最大值池化层梯度反向传播

最大值池化层梯度反向传播类似于ReLU激活层梯度反向传播。

ReLU是求max(0, x)的偏导数，最大池化层是求f = max(a, b, c, d)的偏导数，其中a, b, c, d是输入2*2窗口的数据。对于该函数的偏导数，对a, b, c, d种最大值的梯度为1， 其他为0。

代码实现如下：

首先定义一些超参数并随机生成一些必要的数据，这些数据能在正向计算中获得。

```
import numpy as np
# 设置超参数并随机生成一些必要的数据
(batch, in_height, in_width, in_depth) = (8, 32, 48, 16)
filter_size = 2
filter_size2 = filter_size**2
stride = 2
out_height = (in_height - filter_size) // stride + 1
out_width = (in_width - filter_size) // stride + 1
out_depth = in_depth
 out_size = out_height * out_width

 dout_data = np.random.randn(batch, out_height, out_width, out_depth) # 是上次梯度反向传播得到的，是输入梯度
 matric_data_max_pos = np.random.randn(out_size*in_depth*batch, filter_size)
 # matric_data_max_pos 记录输入特征图中最大值的位置，尺寸是变换后的大矩阵，每行只有局部窗口4个元素， 从正向传播中得到
 matric_data_max_pos = matric_data_max_pos > 0 # 模拟最大值位置，一个元素为bool类型的矩阵
 matric_data_not_max_pos = ~matric_data_max_pos # 对matric_data_max_pos取反，即得到非最大值的位置，因为非最大值位置处的梯度为0
 din_data = np.zeros((batch, in_height, in_width, in_depth), dtype=np.float64) # 输入特征图梯度，需要通过计算得到
```

然后遍历每一个特征图的局部窗口， 将din_data最大值位置处的梯度赋值为dout_data，将非最大值位置的梯度赋值为0

```
# 将din_data最大位置处的梯度赋值为dout_data，将非最大值位置处的梯度赋值为0
height_ef = in_height - filter_size + 1
width_ef = in_width - filter_size + 1
for i_batch in range(batch):
  i_batch_size = i_batch * out_size * in_depth
  for i_h_out, i_height in zip(range(out_height),range(0, height_ef, stride)):
   i_height_size = i_batch_size + i_h_out * out_width * in_depth
```

```
8   for i_w_dout, i_w, i_width in zip(range(out_width), range(0, in_width*ou
t_width, in_depth), range(0, width_ef, stride)): # 遍历每一个窗口
9   md = matric_data_not_max_pos[i_height_size + i_w:i_height_size + i_w + i
n_depth, :]
10  # 获得局部窗口中（包含整个深度维度）非最大值的位置的md，它是一个二维矩阵
11  din = din_data[i_batch, i_height:i_height + filter_size, i_width:i_widt
h + filter_size, :]
12  # 获得输入梯度局部窗口中的数据（整个深度维度）din，是一个三维矩阵
13  dout = dout_data[i_batch, i_h_out, i_w_dout, :]
14  # 获得输出梯度局部窗口数据（整个深度维度）dout，是一个一维矩阵
15  # 对局部窗口每个深度列进行赋值
16  for j in range(filter_size):
17  for i in range(filter_size):
18  din[i, j, :] = dout[:] # din深度列赋值dout的值
19  din[i, j, :][md[:, i*filter_size + j]] = 0 # 对非最大值位置的梯度设置为0
```