第一章 神经网络

创建智能机器

大脑是人体中最不可思议的器官，它掌管着五识：眼识、耳识、鼻识、舌识和身识。使得人们能够存储记忆，体验情感，甚至是做梦。没有大脑，我们将变成原始生物，除了最简单的条件反射什么都做不了。从根本上来说，大脑使我们变得聪明。

婴儿的大脑只有一磅重，但不知道什么原因，它能够解决甚至最强大的超级计算机都无能为力的问题。在出生几个月后，婴儿就可以认出父母的面容，从背景中把各个物体识别出来，甚至能够辨别声音。在一年内，婴儿已经形成了对自然物理的直觉，能够追踪物体，即使它们被部分或全部遮挡；能够把声音与某个特定含义关联起来。在童年早期，它们对所知道的词语中的上千个单词和语法有了复杂的理解。

数十载以来，我们梦想着制造出有像人类一样大脑的智能机器-打扫家庭卫生的机器人助手，无人驾驶的汽车，自动检测疾病的显微镜。但是制造这样的人工智能机器需要解决一些困扰我们已久的最复杂的计算问题，而我们的大脑在几毫秒内就能够解决。为了处理这些问题，我们必须要开发一种与过去几十年大力研发的计算机编程技术完全不同的方法，这就是深度学习(deep learning)，一个相当活跃的人工计算机智能领域。

传统计算机程序的局限性

对计算机来说，为什么“准确”是如此难解决的问题？传统的计算机程序被设计的擅长做两件事情：1)算术运算相当快；2）明确地遵循一组指令。所以如果要处理大量金融数据，那么你是幸运的。传统计算机程序能够处理这些，但是我们想做一些更有趣的事情，例如写一个程序自动识别某人的笔迹。图1-1将作为起点。

图1-1 Image from MNIST handwritten digit dataset

尽管图1-1中的每个数字用稍微不同的方式书写，我们还是能够轻松地识别出第一行的每个数字为0，第二行的每个数字为1等等。让我们尝试写段计算机程序来破解这个任务，使用什么样的规则能够把某个数字与其他数字区分开？

好吧，我们先从简单的规则开始！例如，我们可以认为是0如果图像只有一个闭合的圆圈。图1-1中所有的例子似乎是满足这个规则的，但是这并不是充分条件。要是某人在写0时没有完全闭合圆圈该怎么办？正如图1-2所示，你怎么才能把凌乱的0和6区分开？

你可以为圆圈起点和终点之间的距离设置一个截断值，但是我们并不

图1-2 A zero that's algorithmically difficult to distinguish from a six

清楚要把截断值设为多大。这个两难的窘境只是各种折磨的开始。我们如何区分3和5？或者区分4和9？我们可以通过仔细的观察和数月的试错来不断增加越来越多的规则或者特征，但是很明显这并不是一个非常容易的过程。

许多其他类型的问题也属于同一类问题：目标识别、语言理解、自动翻译等。我们不知道要写什么样的程序，因为大脑工作的机制我们并不清楚。即使我们知道了大脑的工作机制，程序的复杂度也是叹为观止的。

机器学习机理

为了解决这些问题，我们必须要使用全新的方法。在学校的岁月里，我们学习的很多东西与传统计算机程序是一样的。我们学习如何乘数，解方程，通过消化理解一堆规则求导数。但是我们小时候学到的最自然的东西都是通过例子学会的，而不是公式。

例如，在我们两岁的时候，父母并没有教我们如何通过量鼻子的形状或者身体的外形来识别小狗。我们通过多个示范的实例和认错了就纠正来学会识别小狗。换句话说，当我们出生时，大脑给我们提供了如何能够观察世界的模型。随着逐渐长大，模型吸收我们的感官输入，对我们所经历

的进行猜测。如果猜测得到了父母的确认，模型会被强化。如果父母说我们猜错了，我们将会修正模型来包含这个新的信息。在一生当中，随着理解消化越来越多的实例，我们的模型变得越来越准确。显然所有这些活动都是下意识进行的，我们甚至都没有意识到，不过这个能力使我们更有优势。

机器学习是人工智能更加普通化的领域，它基于实例学习来实现预测，深度学习是机器学习的一个分支。在机器学习中，不再用一大堆规则来指示计算机解决问题，而是提供一个可以评估实例的模型和指导计算机计算错误时修正模型的少量指令。我们期望随着时间的流逝，适当的模型将能够非常准确地解决问题。

我们把这个思想数学化，更加严谨地阐述它的涵义。把模型定义为函数$h(x,\theta)$。输入x是用向量表示的实例。例如图1-3所示，如果x是灰度图像，向量的每一项代表每个位置的像素亮度。

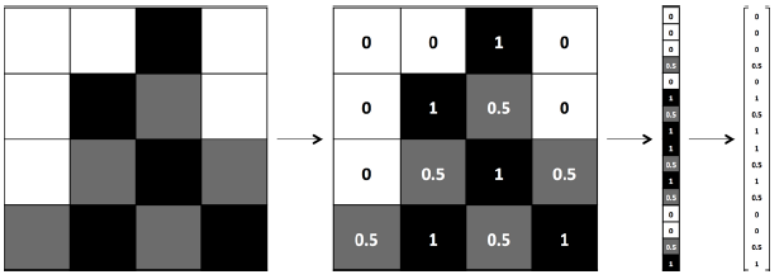

图1-3 The process of vectorizing an image for a machine learning algorithm

输入 θ 是模型的参数向量。随着学习的实例越来越多，机器学习程序努力去优化这些参数的取值。我们将在实战中观察参数优化过程，更多的细节详见第二章。

为了更加直观理解机器学习模型，我们来剖析一个简单的例子。假设我们想确定基于前一晚的学习时长和睡眠时长如何预测考试效果。我们收集大量的数据，对每个数据点 $x=[x_1,x_2]^T$，我们用x_1记录睡眠时长，x_2是花费在学习上的时长，还记录考试结果是否高于或低于班级平均水平。我

们的目标是用参数向量 $\theta = [\theta_0\ \theta_1\ \theta_2]^T$学习到如下的模型$h(x,\theta)$：

$$h(x,\theta) = \begin{cases} -1, & if \quad x^T \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \theta_0 < 0 \\ \\ 1, & if \quad x^T \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \theta_0 \geq 0 \end{cases}$$

换句话说，我们假设模型$h(x,\theta)$ 的设计图如上所述(在几何上，这个特别的设计图描绘了一个线性分类器，将坐标面分为两半)。然后，我们训练参数向量 θ，使得模型对给定输入实例x做出正确的预测(如果考试结果在平均分以下为1，否则为-1)。该模型称为线性感知机，从二十世纪五十年代开始就已经使用了。假设数据如图1-4所示。

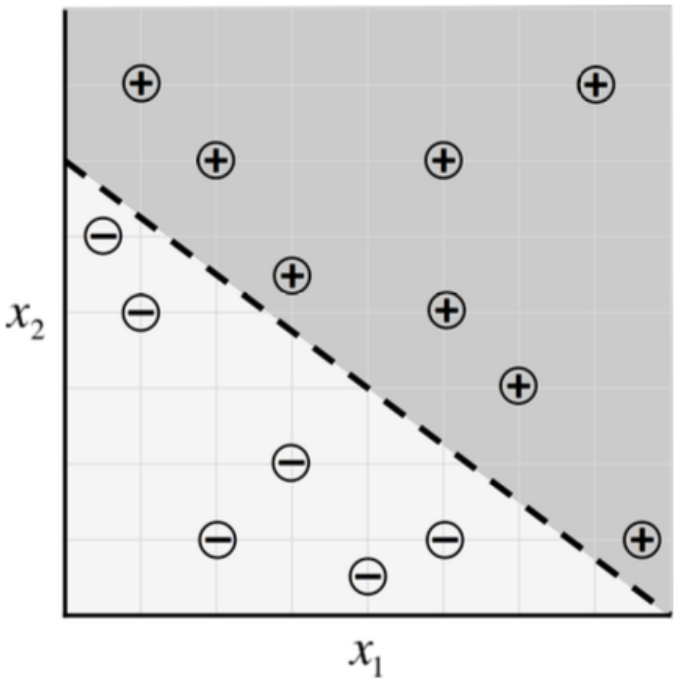

图1-4 Sample data for our exam predictor algorithm and a potential classifier

结果显示，令 $\theta = [-24\ 3\ 4]^T$，我们的机器学习模型在每个数据点上都

做出正确的预测：

$$h(x,\theta)=\begin{cases}-1, & 3x_1+4x_2-24<0\\ 1, & 3x_1+4x_2-24\geq 0\end{cases}$$

最优参数向量 θ 确定分类器的位置使得我们能够做出尽可能多的正确预测。在大部分情况下，θ的最优解有很多种选择(甚至是无穷多种)。但幸运的是，大部分情况下这些最优解彼此间如此相近，它们之间的差异几乎可以忽略不计。如果事实并非如此，那我们则要收集更多的数据来缩小 θ 的选择范围。

建模看起来是合理的，但仍遗留了一些相当重要的问题。首先，最开始我们怎么设法得到参数向量 θ 的最优值？解决整个问题需要众所周知的优化技术。优化器的目标是通过迭代调整参数直到误差最小化来最优化机器学习模型的性能。当在第二章更加详细地描述梯度下降过程时，我们将着手开始处理训练参数向量的问题。在后面的章节，我们将努力去寻找使这个过程更加高效的方法。

其次，显然这个特殊的模型(线性感知机模型)能够学习到的关系非常有限。例如，如图1-5所示的数据分布就不能用线性感知机很好地描述。

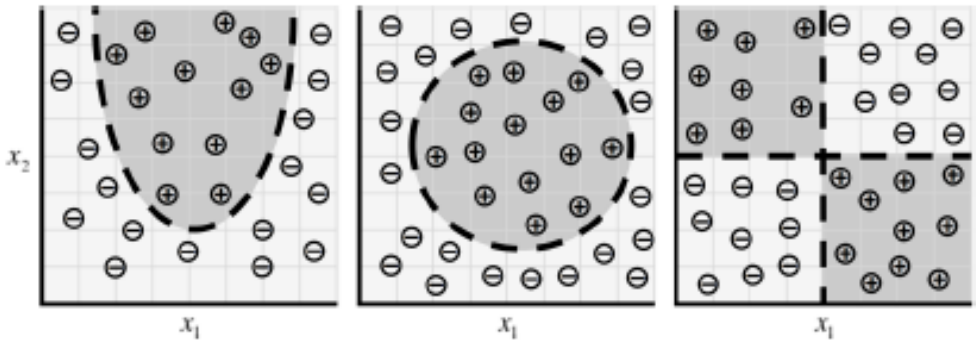

图1-5 As our data takes on more complex forms, we need more complex models to describe them

但是这些情况仅仅是冰山一角。当我们转向更加复杂的问题时，如目标识别和文本分析，数据维度极其高，需要挖掘的关系也变得高度非线性。为了适应这样的复杂性，近期的机器学习研究已经尝试创建类似于人脑结构的模型。这项研究的主体本质上就是俗称的深度学习，它在处理计算机视觉和自然语言处理等问题上已经取得了令人瞩目的成绩。这些算法不仅远超其他类型的机器学习算法，而且达到(甚至超过了)人类的认知准确度。

神经元

人脑的基本单元是神经元。一片小小的大脑，大约稻粒那么大，包含10000多个神经元，每个神经元与其他神经元形成平均6000个连接。正是这个庞大的生物网络使得我们可以感知周围的世界。本节的目标是用这个自然的结构创建机器学习模型，用与大脑类似的方式解决问题。

神经元的核心部位进化后可以接收来自其他神经元的信息，并用独特的方式处理，然后把处理结果传送给其他细胞。处理过程总结在图1-6。神经元沿着天线状结构的树状突(dendrites)接收输入，每个输入连接根据使用多久动态地增强或减弱(这就是我们如何学习新概念的原理！)，每个连接的强度决定了输入对神经元输出的贡献。在被各自连接强度加权后，在细胞体中把输入累加起来。然后累加和被变换成新的信号沿着细胞的轴突(axon)发送到其他神经元。

我们能够把对大脑中神经元的功能性理解转变成可用计算机表示的人工模型。图1-7描述了这样的模型，改善了1943年Warren S. McCulloch和Walter H. Pitts首次提出的方法。正如在生物神经元一样，我们的人工神经元接收一定数量的输入，$x_1, x_2, ..., x_n$，每个输入乘以指定的权重，$w_1, w_2, ..., w_n$。如前所述，加权后的输入相加生成神经元的logit，$z = \sum_{i=0}^{n} w_i x_i$。在很多场合中，logit还包括一个常量-偏置(图中没有表示)。logit通过函数f产生输出 $y = f(z)$，它可以被发送到其他神经元。

通过再次用向量形式表示，我们来结束人工神经元功能的数学讨论。

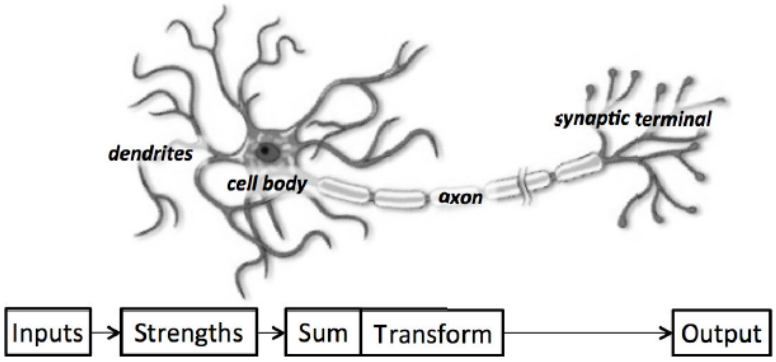

图1-6 A functional description of a biological neuron's structure

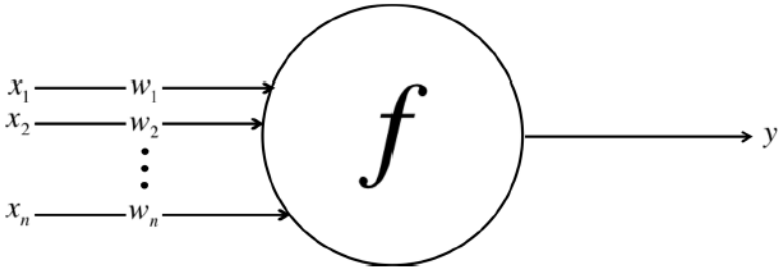

图1-7 Schematic for a neuron in an artificial neural net

将输入重新用向量形式表示为 $x = [x_1\ x_2\ ...\ x_n]$，神经元权重重新表示为 $w = [w1\ w2\ ...\ w_n]$。然后神经元输出重新表示为 $y = f(x \cdot w + b)$，其中 b 是偏置项。换句话说，我们可以用输入和加权向量的点积计算输出，然后加上偏置项生成logit，再应用变换函数。尽管这看起来似乎是普通的重写，但把神经元视为向量操作系列将对在本书后续章节用软件实现神经元至关重要。

把线性感知机表示为神经元

在“机器学习机理”一节，我们讨论了使用机器学习模型获取考试成功与学习/睡眠花费时间之间的关系。为了处理该问题，我们构建线性感知机分类器将平面直角坐标系分为两半：

$$h(x,\theta) = \begin{cases} -1, & if \quad 3x_1 + 4x_2 - 24 < 0 \\ 1, & if \quad 3x_1 + 4x_2 - 24 \geq 0 \end{cases}$$

如图1-4所示，对 θ 存在将数据集中每个数据正确分类的最优解。此处，模型 h 容易地使用神经元表示。考虑图1-8描绘的神经元，有两个输入和一个偏置项，使用函数：

$$f(z) = \begin{cases} -1, & if \quad z < 0 \\ 1, & if \quad z \geq 0 \end{cases}$$

显而易见，线性感知机和神经元模型完美等价。一般来说，可以简单地证明单个神经元比线性感知机具有更强的表达能力。就是说，每个线性感知机可以表示为单个神经元，但单个神经元还能够表示任何线性感知机无法表示的模型。

前馈神经网络

尽管单个神经元比线性感知机更加强大，但表达能力还远不足以解决

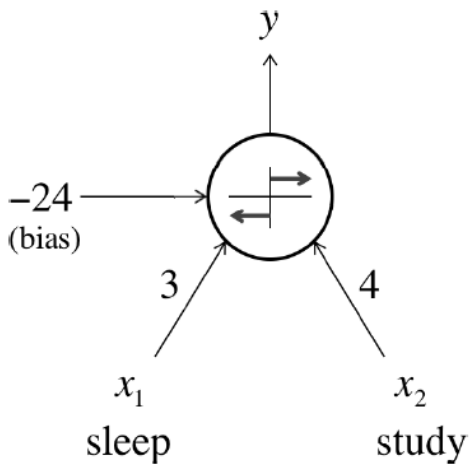

图1-8 Expressing our exam performance perceptron as a neuron

复杂的学习问题，这是我们大脑由不止一个神经元组成的原因。例如，单个神经元不可能区分开手写的数字。为了处理更加负载的任务，我们必须进一步优化机器学习模型。

大脑中的神经元是分层构成的。事实上，大脑皮质(负载人类大部分智商的结构)由6层组成 [7] 。信息流从一层到另一层直到感觉输入被转化为概念性理解。例如，视觉皮质末端层从眼睛接收原始的视觉数据，经每层处理后传递到下一层，直到第六层我们推断出正在看到的是只猫，或者汽水罐，抑或是架飞机。图1-9是这些层更简化的版本。

借助这些概念，我们能够构建人工神经网络。当我们开始把神经元与其他神经元、输入数据和对学习问题相当于网络答复的输出节点连接起来，就实现了神经网络。图1-9展示了一个人工神经网络的简单例子，类似于McCulloch 和Pitt 1943年工作里的架构。网络的最底层吸收输入数据，神经元最顶层(输入节点)计算最终答案。神经元的中间层称为隐藏层，我们令 k^{th} 层的 i^{th} 个神经元与 $k+1^{st}$ 层的 j^{th} 个神经元之间连接的加权为 $w_{i,j}^{(k)}$。这些加权组成了参数向量 θ，如前所述，神经网络解决问题的能力取决于找到最优的值填入 θ。

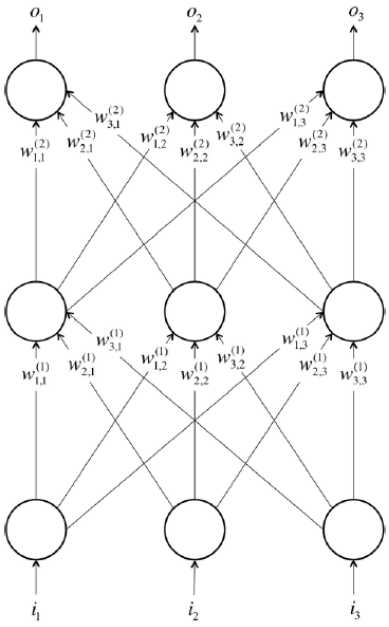

图1-9 A simple example of a feed-forward neural network with three layers (input, one hidden, and output) and three neurons per layer

我们注意到在这个例子中，连接仅仅从较低层横贯到较高层，同层之间的神经元之间没有连接，没有从较高层向较低层传输数据的连接。这样的神经网络叫做前馈网络，由于易于分析，所以从讨论前馈网络开启我们的学习之旅。我们在第二章给出前馈网络分析(即为加权选择最优值的过程)。更多复杂的连接方式将在后续章节阐述。

在最后部分，我们将讨论在前馈神经网络中使用的层的主要类型。但是在讲解之前，必须牢记一些重要内容：

1. 正如我们提到的，夹在神经元第一层(输入层)和最后一层(输出层)之间的神经元层叫做隐藏层。当神经网络试图解决问题时，隐藏层就会产生最最奇妙的反应。(就像在手写数字例子中)，以前我们需要花费大量的精力识别有用的特征，然而现在隐藏层可以自动为我们识别有用特征。通常，看一下隐藏层的活动，它能够告诉你很多从数据中提取出来已经自动学习到的特征信息。

2. 尽管本例中各层有相同数量的神经元，但这既不是必须的也并不建议这么做。隐藏层往往比输入层神经元更少，以此来强迫网络学习到原始输入的压缩表示。例如，当眼睛从周围环境获取到原始像素值时，大脑把它们看做边和轮廓。这是因为大脑中生物神经元的隐藏层强迫我们对觉察到的每件事物提出更好的表示。

3. 不要求每个神经元的输出连接到下一层所有神经元的输入。事实上，选择哪个神经元连接到下一层哪些其他神经元是一项源自经验的艺术。当我们解决各种神经网络案例时再深度讨论这个问题。

4. 输入和输出都被向量化表示。例如，你可以想象一个神经网络，图像中单个像素RGB值作为输入被表示为向量(参考图1-3)。最后一层有两个神经元对应问题的答案：如果图像包含一条狗，输出[1,0]；如果图像包含一只猫，输出[0,1]；如果图像两者都包含，输出[1,1]；如果图像两者都不包含，输出[0,0]。

同样，我们观察到，与神经元的公式重写类似，可以在数学上把神经网络表示为一系列向量和矩阵运算。设网络 i^{th} 层的输入为向量 $x = [x_1\ x_2\ ...\ x_n]$。输入通过神经元生成向量 $y = [y_1\ y_2\ ...\ y_m]$。如果我们构建大

小为 $n \times m$ 的加权矩阵 W 和大小为 m 的偏置向量，这个过程可以表示为简单的矩阵乘法。在矩阵中，每列对应一个神经元，列的 j^{th} 元素对应于来自输入 j^{th} 元素的连接加权。换句话说， $y = f(W^T x + b)$，其中变换函数应用在向量上。当我们开始用软件实现这些网络时，这种公式重写将会变得愈加重要。

7 Mountcastle, Vernon B. “Modality and topographic properties of single neurons of cat’s somatic sensory cortex.” Journal of Neurophysiology 20.4 (1957): 408-434.

线性神经元和它们的局限性

大部分神经元类型是根据应用到它们的logit输出 z 上的函数 f 定义的。我们首先讨论使用形式为 $f(z) = az + b$ 的线性函数的神经元层。例如，尝试估计在快餐店就餐费用的神经元将使用 $a = 1$ 和 $b = 0$ 的线性神经元。换句话说，使用 $f(z) = z$ ，权重等于每道餐品的价格，图1-10中的线性神经元输入已经点好上餐的汉堡包、炸薯条和汽水三种餐品，输出是三者一起的价格。

线性神经元容易计算，但却陷入了严重的局限性。实际上，任何仅由线性神经元组成的前馈神经网络能被表示为没有隐藏层的网络。这是有问题的，因为正如我们前面讨论的，使我们从输入数据中学习到重要特征的是隐藏层。换言之，为了学习到复杂的关系，我们需要使用具有某种非线性的神经元。

Sigmoid, Tanh和ReLU神经元

在实践中主要使用的三种类型神经元在计算中引入了非线性。它们中的第一种是sigmoid神经元，它使用函数：

$$f(z) = \frac{1}{1 + e^{-z}}$$

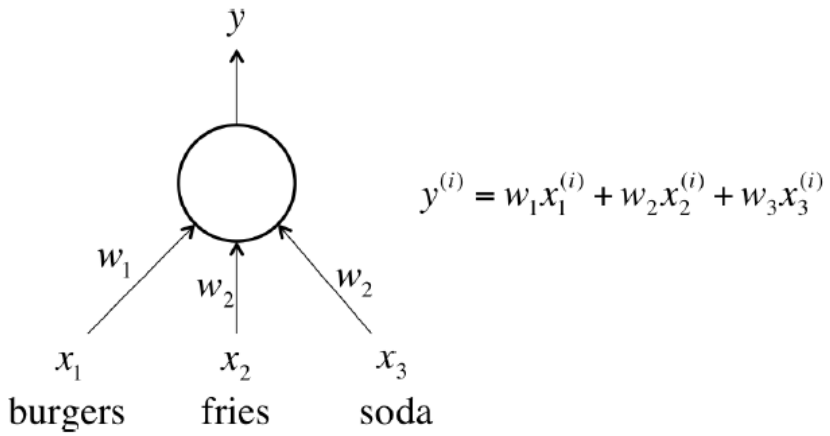

图1-10 An example of a linear neuron

直观地，当logit非常小时，logistic神经元的输出接近于0。当logit非常大时，logistic神经元的输出接近于1。在这两种极端情况之间，如图1-11所示，神经元呈现 S 型。

tanh神经元使用类似于 S 型的非线性，但取值范围不再是0到1，tanh神经元的输出是-1到1。正如所期望的，它令 $f(z) = tanh(z)$。输出y和logit z之间的变换关系如图1-12所示。但使用 S 型非线性关系时，相对于sigmoid神经元，人们更青睐tanh神经元，因此它是以0为中心的。

受限线性单元(ReLU)神经元使用的是完全不同的非线性。它使用函数 $f(z) = max(0, z)$，产生曲棍球形状的响应特性，如图1-13所示。

尽管ReLU有一些缺陷 [8]，但因为许多原因，最近它已成为很多任务选择的神经元(尤其在计算机视觉方面)。在第五章我们将讨论这些原因，以及客服潜在陷阱的策略。

8 Nair, Vinod, and Geoffrey E. Hinton. “Rectified Linear Units Improve Restricted Boltzmann Machines” Pro- ceedings of the 27th International Conference on Machine Learning (ICML-10), 2010.

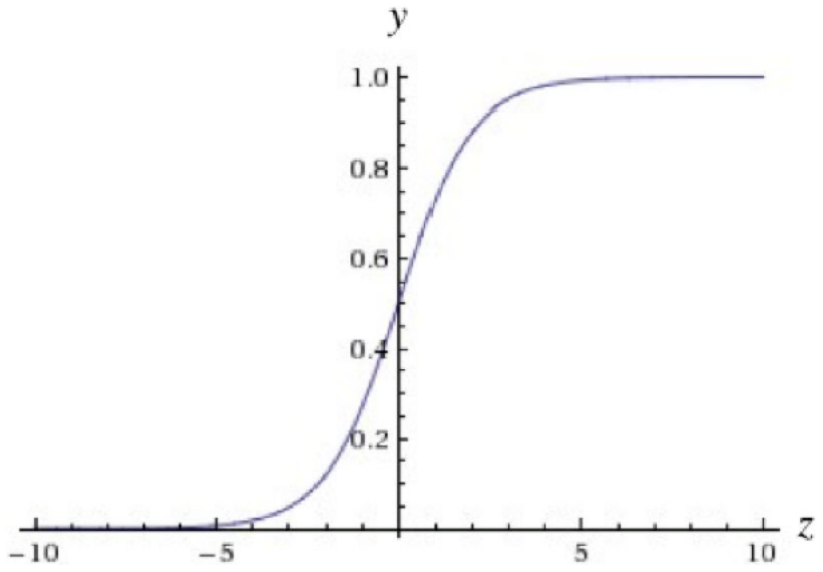

图1-11 The output of a sigmoid neuron as z varies

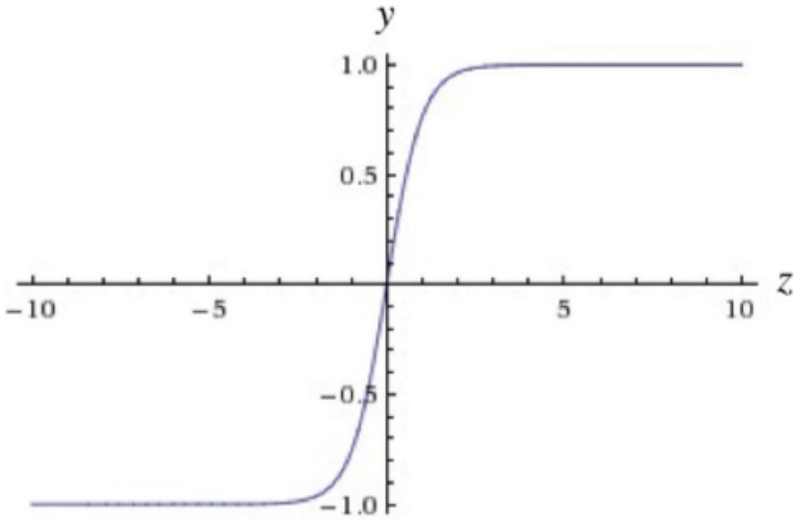

图1-12 The output of a tanh neuron as z varies

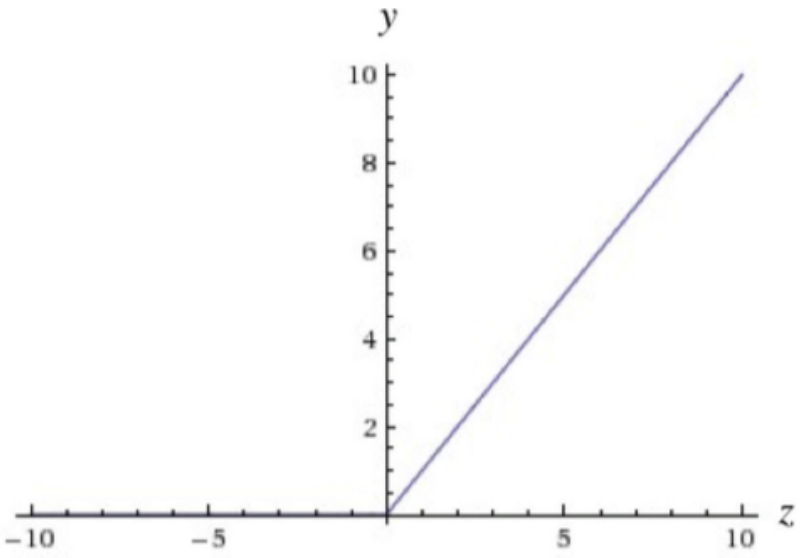

图1-13 The output of a ReLU neuron as z varies

Softmax输出层

通常，我们希望输出向量是对一组互斥标签的概率分布。例如，我们要创建神经网络从MNIST数据集中识别手写的数字。虽然每个标签(从0到9)都是互斥的，但我们不可能100%准确地识别数字。用概率分布让我们对预测的置信区间有了更好的了解。因此，我们想得到的输出向量是下面这种形式的，其中 $\sum_{i=0}^{9} p_i = 1$：

$$[p_0\ p_1\ p_2\ p_3\ \cdots\ p_9]$$

用一种称为softmax层的特殊输出层就能产生这样的输出。不像其他类型的神经元层，softmax层里神经元的输出取决于同层所有其他神经元的输出。这是因为我们要求所有输出的和为1。令 z_i 为 i^{th} softmax神经元的logit，我们可以令它的输出等于下式实现归一化：

$$y_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

强预测将使得向量中的某一项接近1，其他项接近0。弱预测使得多个可能的标签的概率差不多相等。

展望

在本章节，我们对机器学习和神经网络建立了基础的直观认识，探讨了神经元的基本结构，前馈神经网络的工作原理，以及非线性在处理复杂学习问题中的重要性。在下一章，我们将开始构建必要的数学背景来训练神经网络解决问题。特别地，我们将讨论寻找最优参数向量，训练神经网络的最佳实践和主要挑战。在未来的章节，我们将用这些基础思想去构建更加专业的神经架构。