

224x224x3

No.

Date

作业一

一、

ABBAC

二、

1 31x31x1

2 16x16x6

3 乘法 $(\lfloor \frac{w+2pw-kw}{s_w} \rfloor + 1)(\lfloor \frac{h+2ph-kh}{s_h} \rfloor + 1) c_o c_i k_h k_w$

加法 $(\lfloor \frac{w+2pw-kw}{s_w} \rfloor + 1)(\lfloor \frac{h+2ph-kh}{s_h} \rfloor + 1) c_o \cdot c_i k_h k_w$

4 卷积层、池化层 $RF_L = \begin{cases} K^2 & L=1 \\ (RF_{L-1} + (k-1)p)^2 & L>1 \end{cases}$ 其中 $P = P_1 \cdot P_2 \cdots P_{L-1}$

全连接层 ~~RF_L = RF_{L-1}~~ $RF_L = RF_{L-1}$

5

KOKUYO



CS 扫描全能王

3亿人都在用的扫描App

```

class AlexNet(nn.Module):
    def __init__(self, num_classes: int = 1000, dropout: float = 0.5)
        super().__init__()
        _log_api_usage_once(self)
        self.features = nn.Sequential(
            nn.Conv2d(3, 64, kernel_size=11, stride=4, padding=2),
            #感受野: 11 * 11
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
            #感受野: 19 * 19
            nn.Conv2d(64, 192, kernel_size=5, padding=2),
            #感受野: 51 * 51
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
            #感受野: 67 * 67
            nn.Conv2d(192, 384, kernel_size=3, padding=1),
            #感受野: 99 * 99
            nn.ReLU(inplace=True),
            nn.Conv2d(384, 256, kernel_size=3, padding=1),
            #感受野: 131 * 131
            nn.ReLU(inplace=True),
            nn.Conv2d(256, 256, kernel_size=3, padding=1),
            #感受野: 163 * 163
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
            #感受野: 195 * 195
        )
        self.avgpool = nn.AdaptiveAvgPool2d((6, 6))
        #感受野: 195 * 195
        self.classifier = nn.Sequential(
            nn.Dropout(p=dropout),
            nn.Linear(256 * 6 * 6, 4096),
            #感受野: 224 * 224
            nn.ReLU(inplace=True),
            nn.Dropout(p=dropout),
            nn.Linear(4096, 4096),
            #感受野: 224 * 224
            nn.ReLU(inplace=True),
            nn.Linear(4096, num_classes),
            #感受野: 224 * 224
        )

```

