

Chapter 5 卷积神经网络

5.1 二维卷积层

- 二维互相关运算
 - 二维卷积层中一个二维输入数组和一个二维核互相关运算输出一个二维数组
 - 卷积窗口从左上方开始，从左到右从上到下滑动
 - 输入子数组与核数组按元素相乘求和
- 二维卷积层
 - 互相关运算后加一个标量偏差得到输出
 - 模型包括卷积核和标量偏差
 - 通常对卷积核随机初始化，不断迭代卷积核和偏差
- 图像中物体边缘检测
- 通过数据学习核数组
- 互相关运算和卷积运算
 - 卷积运算和互相关运算类似，只需将核数组左右翻转上下翻转，然后与输入数组做互相关运算
 - 之所以互相关运算可以代替卷积运算，是因为核数组是学习出来的
 - 使用互相关运算学习得到核数组A，使用卷积运算学习得到核数组A'，输入数组和A再做互相关运算，输出数组和A'做卷积运算，二者输出相同
- 特征图和感受野
 - 特征图——二维卷积层输出的二维数组可以看作是输入在空间维度（宽和高）上某一级的表征，也叫特征图
 - 感受野——影响元素x的前向计算的所有可能输入区域（可能大于输入的实际尺寸）叫做x的感受野

5.3 多输入通道和多输出通道

- 多输入通道
 - 卷积核输入通道数应与输入的通道数相同
 - 每个通道上做互相关运算，然后按通道相加得到输出
- 多输出通道
 - 设卷积核输入和输出通道数分别为C_I和C_O
 - 若希望得到含多个通道的输出，可以为每个输出通道创建形状为C_I*K_h*K_w的核数组
 - 也就是卷积核形状为C_O*C_I*K_h*K_w
 - 每个输出通道上的结果由卷积核在该输出通道上的核数组与整个输入数组计算而来
- 1*1 卷积层
 - 失去识别高和宽维度上相邻元素的功能
 - 主要计算发生在通道维上
 - 输入和输出具有相同的高和宽
 - 输出中的每个元素来自输入中高和宽上相同位置的元素在不同通道之间的按权重累加
 - 通道维——特征维；高和宽维度——数据样本；1*1卷积层——全连接层
 - 通常用来调整网络层之间的通道数，并控制模型复杂度

5.5 卷积神经网络（LENET）

- 之前实现的多层感知机局限
 - 同一列相邻像素在向量中相距较远
 - 大尺寸输入图像全连接层容易造成模型过大
- 解决——卷积层
 - 保留输入形状
 - 滑动窗口避免参数尺寸过大
- LENET模型
 - 卷积层块
 - 卷积层后接最大池化层
 - 卷积层识别图像里的空间模式
 - 最大池化用来降低卷积层对位置的敏感性
 - 窗口5*5，输出使用sigmoid函数，第一个卷积层输出通道为6，第二个为16
 - 池化层窗口形状为2*2，stride=2
 - 全连接层块
 - 将小批量中每个样本变平（输入为二维）
 - 第一维是批量，第二维是样本向量表示（h*w*channels）
 - 3个全连接层，输出个数120，84，10
 - 获取数据训练模型

5.7 使用重复元素的网络（VGG）

- VGG块
 - 组成规律
 - 连续使用数个相同的填充为1，窗口3*3的卷积层
 - 之后接上步幅为2，窗口形状2*2的最大池化层
 - 卷积层保持输入的高和宽不变，池化层对其减半
 - 对于给定的感受野，堆积的小卷积核优于大卷积核
 - 可以增加网络深度保证学习更复杂的模式，并且代价较小
- VGG网络
 - 超参数conv_arch指定了每个VGG块里卷积层个数和输入输出通道数
 - 实现VGG-11
- 获取数据训练模型

5.2 填充和步幅

- 填充
 - 在输入高和宽两侧填充元素
 - 输出的高和宽分别增加填充的高和宽行（列）数
 - 很多情况下，高和宽分别增加K_h-1, K_w-1, 使得输入和输出具有相同的高和宽
 - 输出Y[i,j]是以输入X[i,j]为中心的窗口和卷积核互相关计算得到的
- 步幅
 - 每次滑动的行数或者列数
 - 输出形状的表达

5.4 池化层

- 二维最大池化层和平均池化层
 - 缓解卷积层对位置的过度敏感性
 - 每次对输入数据的一个固定形状窗口（池化窗口）中的元素计算输出
 - 直接计算窗口内元素最大值或者平均值
 - 边缘检测例子
- 填充和步幅
- 多通道
 - 池化层对每个输入通道分别池化，而不是像卷积层一样相加
 - 输入通道数=输出通道数

5.6 深度卷积神经网络（ALEXNET）

- 学习特征表示
- 与LeNet区别——ALEXNET
 - 5层卷积，2层全连接隐藏层，1层全连接输出层
 - 第一层卷积窗口11*11
 - 第二场卷积窗口5*5
 - 之后全为3*3
 - 第一、二、五卷积层使用3*3 stride=2的最大池化层
 - 全连接层输出为4096
 - ReLU激活函数
 - 丢弃法
 - 图像增广