf.nn.elu(
        tf.nn.bias_add(tf.nn.conv2d(conv1, w_c2, strides=[1, 1, 1, 1], padding='SAME'), b_c2)) # out_size = 40*80*64
    conv2 = tf.nn.max_pool(conv2, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME') # out_size = 20*40*64
    conv2 = tf.nn.dropout(conv2, keep_prob)
    w_c3 = tf.Variable(w_alpha * tf.random_normal([3, 3, 64, 64]))
    b_c3 = tf.Variable(b_alpha * tf.random_normal([64]))
    conv3 = tf.nn.elu(
        tf.nn.bias_add(tf.nn.conv2d(conv2, w_c3, strides=[1, 1, 1, 1], padding='SAME'), b_c3)) # out_size = 20*40*64
    conv3 = tf.nn.max_pool(conv3, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME') # out_size = 10*20*64
    conv3 = tf.nn.dropout(conv3, keep_prob)
    # Fully connected layer
    w_d = tf.Variable(w_alpha * tf.random_normal([10 * 20 * 64, 1024]))
    b_d = tf.Variable(b_alpha * tf.random_normal([1024]))
    # 将这一层的结果变平
    dense = tf.reshape(conv3, [-1, w_d.get_shape().as_list()[0]]) #将10, 20, 64 变成10*20*64
    dense = tf.nn.elu(tf.add(tf.matmul(dense, w_d), b_d))
    dense = tf.nn.dropout(dense, keep_prob)
    w_out = tf.Variable(w_alpha * tf.random_normal([1024, MAX_CAPTCHA * CHAR_SET_LEN]))
    b_out = tf.Variable(b_alpha * tf.random_normal([MAX_CAPTCHA * CHAR_SET_LEN]))
    out = tf.add(tf.matmul(dense, w_out), b_out)
    return out

```

[illegible][illegible]



03

常见问题



过拟合

过拟合是指样本在我们的traing库中效果很好但是在test库里很差，主要的原因是样本有问题



欠拟合

欠拟合就是是指模型不收敛，原因在于模型写法有问题，激活函数有问题，样本有问题



图片的预处理

大小，格式



感谢您的观看

Thank you for your download to watch

<https://www.jianshu.com/u/7109af2617fd>