1.2.1 数据

毋庸置疑，如果没有数据，那么数据科学毫无用武之地。每个数据集由一个个样本（example, sample）组成，大多时候，它们遵循独立同分布(independently and identically distributed, i.i.d.)。样本有时也叫做数据点（data point）或者数据实例（data instance），通常每个样本由一组称为特征（features, 或协变量（covariates））的属性组成。机器学习模型会根据这些属性进行预测。在上面的监督学习问题中，要预测的是一个特殊的属性，它被称为标签（label，或目标（target））。

当处理图像数据时，每一张单独的照片即为一个样本，它的特征由每个像素数值的有序列表表示。比如，200×200彩色照片由$200 \times 200 \times 3 = 120000$个数值组成，其中的“3”对应于每个空间位置的红、绿、蓝通道的强度。再比如，对于一组医疗数据，给定一组标准的特征（如年龄、生命体征和诊断），此数据可以用来尝试预测患者是否会存活。

当每个样本的特征类别数量都是相同的时候，其特征向量是固定长度的，这个长度被称为数据的维数（dimensionality）。固定长度的特征向量是一个方便的属性，它可以用来量化学习大量样本。

然而，并不是所有的数据都可以用“固定长度”的向量表示。以图像数据为例，如果它们全部来自标准显微镜设备，那么“固定长度”是可取的；但是如果图像数据来自互联网，它们很难具有相同的分辨率或形状。这时，将图像裁剪成标准尺寸是一种方法，但这种办法很局限，有丢失信息的风险。此外，文本数据更不符合“固定长度”的要求。比如，对于亚马逊等电子商务网站上的客户评论，有些文本数据很简短（比如“好极了”），有些则长篇大论。与传统机器学习方法相比，深度学习的一个主要优势是可以处理不同长度的数据。

一般来说，拥有越多数据的时候，工作就越容易。更多的数据可以被用来训练出更强大的模型，从而减少对预先设想假设的依赖。数据集的由小变大为现代深度学习的成功奠定基础。在没有大数据集的情况下，许多令人兴奋的深度学习模型黯然失色。就算一些深度学习模型在小数据集上能够工作，但其效能并不比传统方法高。

请注意，仅仅拥有海量的数据是不够的，我们还需要正确的数据。如果数据中充满了错误，或者如果数据的特征不能预测任务目标，那么模型很可能无效。有一句古语很好地反映了这个现象：“输入的是垃圾，输出的也是垃圾。”（“Garbage in, garbage out.”）此外，糟糕的预测性能甚至会加倍放大事态的严重性。在一些敏感应用中，如预测性监管、简历筛选和用于贷款的风险模型，我们必须特别警惕垃圾数据带来的后果。一种常见的问题来自不均衡的数据集，比如在一个有关医疗的训练数据集中，某些人群没有样本表示。想象一下，假设我们想要训练一个皮肤癌识别模型，但它（在训练数据集中）从未“见过”黑色皮肤的人群，这个模型就会顿时束手无策。

再比如，如果用“过去的招聘决策数据”来训练一个筛选简历的模型，那么机器学习模型可能会无意中捕捉到历史残留的不公正，并将其自动化。然而，这一切都可能在不知情的情况下发生。因此，当数据不具有充分代表性，甚至包含了一些社会偏见时，模型就很有可能有偏见。

1.2.2 模型

大多数机器学习会涉及到数据的转换。比如一个“摄取照片并预测笑脸”的系统。再比如通过摄取到的一组传感器读数预测读数的正常与异常程度。虽然简单的模型能够解决如上简单的问题，但本书中关注的问题超出了经典方法的极限。深度学习与经典方法的区别主要在于：前者关注的功能强大的模型，这些模型由神经网络错综复杂的交织在一起，包含层层数据转换，因此被称为深度学习（deep learning）。在讨论深度模型的过程中，本书也将提及一些传统方法。

1.2.3 目标函数

前面的内容将机器学习介绍为“从经验中学习”。这里所说的“学习”，是指自主提高模型完成某些任务的效能。但是，什么才算真正的提高呢？在机器学习中，我们需要定义模型的优劣程度的度量，这个度量在大多数情况是“可优化”的，这被称之为目标函数（objective function）。我们通常定义一个目标函数，并希望优化它到最低点。因为越低越好，所以这些函数有时被称为损失函数（loss function，或cost function）。但这只是一个惯例，我们也可以取一个新的函数，优化到它的最高点。这两个函数本质上是相同的，只是翻转一下符号。

当任务在试图预测数值时，最常见的损失函数是平方误差（squared error），即预测值与实际值之差的平方。当试图解决分类问题时，最常见的目标函数是最小化错误率，即预测与实际情况不符的样本比例。有些目标函数（如平方误差）很容易被优化，有些目标（如错误率）由于不可微性或其他复杂性难以直接优化。在这些情况下，通常会优化替代目标。

通常，损失函数是根据模型参数定义的，并取决于数据集。在一个数据集上，我们可以通过最小化总损失来学习模型参数的最佳值。该数据集由一些为训练而收集的样本组成，称为训练数据集（training dataset，或称为训练集（training set））。然而，在训练数据上表现良好的模型，并不一定在“新数据集”上有同样的性能，这里的“新数据集”通常称为测试数据集（test dataset，或称为测试集（test set））。

综上所述，可用数据集通常可以分成两部分：训练数据集用于拟合模型参数，测试数据集用于评估拟合的模型。然后我们观察模型在这两部分数据集的性能。“一个模型在训练数据集上的性能”可以被想象成“一个学生在模拟考试中的分数”。这个分数用来为一些真正的期末考试做参考，即使成绩令人鼓舞，也不能保证期末考试成功。换言之，测试性能可能会显著偏离训练性能。当一个模型在训练集上表现良好，但不能推广到测试集时，这个模型被称为过拟合（overfitting）的。就像在现实生活中，尽管模拟考试考得很好，真正的考试不一定百发百中。

1.2.4 优化算法

当我们获得了一些数据源及其表示、一个模型和一个合适的损失函数，接下来就需要一种算法，它能够搜索出最佳参数，以最小化损失函数。深度学习中，大多流行的优化算法通常基于一种基本方法–梯度下降（gradient descent）。简而言之，在每个步骤中，梯度下降法都会检查每个参数，看看如果仅对该参数进行少量变动，训练集损失会朝哪个方向移动。然后，它在可以减少损失的方向上优化参数。

1.3 各种机器学习问题

在机器学习的广泛应用中，唤醒词问题只是冰山一角。前面唤醒词识别的例子，只是机器学习可以解决的众多问题中的一个。下面将列出一些常见的机器学习问题和应用，为之后本书的讨论做铺垫。接下来会经常引用前面提到的概念，如数据、模型和优化算法。

1.3.1 监督学习

监督学习（supervised learning）擅长在“给定输入特征”的情况下预测标签。每个“特征-标签”对都称为一个样本（example）。有时，即使标签是未知的，样本也可以指代输入特征。我们的目标是生成一个模型，能够将任何输入特征映射到标签（即预测）。

举一个具体的例子：假设我们需要预测患者的心脏病是否会发作，那么观察结果“心脏病发作”或“心脏病没有发作”将是样本的标签。输入特征可能是生命体征，如心率、舒张压和收缩压等。

监督学习之所以能发挥作用，是因为在训练参数时，我们为模型提供了一个数据集，其中每个样本都有真实的标签。用概率论术语来说，我们希望预测“估计给定输入特征的标签”的条件概率。虽然监督学习只是几大类机器学习问题之一，但是在工业中，大部分机器学习的成功应用都使用了监督学习。这是因为在一定程度上，许多重要的任务可以清晰地描述为，在给定一组特定的可用数据的情况下，估计未知事物的概率。比如：

- 根据计算机断层扫描（Computed Tomography，CT）肿瘤图像，预测是否为癌症；
- 给出一个英语句子，预测正确的法语翻译；
- 根据本月的财务报告数据，预测下个月股票的价格；

监督学习的学习过程一般可以分为三大步骤：

1. 从已知大量数据样本中随机选取一个子集，为每个样本获取真实标签。有时，这些样本已有标签（例如，患者是否在下一年内康复?）；有时，这些样本可能需要被人工标记（例如，图像分类）。这些输入和相应的标签一起构成了训练数据集；
2. 选择有监督的学习算法，它将训练数据集作为输入，并输出一个“已完成学习的模型”；
3. 将之前没有见过的样本特征放到这个“已完成学习的模型”中，使用模型的输出作为相应标签的预测。

整个监督学习过程如 图1.3.1 所示。

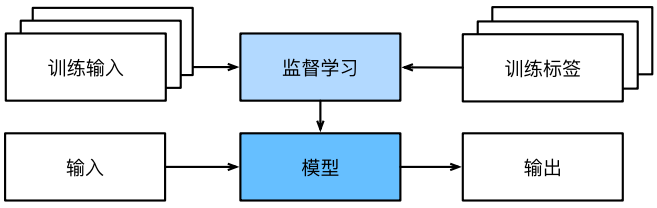

图1.3.1: 监督学习

综上所述，即使使用简单的描述给定输入特征的预测标签，监督学习也可以采取多种形式的模型，并且需要大量不同的建模决策，这取决于输入和输出的类型、大小和数量。例如，我们使用不同的模型来处理“任意

长度的序列”或“固定长度的序列”。

回归

回归（regression）是最简单的监督学习任务之一。假设有一组房屋销售数据表格，其中每行对应一个房子，每列对应一个相关的属性，例如房屋的面积、卧室的数量、浴室的数量以及到镇中心的步行距离，等等。每一行的属性构成了一个房子样本的特征向量。如果一个人住在纽约或旧金山，而且他不是亚马逊、谷歌、微软或Facebook的首席执行官，那么他家的特征向量（房屋面积，卧室数量，浴室数量，步行距离）可能类似于：$[600, 1, 1, 60]$。如果一个人住在匹兹堡，这个特征向量可能更接近$[3000, 4, 3, 10]$……当人们在市场上寻找新房子时，可能需要估计一栋房子的公平市场价值。为什么这个任务可以归类为回归问题呢？本质上是输出决定的。销售价格（即标签）是一个数值。当标签取任意数值时，我们称之为回归问题，此时的目标是生成一个模型，使它的预测非常接近实际标签值。

生活中的许多问题都可归类为回归问题。比如，预测用户对一部电影的评分可以被归类为一个回归问题。这里有一个小插曲：在2009年，如果有人设计了一个很棒的算法来预测电影评分，那可能会赢得100万美元的奈飞奖[12]。再比如，预测病人在医院的住院时间也是一个回归问题。总而言之，判断回归问题的一个很好的经验法则是，任何有关“有多少”的问题很可能就是回归问题。比如：

- 这个手术需要多少小时；
- 在未来6小时，这个镇会有多少降雨量。

即使你以前从未使用过机器学习，可能在不经意间，已经解决了一些回归问题。例如，你让人修理了排水管，承包商花了3小时清除污水管道中的污物，然后他寄给你一张350美元的账单。而你的朋友雇了同一个承包商2小时，他收到了250美元的账单。如果有人请你估算清理污物的费用，你可以假设承包商收取一些基本费用，然后按小时收费。如果这些假设成立，那么给出这两个数据样本，你就已经可以确定承包商的定价结构：50美元上门服务费，另外每小时100美元。在不经意间，你就已经理解并应用了线性回归算法。

然而，以上假设有时并不可取。例如，一些差异是由于两个特征之外的几个因素造成的。在这些情况下，我们将尝试学习最小化“预测值和实际标签值的差异”的模型。本书大部分章节将关注平方误差损失函数的最小化。

分类

虽然回归模型可以很好地解决“有多少”的问题，但是很多问题并非如此。例如，一家银行希望在其移动应用程序中添加支票扫描功能。具体地说，这款应用程序能够自动理解从图像中看到的文本，并将手写字符映射到对应的已知字符之上。这种“哪一个”的问题叫做分类（classification）问题。分类问题希望模型能够预测样本属于哪个类别（category，正式称为类（class））。例如，手写数字可能有10类，标签被设置为数字0～9。最简单的分类问题是只有两类，这被称之为二项分类（binomial classification）。例如，数据集可能由动物图像组成，标签可能是$\{\square, \square\}$两类。回归是训练一个回归函数来输出一个数值；分类是训练一个分类器来输出预测的类别。

然而模型怎么判断得出这种“是”或“不是”的硬分类预测呢？我们可以试着用概率语言来理解模型。给定一个样本特征，模型为每个可能的类分配一个概率。比如，之前的猫狗分类例子中，分类器可能会输出图像

[12] https://en.wikipedia.org/wiki/Netflix_Prize