卷积神经网络

丁文超

LeNet

1

LeNet

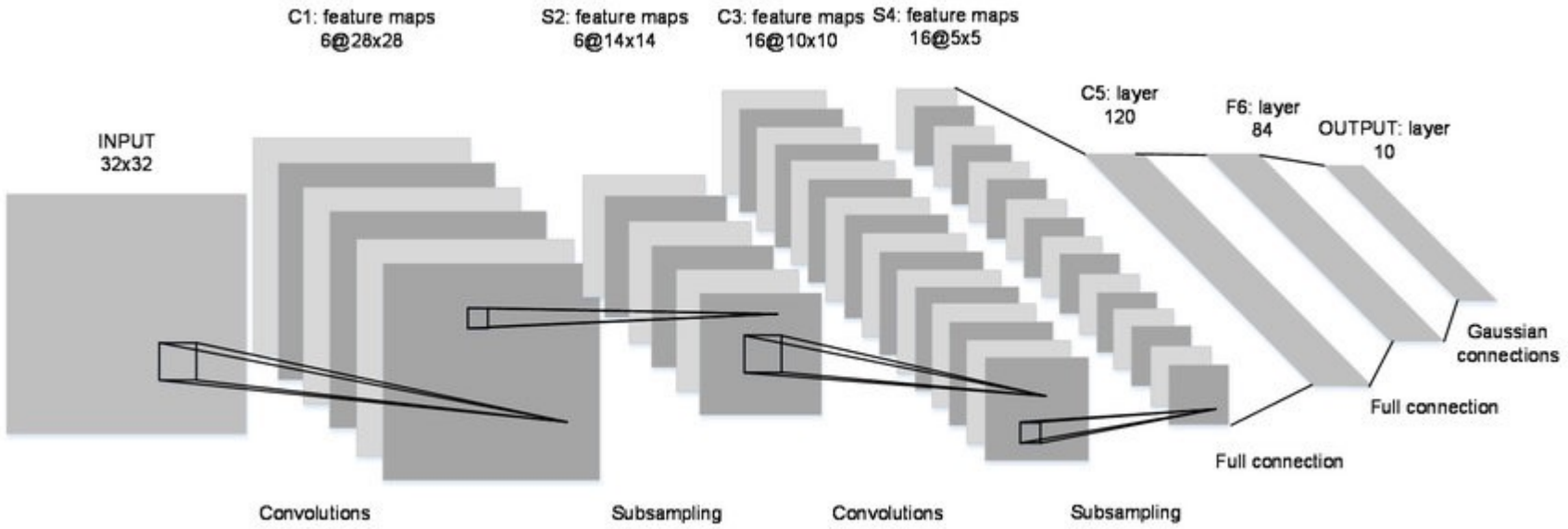
多层感知机模型存在的两个问题：

1. 图像在同一列邻近的像素在这个向量中可能相距较远，它们构成的模式可能难以被模型识别。
2. 对于大尺寸的输入图像，使用全连接层容易导致模型过大。

卷积层尝试解决这两个问题。一方面，卷积层保留输入形状，使图像的像素在高和宽两个方向上的相关性均可能被有效识别；另一方面，卷积层通过滑动窗口将同一卷积核与不同位置的输入重复计算，从而避免参数尺寸过大。

LeNet

LeNet 诞生于 1994 年，是最早的卷积神经网络之一，并且推动了深度学习领域的发展。自从 1988 年开始，在许多次成功的迭代后，这项由 YannLeCun 完成的开拓性成果被命名为 LeNet5.



LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

LeNet

结构细节：

（1）层C1是具有六个5*5的卷积核的卷积层（convolution），特征映射的大小为28*28；

（2）层S2是输出6个大小为14*14的特征图的子采样层（subsampling/pooling）；

（3）层C3是具有16个5-5的卷积核的卷积层；

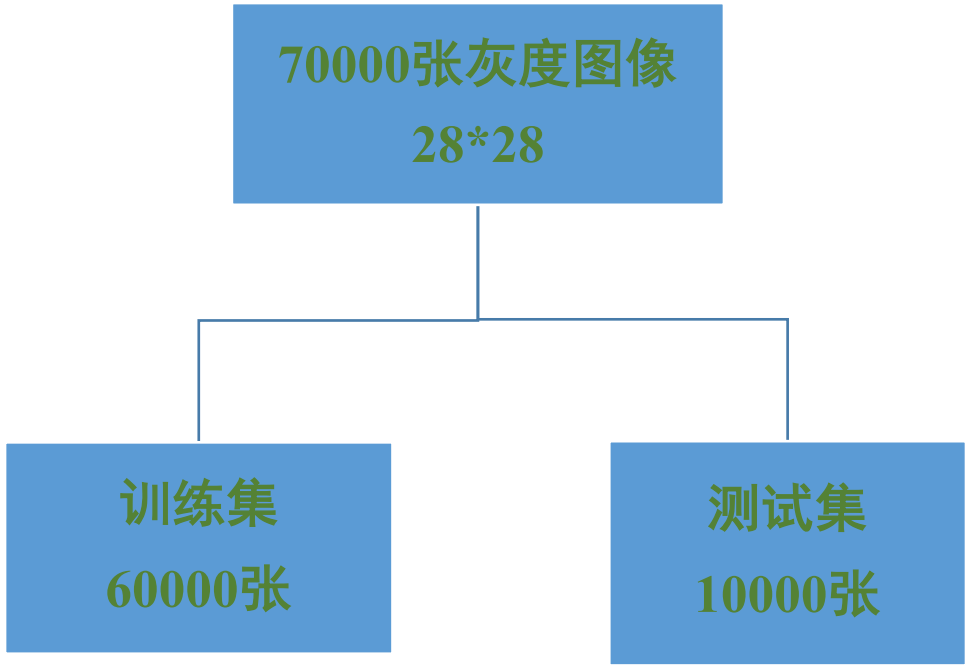
（4）层S4是与S2类似，大小为2*2，输出为16个5*5的特征图；

（5）层C5是具有120个大小为5*5的卷积核的卷积层。每个单元连接到S4的所有16个特征图上的5*5邻域；

（6）F6层完全连接到C5，输出84张特征图.

模型实验：Fashion-MNIST数据集

Fashion-MNIST数据集（10类服饰分类数据集）



Label	Description
0	T恤（T-shirt/top）
1	裤子（Trouser）
2	套头衫（Pullover）
3	连衣裙（Dress）
4	外套（Coat）
5	凉鞋（Sandal）
6	衬衫（Shirt）
7	运动鞋（Sneaker）
8	包（Bag）
9	靴子（Ankle boot）

数据集下载地址：

https://www.worldlink.com.cn/en/osdir/fashion-mnist.html

数据集介绍

数据读取

```
import torch
import torchvision
import torchvision.transforms as transforms

transform = transforms.Compose(
    [transforms.ToTensor(),
    transforms.Normalize((0.5,), (0.5,))])

batch_size = 64

training_data = torchvision.datasets.FashionMNIST('./data', train=True, transform=transform, download=True)
test_data = torchvision.datasets.FashionMNIST('./data', train=False, transform=transform, download=True)

train_dataloader = torch.utils.data.DataLoader(training_data, batch_size=batch_size, shuffle=True)
test_dataloader = torch.utils.data.DataLoader(test_data, batch_size=batch_size, shuffle=False)

# Report split sizes
print('Training set has {} instances'.format(len(training_data)))
print('Validation set has {} instances'.format(len(test_data)))
```

LeNet

与输入图片的初始分辨率有关，Fashion-MNIST 中的图片分辨率为28x28

```python
import torch.nn as nn

class LeNet(nn.Module):
    def __init__(self):
        super(LeNet, self).__init__()
        self.net = nn.Sequential(
            nn.Conv2d(1, 6, kernel_size=5, padding=2), nn.Sigmoid(),
            nn.AvgPool2d(kernel_size=2, stride=2),
            nn.Conv2d(6, 16, kernel_size=5), nn.Sigmoid(),
            nn.AvgPool2d(kernel_size=2, stride=2),
            nn.Flatten(),
            nn.Linear(16 * 5 * 5, 120), nn.Sigmoid(),
            nn.Linear(120, 84), nn.Sigmoid(),
            nn.Linear(84, 10))

    def forward(self, x):
        x = self.net(x)
        return x
```

LeNet

训练代码可以参照前几次实验

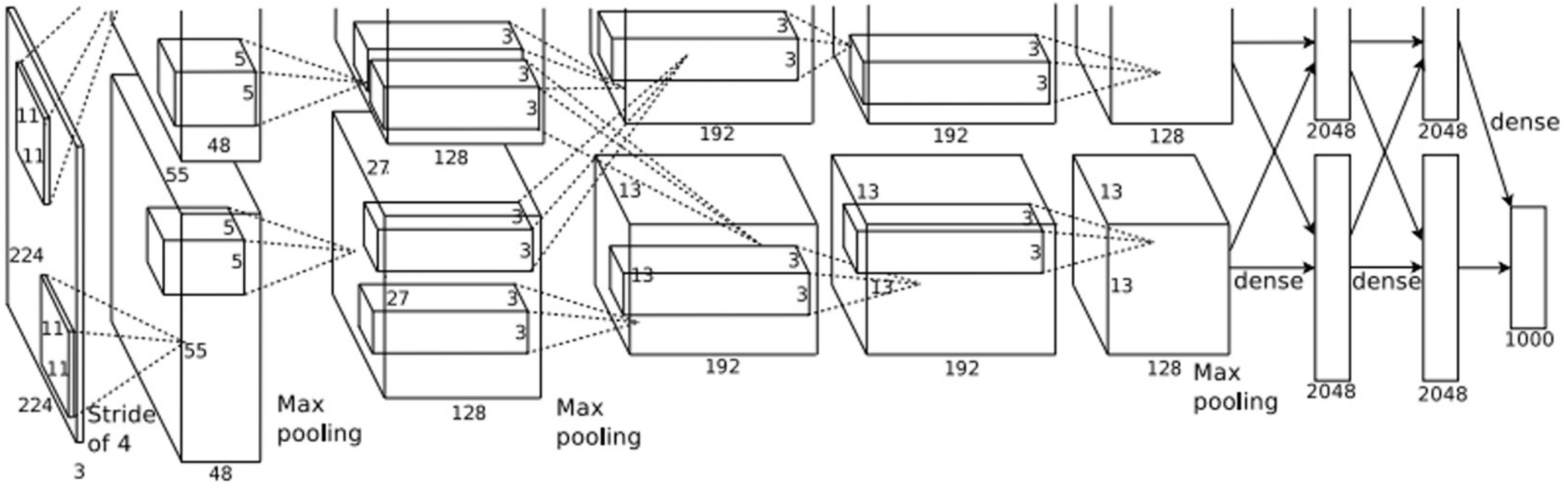
```
Epoch 1
-------------------------------
loss: 2.309355  [   64/60000]
loss: 2.315968  [ 6464/60000]
loss: 2.288191  [12864/60000]
loss: 2.298771  [19264/60000]
loss: 2.278338  [25664/60000]
loss: 2.279574  [32064/60000]
loss: 2.239067  [38464/60000]
loss: 2.222734  [44864/60000]
loss: 2.088865  [51264/60000]
loss: 1.977984  [57664/60000]
Test Error:
 Accuracy: 37.8%, Avg loss: 1.862739

Epoch 2
-------------------------------
loss: 1.863418  [   64/60000]
loss: 1.618481  [ 6464/60000]
loss: 1.074064  [12864/60000]
loss: 1.179384  [19264/60000]
loss: 1.008269  [25664/60000]
loss: 1.204725  [32064/60000]
loss: 1.208350  [38464/60000]
loss: 0.923926  [44864/60000]
...
Test Error:
 Accuracy: 82.7%, Avg loss: 0.477034
```

AlexNet

2

AlexNet

AlexNet 是一个卷积神经网络的名字，最初是与 CUDA 一起使用 GPU 支持运行的，AlexNet 是2012年 ImageNet 竞赛冠军获得者 Alex Krizhevsky 设计的。该网络的错误率与前一届冠军相比减小了 10% 以上，比亚军高出 10.8 个百分点。AlexNet是由多伦多大学 SuperVision 组设计的，由 Alex Krizhevsky , Geoffrey Hinton 和Ilya Sutskever 组成。



AlexNet Vs LeNet

(1)激活函数，sigmoid函数到ReLU函数；

(2)多GPU训练；

(3)局部响应归一化（Local Response Normalization, LRN）；

(4)重叠池化（Overlapping Pooling）；

(5)Data Augmentation（数据增强）；

(6)Dropout

AlexNet Vs LeNet

- AlexNet 第一层中的卷积窗口形状是11×11。因为 ImageNet 中绝大多数图像的高和宽均比 MNIST 图像的高和宽大10倍以上，ImageNet 图像的物体占用更多的像素，所以需要更大的卷积窗口来捕获物体。
- 第二层中的卷积窗口形状减小到5×5，之后全采用3×3。
- 此外，第一、第二和第五个卷积层之后都使用了窗口形状为3×3、步幅为2的最大池化层。而且，AlexNet 使用的卷积通道数也大于 LeNet 中的卷积通道数的数十倍。
- 紧接着最后一个卷积层的是两个输出个数为4096的全连接层。这两个巨大的全连接层带来将近 1 GB的模型参数。由于早期显存的限制，最早的 AlexNet 使用双数据流的设计使一个GPU只需要处理一半模型。幸运的是， 显存在过去几年得到了长足的发展，因此通常我们不再需要这样的特别设计了。

AlexNet Vs LeNet

```python
class AlexNet(nn.Module):
    def __init__(self, num_classes=10):
        super(AlexNet, self).__init__()
        self.features = nn.Sequential(
            nn.Conv2d(1, 64, kernel_size=11, stride=4, padding=2),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
            nn.Conv2d(64, 192, kernel_size=5, padding=2),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
            nn.Conv2d(192, 384, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(384, 256, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(256, 256, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
        )
        self.classifier = nn.Sequential(
            nn.Dropout(),
            nn.Linear(256 * 6 * 6, 4096),  # 根据输入尺寸计算
            nn.ReLU(inplace=True),
            nn.Dropout(),
            nn.Linear(4096, 4096),
            nn.ReLU(inplace=True),
            nn.Linear(4096, num_classes),
        )

    def forward(self, x):
        x = self.features(x)
        x = x.view(x.size(0), 256 * 6 * 6)
        x = self.classifier(x)
        return x
```

AlexNet 输入图片的分辨率为 224x224，

我们需要对数据集做一些处理

```python
# 更改数据处理方式
transform = transforms.Compose([
    transforms.Resize(224),
    transforms.ToTensor(),
    transforms.Normalize((0.5,), (0.5,))
])
```

AlexNet

利用AlexNet完成LeNet对于Fashion-MNIST数据集的分类

（1）数据预处理，尺寸对应，数据增强；

（2）网络模型的构建。

其他过程都是一致的。

对比分析两者的结果如何？


THANKS