一、介绍

深度学习的概念由Hinton等人于2006年提出，但它却有着悠久而丰富的历史。一般认为深度学习的雏形出现在控制论中。50年代中后期，基于神经网络的“连接主义”学习开始出现，但早期的人工智能研究人员偏爱符号主义，所以连接主义并未纳入主流的人工智能研究范畴。80年代中期连接主义重新受到人们关注。首先是神经网络求解“流动推销员问题”取得重大进展，其次反向传播算法的提出，对神经网络产生了深远的影响。21世纪初连接主义又以“深度学习”的名义掀起了人工智能的热潮。

深度学习的概念源于人工神经网络的研究。学习即机器学习，让机器自己能够学习知识。之所以用深度一词，主要因为神经网络的层数变深，狭义上来讲就是“很多层”的神经网络。我们常常用深度学习这个术语来指训练多层神经网络的过程。

二、神经元模型

历史上，科学家一直希望模拟人的大脑，造出可以思考的机器。人为什么能够思考？科学家发现，原因在于人体的神经网络。生物神经网络的工作流程可以简要的概述为如下4步：

1、外部刺激通过神经末梢，转化为电信号，转导到神经细胞（又叫神经元）。

2、电信号最终传导到由无数神经元所构成神经中枢。

3、神经中枢综合各种信号，做出判断。

4、人体根据神经中枢的指令，对外部刺激做出反应。

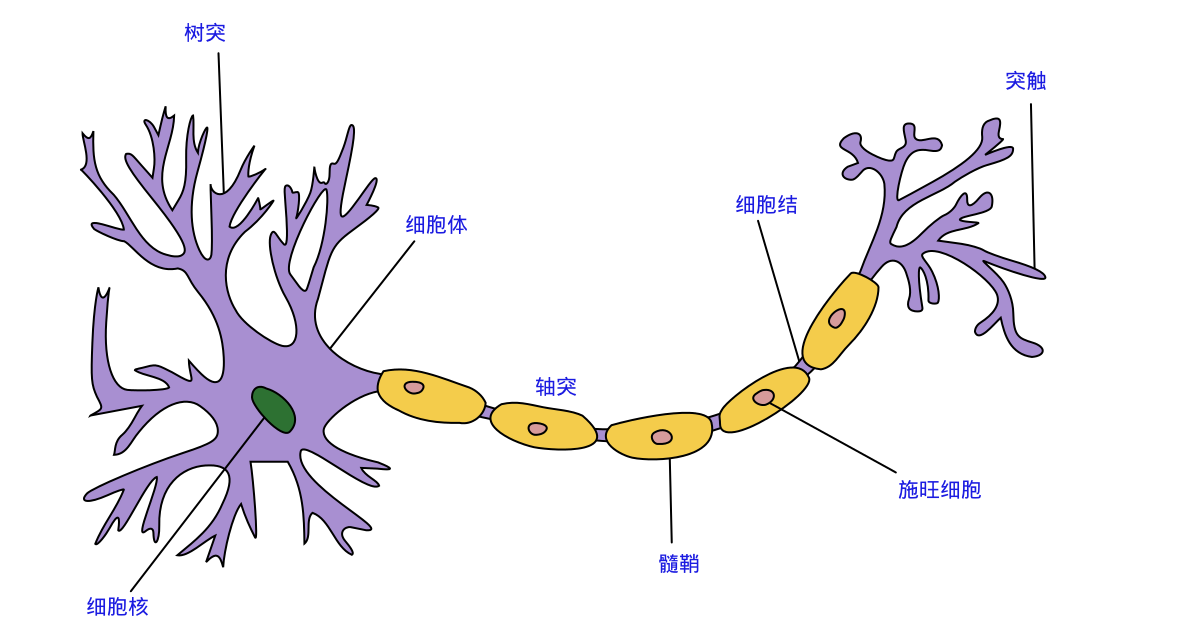


图1 生物神经元

既然思考的基础是神经元，如果能够"人造神经元"（artificial neuron），就能组成人工神经网络，模拟思考。在生物神经网络中，每个神经元都与其他神经元相连。当它“兴奋”时，它会向连接的神经元发送化学物质，从而改变连接神经元中的电位；如果神经元的电位超过了阈值，它就会被激活，并向其他神经元发送化学物质。

1943年McCulloch 和 Pitts将上述的生物神经网络中的神经元抽象为图2所示的简单的模型，这就是一直沿用至今的“M-P神经元模型”。在这个模型中，一个神经元的输入来自与其他n个神经元传递过来的信号，输入信号通过带权重的连接进行传递，神经元计算接收到总的输入值，然后通过“激活函数”产生神经元的输出。

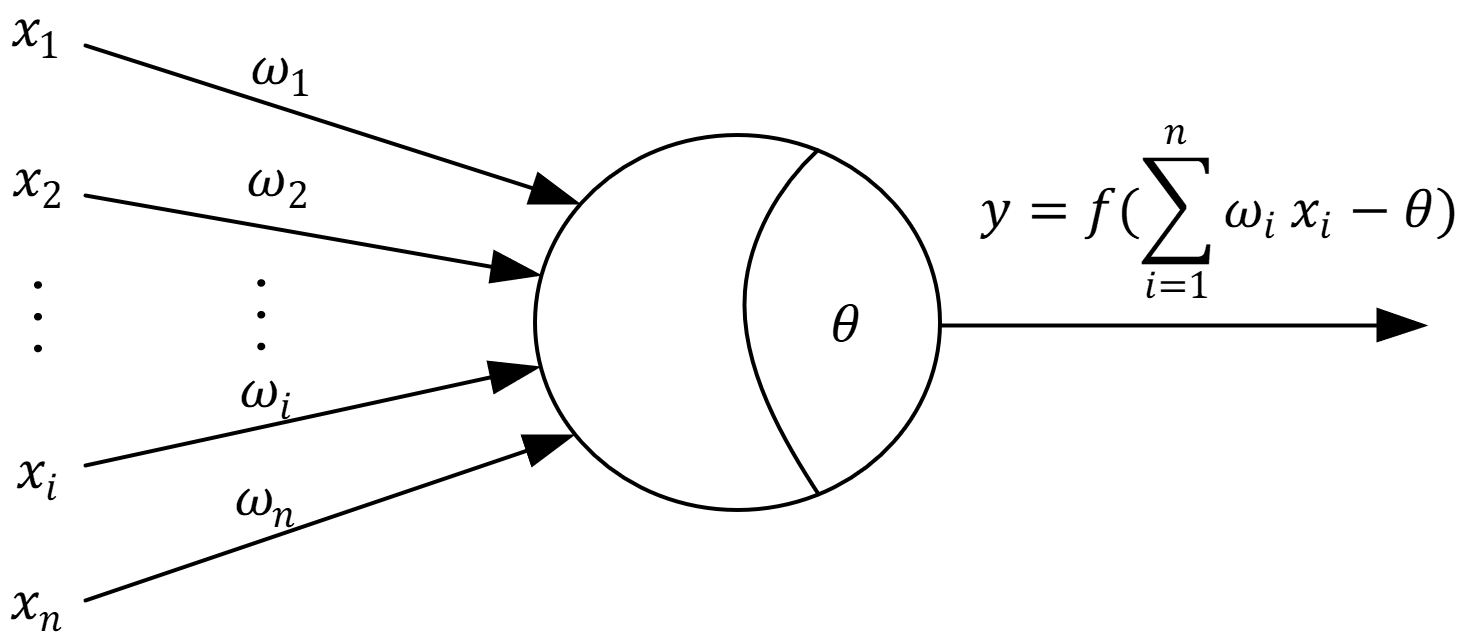


图2 M-P神经元模型

把许多个这样的神经元按一定的层次结构连接起来，就得到了一个神经网络。从数学的角度来看，神经网络就是一个包含了很多参数（）的数学模型（函数）,这个模型接受一个或多个输入，输出一个或多个。

三、感知机

感知机是由两层神经元组成的神经网络结构，如图3所示，输入层接受外界输入信号传递给输出层，输出层（也被称为是感知机的功能层）是一个M-P神经元。

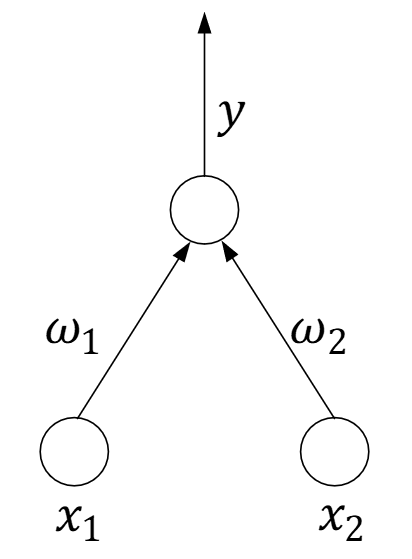


图3 含有2个输入神经元的感知机网络结构

感知机可以很容易的实现逻辑与、或、非运算。与、或、非运算是一个二元函数，其输入为，输出为。表1为与运算的真值表。

表1 与运算真值表

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |
| 0 | 0 | 0 |

从图2中我们知道M-P神经元的输出为，我们令

，,则，仅当时，。

更一般地，在给定的训练数据集上，权重，以及阈值，可以通过梯度下降算法学习得到。感知机的学习规则非常简单，对于训练集上的一个样本，若感知机的输出为，则感知机的参数调整为：

其中，称为学习率，从式(1),(2)可以看出如果感知机输出正确，即，那么感知机的参数不会发生变化，否则感知机将根据错误的程度对参数进行调整。

本节举例说明了感知机如何解决与问题，并通过梯度下降的学习规则，说明感知机如何学习参数。但是感知机的学习能力非常有限，因为其只有输出层神经元具有激活函数，即只有一层功能神经元。事实上，感知机所解决的与、或、非问题都是线性可分问题，即存在着一个超平面将，或的点分开。可以证明对于线性可分的问题，感知机的学习过程一定会收敛而求得适当的参数，否则感知机的学习过程将发生振荡，，难以稳定，无法求得合适的解。例如感知机甚至无法解决异或这样简单的线性不可分问题。如图4所示。

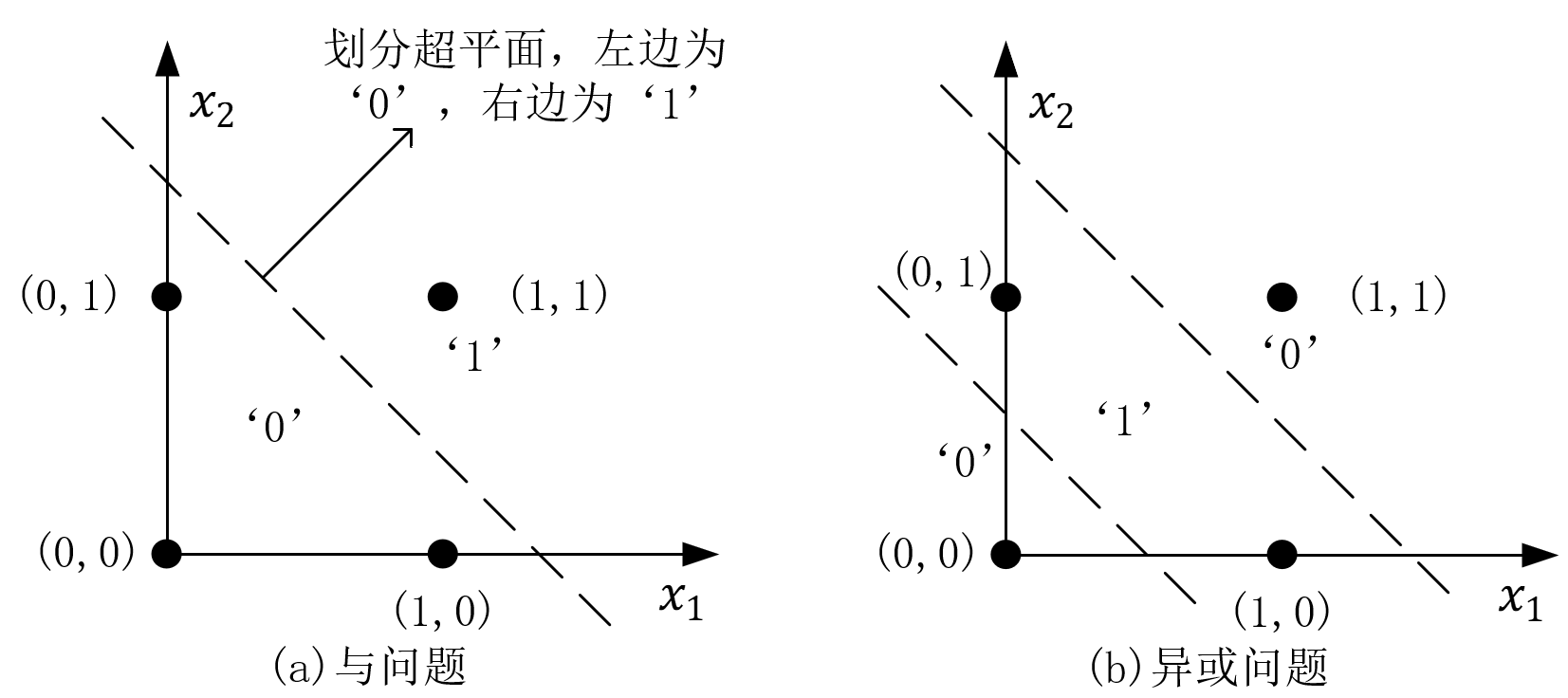


图4 线性可分的与问题与线性不可分的异或问题

要解决的线性不可分的问题，我们需要使用多层功能神经元，即带有激活函数的神经元。多层神经元能够解决线性不可分问题的本质是激活函数是非线性的，它为整个神经网络带来了非线性，最终可以解决线性不可分问题。大多数的问题都是线性不可分，尤其是计算机视觉，自然语言处理领域的问题。因此更一般的网络是形如图5所示的层级结构。其中每层神经元与下一层神经元全互连，同层神经元之间不存在连接，也不存在跨层连接。这样的神经网络结构通常称为“多层前馈神经网络”。

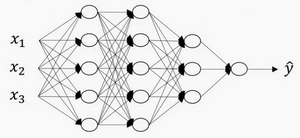


图5 多层前馈神经网络结构

四、深度神经网络

深度神经网络由感知机推广而来，也被称为多层感知机，深度前馈神经网络。其结构如图6所示，我们将第一层称之为输入层，最后一层称之为输出层，中间的层称之为隐层。隐层数量并没有限制，这也是为什么被称之为深度神经网络的原因，因为我们可以增加很多个隐层，使得网络非常的深。

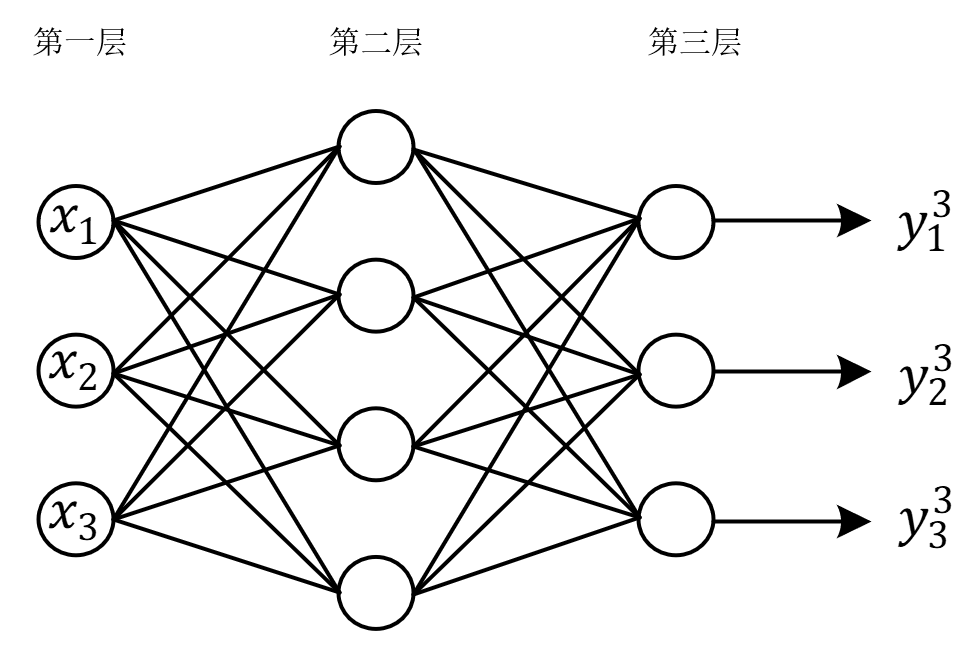


图6 三层深度神经网络

4.1前向传播

前向传播指的是信息从第一层逐渐地向高层进行传递的过程。也就是图6中从输入，到输出，的过程。下面以图6为例说明前向传播的过程。

图6中第一层为输入层，输入信息为。对于第层，我们用表示第层的输出，其输入为前一层的输出，为第层的权重参数。为激活函数。那么图6中第二层的输出可以表示为：

其中（因为第一层为输入层）。具体的，对于第二层的第个神经元来说，其输出为：

其中表示来自第一层神经元的输入，表示连接神经元权重与阈值。我们可以将第二层的前向传播计算过程推广到网络中的任意一层，则：

4.2反向传播

深度神经网络的学习比单层感知机强很多，但要训练一个深度神经网络使用单层感知机的学习规则显然是不行的，因为存在多层的网络结构，无法直接对中间隐层利用损失来进行参数更新，需要更强大的学习算法。反向传播算法就是其中最杰出的代表，它是最成功的神经网络学习算法。

下面来说明神经网络如何通过反向传播算法来学习。假定训练集为

，。对于一个训练样本，根据前向传播算法，可以得到一个层深度神经网络的输出为，那么神经网络的误差可以定义为

根据前向传播我们知道，因此我们可以很容易的使用梯度下降来更新和，具体公式类似于感知机的更新

其中，为学习率。但是对于隐含层的输出，我们无法直接计算它的误差。但是同样的根据前向传播我们知道隐层层的输出为，那么我们可以根据层输出的误差根据链式求导法则从而更新层的参数和。具体更新公式为

同样的我们可以重复这个步骤，更新再前面一层的神经网络的参数。我们可以将这个过程推广到神经网络中的任意一层。对于第层，定义其输入为,输出为，反向传播的过程可以写作

对于最后一层，其输入为，我们使用反向传播计算每一层参数的梯度，然后使用梯度下降法，根据误差来调整每一层的参数。

4.3 为什么深度有效

我们都知道深度神经网络能解决好多问题，其实并不需要很大的神经网络，但是得有深度，得有比较多的隐藏层，这是为什么呢？我们一起来看几个例子来理解，为什么深度神经网络会很好用。

首先，深度网络究竟在计算什么？如果你在建立一个人脸识别或是人脸检测系统。当你输入一张脸部的照片，你可以把深度神经网络的第一层，当成一个特征探测器或者边缘探测器去找这张照片的各个边缘。比如脸部的边缘，眼睛、鼻子等部位的边缘。在找到边缘之后，第二层可以把被探测到的边缘组合成面部的不同部分。比如说，可能有一个神经元会去找眼睛的部分，另外还有别的在找鼻子的部分，然后把这许多的边缘结合在一起，就可以开始检测人脸的不同部分。最后第三层再把这些部分放在一起，比如鼻子眼睛下巴，就可以识别或是检测不同的人脸。

这种从简单到复杂的金字塔状表示方法或者组成方法，也可以应用在图像或者人脸识别以外的其他数据上。比如当你想要建一个语音识别系统的时候，需要解决的就是如何可视化语音，比如你输入一个音频片段，那么神经网络的第一层可能就会去先开始试着探测比较低层次的音频波形的一些特征，比如音调是变高了还是低了，分辨白噪音，咝咝咝的声音，或者音调，可以选择这些相对程度比较低的波形特征，然后把这些波形组合在一起就能去探测声音的基本单元。

所以深度神经网络的这许多隐藏层中，较早的前几层能学习一些低层次的简单特征，等到后几层，就能把简单的特征结合起来，去探测更加复杂的东西。比如你录在音频里的单词、词组或是句子，然后就能运行语音识别了。同时我们所计算的之前的几层，也就是相对简单的输入函数，比如图像单元的边缘什么的。到网络中的深层时，你实际上就能做很多复杂的事，比如探测人脸或是探测单词、短语或是句子。就是为什么深度学习效果拔群的一个直觉解释。

五、卷积神经网络

深度神经网络应用计算机视觉时要面临一个挑战，就是数据的输入可能会非常大。举个例子，一个的小图片，实际上，它的数据量是，因为每张图片都有3个颜色通道。如果计算一下的话，可得知数据量为12288，所以我们的输入向量维度为12288。这其实还好，因为真的是很小的一张图片。如果你要操作更大的图片，比如一张的图片，它足有1M那么大，但是特征向量的维度达到

，因为有3个RGB通道，所以数字将会是300万。

如果你要输入300万的数据量，这就意味着，特征向量的维度高达300万。所以在第一隐藏层中，你也许会有1000个隐藏单元，而所有的参数组成了矩阵。如果你使用了标准的深度神经网络的全连接，这个矩阵的大小将会是1000×300万。因为现在特征向量的维度为，通常用3M来表示300万。这意味着矩阵会有30亿个参数，这是个非常巨大的数字。在参数如此大量的情况下，难以获得足够的数据来防止神经网络发生过拟合和竞争需求，要处理包含30亿参数的神经网络，巨大的内存需求让人不太能接受。

但对于计算机视觉应用来说，你肯定不想它只处理小图片，你希望它同时也要能处理大图。为此，你需要进行卷积计算，它是卷积神经网络中非常重要的一块。下面会为你介绍如何进行这种运算。

5.1卷积运算

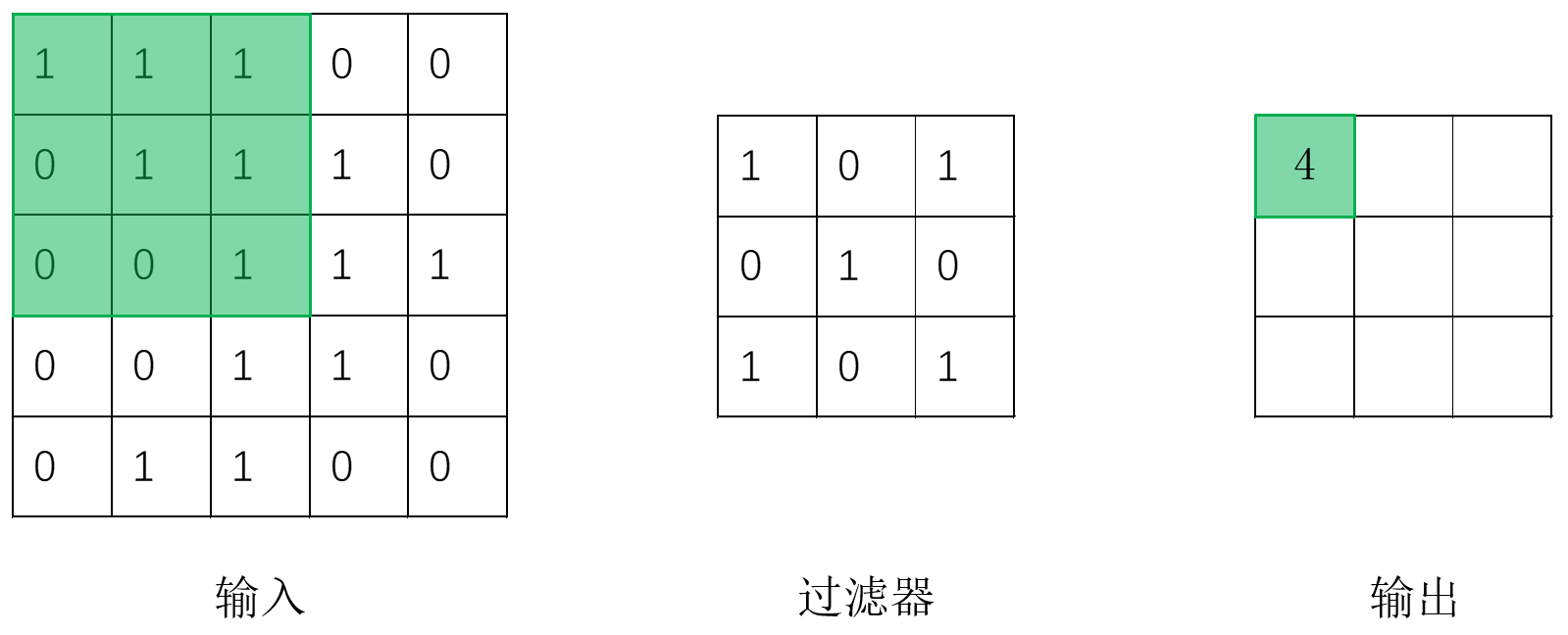


图7 卷积运算I

如图7所示卷积运算输入为的矩阵，为了进行卷积运算，你需要构造一个的矩阵，在卷积神经网络的术语中，它被称为过滤器。这个卷积运算的输出将会是一个3×3的矩阵，你可以将它看成一个的图像。下面来说明是如何计算得到这个矩阵的。

为了计算输出矩阵的第一个元素，在左上角的那个元素，使用的过滤器，将其覆盖在输入图像的绿色部分，如图7所示。然后进行元素乘法运算（点乘），所以

=，然后将该矩阵每个元素相加得到最左上角的元素4。接下来，为了计算第二个元素是什么，你要把绿色的方块，向右移动一步，如图8所示，继续做同样的元素乘法，然后加起来得到第二个位置的值。然后重复这个步骤，计算完矩阵中的其他元素。因此的矩阵和矩阵进行卷积运算得到矩阵。

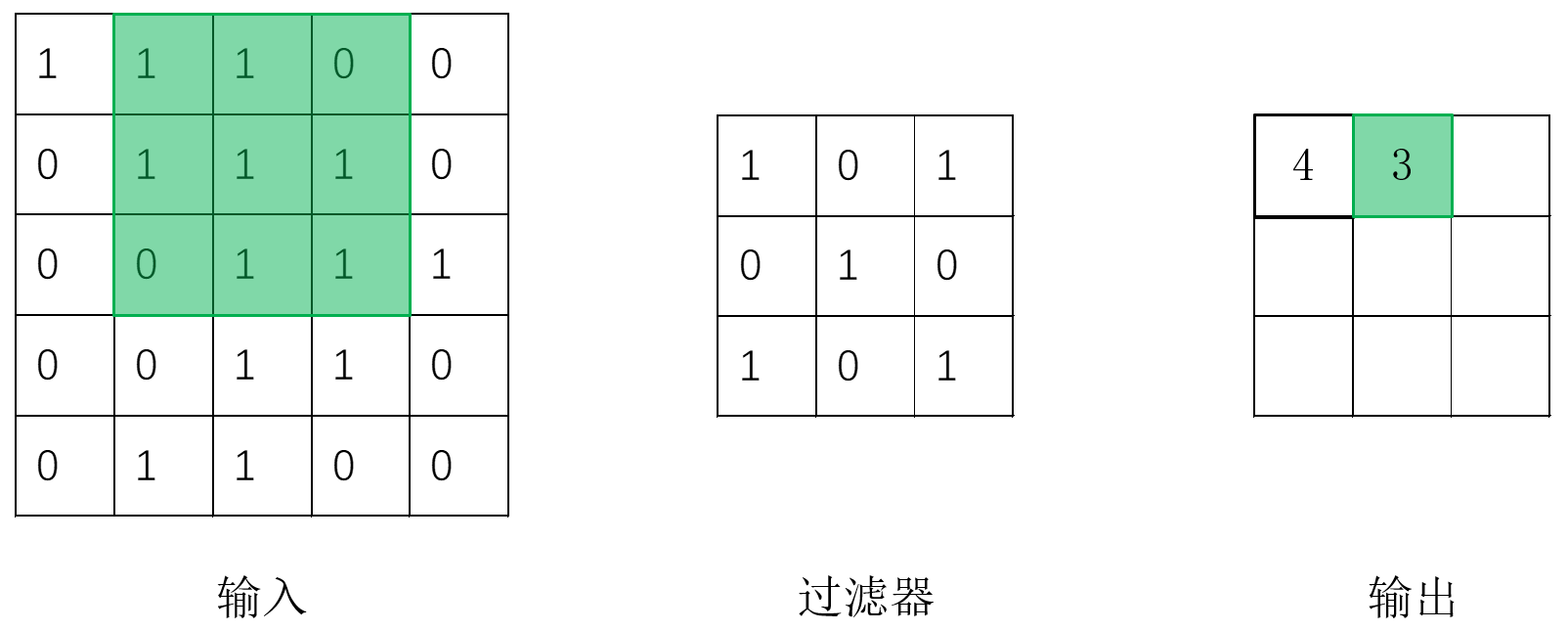


图8 卷积运算II

这些图片和过滤器是不同维度的矩阵，但左边矩阵容易被理解为一张图片，中间的这个被理解为过滤器，右边的图片我们可以理解为另一张图片。

5.2 池化层

除了卷积层，卷积网络也经常使用池化层来缩减模型的大小，提高计算速度，同时提高所提取特征的鲁棒性。

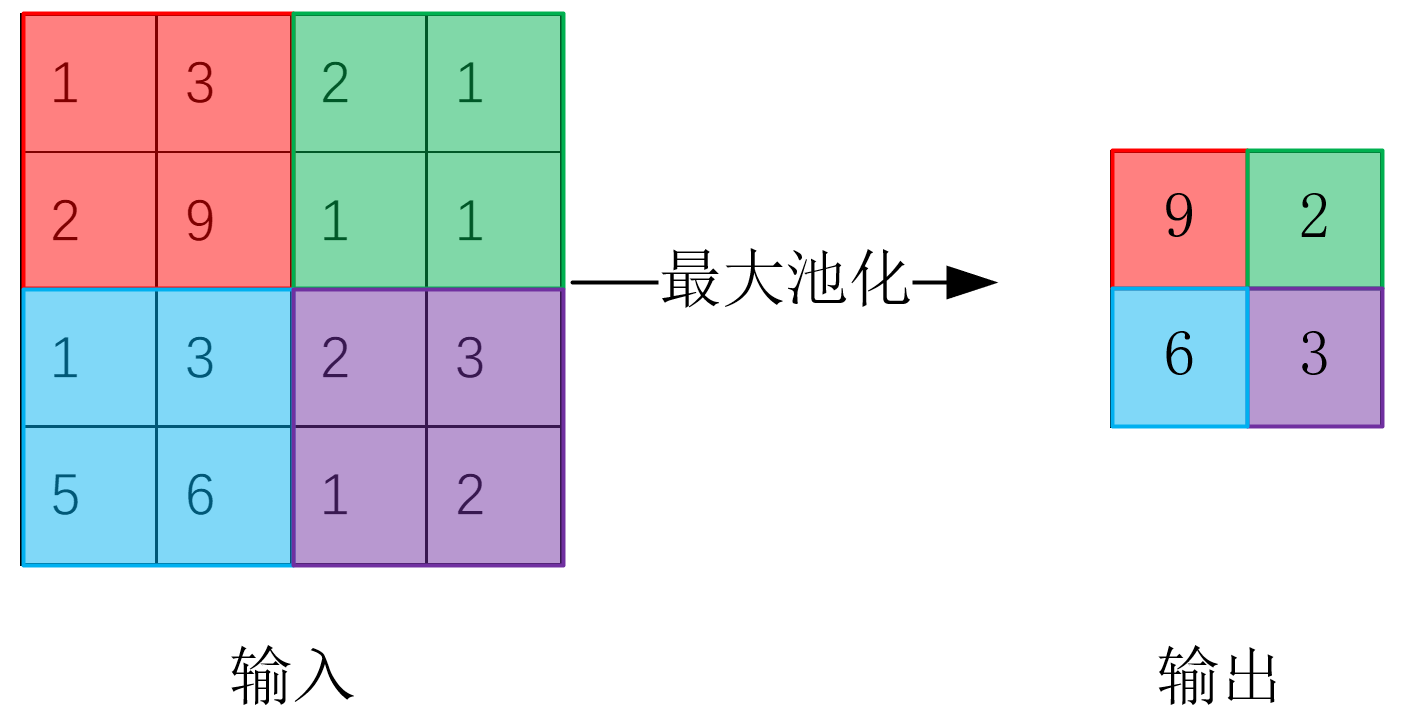


图9 最大池化

先举一个池化层的例子，然后我们再讨论池化层的必要性。如图9所示，输入是一个4×4矩阵，用到的池化类型是最大池化。执行最大池化的是一个2×2矩阵。执行过程非常简单，把4×4的输入拆分成不同的区域，我把这个区域用不同颜色来标记。对于2×2的输出，输出的每个元素都是其对应颜色区域中的最大元素值。

对最大池化功能的一个直观理解是把这个4×4输入看作是某些特征的集合。也就是神经网络中某一层的输出值集合。数字大意味着可能探测到了某些特定的特征，左上象限具有的特征可能是一个垂直边缘，一只眼睛。另外还有一种类型的池化，平均池化，它不太常用。我这种运算顾名思义，选取的不是每个过滤器的最大值，而是平均值。

5.3 为什么使用卷积

深度神经网络中两层神经元之间是全连接的，即第层的一个神经元与层的每一个神经元相连。和只用全连接层相比，卷积层的两个主要优势在于参数共享和稀疏连接。假设我们输入一个大小为的一个RGB三通道的图片，经过神经网络的一层处理后输出为，。我们构建一个深度神经网络，其中一层含有3072个单元，下一层含有4074个单元，两层中的每个神经元彼此相连，然后计算权重矩阵，它等于万，要训练的参数很多。虽然以现在的技术，我们可以用1400多万个参数来训练网络，因为这张的图片非常小，训练这么多参数没有问题。如果这是一张的图片，权重矩阵会变得非常大。但是如果我们使用卷积层的话仅仅需要6个大小为的过滤器。一个过滤器有25个参数，再加上阈值参数，那么每个过滤器就有26个参数，一共有6个过滤器，所以参数共计156个。参数数量是远远小于1400万的。

卷积网络这么少参数有两个原因：

一是参数共享。观察发现，特征检测如垂直边缘检测如果适用于图片的某个区域，那么它也可能适用于图片的其他区域。也就是说，如果你用一个3×3的过滤器检测垂直边缘，那么图片的左上角区域，以及旁边的各个区域都可以使用这个3×3的过滤器。每个特征检测器以及输出都可以在输入图片的不同区域中使用同样的参数，以便提取垂直边缘或其它特征。它不仅适用于边缘特征这样的低阶特征，同样适用于高阶特征，例如提取脸上的眼睛，猫或者其他特征对象。因此在计算图片左上角和右下角区域时，你不需要添加其它特征检测器。

第二个方法是使用稀疏连接。在图7中这个输出“4”是通过3×3的卷积计算得到的，它只依赖于这个3×3的输入的单元格，右边这个输出单元（元素4）仅与25个输入特征中9个相连接。而且其它输入值都不会对输出产生任影响，这就是稀疏连接的概念。神经网络可以通过这两种机制减少参数，以便我们用更小的训练集来训练它，从而预防过度拟合。

六 循环神经网络

循环神经网络（RNN）的提出是为了解决序列问题。序列的维度不像图片那样是固定的，比如对一个句子来说，其长度是变化的。当我们尝试用神经网络来解决一个序列问题，比如机器翻译，可以尝试的方法之一是使用标准的深度神经网络。假设我们有9个输入单词。想象一下，把这9个输入单词，可能是9个**one-hot**向量，然后将它们输入到一个标准神经网络中，经过一些隐藏层，最终会输出9个值为0或1的项，它表明每个输入单词是否是人名的一部分。

但结果表明这个方法并不好，主要有两个问题：一是输入和输出数据在不同例子中可以有不同的长度，不是所有的例子都有着同样输入长度或是同样输出长度的。即使每个句子都有最大长度，也许你能够填充或零填充使每个输入语句都达到最大长度，但这仍然看起来不是一个好的处理方式；二是一个像这样单纯的神经网络结构，它并不共享从文本的不同位置上学到的特征。具体来说，如果神经网络已经学习到了在位置1出现的**Harry**可能是人名的一部分，那么如果**Harry**出现在其他位置时，它也能够自动识别其为人名的一部分。这可能类似于你在卷积神经网络中看到的，将部分图片里学到的内容快速推广到图片的其他部分，而我们希望对序列数据也有相似的效果。和你在卷积网络中学到的类似，用一个更好的表达方式也能够减少模型中参数的数量。

图10是一个非常简单的循环神经网络，它含有一个输入节点、一个输出节点、一个隐层节点。可以看到循环神经网络包含了节点到自身节点的连接，不再是一个前馈神经网络。我们可以这样来理解，如果把上面有的那个带箭头的圈去掉，它就变成了最普通的全连接神经网络。是一个向量，它表示输入层的值；是一个向量，它表示隐藏层的值。是输入层到隐藏层的权重矩阵，也是一个向量，它表示输出层的值；是隐藏层到输出层的权重矩阵。那么，现在我们来看看是什么。循环神经网络的隐藏层的值不仅仅取决于当前这次的输入，还取决于上一次隐藏层的值。权重矩阵 就是隐藏层上一次的值作为这一次的输入的权重。

在一个循环神经网络中，我们是将系列中的数据按照顺序一个一个输入网络的，具有时间上的先后关系。因此我们也可以按照时间线来展开这个神经网络。如图11所示

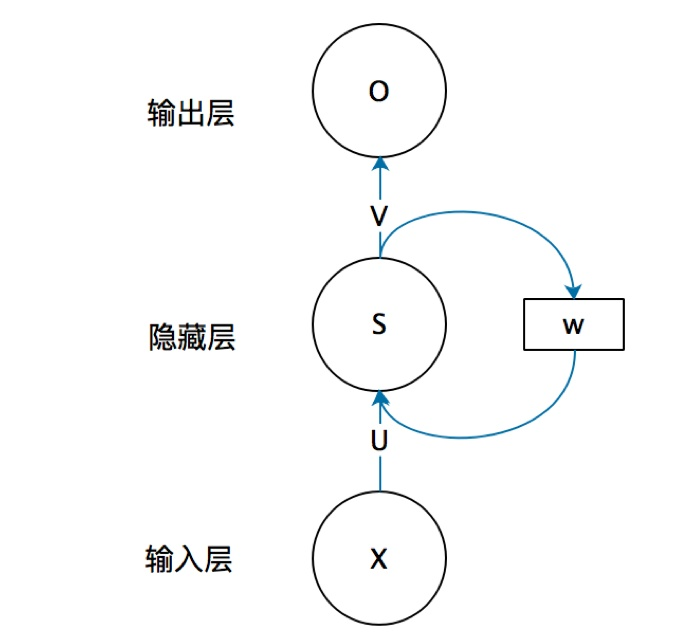


图10 循环神经网络

现在看上去就比较清楚了，这个网络在时刻接收到输入之后，隐藏层的值是，输出值是。关键点是，的值不仅仅取决于，还取决于。我们可以用下面的公式来表示循环神经网络的计算方法：

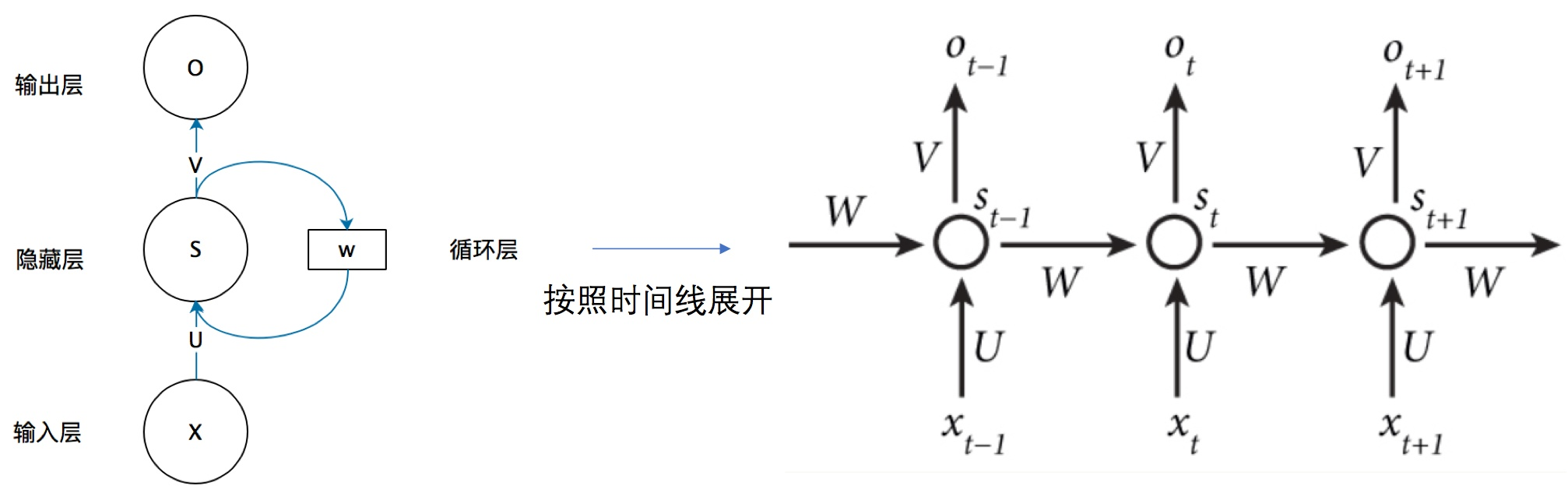


图11 循环神经网络按时间线展开

七 其他常见神经网络

7.1径向基函数网络（RBF）

径向基函数是中心点径向对称、取值仅依赖于距中心点距离的非负实值函数。简单地说，就是该点的函数值只与该点距离中心点的距离有关。它使用径向基函数作为隐层神经元的激活函数，输出层是对隐层神经元输出的线性组合。RBF网络二点输出可以表示为

其中为隐层神经元个数，为第个神经元所对应的权重和中心，为径向基函数。

7.2 自适应谐振理论网络（ART）

竞争型学习中网络的输出神经元互相竞争，每一时刻仅有一个竞争获胜的神经元被激活，其他神经元都被抑制。自适应谐振理论网络是竞争型学习的代表，它由比较层、识别层、识别阈值和重置模块构成。比较层接受输入样本传递给识别层神经元。在接收到比较层的输入后，识别层神经元之间互相竞争易产生获胜神经元。ART比较好的缓解了竞争型学习中的“可塑性-稳定性窘境”，可塑性是指神经网络对新知识的学习能力，而稳定性是指神经网络对旧知识的记忆。这就使得ART网络可以进行增量学习或在线学习。

7.3 玻尔兹曼机（BM）

玻尔兹曼机为神经网络的状态定义了一个“能量”，当能量最小化时网络达到理想的状态，网络的训练过程就是最小化这个能量函数。其神经元分为两层：显层与隐层。显层用于表示数据的输入输出，隐层可以理解为数据的内在表达。玻尔兹曼机中的神经元都是布尔类型的，即取值只能为0、1。状态1表示激活，0则表示抑制。

玻尔兹曼机是一种随机神经网络，借鉴了模拟退火思想。随机神经网络与其他神经网络相比有两个主要区别：

一、在训练阶段，随机网络不像其他网络那样基于某种确定性算法调整权值，而是按某种概率分布进行修改。

二、在预测阶段，随机网络不是按某种确定性的网络方程进行状态演变，而是按某种概率分布决定其状态的转移。神经元的净输入不能决定其状态取１还是取０，但能决定其状态取１还是取０的概率。这就是随机神经网络算法的基本概念。