# 深度学习

## 深度学习基础

### 1.1机器学习和深度学习

机器学习：“计算机程序可以在给定某种类别的任务 T 和性能度量 P 下学习经验 E ，如果其在任务 T 中的性能恰好可以用 P 度量，则随着经验 E 而提高。”让我们根据前边的解释来定义我们的预测系统：我们的任务是确定可能产生飓风的气象条件。性能P是在系统所有给定的条件下有多少次正确预测飓风。经验E是我们的系统的迭代次数。

深度学习：深度学习是一种特殊的机器学习，通过学习将世界使用嵌套的概念层次来表示并实现巨大的功能和灵活性，其中每个概念都定义为与简单概念相关联，而更为抽象的表示则以较不抽象的方式来计算。

机器学习和深度学习的对比：深度学习与传统的机器学习最主要的区别在于随着数据规模的增加其性能也不断增长。当数据很少时，深度学习算法的性能并不好。这是因为深度学习算法需要大量的数据来完美地理解它。另一方面，在这种情况下，传统的机器学习算法使用制定的规则，性能会比较好。传统机器学会将问题分解为两步：物体检测和物体识别。首先，使用一个边界框检测算法扫描整张图片找到可能的是物体的区域；然后使用物体识别算法(例如 SVM 结合 HOG )对上一步检测出来的物体进行识别。相反，深度学习会直接将输入数据进行运算得到输出结果。例如可以直接将图片传给 YOLO 网络(一种深度学习算法)，YOLO 网络会给出图片中的物体和名称。

### 1.2 深度学习简介



深度学习将所需的复杂映射分解为一系列嵌套的简单映射（每个由模型的不同层描述）

来解决这一难题。输入展示在可见层 (visible layer)，这样命名的原因是因为它包含我们能观察到的变量。然后是一系列从图像中提取越来越多抽象特征的隐藏层 (hidden layer)。因为它们的值不在数据中给出，所以将这些层称为 ‘‘隐藏”; 模型必须确定哪些概念有利于解释观察数据中的关系。

一种是在深度概率模型中使用的方法，将描述概念如何彼此相关的图的深度视为模型深度。一种将计算图的深度视为模型深度。



从线性模型（线性加权重）到连接机制（当网络将大量简单计算单元连接在一起时可以实现智能行为）。连接机制其中一个概念是分布式表示，这一想法是系统每个的输入应该由许多特征表示的，并且每个特征应参与许多可能输入的表示。连接机制运动的另一个重要成就是反向传播算法的成功运用（训练具有内部表示的深度神经网络）和普及 (Rumelhart et al., 1986c; LeCun, 1987)。这个算法虽然曾黯然失色不再流行，但截至写书之时，仍是训练深度模型的主要方法。神经网络研究的第三次浪潮始于 2006 年的突破。名为深度信念网络的神经网络可以使用一种称为贪心逐层训练的策略进行有效地训练。

### 1.3 深度学习数学基础

#### 1.3.1 线性代数

##### 1.3.1.1 标量，向量，矩阵和张量

标量 (scalar)：一个标量就是一个单独的数，不同于线性代数中大多数概念会涉及到多个数。我们用斜体表示标量。标量通常赋予小写的变量名称。当我们介绍标量时，会明确它们是哪种类型的数。比如，在定义实数标量时，我们可能会说‘让 s ∈ R 表示一条线的斜率’。在定义自然数标量时，我们可能会说‘让 n ∈ N 表示元素的数目’。

向量 (vector)：一个向量是一列数。这些数是有序排列的。通过次序中的索引，我们可以确定每个单独的数。通常我们赋予向量粗体的小写变量名称，比如 x。向量中的元素可以通过带脚标的斜体表示。向量 x 的第一个元素是 x1，第二个元素是 x2，等等。有时我们需要指定向量中某个集合的元素。在这种情况下，我们定义一个包含这些索引的集合，然后将该集合写在脚标处。比如，指定 x1，x3 和 x6，我们定义集合 S = {1, 3, 6} xS。比如 x\_1 表示 x 中除 x1 外的所有元素，x\_S 表示 x 中除 x1，x3，x6 外所有元素构成的向量。

矩阵 (matrix)：矩阵是二维数组，其中的每一个元素被两个索引而非一个所确定。我们通常会赋予矩阵粗体的大写变量名称，比如 A。

[ A1,1 A1,2 ]

[ A2,1 A2,2 ]

张量 (tensor)：在某些情况下，我们会讨论不只两维坐标的数组。一般地，一组数组中的元素分布在若干维坐标的规则网格中，我们将其称之为张量。我们使用这种字体 A 来表示张量 “A’’。张量 A 中坐标为 (i, j, k) 的元素记作 Ai,j,k。

注意：

1.转置 (transpose) 是矩阵的重要操作之一。矩阵的转置是以对角线为轴的镜像，这条从左上角到右下角的对角线被称为主对角线 (main diagonal)。。我们将矩阵 A 的转置表示为 A⊤，定义如下(A⊤)i,j = Aj,i.



即可以这么理解，按照主对角线进行镜像。

2.向量可以看作是只有一列的矩阵。对应地，向量的转置可以看作是只有一行的矩阵。有时，我们将向量表示成行矩阵的转置，写在行中，然后使用转置将其变为标准的列向量，比如 x = [x1, x2, x3]⊤.

3.标量可以看作是只有一个元素的矩阵。因此，标量的转置等于它本身，a = a⊤。

4.只要矩阵的形状一样，我们可以把两个矩阵相加。两个矩阵相加是指对应位置的元素相加，比如 C = A + B，其中 Ci,j = Ai,j + Bi,j。

5.标量和矩阵相乘，或是和矩阵相加时，我们将其与矩阵的每个元素相乘或相加，比如 D = a · B + c，其中 Di,j = a · Bi,j + c。在深度学习中，我们也使用一些不那么常规的符号。我们允许矩阵和向量相加，产生另一个矩阵：C = A + b，其中 Ci,j = Ai,j + bj。换言之，向量 b 和矩阵 A 的每一行相加。这个速记方法使我们无需在加法操作前定义复制向量 b 到矩阵的每一行。这种隐式地复制向量 b 到很多位置的方式，被称为广播 (broadcasting)。

##### 1.3.1.2 矩阵和向量相乘

两个矩阵 A 和 B 的矩阵乘积 (matrixproduct) 是第三个矩阵 C。为了使乘法定义良好，矩阵 A 的列数必须和矩阵 B 的行数相等。如果矩阵 A 的形状是 m × n，矩阵 B 的形状是 n × p，那么矩阵 C 的形状是 m × p。矩阵乘积可以作用于两个或多个并在一起的矩阵，C = AB，该乘法操作定义为Ci,j =∑Ai,k Bk,j .需要注意的是，两个两个矩阵的标准乘积不是指两个矩阵中对应元素的乘积。不过，那样的矩阵操作确实是存在的，被称为元素对应乘积 或者哈达玛乘积A◎B。两个相同维数的向量 x 和 y 的点积 (dot product) 可看作是矩阵乘积 x⊤y。我们可以把矩阵乘积 C = AB 中计算 Ci,j 的步骤看作是 A 的第 i 行和 B 的第 j 列之间的点积。矩阵支持分配A（B+C）=AB+AC 结合律ABC=ACB 不支持结合律。两个向量的点积 (dot product) 满足交换律x⊤y = y⊤x（x⊤y =（x⊤y)⊤= y⊤x）。矩阵乘积的转置有着简单的形式：(AB)⊤ = B⊤A⊤。

##### 1.3.1.3 单位矩阵和逆矩阵

单位矩阵：I3

[ 1 0 0 ]

[ 0 1 0]

[ 0 0 1]

A\*A- =I3 A乘以A逆矩阵=单位矩阵 Ax=b A A-X = b A- x=b A-