Pytorch学习笔记

## 构建神经网络及调用

class Flatten(nn.Module):

    """\_summary\_

    将多维的数组变成二维数组,行数不变

    比如

        rand(3,4,5)->tensor(3,20)

    Args:

        nn (\_type\_): \_description\_

    """

    def forward(self, x):

        return x.reshape(x.size(0), -1)

调用时直接使用

a=Flatten()

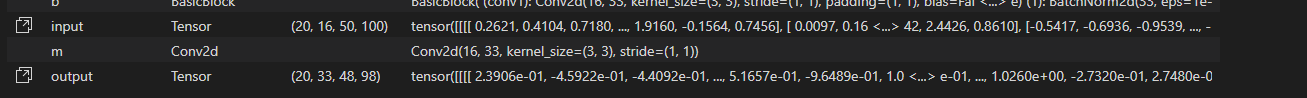
a(x)

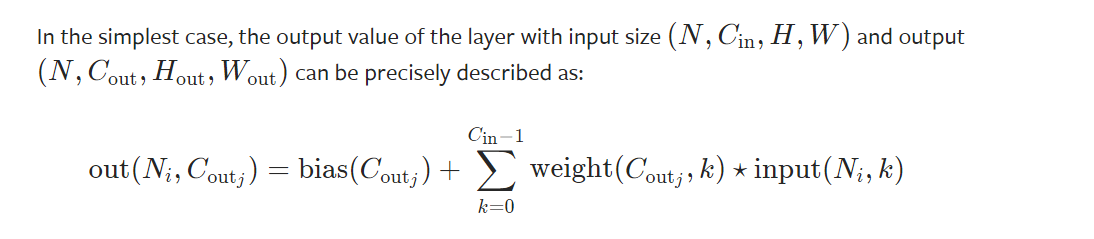
# With square kernels and equal stride

m = nn.Conv2d(16, 33, 3, stride=1)

input = torch.randn(20, 16, 50, 100)

output = m(input)





N是样本的数量batch size

*N* is a batch size,

*C* denotes a number of channels,

*H* is a height of input planes in pixels,

*W* is width in pixels.

# non-square kernels and unequal stride and with padding

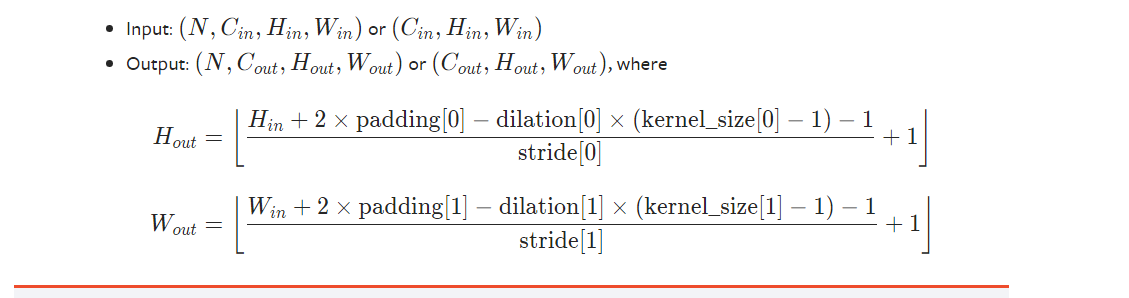
#m = nn.Conv2d(16, 33, (3, 5), stride=(2, 1), padding=(4, 2))

m = nn.Conv2d(16, 33, (3, 5), stride=1)

input = torch.randn(20, 16, 50, 100)

output = m(input)

输入输出之间的关系



# non-square kernels and unequal stride and with padding and dilation

m = nn.Conv2d(16, 33, (3, 5), stride=(2, 1), padding=(4, 2), dilation=(3, 1))

## BatchNorm2d

**>>>** *# With Learnable Parameters*

**>>> m** **=** **nn.BatchNorm2d(**100**)**

**>>>** *# Without Learnable Parameters*

**>>> m** **=** **nn.BatchNorm2d(**100**,** **affine=False)**

**>>>** input **=** **torch.randn(**20**,** 100**,** 35**,** 45**)**

**>>> output** **=** **m(**input**)**