

2.AlexNet

2012年，由Hinton和他的学生Alex提出，论文地址[ImageNet Classification with Deep Convolutional Neural Networks](https://arxiv.org/abs/1207.4186)

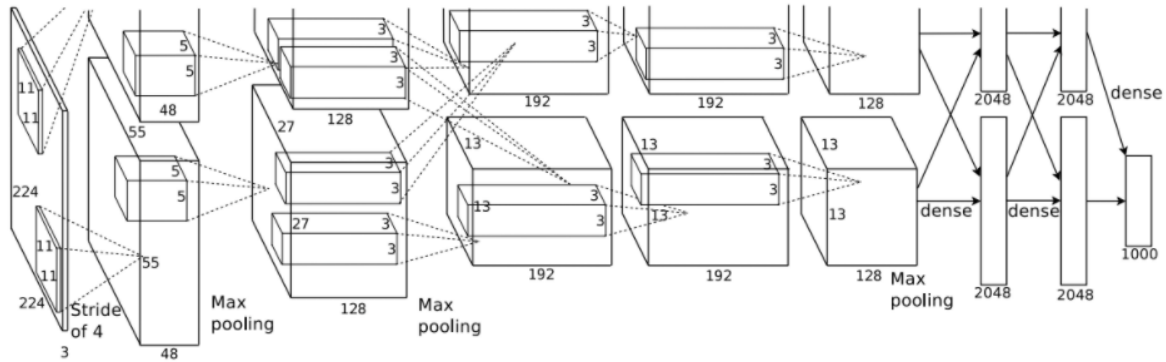


图2 AlexNet网络结构

pytorch实现

```
1 class AlexNet(nn.Module):
2
3     def __init__(self, num_classes=1000):
4         super(AlexNet, self).__init__()
5         self.features = nn.Sequential(
6             nn.Conv2d(3, 64, kernel_size=11, stride=4,
7 padding=2),
8             nn.ReLU(inplace=True),
9             nn.MaxPool2d(kernel_size=3, stride=2),
10            nn.Conv2d(64, 192, kernel_size=5,
11 padding=2),
12            nn.ReLU(inplace=True),
13            nn.MaxPool2d(kernel_size=3, stride=2),
14            nn.Conv2d(192, 384, kernel_size=3,
15 padding=1),
16            nn.ReLU(inplace=True),
17            nn.Conv2d(384, 256, kernel_size=3,
18 padding=1),
19            nn.ReLU(inplace=True),
20            nn.Conv2d(256, 256, kernel_size=3,
21 padding=1),
22            nn.ReLU(inplace=True)
23        )
24        self.classifier = nn.Sequential(
25            nn.Dropout(),
26            nn.Linear(256 * 4 * 4, 2048),
27            nn.ReLU(inplace=True),
28            nn.Dropout(),
29            nn.Linear(2048, 2048),
30            nn.ReLU(inplace=True),
31            nn.Linear(2048, num_classes)
32        )
```

```

17         nn.ReLU(inplace=True),
18         nn.MaxPool2d(kernel_size=3, stride=2),
19     )
20     self.avgpool = nn.AdaptiveAvgPool2d((6, 6))
21     self.classifier = nn.Sequential(
22         nn.Dropout(),
23         nn.Linear(256 * 6 * 6, 4096),
24         nn.ReLU(inplace=True),
25         nn.Dropout(),
26         nn.Linear(4096, 4096),
27         nn.ReLU(inplace=True),
28         nn.Linear(4096, num_classes),
29     )
30
31     def forward(self, x):
32         x = self.features(x)
33         x = self.avgpool(x)
34         x = torch.flatten(x, 1)
35         x = self.classifier(x)
36         return x

```

每个Conv后都要接激活函数，Dropout放在全连接层之前，且全连接层后也要接激活函数，最后一个全连接层不需要Dropout和激活函数。

nn.AdaptiveAvgPool2d作用：只需要给定特征图的大小，其前后通道数不变