第 5 单元　卷积神经网络与计算机视觉

支撑的课程目标

1. 能够基于智能信息处理的基本理论和技术，识别和理解数据处理与分析等问题的相关特性.
2. 能够运用智能信息处理的相关原理和专业知识，设计实验方案，为解决数据处理与分析等问题提供支持.

基本要求

1. 能够详述卷积神经网络的基本概念，理解卷积神经网络模型的特点.
2. 应用卷积神经网络的基本原理，解决数据分析领域的分类问题.

教学重点与难点

重点：　卷积；池化；特征图；卷积网络的构建.

难点：　卷积.

教学过程设计

新课导入、知识讲授、教学目标达成考核、总结.

教学过程设计

本单元教学通过“互动、开放”的课堂形式，采用探究式学习、问题导入的教学方法，激发学生的学习兴趣，促成课程目标的达成.

教学学时

6 学时.

一、导入新课（5 分钟）

前面介绍的全连接神经网络处理的模式特征为向量形式，所以在像图像类的数据在输入到神经网络之前是将图像数据转换成向量的形式。但是，这种方

法没有利用图像中像素之间可能存在的任何空间关系，例如像素排列成角，边缘段的存在以及其他可能有助于解决问题的特征。

二、知识讲授（240 分钟）

本单元要点：

* 基础概念
* CNN 的神经计算
* CNN 的前向传播
* CNN 的反向传播

1 基础概念

下面将 LeNet 为例介绍深度卷积神经网络（简称 CNN），它接受图像作为输入，非常适合自动学习和图像分类。

CNN 网络与全连接网络执行的计算非常相似，包括四个步骤：（1）获得乘积之和，（2）添加一个偏差值，（3）结果通过激活函数，（4）激活值成为下一层的单一输入。尽管 CNN 和全连接神经网络执行的计算相似，但两者之间存在一些基本区别，除了它们的输入格式是 2D 还是矢量。一个重要的区别是，CNN 能够直接从原始图像数据中学习二维特征。另一个主要区别在于层的连接方式。在完全连接的神经网络中，将一层中每个神经元的输出直接馈送到下一层中每个神经元的输入。相比之下，在 CNN 中，将一层中每个输入传递单个值，该值是通过在前一层的输出的空间邻域上做卷积得到的。因此，在全连接网络的意义上，CNN 并不是完全连接的。另一个不同之处在于，从一层传递到下一层的二维阵列要经过下采样以降低对输入中平移的敏感性。

CNN 中的邻域处理类型是空间卷积，卷积计算出像素与一组内核权重之间的乘积总和。在输入图像的每一个空间位置进行卷积操作，在每个空间位置可以获得一个标量值。如果将该值看作是一个全连接网络中一层的一个神经元的输出，再加上偏差并将结果传递激活函数，可以看出 CNN 的基本计算和全连接神经网络的完全类似。在 CNN 中，空间邻域被称为感受野。感受野的作用只是在输入图像中选择一个区域。如图 1 所示，CNN 执行的第 1 个操作是卷积，它的值是通过在图像上移动感受野得到的，该值是一组权值和感受野内的像素的乘积之和。以感受野形状排列的一组权值称为一个核。感受野移动的空间位置

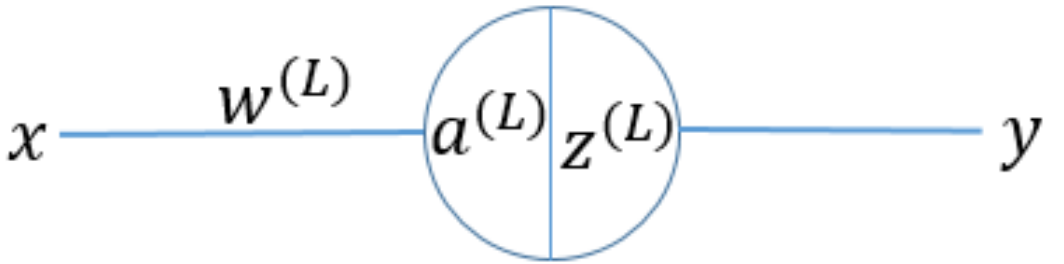

图 1: 卷积神经网络示意图

称为步幅。在 CNN 中，使用大于 1 的步幅的重要动机是数据规约。另一个重要动机是作为下采样的替代，用于降低系统对空间偏移的灵敏度。

对于每个卷积值（乘积和），添加一个偏差，然后传递结果给激活函数来生成单个值。然后，该值被馈送到下一层输入中的相应位置。当对输入图像所有位置都处理完毕，将产生二维值集，该值集在下一层将被存储为二维数组，称为特征图。在输入图像的感受野的所有位置使用相同的权重和单个偏差获取卷积值（特征图）。这样做是为了使图像中所有点的相同特征被检测到。为此使用相同的权重和偏差称为权重（或参数）共享。在 CNN 中，特征图统称为卷积层。

卷积和激活后的过程是下采样（也称为汇聚），这是由 Hubel 和 Wiesel 在 1959 年提出的哺乳动物视觉皮层模型激发的。汇聚是对降维进行建模的一种方式。当使用大型图像数据库训练 CNN 时，汇聚具有减少正在处理的数据量的附加优势。可以认为下采样的结果会产生汇聚特征图。汇聚操作是这样的，将一个特征图分成不同的小区域（通常为 2x2），称为汇聚邻域，然后利用单个值替换汇聚邻域内所有的元素。假设汇聚邻域是相邻的，有几种计算汇聚值的方法。三种常见的汇聚方法是：（1）平均汇聚，其中每个邻域中的值都被替换为邻域内元素均值；（2）最大汇聚，其中每个邻域中的值被替换为邻域中的元素的最大值；（3）汇聚，其中汇聚的结果是邻域值平方和的平方根。对每个特征图都可以得到一个汇聚的特征图。汇聚的特征图统称为汇聚层。感受野、卷积、参数共享和汇聚的使用是 CNN 独有的特征。

因为特征图是空间卷积的结果，它们只是经过滤波的图像。因此，可知汇聚

的特征图是低分辨率图像的滤波结果。在 CNN 中，在上一层汇聚特征图是下一层的输入。因此，除了第一层的输入外，每个卷积层的输入通过会有多个。由于最终目标是使用特征进行分类，因此需要设计一个分类器。在 CNN 中，通过将最后的汇聚层的值输入到全连接的神经网络中来执行分类。但是 CNN 的输出是 2D 数组（即分辨率降低的滤波图像），而全连接网络的输入是向量。因此，必须在最后一层向量化 2D 汇聚特征图。CNN 中最后一层的每个二维数组都转换为向量，然后将所有得到的向量以列形式连接起来形成单个向量。该向量通过神经网络传播。在任何给定的应用中，全连接网络中输出数量等于要分类的模式类别的数量，具有最高值的输出决定输入的类别。

2 CNN 网络的神经计算

在 CNN 中，用来生成特征图中单个值的计算是二维卷积，它是卷积核的系数与卷积核在图像中覆盖的相应元素的乘积之和。设 w 表示卷积核，它是按感受野的形状排列而成的权值；$z_{x,y}$ 表示图像或池化后的特征值，则输入图像中任意点 (x,y) 的卷积值由下式给出

$$w * z_{x,y} = \sum_{l}\sum_{k} w_{m,n} z_{x-m,y-n} \tag{1}$$

其中 l,k 张成了卷积核的两个维度。如果给上式增加偏差，并用 a 表示结果，可以得到

$$a = w * z_{x,y} = \sum_{l}\sum_{k} w_{m,n} z_{x-m,y-n} + b = w * z \tag{2}$$

如果对任意点 (x,y) 的空间卷积添加偏差，则可以得到与全连接网络中人工神经元相同的计算形式。如果将 a 看作神经元的总输入，则将 a 传递给激活函数 h 得到神经元的输出：

$$z = h(a) \tag{3}$$

3 CNN 的前向传播方程

根据前面的讨论可知，卷积核 w 与输入 $z_{x,y}$ 的卷积结果表示为

$$\begin{aligned} a_{x,y} &= \sum_{m}\sum_{n} w_{m,n} z_{x-m,y-n} + b \\ &= w * z_{x,y} + b \end{aligned} \tag{4}$$

其中 l, k 张成了卷积核的两个维度，x, y 张成了输入的两个维度，b 是偏置。则激活函数的值 $z_{x,y}$ 表示为

$$z_{x,y} = h(a_{x,y}) \tag{5}$$

为了区分各层特征，可以使用 l 表示层数，上述式子可以重新表示为

$$\begin{aligned} a_{x,y}^{(l)} &= \sum_m \sum_n w_{m,n}^{(l)} z_{x-m,y-n}^{(l-1)} + b^{(l)} \\ &= w^{(l)} * z_{x,y}^{(l-1)} + b^{(l)} \end{aligned} \tag{6}$$

$$z_{x,y}^{(l)} = h(a_{x,y}^{(l)}) \tag{7}$$

其中 $l = 1, 2, \cdots, L$，L 表示卷积层的个数，$z_{x,y}^{(l)}$ 表示卷积层 l 的汇聚特征的值。当 $l = 1$ 时，

$$z_{x,y}^{(0)} = \{\text{表示输入图像的像素值}\} \tag{8}$$

当 $l = L$ 时，

$$z_{x,y}^{(L)} = \{\text{表示最后一个卷积层的汇聚特征的值}\} \tag{9}$$

注意：l 从 1 开始而不是从 2 开始。原因是这样做，l 与卷积层的序号一一对应，卷积层从 2 开始有点混淆。最后，我们注意池化不需要任何卷积操作。池化的唯一作用是降低特征图的分辨率，在这里我们没有显示给出池化的公式。

在卷积神经网络的卷积部分，我们需要利用式 (10)、(7)、(8)、(9) 计算前向传播的所有值。在神经网络中，最后一个卷积层的汇聚特征值被向量化，作为前馈神经网络的输入传递到全连接的前馈神经网络中，前向传播过程参考第 4 单元的神经网络内容。

4 训练 CNN 的反向转播方程

CNN 的前馈方程与全连接的神经网络的前馈方程相似，仅是乘法被卷积代替，其符号反映 CNN 不是完全连接的。反向传播方程在许多方面也与全连接的

神经网络中的方程相似。首先定义 CNN 的输出误差相对于网络中每个神经元的变化率。输出误差的形式与完全连接的神经网络相同，被定义为

$$\delta_{x,y}^{(l)} \equiv \frac{\partial E}{\partial a_{x,y}^{(l)}} \tag{10}$$

与全连接网络一样，我们希望寻找第 l 层的误差与第 $l+1$ 层的误差之间的关系。使用链式法则，可以得到

$$\delta_{x,y}^{(l)} = \frac{\partial E}{\partial a_{x,y}^{(l)}} = \sum_u \sum_v \frac{\partial E}{\partial a_{u,v}^{(l+1)}} \frac{\partial a_{u,v}^{(l+1)}}{\partial a_{x,y}^{(l)}} \tag{11}$$

其中 u, v 表示所有与 $a_{x,y}^{(l)}$ 相关的值的下标。这种求和可以通过链式法则求解。

根据定义，式 (11) 的第一项为 $\delta_{x,y}^{(l+1)}$。因此，式 (11) 可以写为

$$\delta_{x,y}^{(l)} = \frac{\partial E}{\partial a_{x,y}^{(l)}} = \sum_u \sum_v \delta_{u,v}^{(l+1)} \frac{\partial a_{u,v}^{(l+1)}}{\partial a_{x,y}^{(l)}} \tag{12}$$

将式 (7) 代入式 (12)，并使用式 (10) 替换 $a_{u,v}^{(l+1)}$，可以得到

$$\delta_{x,y}^{(l)} = \sum_u \sum_v \delta_{u,v}^{(l+1)} \frac{\partial}{\partial a_{x,y}^{(l)}} \left[\sum_m \sum_n w_{m,n}^{(l+1)} h(a_{u-m,v-n}^{(l)}) + b^{(l+1)} \right] \tag{13}$$

式 (13) 中，括号中的项的导数都等于零除了 $u-m=x, v-n=y$ 的项，因为 $b^{(l+1)}$ 关于 $a_{x,y}^{(l)}$ 等于零。如果 $u-m=x, v-n=y$，那么 $m=u-x, n=v-y$。因此，对括号中的表达式求导，将式 (13) 中写为

$$\delta_{x,y}^{(l)} = \sum_u \sum_v \delta_{u,v}^{(l+1)} \left[\sum_{u-x} \sum_{v-y} w_{u-x,v-y}^{(l+1)} h'(a_{x,y}^{(l)}) \right] \tag{14}$$

x, y, u, v 的值是在括号内的项的范围之外确定的。一旦这些变量的值确定后，括号内的 $u-x, v-y$ 就是两个常数。因此，双重求和的值为 $w_{u-x,v-y}^{(l+1)} h'(a_{x,y}^{(l)})$，可以将式 (14) 写为

$$\begin{aligned} \delta_{x,y}^{(l)} &= \sum_u \sum_v \delta_{u,v}^{(l+1)} w_{u-x,v-y}^{(l+1)} h'(a_{x,y}^{(l)}) \\ &= h'(a_{x,y}^{(l)}) \sum_u \sum_v \delta_{u,v}^{(l+1)} w_{u-x,v-y}^{(l+1)} \end{aligned} \tag{15}$$

式 (15) 中第 2 行的双重求和是卷积的形式，但是其下标与式 (10) 的下标的负数。因此，可以把式 (15) 写为

$$\delta_{x,y}^{(l)} = h'(a_{x,y}^{(l)}) \left[\delta_{u,v}^{(l+1)} * w_{-x,-y}^{(l+1)}\right] \tag{16}$$

式 (16) 中下标的负号表示对 w 沿着两个坐标轴进行反折。这相当于对 w 旋转 180 度。根据这一个结果，我们得出第 l 层误差的表达式，即式 (16) 等价的写为

$$\delta_{x,y}^{(l)} = h'(a_{x,y}^{(l)}) \left[\delta_{u,v}^{(l+1)} * rot180(w_{x,y}^{(l+1)})\right] \tag{17}$$

由于卷积核不依赖于 x, y，因此可以将式 (17) 写为

$$\delta_{x,y}^{(l)} = h'(a_{x,y}^{(l)}) \left[\delta_{u,v}^{(l+1)} * rot180(w^{(l+1)})\right] \tag{18}$$

与全连接网络一样，我们最终的目标是计算误差函数 E 关于权值和偏差的导数。通过上述推导过程，可以得到

$$\begin{aligned}
\frac{\partial E}{\partial w_{m,n}^{(l)}} &= \sum_x \sum_y \frac{\partial E}{\partial a_{x,y}^{(l)}} \frac{\partial a_{x,y}^{(l)}}{\partial w_{m,n}} \\
&= \sum_x \sum_y \delta_{x,y}^{(l)} \frac{\partial a_{x,y}^{(l)}}{\partial w_{m,n}} \\
&= \sum_x \sum_y \delta_{x,y}^{(l)} \frac{\partial}{\partial w_{m,n}} \left[\sum_m \sum_n w_{m,n}^{(l)} h(a_{x-m,y-n}^{(l-1)}) + b^{(l)}\right] \\
&= \sum_x \sum_y \delta_{x,y}^{(l)} h(a_{x-m,y-n}^{(l-1)}) \\
&= \sum_x \sum_y \delta_{x,y}^{(l)} z_{x-m,y-n}^{(l-1)}
\end{aligned} \tag{19}$$

其中，最后一步从式 (7) 得出。这一行是一个卷积的形式，但是与式 (10) 相比，求和变量和对应的下标之间有一个符号逆转。把它写成卷积的形式，式 (19)

的最后一行可以写为

$$\begin{aligned}\frac{\partial E}{\partial w_{m,n}^{(l)}} &= \sum_x \sum_y \delta_{x,y}^{(l)} z_{-(m-x),-(n-y)}^{(l-1)} \\ &= \delta_{m,n}^{(l)} * z_{-m,-n}^{(l-1)} \\ &= \delta_{m,n}^{(l)} * rot180(z^{(l-1)})\end{aligned} \tag{20}$$

与上述推导类似，偏置的导数为

$$\begin{aligned}\frac{\partial E}{\partial b^{(l)}} &= \sum_x \sum_y \frac{\partial E}{\partial a_{x,y}^{(l)}} \frac{\partial a_{x,y}^{(l)}}{\partial b^{(l)}} \\ &= \sum_x \sum_y \delta_{x,y}^{(l)} \frac{\partial a_{x,y}^{(l)}}{\partial b^{(l)}} \\ &= \sum_x \sum_y \delta_{x,y}^{(l)} \frac{\partial}{\partial b^{(l)}} \left[\sum_m \sum_n w_{m,n}^{(l)} h(a_{x-m,y-n}^{(l-1)}) + b^{(l)} \right] \\ &= \sum_x \sum_y \delta_{x,y}^{(l)}\end{aligned} \tag{21}$$

利用梯度下降公式，可以得到

$$\begin{aligned}w_{m,n}^{(l)} &= w_{m,n}^{(l)} - \alpha \frac{\partial E}{\partial w_{m,n}^{(l)}} \\ &= w_{m,n}^{(l)} - \alpha \delta_{m,n}^{(l)} * rot180(z^{(l-1)})\end{aligned} \tag{22}$$

$$\begin{aligned}b^{(l)} &= b^{(l)} - \alpha \frac{\partial E}{\partial b^{(l)}} \\ &= b^{(l)} - \alpha \sum_x \sum_y \delta_{x,y}^{(l)}\end{aligned} \tag{23}$$

在 CNN 中，利用上述两个式子可以计算每个卷积层的权值和偏差。在前向传播中，从卷积层转到池化层。在反向传播中，朝着相反的方向前进。但是汇聚特征图小于其相应的特征图。因此，当反向传播时，对每个汇聚特征图进行上采样以匹配生成它的特征图的尺寸，每个汇聚的特征图对应于唯一的特征图，

因此反向传播的路径是清晰定义的。反向传播从全连接网络的输出开始，当到达神经网络和 CNN 之间的“接口”时，需要求助向量化方法生成输入矢量。也就是说，在进行反向传播之前，必须将全连接网络传递过来的单个向量转换成单个汇聚特征图。

三、教学目标考核（50 分钟）

讨论：

1. 卷积神经网络的卷积操作和池化操作是如何实现的？各有什么作用？
2. 卷积神经网络是如何构建的？
3. 卷积神经网络是如何实现自我学习的？

四、总结（5 分钟）

感知器模型可以算得上是深度学习的基石。最初的单层感知器模型就是为了模拟人脑神经元提出的，但是就连异或运算都无法模拟。经过多年的研究，人们终于提出了多层感知器模型，用于拟合任意函数。结合高效的 BP 算法，神经网络终于诞生。尽管目前看来，BP 神经网络已经无法胜任许多工作，但是从发展的角度来看，BP 神经网络仍是学习深度学习不可不知的重要部分。