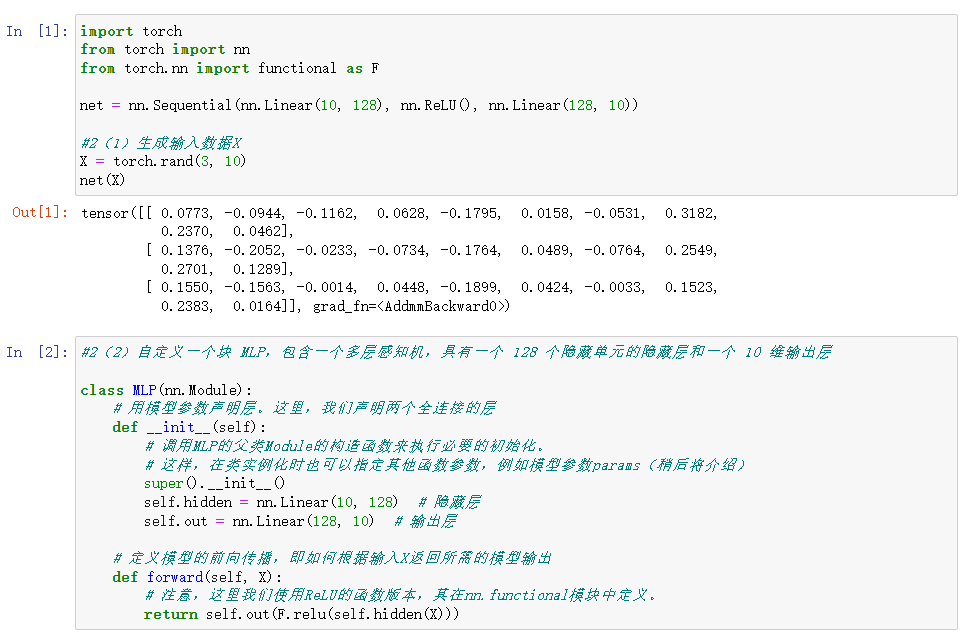
**深度学习第三次作业**

**胡雅娴-2020311448**

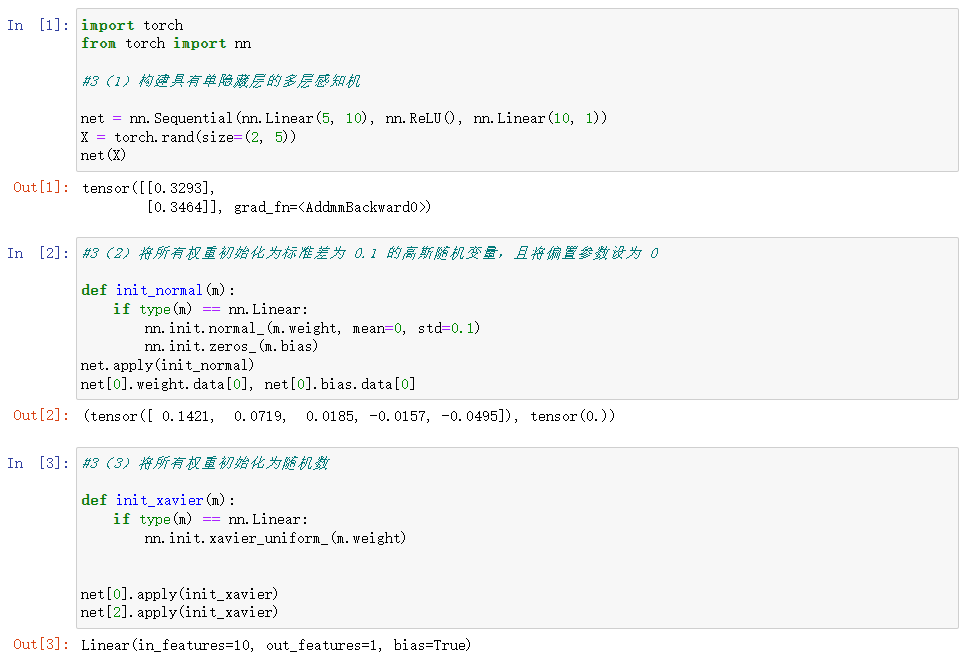
2.代码即运行结果截图如下：

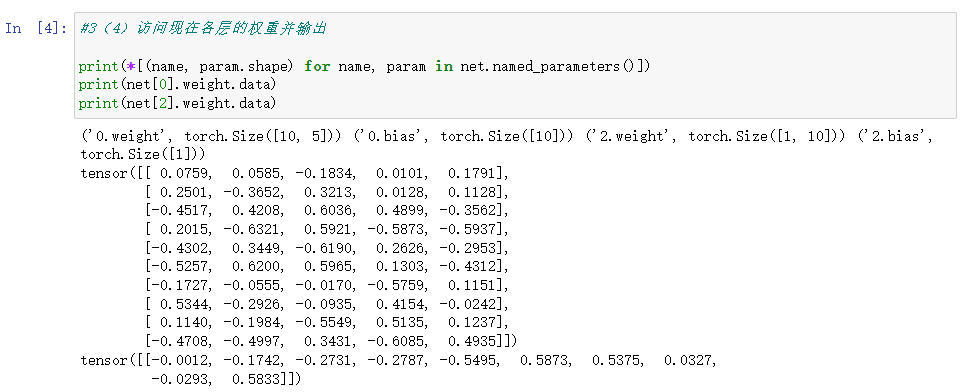


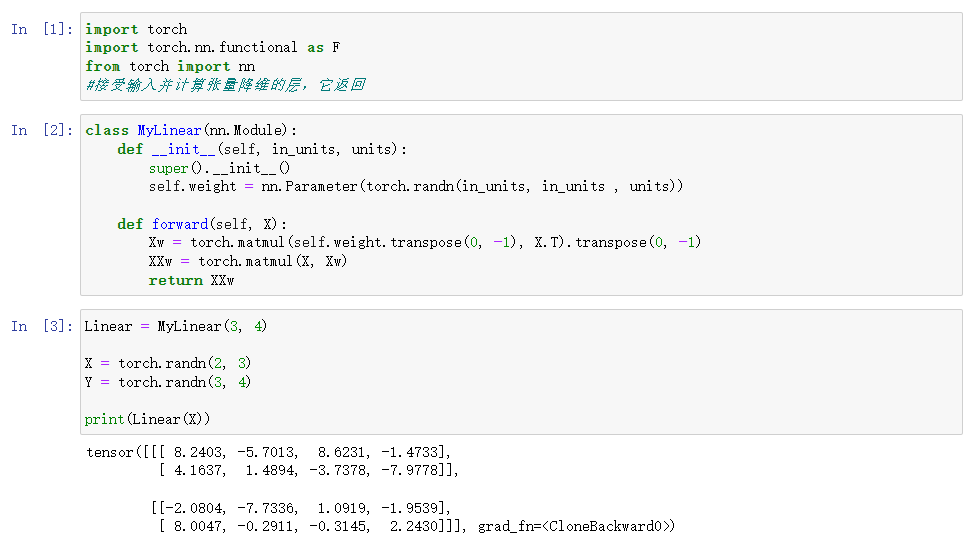




3.运行结果如下：





4. 运行结果如下：

5.

（1）无填充、步幅为1时，，

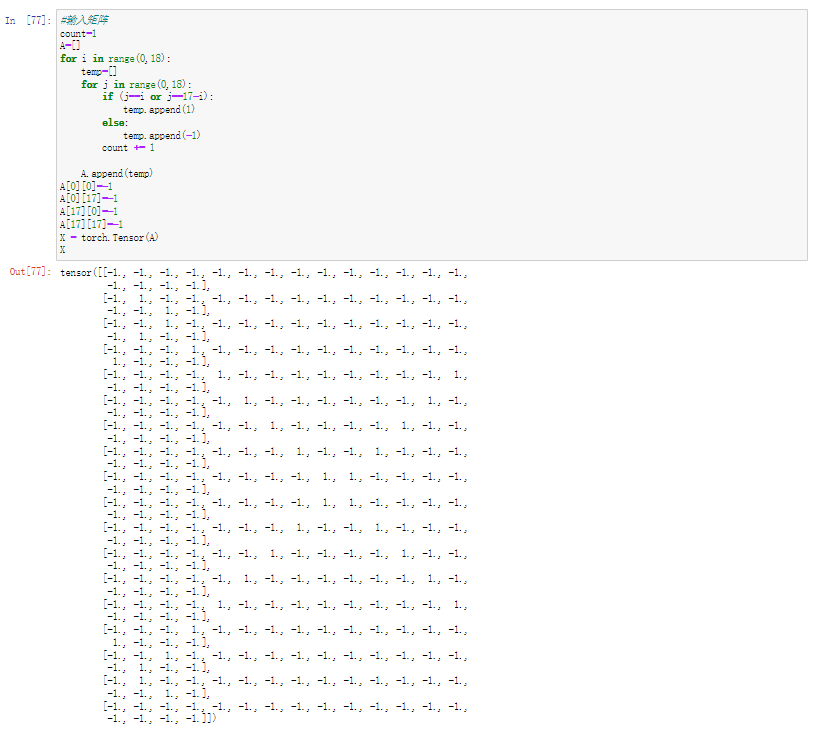
故需要层。答案为111。

（2）计算步骤如下：

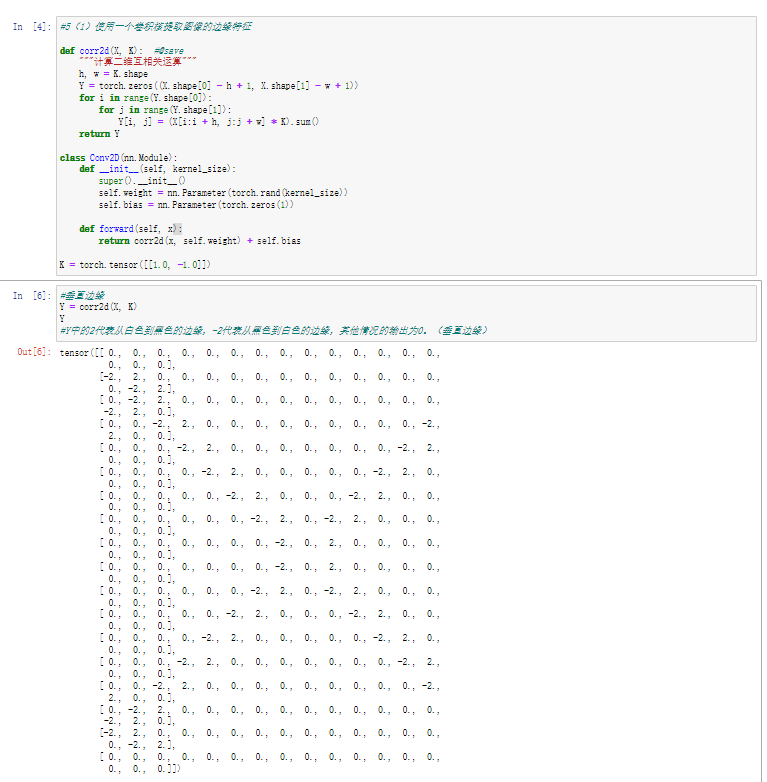
**

故形状是34\*50。

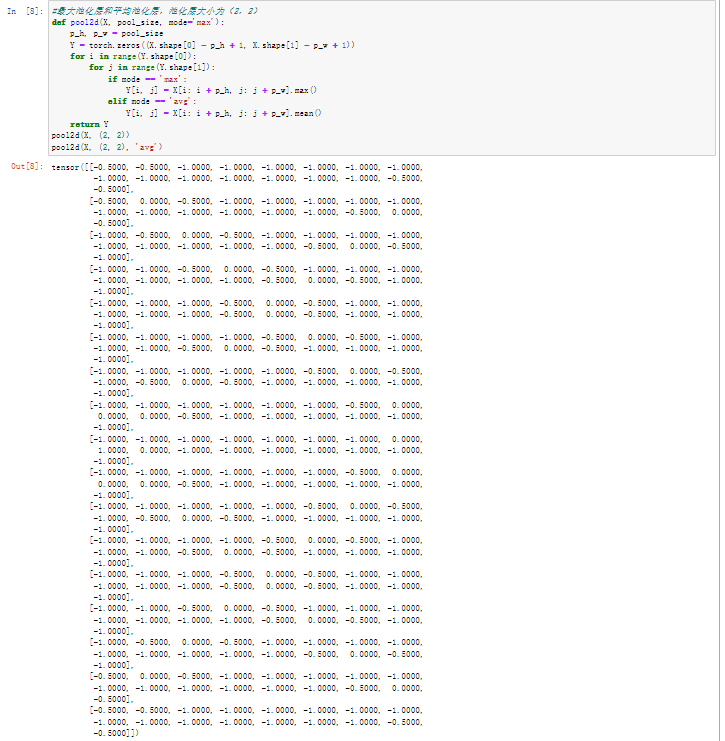
6.刻画图像如下：

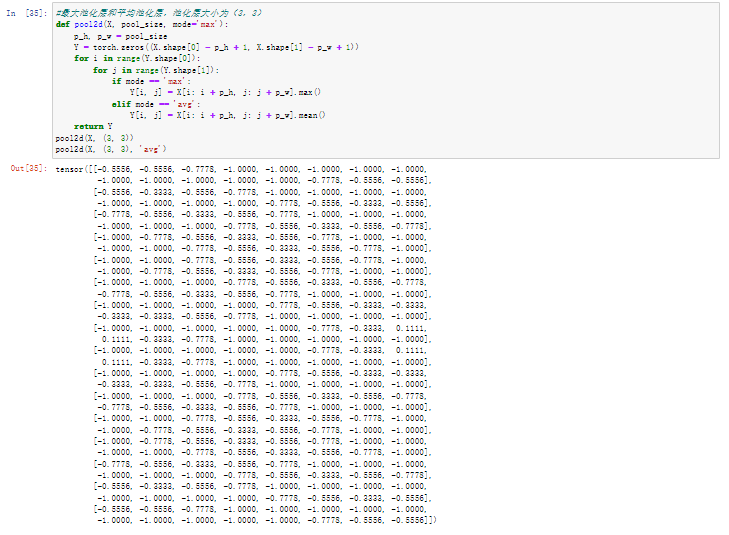


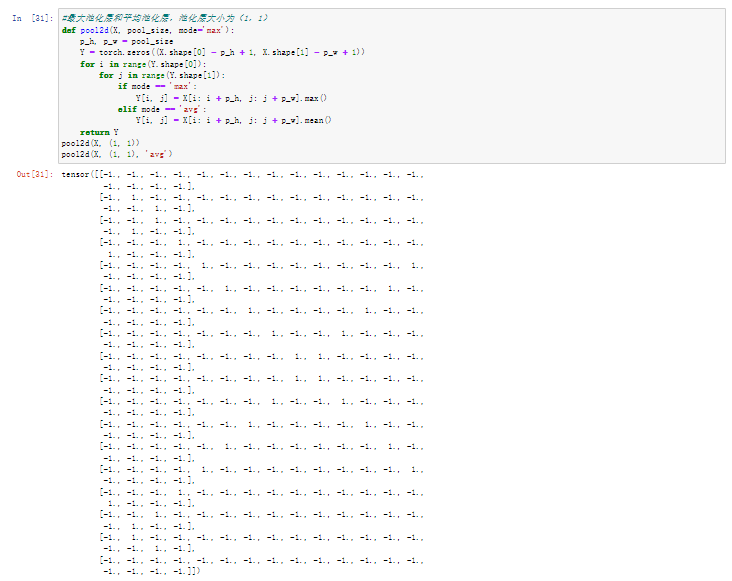
使用一个卷积核提取图像的边缘特征



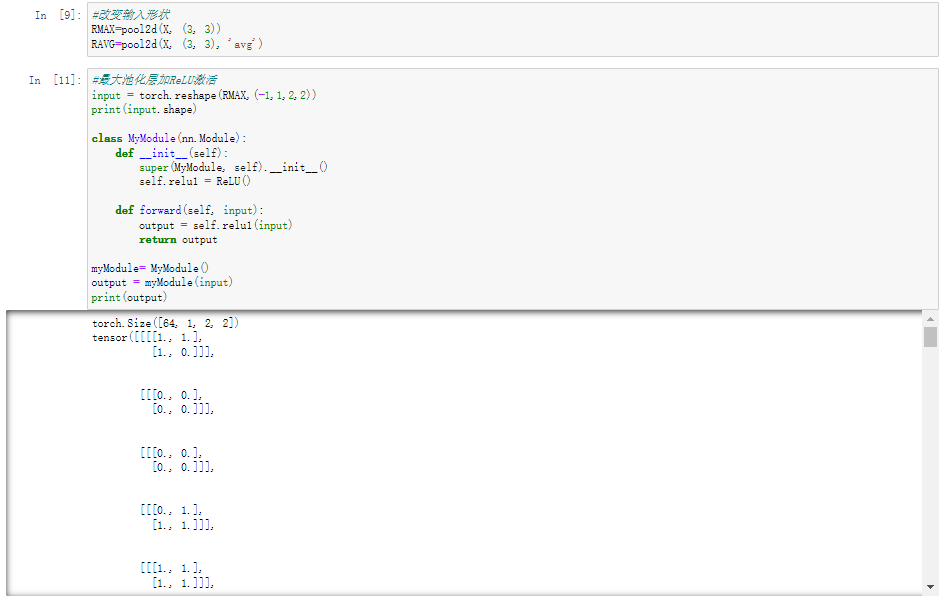
在卷积层后分别加入最大池化层、平均池化层，自定义池化层大小，观察输出结果：







在池化层后加入 ReLU 激活，观察输出结果：



输出结果为：

|  |
| --- |
| torch.Size([64, 1, 2, 2])  tensor([[[[1., 1.],  [1., 0.]]],  [[[0., 0.],  [0., 0.]]],  [[[0., 0.],  [0., 0.]]],  [[[0., 1.],  [1., 1.]]],  [[[1., 1.],  [1., 1.]]],  [[[0., 0.],  [0., 0.]]],  [[[0., 0.],  [0., 0.]]],  [[[1., 1.],  [1., 1.]]],  [[[1., 1.],  [1., 1.]]],  [[[1., 0.],  [0., 0.]]],  [[[0., 0.],  [0., 1.]]],  [[[1., 1.],  [1., 1.]]],  [[[0., 1.],  [1., 1.]]],  [[[1., 1.],  [0., 0.]]],  [[[0., 0.],  [1., 1.]]],  [[[1., 1.],  [1., 0.]]],  [[[0., 0.],  [1., 1.]]],  [[[1., 1.],  [1., 0.]]],  [[[0., 1.],  [1., 1.]]],  [[[1., 1.],  [0., 0.]]],  [[[0., 0.],  [0., 1.]]],  [[[1., 1.],  [1., 1.]]],  [[[1., 1.],  [1., 1.]]],  [[[1., 0.],  [0., 0.]]],  [[[0., 0.],  [0., 0.]]],  [[[1., 1.],  [1., 1.]]],  [[[1., 1.],  [1., 1.]]],  [[[0., 0.],  [0., 0.]]],  [[[0., 0.],  [0., 0.]]],  [[[0., 1.],  [1., 1.]]],  [[[1., 1.],  [1., 0.]]],  [[[0., 0.],  [0., 0.]]],  [[[0., 0.],  [0., 0.]]],  [[[0., 1.],  [1., 1.]]],  [[[1., 1.],  [1., 0.]]],  [[[0., 0.],  [0., 0.]]],  [[[0., 0.],  [0., 0.]]],  [[[1., 1.],  [1., 1.]]],  [[[1., 1.],  [1., 1.]]],  [[[0., 0.],  [0., 0.]]],  [[[0., 0.],  [0., 1.]]],  [[[1., 1.],  [1., 1.]]],  [[[1., 1.],  [1., 1.]]],  [[[1., 0.],  [0., 0.]]],  [[[0., 0.],  [1., 1.]]],  [[[1., 1.],  [1., 0.]]],  [[[0., 1.],  [1., 1.]]],  [[[1., 1.],  [0., 0.]]],  [[[0., 1.],  [1., 1.]]],  [[[1., 1.],  [0., 0.]]],  [[[0., 0.],  [1., 1.]]],  [[[1., 1.],  [1., 0.]]],  [[[1., 1.],  [1., 1.]]],  [[[1., 0.],  [0., 0.]]],  [[[0., 0.],  [0., 1.]]],  [[[1., 1.],  [1., 1.]]],  [[[1., 1.],  [1., 1.]]],  [[[0., 0.],  [0., 0.]]],  [[[0., 0.],  [0., 0.]]],  [[[1., 1.],  [1., 1.]]],  [[[1., 1.],  [1., 0.]]],  [[[0., 0.],  [0., 0.]]],  [[[0., 0.],  [0., 0.]]],  [[[0., 1.],  [1., 1.]]]]) |

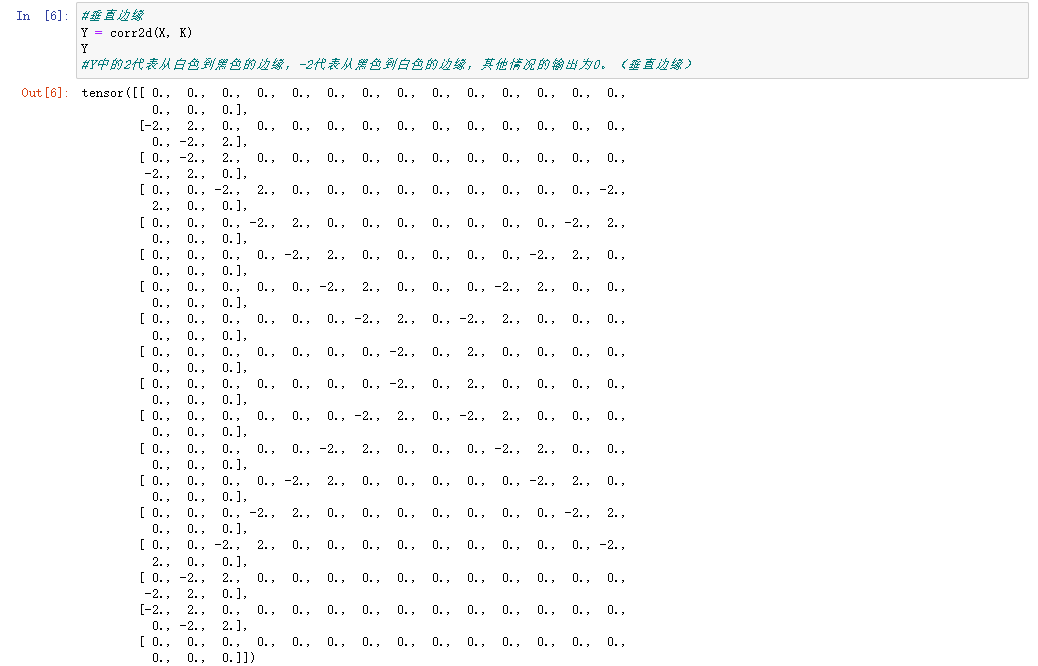


输出结果为：

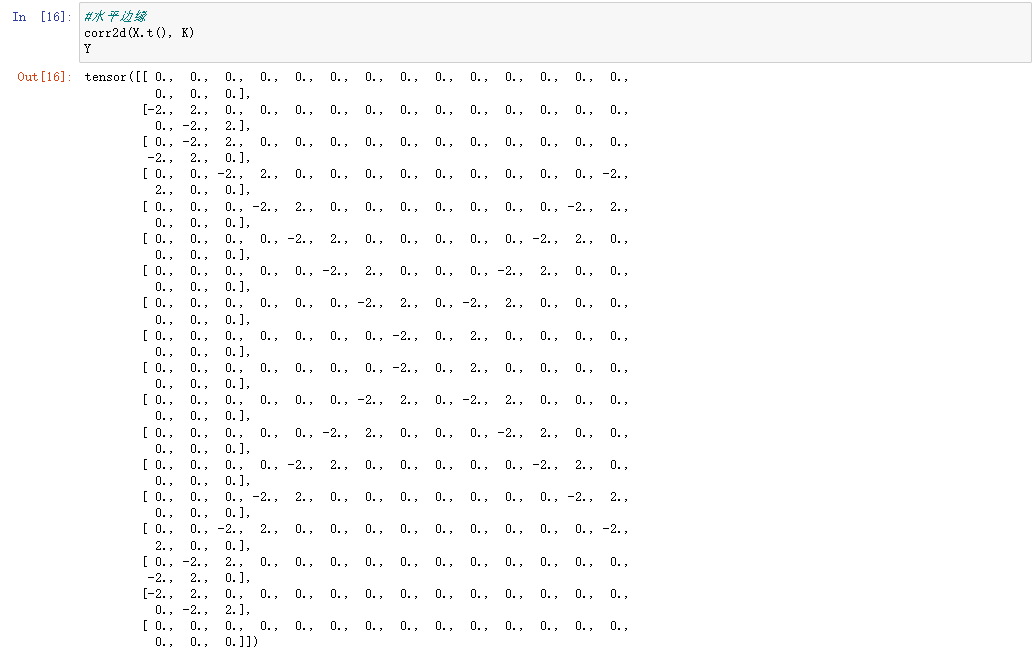
|  |
| --- |
| tensor([[[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.1111]]],  [[[0.1111, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.1111]]],  [[[0.1111, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]],  [[[0.0000, 0.0000],  [0.0000, 0.0000]]]]) |

使用三个不同数值的卷积核，提取图片的不同特征信息，并阐述观察到的特征：

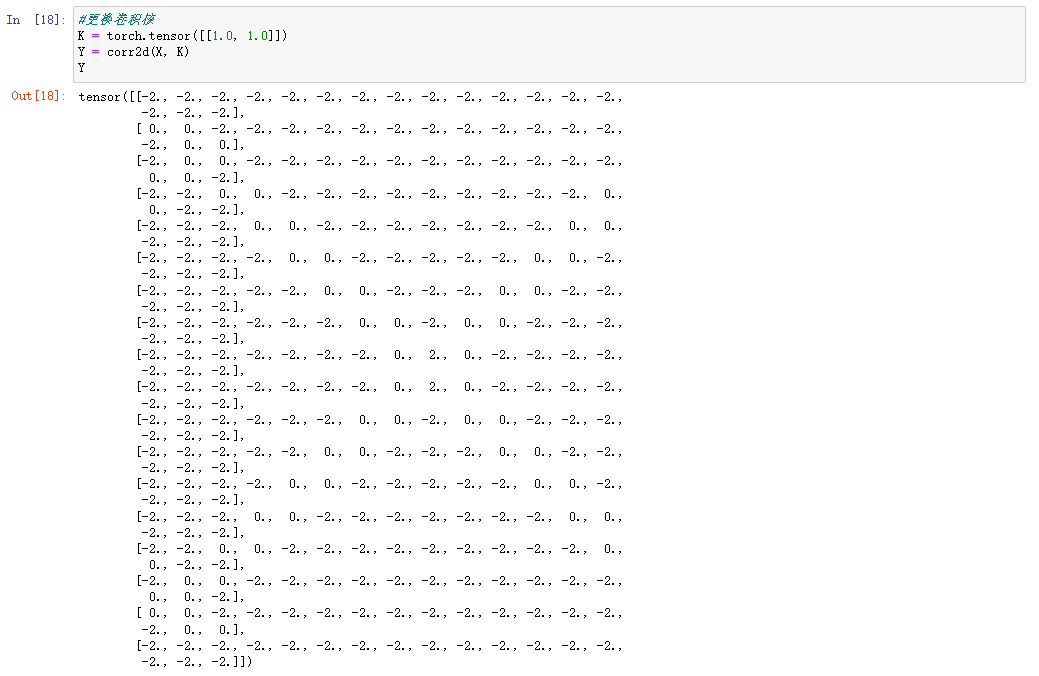
垂直边缘：2代表从白色到黑色的边缘，-2代表从黑色到白色的边缘，其他情况的输出为0。



水平边缘：2代表从白色到黑色的边缘，-2代表从黑色到白色的边缘，其他情况的输出为0。



更换卷积核：2水平或垂直方向上从白色到黑色，-2代表水平或垂直方向上从黑色到白色，其他情况的输出为0。

****

写一个两层的卷积神经网络，每层包括池化层和激活，其中至少一层含多输出通道，至少一层步幅大于1。观察上述图片经过神经网络后的输出：

