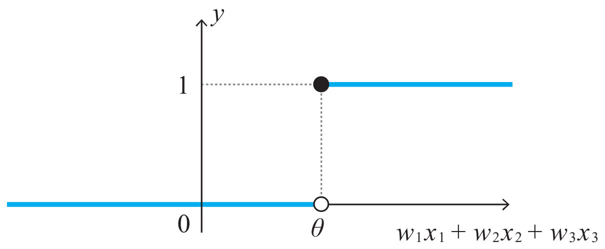
例 1 来自两个神经元 1、2 的输入信号分别为变量 x_1、x_2，权重为 w_1、w_2，神经元的阈值为 θ。当 $w_1=5$，$w_2=3$，$\theta=4$ 时，考察信号之和 $w_1x_1+w_2x_2$ 的值与表示点火与否的输出信号 y 的值。

输入 x_1	输入 x_2	和 $w_1x_1+w_2x_2$	点　火	输出信号 y
0	0	$5\times0+3\times0=0<4$	无	0
0	1	$5\times0+3\times1=3<4$	无	0
1	0	$5\times1+3\times0=5\geqslant4$	有	1
1	1	$5\times1+3\times1=8\geqslant4$	有	1

点火条件的图形表示

下面我们将表示点火条件的式 (2) 图形化。以神经元的输入信号之和为横轴，神经元的输出信号 y 为纵轴，将式 (2) 用图形表示出来。如下图所示，当信号之和小于 θ 时，y 取值 0，反之 y 取值 1。

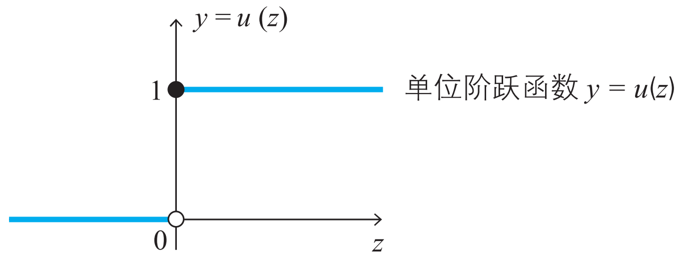


将点火条件图形化。横轴表示信号之和 $w_1x_1+w_2x_2+w_3x_3$。

如果用函数式来表示这个图形，就需要用到下面的**单位阶跃函数**。

$$u(z)=\begin{cases}0 & (z<0)\\ 1 & (z\geqslant 0)\end{cases}$$

单位阶跃函数的图形如下所示。



利用单位阶跃函数 $u(z)$，式 (2) 可以用一个式子表示如下。

点火的式子：$y = u(w_1x_1 + w_2x_2 + w_3x_3 - \theta)$　　(3)

通过下表可以确认式 (3) 和式 (2) 是一样的。

y	$w_1x_1 + w_2x_2 + w_3x_3$	$z = w_1x_1 + w_2x_2 + w_3x_3 - \theta$	$u(z)$
0（无点火）	小于 θ	$z < 0$	0
1（点火）	大于等于 θ	$z \geqslant 0$	1

此外，该表中的 z（式 (3) 的阶跃函数的参数）的表达式

$$z = w_1x_1 + w_2x_2 + w_3x_3 - \theta \tag{4}$$

称为该神经元的**加权输入**。

Memo

备注　$w_1x_1 + w_2x_2 + w_3x_3 = \theta$ 的处理

有的文献会像下面这样处理式 (2) 的不等号。

$$\begin{cases} \text{无输出信号}（y=0）：w_1x_1 + w_2x_2 + w_3x_3 \leqslant \theta \\ \text{有输出信号}（y=1）：w_1x_1 + w_2x_2 + w_3x_3 > \theta \end{cases}$$

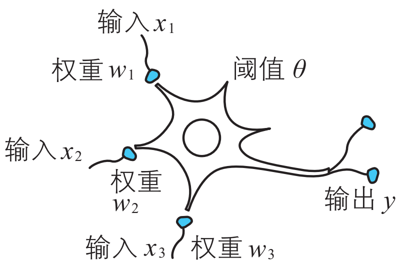
在生物上这也许是很大的差异，不过对于接下来的讨论而言是没有问题的。因为我们的主角是 Sigmoid 函数，所以不会发生这样的问题。

1-3 激活函数：将神经元的工作一般化

1-2 节中用数学式表示了神经元的工作。本节我们试着将其在数学上一般化。

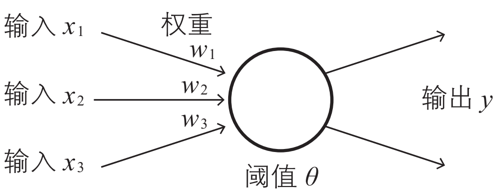
简化神经元的图形

为了更接近神经元的形象，1-2 节中将神经元表示为了下图的样子。



神经元的示意图（3 个输入、2 个输出的情况）。轴突分岔为两个输出端，其输出值相同。

然而，为了画出网络，需要画很多的神经元，在这种情况下上面那样的图就不合适了。因此，我们使用如下所示的简化图，这样很容易就能画出大量的神经元。



该图是神经元的简化图。用箭头方向区分输入和输出。神经元的输出由两个箭头指出，其值是相同的。

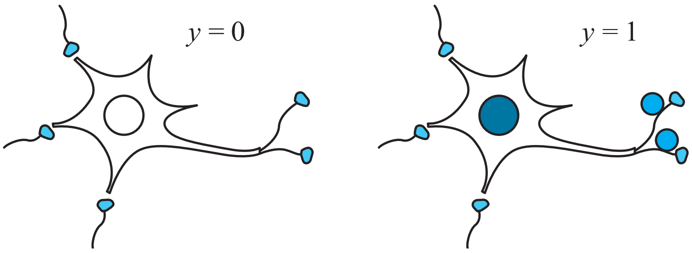
为了与生物学的神经元区分开来，我们把经过这样简化、抽象化的神经元称为**神经单元**（unit）。

注：很多文献直接称为“神经元”。本书为了与生物学术语“神经元”区分，使用“神经单元”这个称呼。另外，也有文献将“神经单元”称为“人工神经元”，但是由于现在也存在生物上的人工神经元，所以本书中也不使用“人工神经元”这个称呼。

激活函数

将神经元的示意图抽象化之后，对于输出信号，我们也对其生物上的限制进行一般化。

根据点火与否，生物学上的神经元的输出 y 分别取值 1 和 0（下图）。



点火与否用 1 和 0 表示。

然而，如果除去“生物”这个条件，这个“0 和 1 的限制”也应该是可以解除的。这时表示点火与否的下式（1-2 节式 (3)）就需要修正。

$$点火的式子：y = u(w_1x_1 + w_2x_2 + w_3x_3 - \theta) \tag{1}$$

这里，u 是单位阶跃函数。我们将该式一般化，如下所示。

$$\boxed{y = a(w_1x_1 + w_2x_2 + w_3x_3 - \theta) \qquad (2)}$$

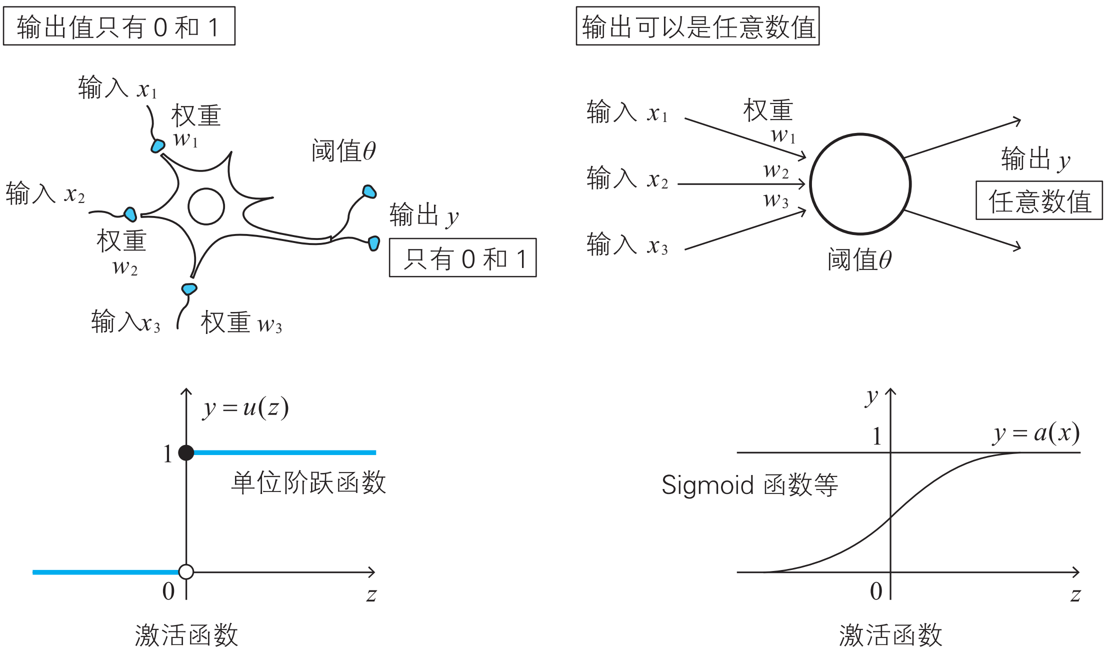
这里的函数 a 是建模者定义的函数，称为**激活函数**（activation function）。x_1、x_2、x_3 是模型允许的任意数值，y 是函数 a 能取到的任意数值。这个式 (2) 就是今后所讲的神经网络的出发点。

注：虽然式 (2) 只考虑了 3 个输入，但这是很容易推广的。另外，式 (1) 使用的单位阶跃函数 $u(z)$ 在数学上也是激活函数的一种。

请注意，式 (2) 的输出 y 的取值并不限于 0 和 1，对此并没有简单的解释。一定要用生物学来比喻的话，可以考虑神经单元的“兴奋度”“反应度”“活性度”。

我们来总结一下神经元和神经单元的不同点，如下表所示。

	神经元	神经单元
输出值y	0或1	模型允许的任意数值
激活函数	单位阶跃函数	由分析者给出，其中著名的是Sigmoid函数（后述）
输出的解释	点火与否	神经单元的兴奋度、反应度、活性度



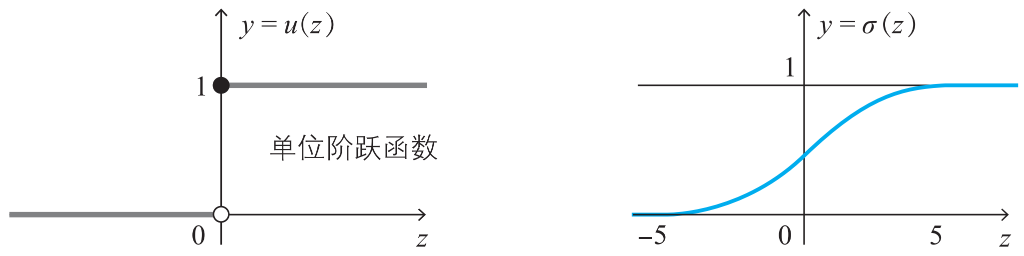
将神经元点火的式 (1) 一般化为神经单元的激活函数式 (2)，要确认这样做是否有效，就要看实际做出的模型能否很好地解释现实的数据。实际上，式 (2) 表示的模型在很多模式识别问题中取得了很好的效果。

Sigmoid 函数

激活函数的代表性例子是 Sigmoid 函数 $\sigma(z)$，其定义如下所示。

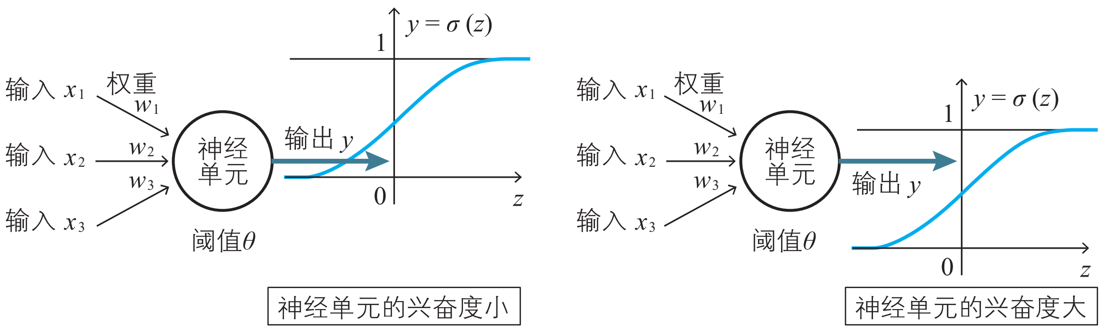
$$\sigma(z)=\frac{1}{1+\mathrm{e}^{-z}} \quad (\mathrm{e}=2.718281\ldots) \tag{3}$$

关于这个函数，我们会在后面详细讨论（2-1 节）。这里先来看看它的图形，Sigmoid 函数 $\sigma(z)$ 的输出值是大于 0 小于 1 的任意值。此外，该函数连续、光滑，也就是说可导。这两种性质使得 Sigmoid 函数很容易处理。



右图是激活函数的代表性例子 Sigmoid 函数 $\sigma(z)$ 的图形。除了原点附近的部分，其余部分与单位阶跃函数（左图）相似。Sigmoid 函数具有处处可导的性质，很容易处理。

单位阶跃函数的输出值为 1 或 0，表示点火与否。然而，Sigmoid 函数的输出值大于 0 小于 1，这就有点难以解释了。如果用生物学术语来解释的话，如上文中的表格所示，可以认为输出值表示神经单元的兴奋度等。输出值接近 1 表示兴奋度高，接近 0 则表示兴奋度低。



本书中将 Sigmoid 函数作为标准激活函数使用，因为它具有容易计算的漂亮性质。如果用数学上单调递增的可导函数来代替，其原理也是一样的。

偏置

再来看一下激活函数的式 (2)。

$$y = a(w_1x_1 + w_2x_2 + w_3x_3 - \theta) \tag{2}$$

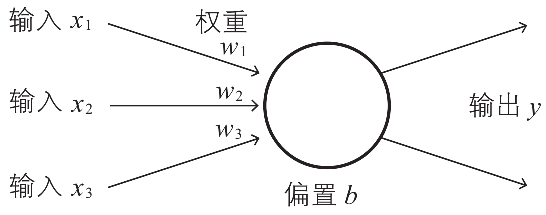
这里的 θ 称为阈值，在生物学上是表现神经元特性的值。从直观上讲，θ 表示神经元的感受能力，如果 θ 值较大，则神经元不容易兴奋（感觉迟

钝），而如果值较小，则神经元容易兴奋（敏感）。

然而，式 (2) 中只有 θ 带有负号，这看起来不漂亮。数学不喜欢不漂亮的东西。另外，负号具有容易导致计算错误的缺点，因此，我们将 $-\theta$ 替换为 b。

$$y = a(w_1x_1 + w_2x_2 + w_3x_3 + b) \tag{4}$$

经过这样处理，式子变漂亮了，也不容易发生计算错误。这个 b 称为**偏置**（bias）。



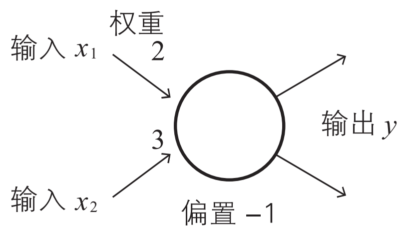
本书将式 (4) 作为标准使用。另外，此时的加权输入 z（1-2 节）如下所示。

$$z = w_1x_1 + w_2x_2 + w_3x_3 + b \tag{5}$$

式 (4) 和式 (5) 是今后所讲的神经网络的出发点，非常重要。

另外，生物上的权重 w_1、w_2、w_3 和阈值 θ（$=-b$）都不是负数，因为负数在自然现象中实际上是不会出现的。然而，在将神经元一般化的神经单元中，是允许出现负数的。

问题 右图是一个神经单元。如图所示，输入 x_1 的对应权重是 2，输入 x_2 的对应权重是 3，偏置是 -1。根据下表给出的输入，求出加权输入 z 和输出 y。注意这里的激活函数是 Sigmoid 函数。



输入 x_1	输入 x_2	加权输入 z	输出 y
0.2	0.1		
0.6	0.5		

解 结果如下表所示（式 (3) 中的 e 取 e = 2.7 进行计算）。

输入 x_1	输入 x_2	加权输入 z	输出 y
0.2	0.1	$2\times0.2+3\times0.1-1=-0.3$	0.43
0.6	0.5	$2\times0.6+3\times0.5-1=1.7$	0.84

Memo 备注 改写式 (5)

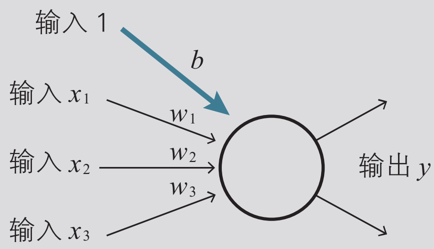
我们将式 (5) 像下面这样整理一下。

$$z=w_1x_1+w_2x_2+w_3x_3+b\times1 \tag{6}$$

这里增加了一个虚拟的输入，可以理解为以常数 1 作为输入值（右图）。

于是，加权输入 z 可以看作下面两个向量的内积。

$$(w_1,\ w_2,\ w_3,\ b)\,(x_1,\ x_2,\ x_3,\ 1)$$



计算机擅长内积的计算，因此按照这种解释，计算就变容易了。

1-4 什么是神经网络

神经网络作为本书的主题，它究竟是什么样的呢？下面让我们来看一下其概要。

神经网络

上一节我们考察了神经单元，它是神经元的模型化。那么，既然大脑是由神经元构成的网络，如果我们模仿着创建神经单元的网络，是不是也能产生某种“智能”呢？这自然是让人期待的。众所周知，人们的期待没有被辜负，由神经单元组成的网络在人工智能领域硕果累累。

在进入神经网络的话题之前，我们先来回顾一下上一节考察过的神经单元的功能。

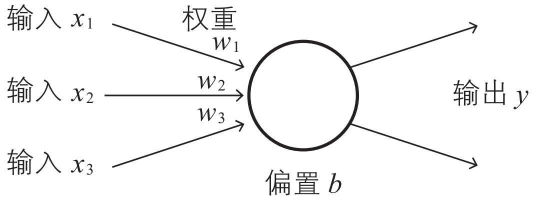
· 将神经单元的多个输入 $x_1, x_2, \cdots, x_n$ 整理为加权输入 z。

$$z = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b \quad (1)$$

其中 $w_1, w_2, \cdots, w_n$ 为权重，b 为偏置，n 为输入的个数。

· 神经单元通过激活函数 $a(z)$，根据加权输入 z 输出 y。

$$y = a(z) \quad (2)$$



神经单元具有如上总结的运算功能。另外，即使有多个输出，其值也相同。

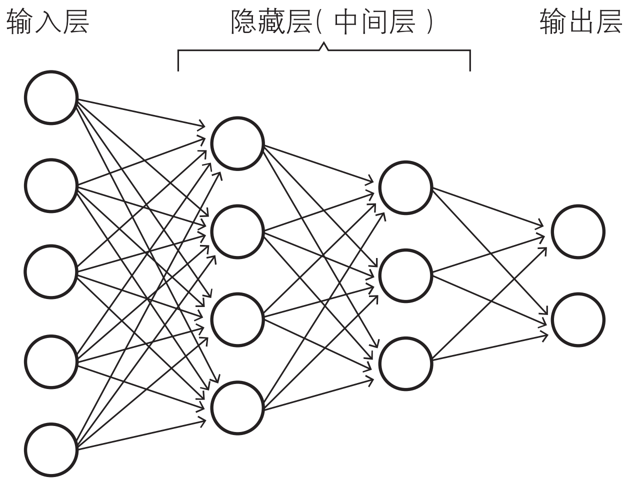
将这样的神经单元连接为网络状，就形成了**神经网络**。

网络的连接方法多种多样，本书将主要考察作为基础的**阶层型神经网络**以及由其发展而来的**卷积神经网络**。

注：为了与生物学上表示神经系统的神经网络区分开来，有的文献使用“人工神经网络”这个称呼。本书中为了简便，省略了“人工”二字。

神经网络各层的职责

阶层型神经网络如下图所示，按照**层**（layer）划分神经单元，通过这些神经单元处理信号，并从输出层得到结果，如下图所示。



阶层型神经网络的示例。除了阶层型以外，还有“互相连接型”等各种类型的网络。

构成这个网络的各层称为**输入层**、**隐藏层**、**输出层**，其中隐藏层也被称为**中间层**。

各层分别执行特定的信号处理操作。

输入层负责读取给予神经网络的信息。属于这个层的神经单元没有输入箭头，它们是简单的神经单元，只是将从数据得到的值原样输出。

隐藏层的神经单元执行前面所复习过的处理操作 (1) 和 (2)。在神经网络中，这是实际处理信息的部分。

输出层与隐藏层一样执行信息处理操作 (1) 和 (2)，并显示神经网络计算出的结果，也就是整个神经网络的输出。

深度学习

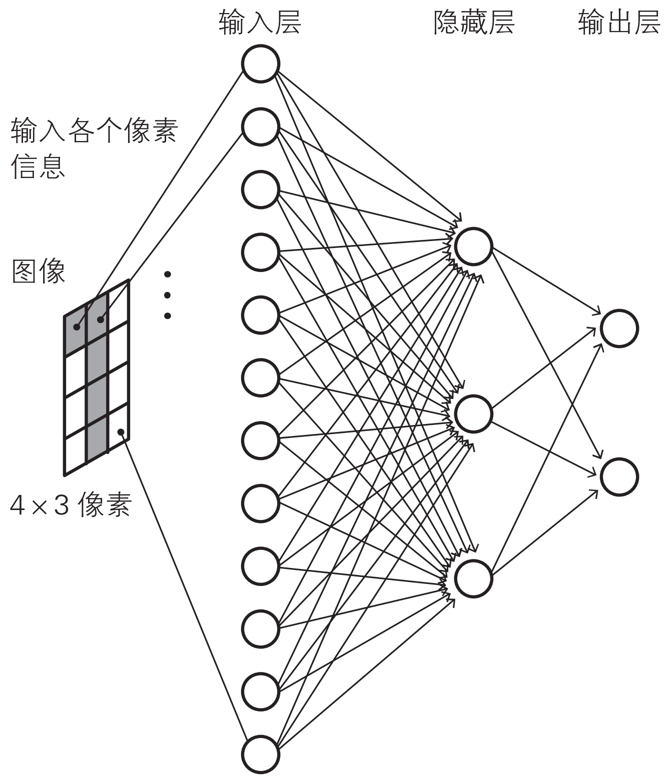
深度学习，顾名思义，是叠加了很多层的神经网络。叠加层有各种各样的方法，其中著名的是**卷积神经网络**（第 5 章）。

考察具体的例子

从现在开始一直到第 4 章，我们都将围绕着下面这个简单的例子来考察神经网络的结构。

> **例题** 建立一个神经网络，用来识别通过 4×3 像素的图像读取的手写数字 0 和 1。学习数据是 64 张图像，其中像素是单色二值。

解 我们来示范一下这个**例题**如何解答。



作为**例题**解答的神经网络示例。这个示例将手写数字 1 作为单色二值图像读入。

这个解答是演示实际的神经网络如何发挥功能的最简单的神经网络