【TensorFlow】tf.nn.conv2d是怎样实现卷积的?

原创 2017年09月16日 16:30:44

标签: python (http://so.csdn.net/so/search/s.do?q=python&t=blog) / Tensorflow (http://so.csdn.net/so/search/s.do?q=Tensorflow&t=blog) / 机器学习 (http://so.csdn.net/so/search/s.do?q=机器学习&t=blog) / 深度学习 (http://so.csdn.net/so/search/s.do?q=深度学习&t=blog) / 卷积神经网络 (http://so.csdn.net/so/search/s.do?q=卷积神经网络&t=blog)

5683

实验环境：tensorflow版本1.2.0，python2.7

介绍

惯例先展示函数：

```
1 tf.nn.conv2d(input, filter, strides, padding, use_cudnn_on_gpu=None, name=None)
```

除去name参数用以指定该操作的name，与方法有关的一共五个参数：

- input：
  指需要做卷积的输入图像，它要求是一个Tensor，具有[batch, in_height, in_width, in_channels]这样的shape，具体含义是[训练时一个batch的图片数量, 图片高度, 图片宽度, 图像通道数]，注意这是一个4维的Tensor，要求类型为float32和float64其中之一

- filter：
  相当于CNN中的卷积核，它要求是一个Tensor，具有[filter_height, filter_width, in_channels, out_channels]这样的shape，具体含义是[卷积核的高度，卷积核的宽度，图像通道数，卷积核个数]，要求类型与参数input相同，有一个地方需要注意，第三维in_channels，就是参数input的第四维

- strides：卷积时在图像每一维的步长，这是一个一维的向量，长度4

- padding：
  string类型的量，只能是"SAME","VALID"其中之一，这个值决定了不同的卷积方式（后面会介绍）

- use_cudnn_on_gpu：
  bool类型，是否使用cudnn加速，默认为true

结果返回一个Tensor，这个输出，就是我们常说的feature map

xf__mao (http://blog.csd...
+关注
(http://blog.csdn.net/mao_xiao_feng)
码云 未开通 (https://gite
原创 40 粉丝 241 喜欢 6

他的最新文章
更多文章 (http://blog.csdn.net/mao_xiao_feng)
【Tensorflow】tf.nn.atrous_conv2d如何实现空洞卷积? (http://blog.csdn.net/mao_xiao_feng/article/details/78003730)
【Tensorflow】tf.nn.depthwise_conv2d如何实现深度卷积? (http://blog.csdn.net/mao_xiao_feng/article/details/78003476)
【Tensorflow】tf.nn.separable_conv2d如何实现深度可分卷积? (http://blog.csdn.net/mao_xiao_feng/article/details/78002811)
【Tensorflow】tf.nn.depthwise_conv2d如何实现深度卷积? (http://blog.csdn.net/mao_xiao_feng/article/details/77938385)
【Tensorflow】tf.nn.atrous_conv2d如何实现空洞卷积? (http://blog.csdn.net/mao_xiao_feng/article/details/77924003)

博主专栏

如何为Tensorflow训练打包数据和预处理? (http://blog.csdn.net/column/details/15035.html)

实验

那么TensorFlow的卷积具体是怎样实现的呢，用一些例子去解释它：

1.考虑一种最简单的情况，现在有一张3×3单通道的图像（对应的shape：[1，3，3，1]），用一个1×1的卷积核（对应的shape：[1，1，1，1]）去做卷积，最后会得到一张3×3的feature map

2.增加图片的通道数，使用一张3×3五通道的图像（对应的shape：[1，3，3，5]），用一个1×1的卷积核（对应的shape：[1，1，1，1]）去做卷积，仍然是一张3×3的feature map，这就相当于每一个像素点，卷积核都与该像素点的每一个通道做点积

```
input = tf.Variable(tf.random_normal([1,3,3,5]))
filter = tf.Variable(tf.random_normal([1,1,5,1]))

op = tf.nn.conv2d(input, filter, strides=[1, 1, 1, 1], padding='VALID')
```

3.把卷积核扩大，现在用3×3的卷积核做卷积，最后的输出是一个值，相当于情况2的feature map所有像素点的值求和

```
input = tf.Variable(tf.random_normal([1,3,3,5]))
filter = tf.Variable(tf.random_normal([3,3,5,1]))

op = tf.nn.conv2d(input, filter, strides=[1, 1, 1, 1], padding='VALID')
```

4.使用更大的图片将情况2的图片扩大到5×5，仍然是3×3的卷积核，令步长为1，输出3×3的feature map

```
.....
.xxx.
.xxx.
.xxx.
.....
```

5.上面我们一直令参数padding的值为'VALID'，当其为'SAME'时，表示卷积核可以停留在图像边缘，如下，输出5×5的feature map

```
input = tf.Variable(tf.random_normal([1,5,5,5]))
filter = tf.Variable(tf.random_normal([3,3,5,1]))

op = tf.nn.conv2d(input, filter, strides=[1, 1, 1, 1], padding='SAME')
```

```
xxxxx
xxxxx
xxxxx
xxxxx
xxxxx
```

6.如果卷积核有多个

```
input = tf.Variable(tf.random_normal([1,5,5,5]))
filter = tf.Variable(tf.random_normal([3,3,5,7]))

op = tf.nn.conv2d(input, filter, strides=[1, 1, 1, 1], padding='SAME')
```

/details.29
/16035.html)

他的热门文章

【TensorFlow】tf.nn.conv2d是怎样实现卷积的？ (http://blog.csdn.net/mao_xiao_feng/article/details/53444333)
33270

【TensorFlow】tf.nn.softmax_cross_entropy_with_logits的用法 (http://blog.csdn.net/mao_xiao_feng/article/details/53382790)
25670

【TensorFlow】tf.nn.max_pool实现池化操作 (http://blog.csdn.net/mao_xiao_feng/article/details/53453926)
19299

【TensorFlow】tf.concat的用法 (http://blog.csdn.net/mao_xiao_feng/article/details/53366163)
16640

对于随机森林的通俗理解 (http://blog.csdn.net/mao_xiao_feng/article/details/52728164)
14096

此时输出7张5×5的feature map

7.步长不为1的情况，文档里说了对于图片，因为只有两维，通常strides取[1，stride，stride，1]

```
input = tf.Variable(tf.random_normal([1,5,5,5]))
filter = tf.Variable(tf.random_normal([3,3,5,7]))

op = tf.nn.conv2d(input, filter, strides=[1, 2, 2, 1], padding='SAME')
```

此时，输出7张3×3的feature map

```
x.x.x
.....
x.x.x
.....
x.x.x
```

8.如果batch值不为1，同时输入10张图

```
input = tf.Variable(tf.random_normal([10,5,5,5]))
filter = tf.Variable(tf.random_normal([3,3,5,7]))

op = tf.nn.conv2d(input, filter, strides=[1, 2, 2, 1], padding='SAME')
```

每张图，都有7张3×3的feature map，输出的shape就是[10，3，3，7]

代码清单

最后，把程序总结一下：

```
import tensorflow as tf
#case 2
input = tf.Variable(tf.random_normal([1,3,3,5]))
filter = tf.Variable(tf.random_normal([1,1,5,1]))

op2 = tf.nn.conv2d(input, filter, strides=[1, 1, 1, 1], padding='VALID')
#case 3
input = tf.Variable(tf.random_normal([1,3,3,5]))
filter = tf.Variable(tf.random_normal([3,3,5,1]))

op3 = tf.nn.conv2d(input, filter, strides=[1, 1, 1, 1], padding='VALID')
#case 4
input = tf.Variable(tf.random_normal([1,5,5,5]))
filter = tf.Variable(tf.random_normal([3,3,5,1]))

op4 = tf.nn.conv2d(input, filter, strides=[1, 1, 1, 1], padding='VALID')
#case 5
input = tf.Variable(tf.random_normal([1,5,5,5]))
filter = tf.Variable(tf.random_normal([3,3,5,1]))

op5 = tf.nn.conv2d(input, filter, strides=[1, 1, 1, 1], padding='SAME')
#case 6
input = tf.Variable(tf.random_normal([1,5,5,5]))
filter = tf.Variable(tf.random_normal([3,3,5,7]))

op6 = tf.nn.conv2d(input, filter, strides=[1, 1, 1, 1], padding='SAME')
#case 7
input = tf.Variable(tf.random_normal([1,5,5,5]))
filter = tf.Variable(tf.random_normal([3,3,5,7]))

op7 = tf.nn.conv2d(input, filter, strides=[1, 2, 2, 1], padding='SAME')
#case 8
input = tf.Variable(tf.random_normal([10,5,5,5]))
filter = tf.Variable(tf.random_normal([3,3,5,7]))

op8 = tf.nn.conv2d(input, filter, strides=[1, 2, 2, 1], padding='SAME')

init = tf.initialize_all_variables()
with tf.Session() as sess:
    sess.run(init)
    print("case 2")
    print(sess.run(op2))
    print("case 3")
    print(sess.run(op3))
    print("case 4")
    print(sess.run(op4))
    print("case 5")
    print(sess.run(op5))
    print("case 6")
    print(sess.run(op6))
    print("case 7")
    print(sess.run(op7))
    print("case 8")
    print(sess.run(op8))
```

因为是随机初始化，我的结果是这样的：

```
case 2
[[[[-0.64064658]
   [-1.82183945]
   [-2.63191342]]

  [[ 8.05008984]
   [ 1.66023612]
   [ 2.53465152]]

  [[-3.51703644]
   [-5.92647743]
   [ 0.55595356]]]]
case 3
[[[[ 10.53139973]]]]
case 4
[[[[ 10.45460224]
   [  6.23760509]
   [  4.97157574]]

  [[  3.05653667]
   [-11.43907833]
   [ -2.05077457]]

  [[ -7.48340607]
   [ -0.90697062]
   [  3.27171206]]]]
case 5
[[[[  5.30279875]
   [ -2.75329947]
   [  5.62432575]
   [-10.24609661]
   [  0.12603235]]

  [[  0.2113893 ]
   [  1.73748684]
   [ -3.04372549]
   [ -7.2625494 ]
   [-12.76445198]]

  [[ -1.57414591]
   [ -3.39802694]
   [ -6.01582575]
   [ -1.73042905]
   [ -3.07183361]]

  [[  1.41795194]
   [ -2.02815866]
   [-17.08983231]
   [ 11.98958111]
   [  2.44879103]]

  [[  0.29902667]
   [ -3.19712877]
   [ -2.84978414]
   [ -2.71143317]
   [  5.99366283]]]]
case 6
[[[[ 12.02504349   4.35077286   2.67207813   5.77893162   6.98221684
     -0.96858567  -8.1147871 ]
   [ -0.02988982  -2.52141953  15.24755192   6.39476395  -4.36355495
     -3.24515095  -5.55748504]
   [ -2.74448752  -1.62703776  -6.84849405  10.12248802   3.7408421
```

```
    4.71439075   6.13722801]
  [  0.82365227  -1.00546622  -3.29460764   5.12690163  -0.75699937
    -2.60097408  -8.33882809]
  [  0.76171923  -0.86230004  -6.30558443  -5.58426857   2.70478535
     8.98232937  -2.45504045]]

 [[  3.13419819 -13.96483231   0.42031103   2.97559547   6.86646557
    -3.44916964  -0.10199898]
  [ 11.65359879  -5.2145977    4.28352737   2.68335319   3.21993709
    -6.77338028   8.08918095]
  [  0.91533852  -0.31835344  -1.06122255  -9.11237717   5.05267143
     5.6913228   -5.23855162]
  [ -0.58775592  -5.03531456  14.70254898   9.78966522 -11.00562763
    -4.08925819  -3.29650426]
  [ -2.23447251  -0.18028721  -4.80610704  11.2093544   -6.72472
    -2.67547607   1.68422937]]

 [[ -3.40548897  -9.70355129  -1.05640507  -2.55293012  -2.78455877
   -15.05377483  -4.16571808]
  [ 13.66925812   2.87588191   8.29056358   6.71941566   2.56558466
    10.10329056   2.88392687]
  [ -6.30473804  -3.3073864   12.43273926  -0.66088223   2.94875336
     0.06056046  -2.78857946]
  [ -7.14735603  -1.44281793   3.3629775   -7.87305021   2.00383091
    -2.50426936  -6.93097973]
  [ -3.15817571   1.85821593   0.60049552  -0.43315536  -4.43284273
     0.54264796   1.54882073]]

 [[  2.19440389  -0.21308756  -4.35629082  -3.62100363  -0.08513772
    -0.80940366   7.57606506]
  [ -2.65713739   0.45524287 -16.04298019  -5.19629049  -0.63200498
     1.13256514  -6.70045137]
  [  8.00792599   4.09538221  -6.16250181   8.35843849  -4.25959206
    -1.5945878   -7.60996151]
  [  8.56787586   5.85663748  -4.38656425   0.12728286  -6.53928804
     2.3200655    9.47253895]
  [ -6.62967777   2.88872099  -2.76913023  -0.86287498  -1.4262073
    -6.59967232   5.97229099]]

 [[ -3.59423327   4.60458899  -5.08300591   1.32078576   3.27156973
     0.5302844   -5.27635145]
  [ -0.87793881   1.79624665   1.66793108  -4.70763969  -2.87593603
    -1.26820421  -7.72825718]
  [ -1.49699068  -3.40959787  -1.21225107  -1.11641395  -8.50123024
    -0.59399474   3.18010235]
  [ -4.4249506   -0.73349547  -1.49064219  -6.09967899   5.18624878
    -3.80284953  -0.55285597]
  [ -1.42934585   2.76053572  -5.19795799   0.83952439  -0.15203482
     0.28564462   2.66513705]]]]
case 7
[[[[  2.66223097   2.64498258  -2.93302107   3.50935125   4.62247562
     2.04241085  -2.65325522]
  [ -0.03272867  -1.00103927  -4.3691597    2.16724801   7.75251007
    -4.6788125   -0.89318085]
  [  4.74175072  -0.80443329  -1.02710629  -6.68772554   4.57605314
    -3.72993755   4.79951382]]

 [[  5.249547    8.92288399   7.10703182  -9.10498428  -7.43814278
    -8.69616318   1.78862095]
  [  7.53669024 -14.52316284  -2.55870199  -1.11976743   3.81035042
     2.45559502  -2.35436153]
  [  3.93275881   5.11939669  -4.7114296  -11.96386623   2.11866689
```

```
    0.57433248  -7.19815397]]

  [[  0.25111672   1.40801668   1.28818977  -2.64093828   0.98182392
     3.69512987   4.78833389]
   [  0.30391204 -10.26406097   6.05877018  -6.04775047   8.95922089
     0.80235004  -5.4520669 ]
   [ -7.24697018  -2.33498096 -10.20039558  -1.24307609   3.99351597
    -8.1029129    2.44411373]]]]
case 8
[[[[ -6.84037447e+00   1.33321762e-01  -5.09891272e+00   5.55682087e+00
     8.22002888e+00  -4.94586229e-02   4.19012117e+00]
   [  6.79884481e+00   1.21652853e+00  -5.69557810e+00  -1.33555794e+00
     3.24849486e-01   4.88868570e+00  -3.90220714e+00]
   [ -3.53190374e+00  -4.11765718e+00   4.54340839e+00   1.85549557e+00
    -3.38682461e+00   2.62719369e+00  -4.98658371e+00]]

  [[ -9.86354351e+00  -6.76713943e+00   3.62617874e+00  -6.16720629e+00
     1.96754158e+00  -4.54203081e+00  -1.37485743e+00]
   [ -1.76783955e+00   2.35163045e+00  -2.21175838e+00   3.83091879e+00
     3.16964531e+00  -7.58307219e+00   4.71943617e+00]
   [  1.20776439e+00   4.86006308e+00   1.04233503e+01  -7.82327271e+00
     5.39195156e+00  -6.31672382e+00   1.35577369e+00]]

  [[ -3.65947580e+00  -1.98961139e+00   7.53771305e+00   2.79224634e-01
    -2.90050888e+00  -3.57466817e+00  -6.33232594e-01]
   [  5.89931488e-01   2.83219159e-01  -1.65850735e+00  -6.45545387e+00
    -1.17044592e+00   1.40343285e+00   5.74970901e-01]
   [ -8.58810043e+00  -1.25172977e+01   6.84177876e-01   3.80004168e+00
    -1.54420209e+00  -3.32161427e+00  -1.05423713e+00]]]


 [[[ -4.82677078e+00   3.11167526e+00  -4.32694483e+00  -4.77198696e+00
     2.32186103e+00   1.65402293e-01  -5.32707453e+00]
   [  3.91779566e+00   6.27949667e+00   2.32975650e+00  -1.06336937e+01
     4.44044876e+00   8.08288479e+00  -5.83346319e+00]
   [ -2.82141399e+00  -9.16103745e+00   6.98908520e+00  -5.66505909e+00
    -2.11039782e+00   2.27499461e+00  -5.74120235e+00]]

  [[  6.71680808e-01  -4.01104212e+00  -4.61760712e+00   1.02667952e+01
    -8.21200657e+00  -8.57054043e+00   1.71461976e+00]
   [  2.40794683e+00  -2.63071585e+00   9.68963623e+00  -4.51778412e+00
    -3.91073084e+00  -5.91874409e+00   9.96273613e+00]
   [  2.67705870e+00   2.85607010e-01   2.45853162e+00   4.44810390e+00
    -2.11300468e+00  -5.77583075e+00   2.83322239e+00]]

  [[ -8.21949577e+00  -7.57754421e+00   3.93484974e+00   2.26189137e+00
    -3.49395227e+00  -6.40283823e+00  -6.00450039e-01]
   [  2.95964479e-02  -1.19976890e+00   5.38537979e+00   4.62369967e+00
     3.89780998e+00  -6.36872959e+00   7.12107182e+00]
   [ -8.85006547e-01   1.92706418e+00   3.26668215e+00   2.03566647e+00
     1.44209075e+00  -6.48463774e+00  -8.33671093e-02]]]


 [[[ -2.64583921e+00   3.86011934e+00   4.18198538e+00   3.50338411e+00
     6.35944796e+00  -4.28423309e+00   4.87355423e+00]
   [  4.42271233e+00   3.92883778e+00  -5.59371090e+00   4.98251200e+00
    -3.45068884e+00   2.91921115e+00   1.03779554e+00]
   [  1.36162388e+00  -1.06808968e+01  -3.92534947e+00   1.85111761e-01
    -4.87255526e+00   1.66666222e+01  -1.04918976e+01]]

  [[ -4.34632540e+00   1.74614882e+00  -2.89012527e+00  -8.74067783e+00
     5.06610107e+00   1.24989772e+00  -3.06433105e+00]
```

```
 [ 2.49973416e+00  2.14041996e+00 -4.71008825e+00  7.39326143e+00
   3.94770741e+00  8.23049164e+00 -1.67046225e+00]
 [-2.94665837e+00 -4.58543825e+00  7.21219683e+00  1.09780006e+01
   5.17258358e+00  7.90257788e+00 -2.13929534e+00]]

[[ 4.20402241e+00 -2.98926830e+00 -3.89006615e-01 -8.16001511e+00
  -2.38355541e+00  1.42584383e+00 -5.46632290e+00]
 [ 5.52395058e+00  5.09255171e+00 -1.08742390e+01 -4.96262169e+00
  -1.35298109e+00  3.65663052e-01 -3.40589857e+00]
 [-6.95647061e-01 -4.12855625e+00  2.66609401e-01 -9.39565372e+00
  -3.85058141e+00  2.51248240e-01 -5.77149725e+00]]]


[[[ 1.22103825e+01  5.72040796e+00 -3.56989503e+00 -1.02248180e+00
  -5.20942688e-01  7.15008640e+00  3.43482435e-01]
 [ 6.01409674e+00 -1.59511256e+00 -6.48080063e+00 -1.82889538e+01
  -1.03537569e+01 -1.48270035e+01 -5.26662111e+00]
 [ 5.51758146e+00 -2.91831636e+00  3.75461340e-01 -9.23893452e-02
  -9.22101116e+00  7.16952372e+00 -6.86479330e-01]]

[[-3.03645611e+00  6.68620300e+00 -3.31973934e+00 -4.91346550e+00
   9.20719814e+00 -2.55552864e+00 -2.16087699e-02]
 [-3.02986956e+00 -1.29726543e+01  1.53023469e+00 -8.19733238e+00
   5.68085670e+00 -1.72856820e+00 -4.69369221e+00]
 [-6.67176056e+00  8.76355553e+00  2.18996063e-01 -4.38777208e+00
  -6.35764122e-01 -1.37812555e+00 -4.41474581e+00]]

[[ 2.25345469e+00  1.02142305e+01 -1.71714854e+00 -5.29060185e-01
   2.27982092e+00 -8.75302982e+00  7.13998675e-02]
 [-6.67547846e+00  3.67722750e+00 -3.44172812e+00  5.69674826e+00
  -2.28723526e+00  5.92991543e+00  5.53608060e-01]
 [-1.01174891e-01 -2.73731589e+00 -4.06187654e-01  6.54158068e+00
   2.59603882e+00  2.99202776e+00 -2.22350287e+00]]]


[[[-1.81271315e+00  2.47674489e+00 -2.90284491e+00  1.34291325e+01
   7.69864845e+00 -1.27134466e+00  3.02233839e+00]
 [-2.08135307e-01  1.03206539e+00  1.90775347e+00  9.01517391e+00
  -3.52140331e+00  9.05393791e+00 -9.12732124e-01]
 [ 1.12128162e+00  5.98179293e+00 -2.27206993e+00 -5.21281779e-01
   6.20835352e+00  3.73474598e+00  1.18961644e+00]]

[[ 3.17242837e+00 -6.00571585e+00  2.37661076e+00 -5.64483738e+00
  -6.45412731e+00  8.75251675e+00  7.33790398e-02]
 [ 3.08957529e+00 -1.06855690e-01 -5.16810894e-01 -9.41085911e+00
   8.23878098e+00  6.79738426e+00 -1.23478663e+00]
 [-9.20640087e+00 -6.82801771e+00 -5.96975613e+00  7.61030674e-01
  -4.35995817e+00 -3.54818010e+00 -2.56281614e+00]]

[[ 4.69872713e-01  8.36402321e+00  5.37103415e-01 -1.68033957e-01
  -3.21731424e+00 -7.34270859e+00 -3.14253521e+00]
 [ 6.69656086e+00 -5.27954197e+00 -8.57314682e+00  4.84328842e+00
  -2.96387672e+00  2.47114658e+00  2.85376692e+00]
 [-7.86032295e+00 -7.18845367e+00 -3.27161223e-01  9.27330971e+00
  -6.14093494e+00 -4.49041557e+00  3.47160912e+00]]]


[[[-1.89188433e+00  5.43082857e+00  6.04252160e-01  6.92894220e+00
   8.59178162e+00  1.02003086e+00  5.31300211e+00]
 [-8.97491455e-01  6.52438164e+00 -4.43710327e+00  7.10509634e+00
   8.84234428e+00  3.08552694e+00  2.78152227e+00]
 [-9.40537453e-02  2.34666920e+00 -5.57496691e+00 -8.62346458e+00
```

```
    -1.32807600e+00  -8.12027454e-02  -9.00946975e-01]]

  [[ -3.53673506e+00   8.93675327e+00   3.27456236e-01  -3.41519475e+00
     7.69804525e+00  -5.18698692e+00  -3.96991730e+00]
   [  1.99988627e+00  -9.16149998e+00  -7.49944544e+00   5.02162695e-01
     3.57059622e+00   9.17566013e+00  -1.77589107e+00]
   [ -1.18147678e+01  -7.68992901e+00   1.88449645e+00   2.77643538e+00
    -1.11342735e+01  -3.12916255e+00  -3.34161663e+00]]

  [[ -3.62668943e+00  -3.10993242e+00   3.60834384e+00   4.69678783e+00
    -1.73794723e+00  -1.27035933e+01   3.65882218e-01]
   [ -8.97550106e+00  -4.33533072e-01   4.41743970e-01  -5.83433771e+00
    -4.85818958e+00   9.56629372e+00   3.56375504e+00]
   [ -6.87092066e+00   1.96412420e+00   5.14182663e+00  -8.97769547e+00
     3.61136627e+00   5.91387987e-01  -2.95224571e+00]]]


 [[[ -1.11802626e+00   3.24175072e+00   5.94067669e+00   9.29727936e+00
     9.28199863e+00  -4.80889034e+00   6.96202660e+00]
   [  7.23959684e+00   3.11182523e+00   1.84116721e+00   5.12095928e-01
    -7.65049171e+00  -4.05325556e+00   5.38544941e+00]
   [  4.66621685e+00  -1.61665392e+00   9.76448345e+00   2.38519001e+00
    -2.06760812e+00  -6.03633642e-01   3.66192675e+00]]

  [[  1.52149725e+00  -1.84441996e+00   4.87877655e+00   2.96750760e+00
     2.37311172e+00  -2.98487616e+00   9.98114228e-01]
   [  9.20035839e+00   5.24396753e+00  -2.57312679e+00  -7.26040459e+00
    -1.17509928e+01   6.85688591e+00   3.37383580e+00]
   [  6.17629957e+00  -5.15294194e-01  -1.64212489e+00  -5.70274448e+00
    -2.36294913e+00   2.60432816e+00   2.63957453e+00]]

  [[  7.91168213e-03  -1.15018034e+00   3.05471039e+00   3.31086922e+00
     5.35744762e+00   1.14832592e+00   9.56500292e-01]
   [  4.86464739e+00   5.37348413e+00   1.42920148e+00   1.62809372e+00
     2.61656570e+00   7.88479471e+00  -6.09324336e-01]
   [  7.71319962e+00  -1.73930550e+00  -2.99925613e+00  -3.14857435e+00
     3.19194889e+00   1.70928288e+00   4.90955710e-01]]]


 [[[ -1.79046512e+00   8.54369068e+00   1.85044312e+00  -9.88471413e+00
     9.52995300e-01  -1.34820042e+01  -1.13713551e+01]
   [  8.37582207e+00   6.64692163e+00  -3.22429276e+00   3.37997460e+00
     3.91468263e+00   6.96061993e+00  -1.18029404e+00]
   [ -2.13278866e+00   4.36152029e+00  -4.14593410e+00  -2.15160155e+00
     1.90767622e+00   1.16321917e+01  -3.72644544e+00]]

  [[ -5.03508925e-01  -6.33426476e+00  -1.06393566e+01  -6.49301624e+00
    -6.31036520e+00   3.13485146e+00  -5.77433109e-01]
   [  7.41444230e-01  -4.87326956e+00  -5.98253345e+00  -9.14121056e+00
    -8.64077091e-01   2.06696177e+00  -7.59688473e+00]
   [  1.38767815e+00   1.84418947e-01   5.72539902e+00  -2.07557893e+00
     9.70911503e-01   1.16765432e+01  -1.40111232e+00]]

  [[ -1.21869087e+00   2.44499159e+00  -1.65706706e+00  -6.19807529e+00
    -5.56950712e+00  -1.72372568e+00   3.62687564e+00]
   [  2.23708963e+00  -2.87862611e+00   2.71666467e-01   4.35115099e+00
    -8.85548592e-01   2.91860628e+00   8.10848951e-01]
   [ -5.33635712e+00   7.15072036e-01   5.21240902e+00  -3.11152220e+00
     2.01623154e+00  -2.28398323e-01  -3.23233747e+00]]]


  [[[ 3.77991509e+00   5.53513861e+00  -1.82022047e+00   4.22430277e+00
```

```
    5.60331726e+00 -4.28308249e+00  4.54524136e+00]
 [-5.30983162e+00 -3.45605731e+00  2.69374561e+00 -6.16836596e+00
   -9.18601036e+00 -1.58697796e+00 -5.73809910e+00]
 [ 2.18868661e+00  6.96338892e-01  1.88057957e+01 -4.21353197e+00
    1.20818818e+00  2.85108542e+00  6.62180042e+00]]

[[ 1.01285219e+01 -4.86819077e+00 -2.45067930e+00  7.50106812e-01
    4.37201977e+00  4.78472042e+00  1.19103444e+00]
 [-3.26395583e+00 -5.59358537e-01  1.52001972e+01 -5.93994498e-01
   -1.49040818e+00 -7.02547312e+00 -1.29268813e+00]
 [ 1.02763653e+01  1.31108007e+01 -2.91605043e+00 -1.37688947e+00
    3.33029580e+00  1.96966705e+01  2.55259371e+00]]

[[ 4.58397627e+00 -3.19160700e+00 -6.51985502e+00  1.02908373e+01
   -4.17618275e+00 -9.69347239e-01  7.46259832e+00]
 [ 6.09876537e+00  1.33044279e+00  5.04027081e+00 -6.87740147e-01
    4.14770365e+00 -2.26751328e-01  1.54876924e+00]
 [ 2.70127630e+00 -1.59834003e+00 -1.82587504e+00 -5.92888784e+00
   -5.65038967e+00 -6.46078014e+00 -1.80765367e+00]]]


[[[-1.57899165e+00  3.39969063e+00  1.02308102e+01 -7.77082300e+00
   -8.02129686e-01 -3.67387819e+00 -1.37204361e+00]
 [ 3.93093729e+00  6.17498016e+00 -1.41695750e+00 -1.26903206e-01
    2.18985319e+00  5.83657503e-01  7.39725351e-01]
 [ 5.53898287e+00  2.22283316e+00 -1.10478985e+00  2.68644023e+00
   -2.59913635e+00  3.74231935e+00  4.85016155e+00]]

[[ 4.05368614e+00 -3.74058294e+00  7.32348633e+00 -1.17656231e+00
    3.71810269e+00 -1.63957381e+00  9.91670132e-01]
 [-1.29317007e+01  1.12296543e+01 -1.13844347e+01 -7.13933802e+00
   -8.65884399e+00 -5.56065178e+00 -1.46718264e+00]
 [-8.08718109e+00 -1.98826480e+00 -4.07488203e+00  2.06440473e+00
    1.13524094e+01  5.68703651e+00 -2.18706942e+00]]

[[ 1.51166654e+00 -6.84034204e+00  9.33474350e+00 -4.80931902e+00
   -6.24172688e-02 -4.21381521e+00 -5.73313046e+00]
 [-1.35943902e+00  5.27799511e+00 -3.77813816e+00  6.88291168e+00
    4.35068893e+00 -1.02540245e+01  8.86861205e-01]
 [-4.49999619e+00 -2.97630525e+00 -6.18604183e-01 -2.49702692e+00
   -6.76169348e+00 -2.55930996e+00 -2.71291423e+00]]]]
```

pkokocl (/pkokocl) 2017-11-21 22:46非常不错，终于弄懂了 3楼
(/pkokocl)
回复

查看 6 条热评

【TensorFlow】tf.nn.conv2d是怎样实现卷积的？ (http://blog.csdn.net/guvcolie/article/...

文章出处：http://blog.csdn.net/mao_xiao_feng/article/details/53444333 tf.nn.conv2d是TensorFlow里面实...

guvcolie (http://blog.csdn.net/guvcolie) 2017年08月06日 18:17 253

【TensorFlow】tf.nn.conv2d_transpose是怎样实现反卷积的？ (http://blog.csdn.net/ma...

三个月没更新了啊，回来更一发～～ csdn上主要讲一些coding过程中遇到的函数，问题，解决方案。偏实践另外，如果你想看一些理论方面的东西，欢迎加我的知乎 csdn私信几乎不看，有问题交流可以发邮...

mao_xiao_feng (http://blog.csdn.net/mao_xiao_feng) 2017年05月12日 15:08 7225

【TensorFlow】tf.nn.conv2d是怎样实现卷积的？ (http://blog.csdn.net/daodaipsrenshe...

本文转自 tf.nn.conv2d是TensorFlow里面实现卷积的函数，参考文档对它的介绍并不是很详细，实际上这是搭建卷积神经网络比较核心的一个方法，非常重要 tf.nn....

daodaipsrensheng (http://blog.csdn.net/daodaipsrensheng) 2017年11月20日 09:19 99

【TensorFlow】tf.nn.conv2d是怎样实现卷积的？ (http://blog.csdn.net/mao_xiao_feng/...

tf.nn.conv2d是TensorFlow里面实现卷积的函数，参考文档对它的介绍并不是很详细，实际上这是搭建卷积神经网络比较核心的一个方法，非常重要 tf.nn.conv2d(input, fi...

mao_xiao_feng (http://blog.csdn.net/mao_xiao_feng) 2016年12月03日 17:17 33445

tensorflow:tf.nn.conv2d是怎样实现卷积的？ (http://blog.csdn.net/xbcReal/article/detail...

tf.nn.conv2d是TensorFlow里面实现卷积的函数，参考文档对它的介绍并不是很详细，实际上这是搭建卷积神经网络比较核心的一个方法，非常重要 tf.nn.conv2d(input, fi...

xbcReal (http://blog.csdn.net/xbcReal) 2017年05月13日 14:58 647

【TensorFlow】tf.nn.conv2d是怎样实现卷积的？有1*1（1×1）卷积介绍 (http://blog.cs...

版权声明：本文为博主原创文章，转载请注明出处。文章太好了，怕丢!，转自xf_mao的： http://blog.csdn.net/mao_xiao_feng/article/details/53...

yewei11 (http://blog.csdn.net/yewei11) 2017年08月03日 10:24 435

[TensorFlow 学习笔记-04]卷积函数之tf.nn.conv2d (http://blog.csdn.net/caicaiatnbu/arti…

[版权说明] TensorFlow 学习笔记参考： 李嘉璇 著 TensorFlow技术解析与实战黄文坚 唐源 著 TensorFlow实战郑泽宇 顾思宇 著 TensorFlow实战Goog...

caicaiatnbu (http://blog.csdn.net/caicaiatnbu) 2017年05月28日 13:08 1852

TensorFlow实例(5.2)--MNIST手写数字进阶算法(卷积神经网络CNN) 之 卷积tf.nn.conv2…

本文是MNIST手写数字进阶算法(卷积神经网络CNN)的扩展篇主要通过数据演算，理解卷积函数tf.nn.conv2d...

carmelcarmen (http://blog.csdn.net/carmelcarmen) 2017年12月30日 13:36 19

tensorflow conv2d padding,tf图像卷积边缘扩展问题 (http://blog.csdn.net/wkk15903468…

初学tensorflow的conv2d的时候，一般书上会说conv2d的扩展可以选择两种，SAME和VALID。这两种要么导致图像变小（valid），要么导致边缘变黑（same），因为边缘只补0。曾一...

wkk15903468980 (http://blog.csdn.net/wkk15903468980) 2017年07月12日 16:27 695

tensorflow:tf.nn.conv2d 参数详解 (http://blog.csdn.net/shenhuaifeng/article/details/760…

tensorflow中的卷积运算参数详解.

shenhuaifeng (http://blog.csdn.net/shenhuaifeng) 2017年07月25日 15:56 332

Tensorflow系列： tf.nn.conv2d (http://blog.csdn.net/YoungDreamNJU/article/details/539…

TensorFlow的CNN代码中有tf.nn.conv2d(X, W1, strides=[1, 1, 1, 1], padding='SAME')这样一句，本文介绍tf.nn.conv2d的用法：...

YoungDreamNJU (http://blog.csdn.net/YoungDreamNJU) 2017年01月03日 14:06 872

TensorFlow 从入门到精通（八）： TensorFlow tf.nn.conv2d 一路追查 (http://blog.csdn…

读者可能还记得本系列博客（二）和（六）中 tf.nn 模块，其中最关心的是 conv2d 这个函数。首先将博客（二） MNIST 例程中 convolutional.py 关键源码列出： d...

kkk584520 (http://blog.csdn.net/kkk584520) 2016年06月08日 13:13 29041

利用tf.nn.conv2d_transpose实现图片分辨率按指定倍数扩展 (http://blog.csdn.net/wm6…

我有输入尺寸为 64x64x3 的图片，即高64， 宽64， 通道数3 我想通过transposed convolution 来使图片的高宽放大到2倍，即 128x128import tensorf...

wm6274 (http://blog.csdn.net/wm6274) 2017年06月16日 11:26 165

tf.nn.conv2d_transpose 实例 及 解析 (http://blog.csdn.net/guotong1988/article/details/5…

这个程序完全按照 https://github.com/tensorflow/tensorflow/blob/r0.11/tensorflow/core/kernels/conv_grad_ops.c...

guotong1988 (http://blog.csdn.net/guotong1988) 2016年10月28日 12:21 6725

tf.nn.atrous_conv2d(value, filters, rate, padding, name=None) {#atrous_conv2d} (http://…

tf.nn.atrous_conv2d(value, filters, rate, padding, name=None) {#atrous_conv2d} Atrous convolution (...

liyaoqing (http://blog.csdn.net/liyaoqing) 2017年01月20日 17:09 988

【TensorFlow】tf.nn.max_pool实现卷积操作 (http://blog.csdn.net/mao_xiao_feng/articl…

max pooling是CNN当中的最大值池化操作，其实用法和卷积很类似有些地方可以从卷积去参考【TensorFlow】tf.nn.conv2d是怎样实现卷积的？ tf.nn.max_pool(...

mao_xiao_feng (http://blog.csdn.net/mao_xiao_feng) 2016年12月04日 14:28 19399

【TensorFlow】tf.nn.max_pool实现池化操作 (http://blog.csdn.net/zj360202/article/detai…

原文地址：http://blog.csdn.net/mao_xiao_feng/article/details/53453926 max pooling是CNN当中的最大值池化操作，其实用法和卷积很...

zj360202 (http://blog.csdn.net/zj360202) 2017年04月19日 18:34 1185

TensorFlow学习笔记（四）：tf.nn.max_pool如何实现池化操作？ (http://blog.csdn.net/…

max pooling是CNN当中的最大值池化操作，其实用法和卷积很类似 tf.nn.max_pool(value,ksize, strides, padding, name=None) 参数是四...

zeuseign (http://blog.csdn.net/zeuseign) 2017年05月25日 19:26 378

TensorFlow 用 tf.nn.max_pool 实现最大池化操作 (http://blog.csdn.net/lyc_yongcai/artic…

max pooling是CNN当中的最大值池化操作，其实用法和卷积很类似 tf.nn.max_pool(value, ksize, strides, padding, name=None) ...

lyc_yongcai (http://blog.csdn.net/lyc_yongcai) 2017年06月14日 14:57 245

TensorFlow学习---tf.nn.max_pool实现池化操作 (http://blog.csdn.net/huahuazhu/article/…

max pooling是CNN当中的最大值池化操作，其实用法和卷积很类似有些地方可以从卷积去参考【TensorFlow】tf.nn.conv2d是怎样实现卷积的？ tf.nn.max_po...

huahuazhu (http://blog.csdn.net/huahuazhu) 2017年06月21日 15:33 189