

仅供课堂学习使用，请勿转发、转载

深度学习的数学

作者: [日]涌井良幸 / [日]涌井贞美

译者: 杨瑞龙

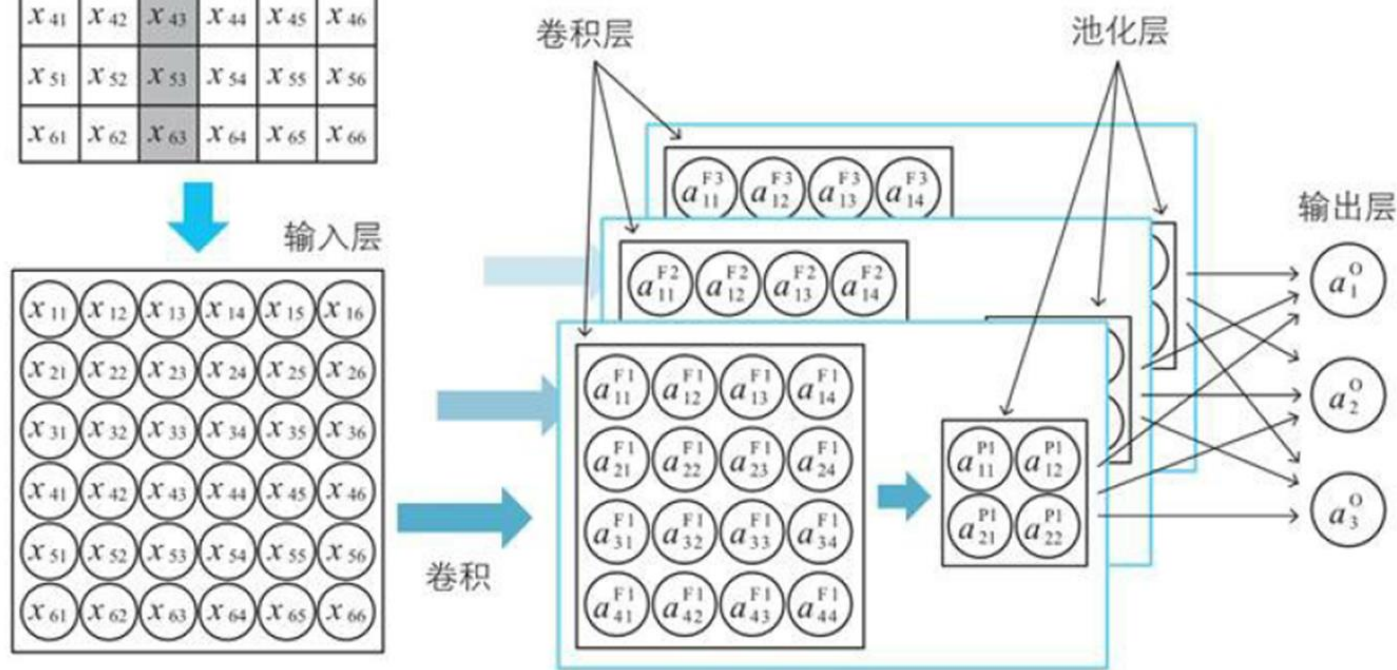
深度学习和卷积神经网络

- 深度学习是重叠了很多层的隐藏层（中间层）的神经网络。这样的神经网络使隐藏层具有一定的结构，从而更加有效地进行学习。

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}
x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}

图像
(下标为像素编号)

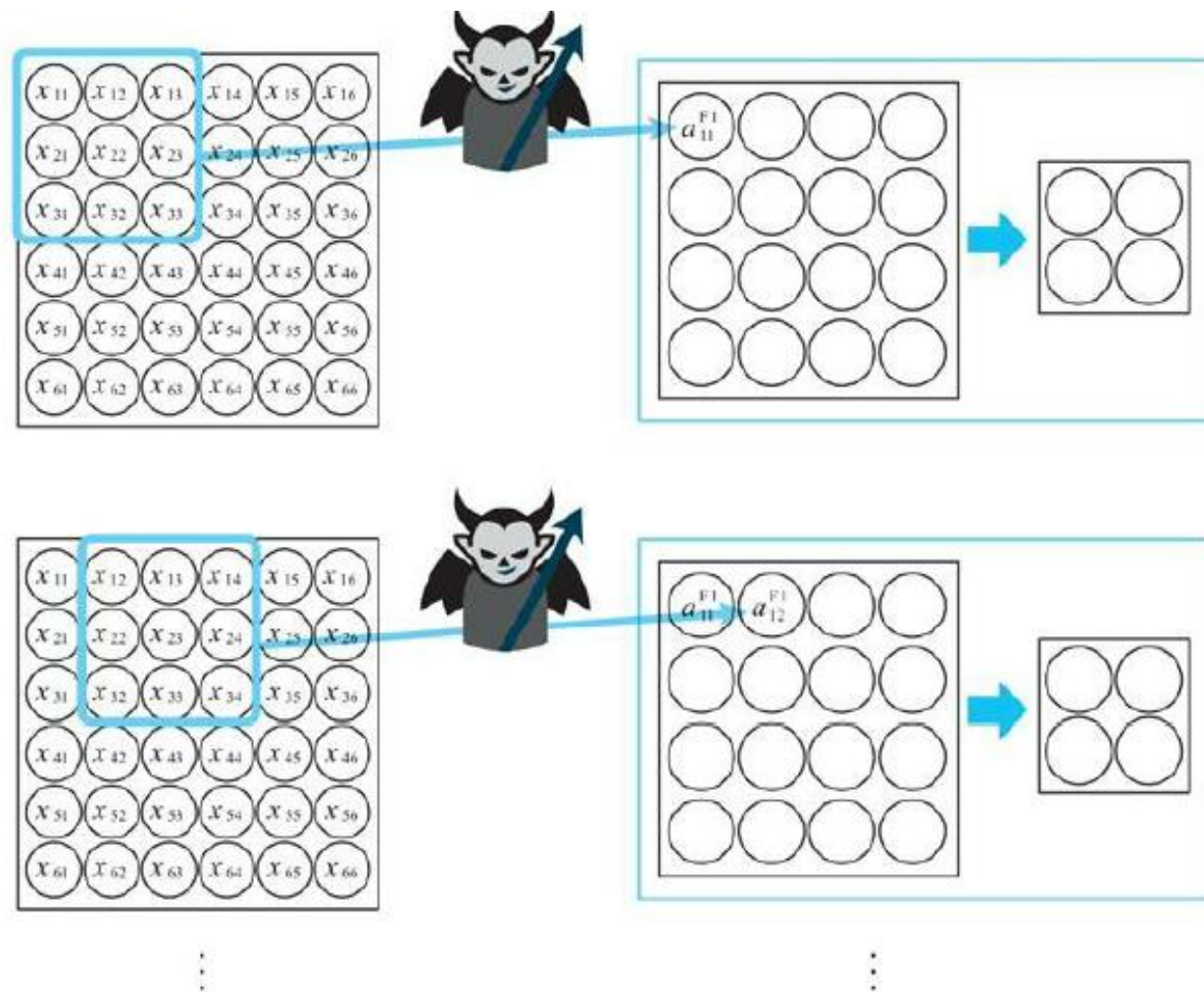
作为例题的解答示例的卷积神经网络的图。本章我们将讲解这个神经网络的图。图中的神经元名是后面将要考察的输出变量名(5-3节)。



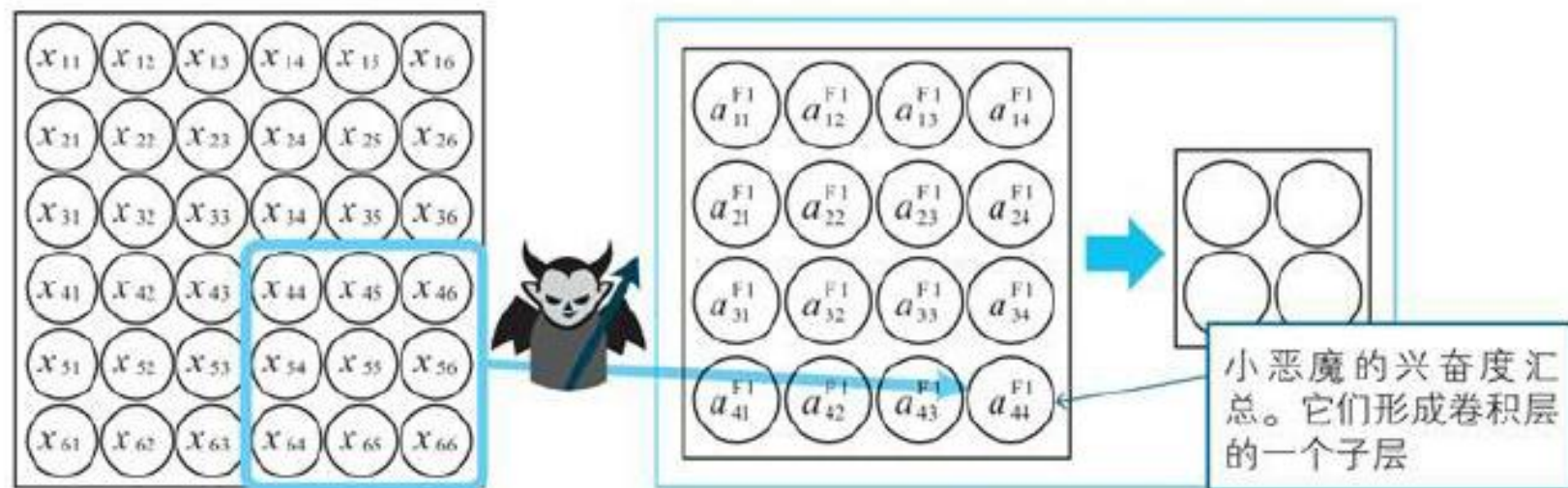
思路

- 住在隐藏层的恶魔具有各自偏好的模式。恶魔对自己偏好的模式做出反应，输出层接收这些信息，从而使神经网络进行模式识别成为可能。
- 本节登场的恶魔与之前的恶魔性格稍微有点不同。虽然他们的共同点都是具有自己偏好的模式，但是相比第 3 章登场的恶魔坐着一动不动，这里的恶魔是活跃¹的，他们会积极地从图像中找出偏好的模式，我们称之为小恶魔。
- 为了让这些小恶魔能够活动，我们为其提供工作场所，那就是由卷积层与池化层构成的隐藏子层。我们为每个小恶魔准备一个隐藏子层作为工作场所。

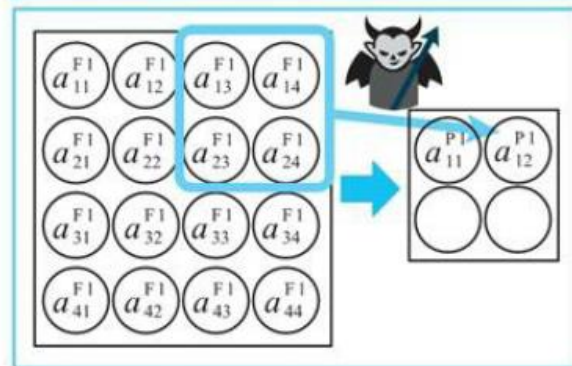
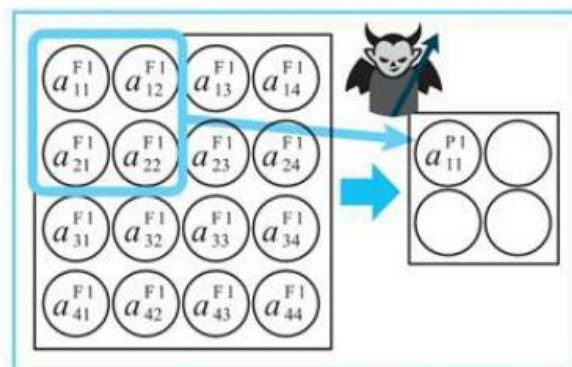
- 活跃的小恶魔积极地扫描图像，检查图像中是否含有自己偏好的模式。
- 如果图像中含有较多偏好的模式，小恶魔就很兴奋，反之就不兴奋。
- 此外，由于偏好的模式的大小比整个图像小，所以兴奋度被记录在多个神经元中。



小恶魔扫描图像数据，根据检测到的偏好模式的多少而产生兴奋，其兴奋度会被记录在卷积层的神经元中。神经元名中 F1 的 F 为 Filter 的首字母，1 为隐藏子层的编号。



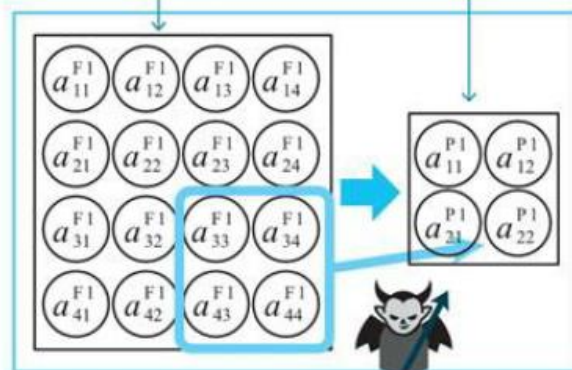
注：一般用于扫描的过滤器的大小是 5×5 。这里为了使结果变简单，我们使用如图所示的 3×3 的大小。



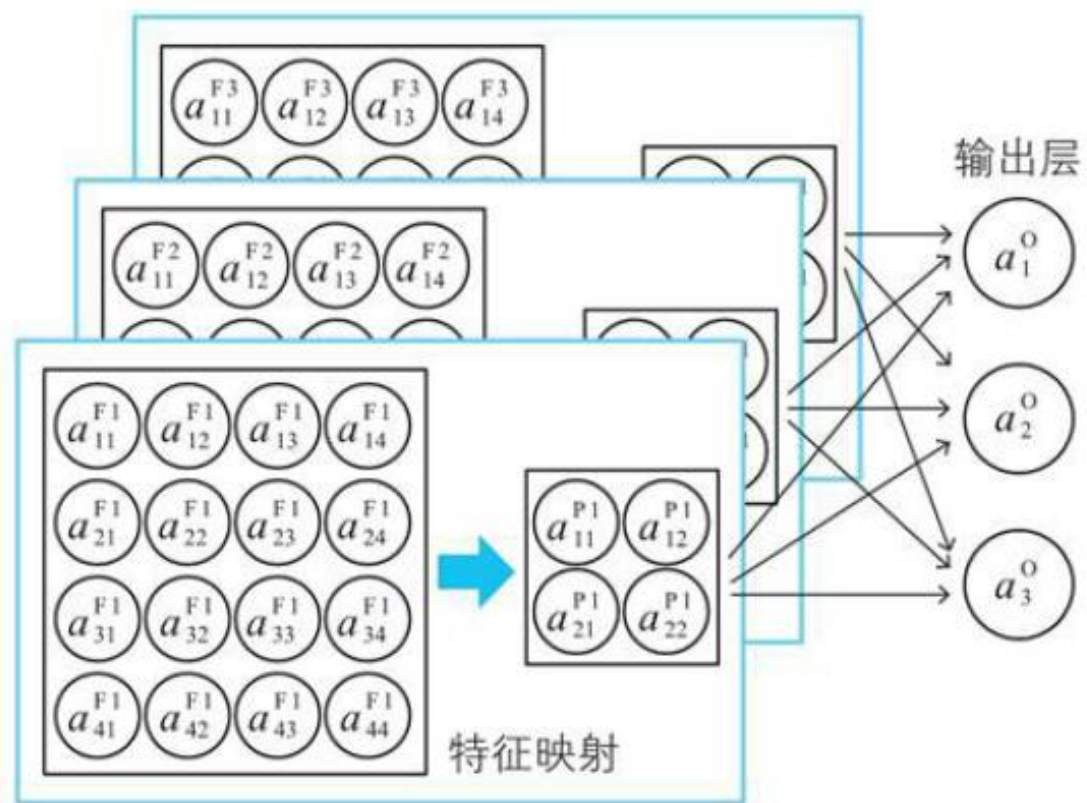
卷积层的
一个子层

:

池化层的
一个子层



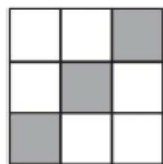
池化层的建立。小恶魔将扫描结果的兴奋度 (a_{11}^{F1} 等) 进一步集中起来, 整理为池化层的神经元。池化层中浓缩了小恶魔所偏好的模式的信息。神经元名中 P1 的 P 为 Pooling 的首字母, 1 为隐藏子层的编号。



输出层将3个小恶魔的报告进行汇总。为了分别对手写数字1、2、3产生较大反应，需要3个输出恶魔。

- 建立一个神经网络，用来识别通过 6×6 像素的图像读取的手写数字1、2、3。图像像素为单色二值。

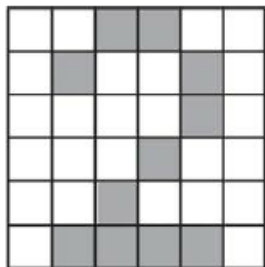
下面我们从数学角度来考察 5 - 1 节的小恶魔的工作。首先我们请小恶魔 S 登场。假定这个小恶魔 S 喜欢如下的模式 S。



小恶魔 S 偏好的模式 S。
(S 为 Slash (/) 的首字母。)

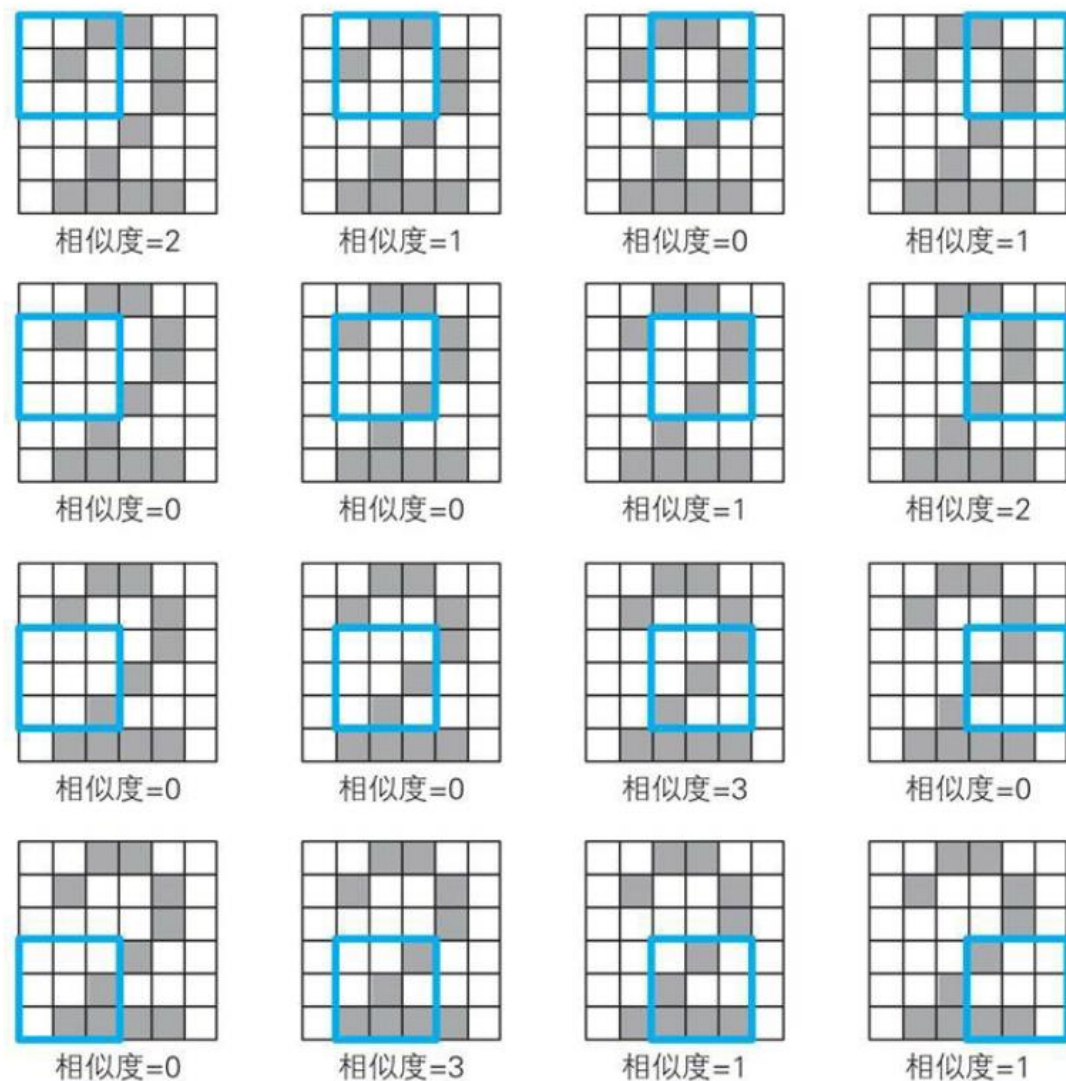
注：模式的大小通常为 5×5 。这里为了使结果变简单，我们使用图中所示的小的 3×3 模式。

假设下面的图像“2”就是要考察的图像。我们将手写数字 2 作为它的正解。



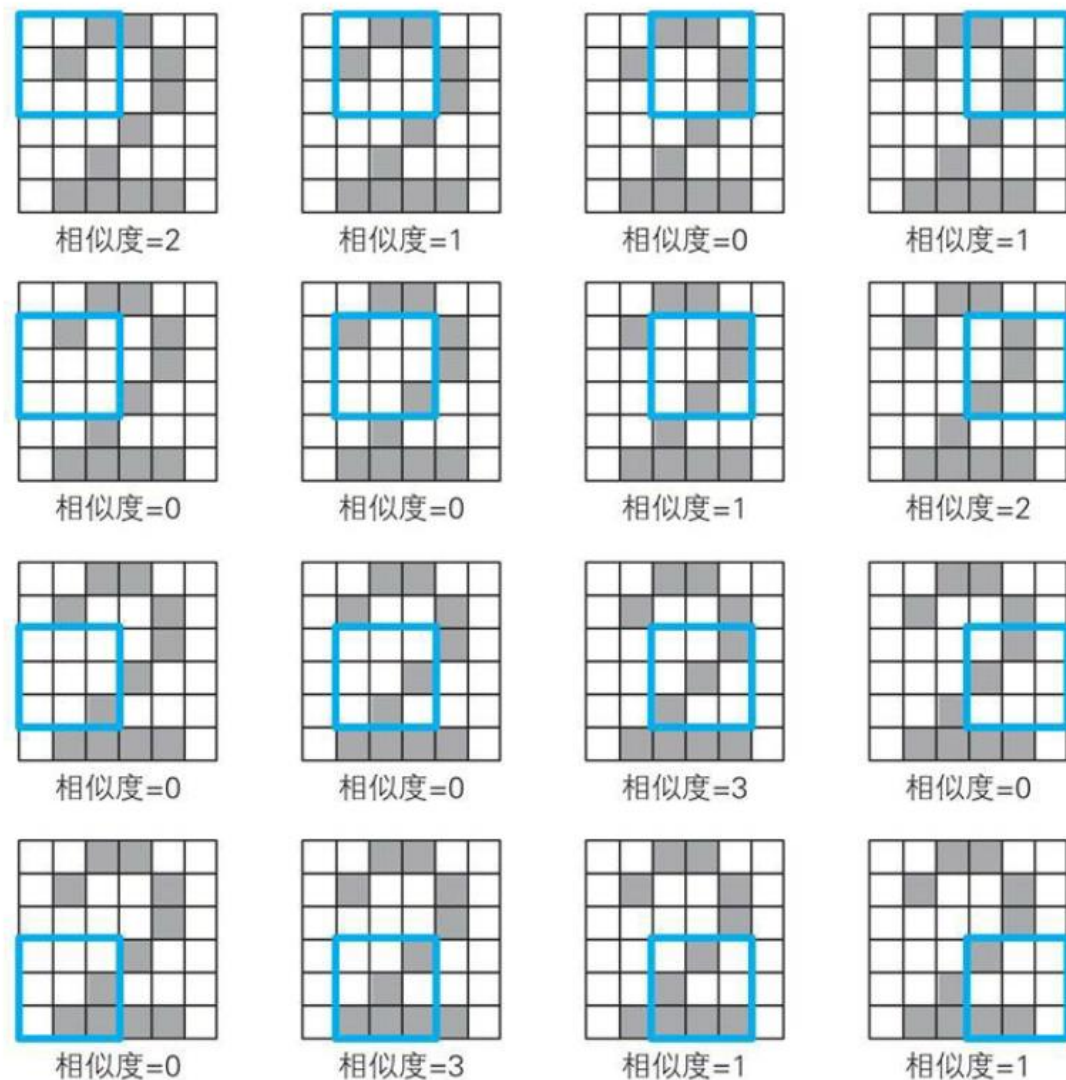
图像“2”。从数学角度考察小恶魔处理这个图像的过程。

- 小恶魔 S 首先将偏好的模式 S 作为过滤器对图像进行扫描。我们将这个过滤器命名为过滤器 S。接下来，我们实际用过滤器 S 扫描整个图像“2”。

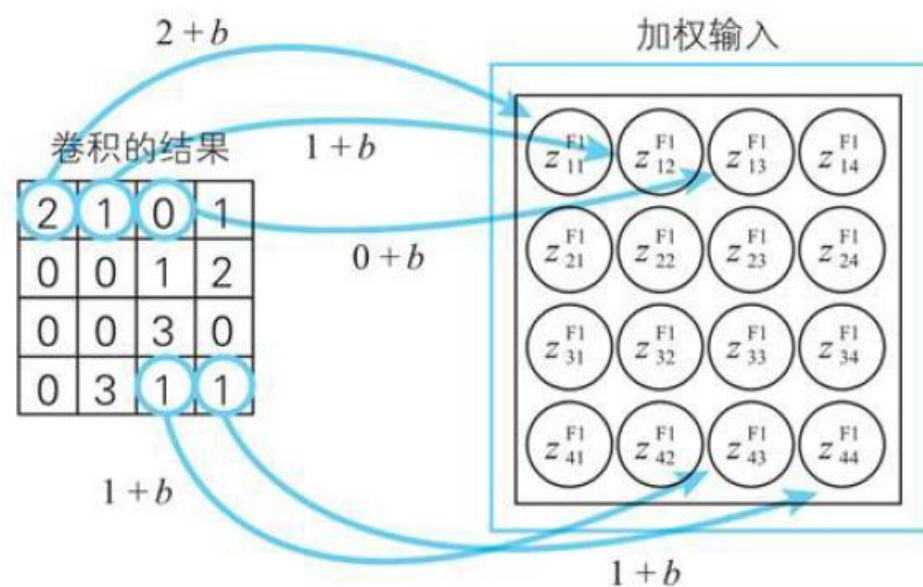


- 各个图像下面的“相似度”表示过滤器 S 的灰色格子部分与扫描图像块的灰色格子部分吻合的地方的个数。这个值越大，就说明越符合小恶魔偏好的模式。
- 我们将这个相似度汇总一下，如下表所示。这就是根据过滤器 S 得到的卷积（convolution）的结果，称为特征映射（feature map）。

2	1	0	1
0	0	1	2
0	0	3	0
0	3	1	1

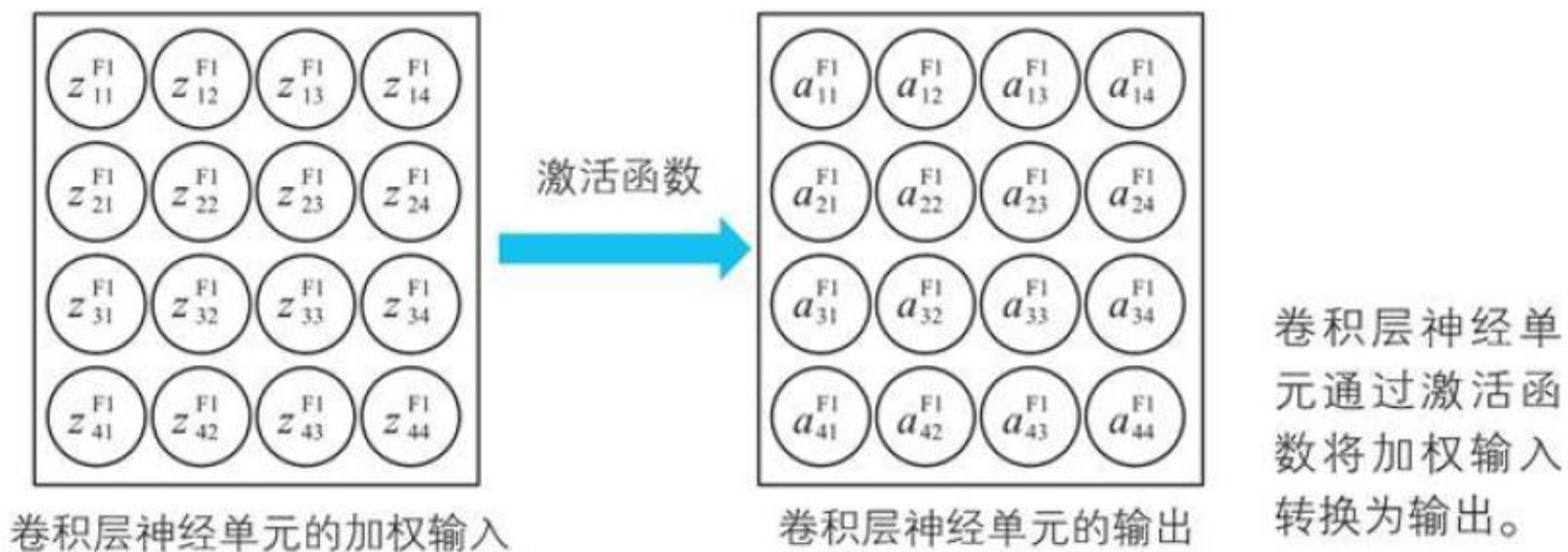


- 卷积层中的神经元将这一卷积的结果作为输入信息。各神经元将对应的卷积的值加上特征映射固有的偏置作为加权输入



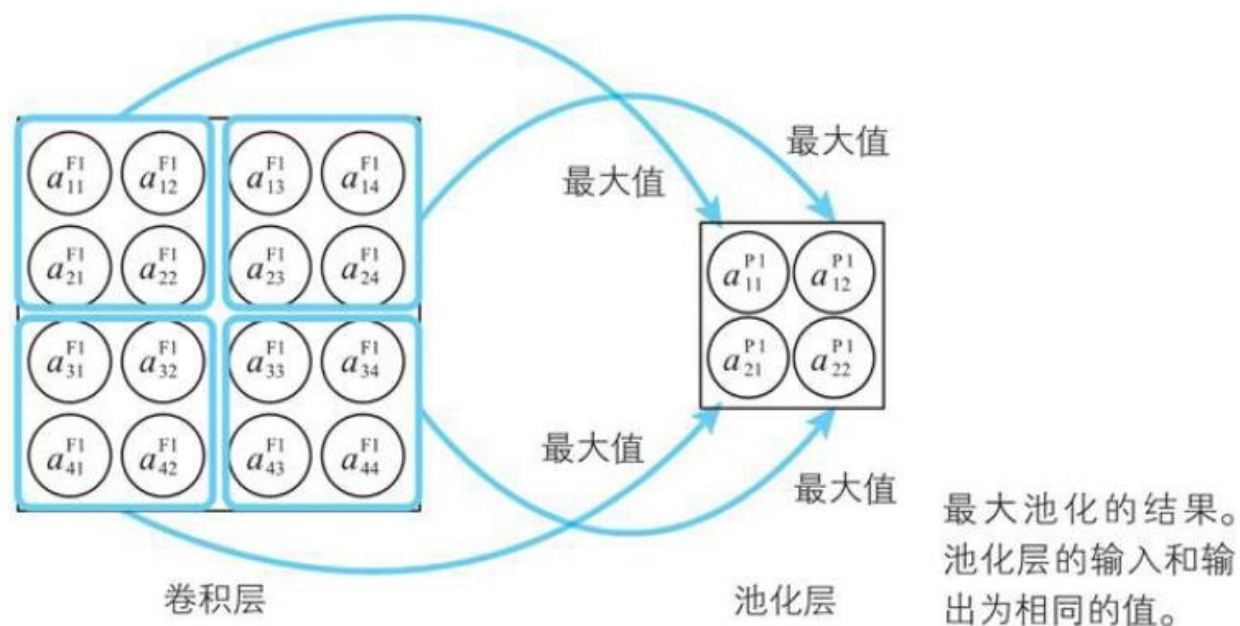
卷积层的神经单元的加权输入。请注意偏置 b 是相同的。此外，小恶魔 S 在编号 1 的隐藏子层中活动。

- 卷积层的各个神经单元通过激活函数来处理加权输入，并将处理结果作为神经单元的输出。这样卷积层的处理就完成了。

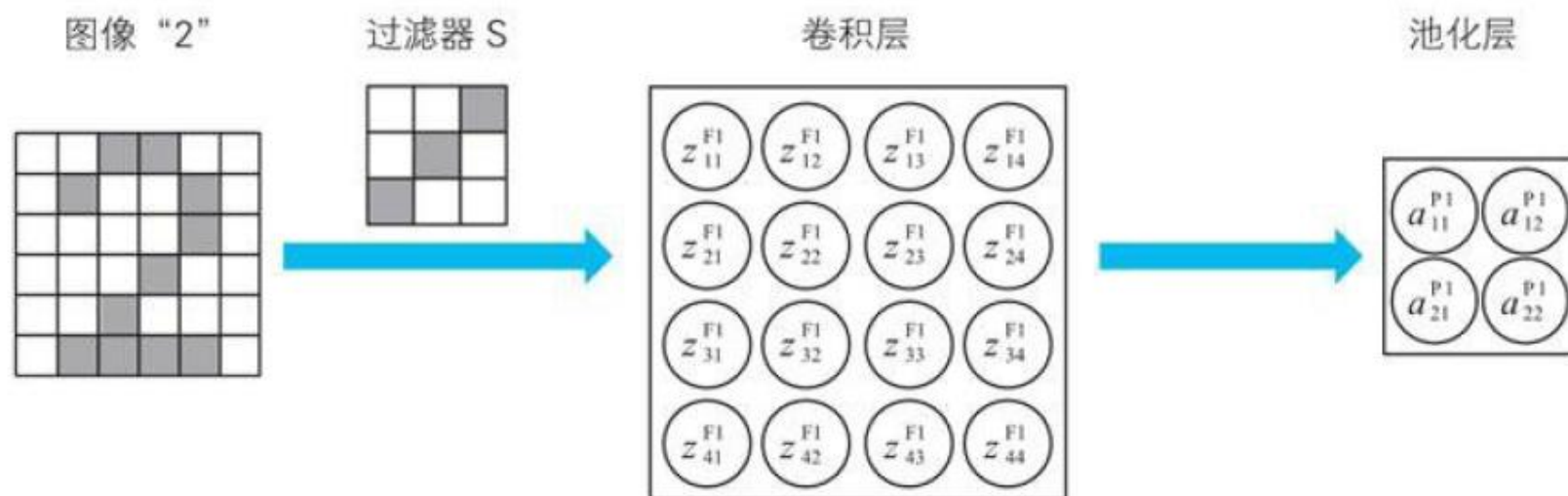


- 通过池化进行信息压缩

压缩的方法十分简单，只需要将卷积层神经单元划分为不重叠的 2×2 的区域，然后在各个区域中计算出代表值即可。本书中我们使用最有名的信息压缩方法最大池化（max pooling），具体来说就是将划分好的各区域的最大值提取出来。



这样一来，一张图像的信息就被集中在紧凑的神经元集合中了。

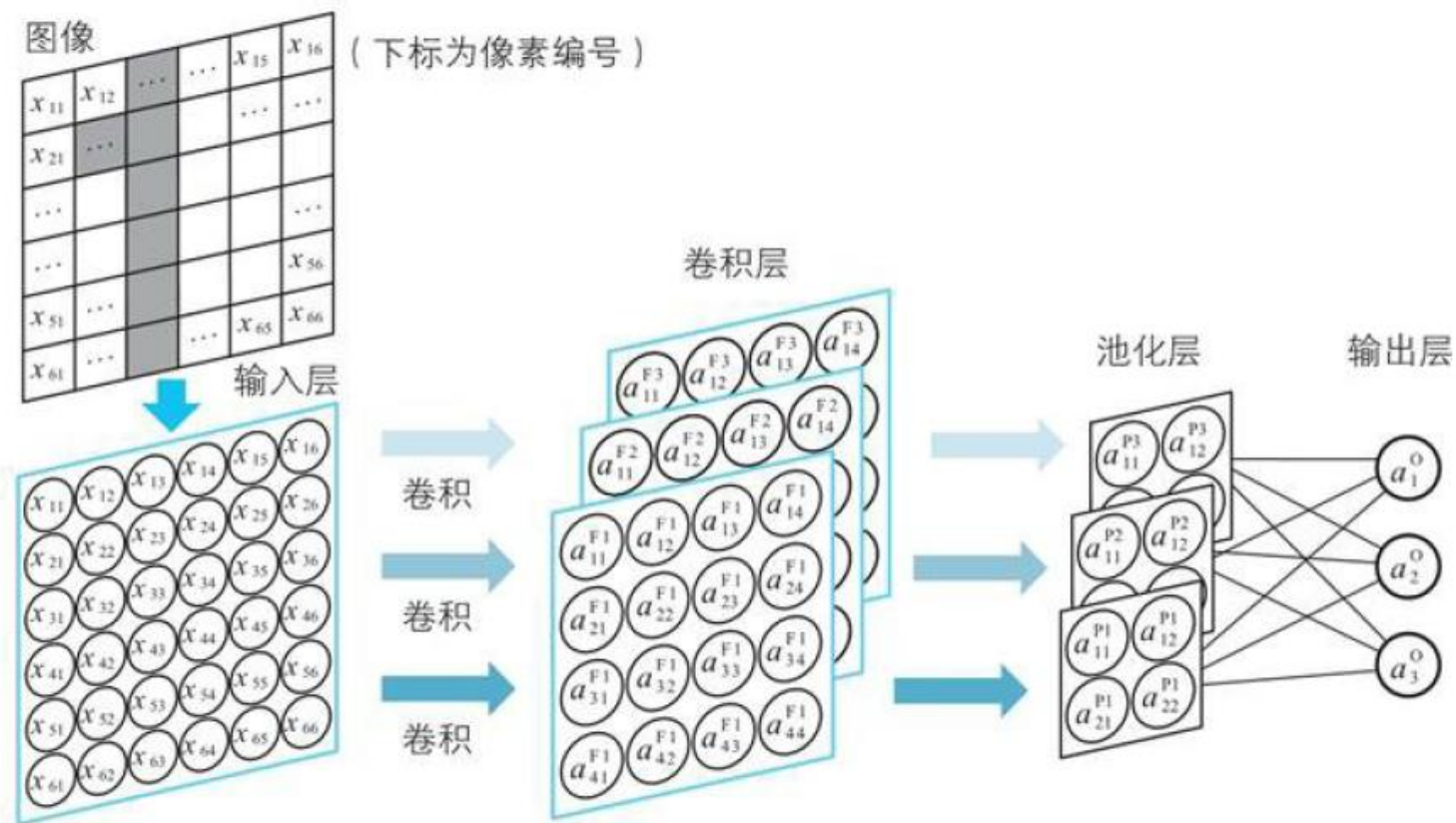


- 利用前面所示的图像“2”和过滤器 S 来实际计算卷积层和池化层中神经单元的输入输出值。设特征映射的偏置为 -1（阈值为 1），激活函数为 Sigmoid 函数。
- 如果池化层神经单元的输出值较大，就表示原始图像中包含较多的过滤器 S 的模式。



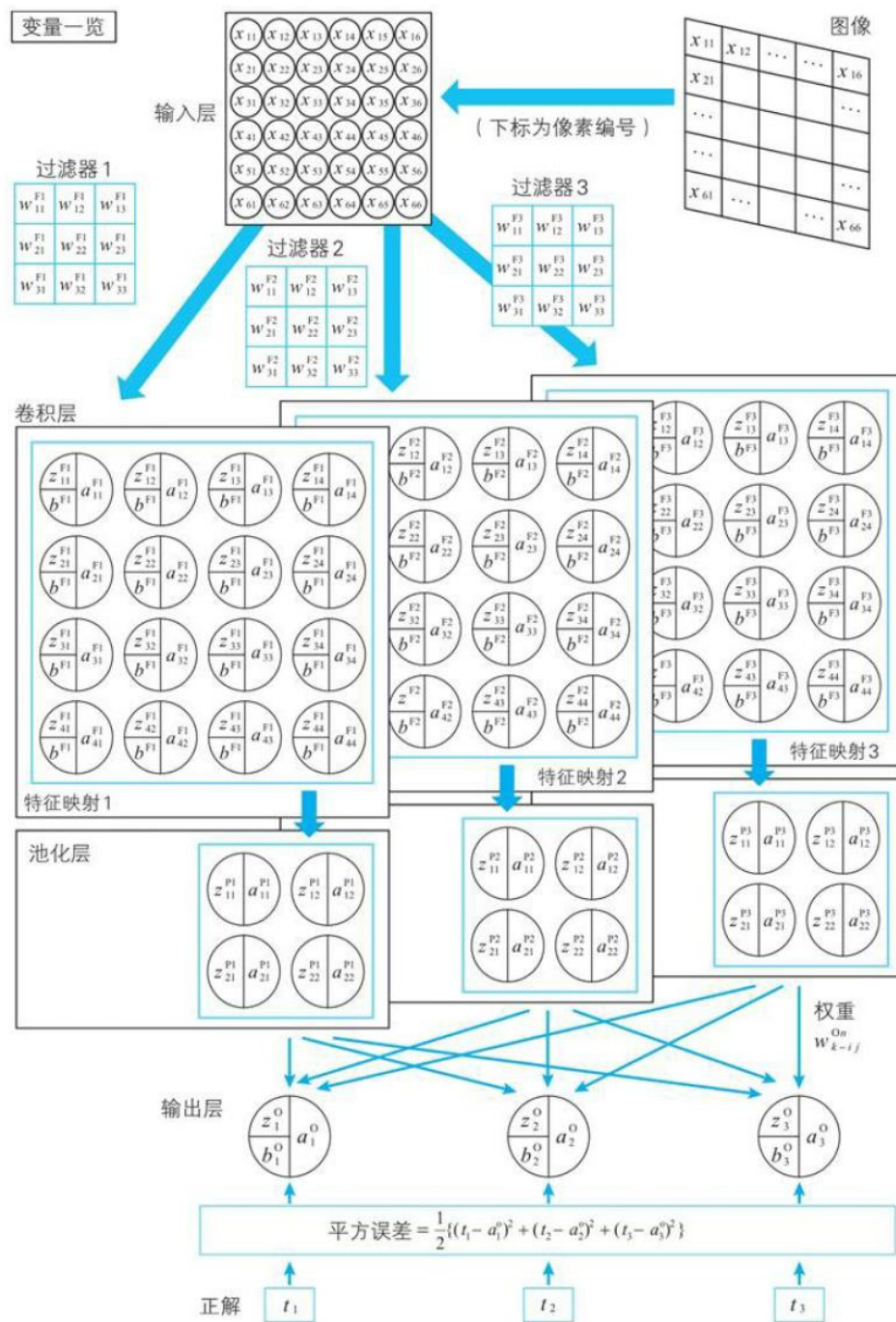
卷积神经网络的变量关系式

- 建立一个神经网络，用来识别通过 6×6 像素的图像读取的手写数字1、2、3。图像像素为单色二值。学习数据为 96 张图像。



位 置	符 号	含 义
输入层	x_{ij}	神经元中输入的图像像素 (i 行 j 列) 的值。与输出值相同
过滤器	w_{ij}^{Fk}	用于建立第 k 个特征映射的过滤器的 i 行 j 列的值。这里为了简化, 考虑 3×3 大小的过滤器 (通常采用 5×5 大小)
卷积层	z_{ij}^{Fk}	卷积层第 k 个子层的 i 行 j 列的神经单元的加权输入
	b^{Fk}	卷积层第 k 个子层的 i 行 j 列的神经单元的偏置。注意这些偏置在各特征映射中是相同的
	a_{ij}^{Fk}	卷积层第 k 个子层的 i 行 j 列的神经单元的输出 (激活函数的值)
池化层	z_{ij}^{Pk}	池化层第 k 个子层的 i 行 j 列的神经单元的输入。通常是前一层输出值的非线性函数值
	a_{ij}^{Pk}	池化层第 k 个子层的 i 行 j 列的神经单元的输出。与输入值 z_{ij}^{Pk} 一致
输出层	w_{k-ij}^{On}	从池化层第 k 个子层的 i 行 j 列的神经单元指向输出层第 n 个神经单元的箭头的权重
	z_n^o	输出层第 n 个神经单元的加权输入
	b_n^o	输出层第 n 个神经单元的偏置
	a_n^o	输出层第 n 个神经单元的输出 (激活函数的值)
学习数据	t_n	正解为1时, $t_1 = 1, t_2 = 0, t_3 = 0$ 正解为2时, $t_1 = 0, t_2 = 1, t_3 = 0$ 正解为3时, $t_1 = 0, t_2 = 0, t_3 = 1$

过滤器也称为核
(kernel)。



池化层的神经单元不存在权重和偏置的概念，也就是不具有模型参数。