第 1 章　概述

机器学习赋予计算机从数据中学习的能力

学习一个学科或一门课程，要避免“瞎子摸象，不得要领”，就需要先知其全貌、了解其大概、以及各部分之间的关系。这样在脑海中先形成一个全局的图景，之后就是让这个图景逐渐变得清晰、完整。本章所起的作用就是让读者先“知其概貌”。为此目的，本章将从整体上对“机器学习”进行一个概要性的介绍，从机器学习的基本概念开始，以机器学习的历史与现状收尾，谈及其术语、任务、分类等重要部分及其相互间的内在联系。虽然比较粗略，然期望达到“知其概貌”就足矣。细节的详细阐述是后续章节的任务。

1.1　什么是机器学习

什么是“机器学习”？可以用一句话来概括：机器学习赋予计算机从数据中学习的能力。这句话里面的关键词汇是“学习”。这个词汇当然读者再熟悉不过了，读者不就是正在“学习”机器学习这门课程吗？那么，读者学习的目的是什么呢？当然是希望通过这个过程来提升自己解决相关问题的能力。所以可以说，如果一个系统能够通过执行某个过程来提升他的能力，这就是学习。

类比人的学习，可以在老师的指导下进行学习，也可以在没有老师的指导下自学，还可以在实践试错的过程中来学习，等等。所以，据此可以将机器学习粗略划分为几大类：有监督、无监督、自监督、环境监督[1]。接下来分别谈一谈。

1.1.1　有监督学习

类比人的学习，所谓“有监督学习”就是指在老师指导下的学习，或者说给定“正确答案”的学习。比如从学习材料中学习了一个新的知识点，就会去做题，做完以后通过与正确答案进行比较，就会发现哪里存在问题，从而去纠错，下次遇到了新的类似的问题，就有很大可能性不会犯错了，也就是解题能力或者说对知识的运用能力就得到提升了。

用机器学习的术语来讲，“学习材料（包括习题及其正确答案）”就是事先收集好的训练数据（称为训练集），“新的类似的问题”就是测试数据（称为测试集）。“做题、对答案、纠错”就是有监督学习过程。

更正式地，用集合$D=\{(\boldsymbol{x_1},y_1),\dots,(\boldsymbol{x_N},y_N)\}$表示有$N$个样本的训练集，其中第$i$个元素$(\boldsymbol{x_i},y_i),i=1,\dots,N$是一个有序对（可以对应为 Python 中的元组），$\boldsymbol{x_i}$表示第i个数据（第i个习题）——一般称为“特征向量”，y_i表示第i个数据对应的标签（第i个习题的正确答案）——一般是一个标量。类似地，用集合$T=\{(\boldsymbol{x_1},y_1),\dots\dots,(\boldsymbol{x_M},y_M)\}$表示有$M$个样本的测试集。那么，有监督学习就是要在训练集$D$上学习一个函数$y=f(\boldsymbol{x})$，并在测试集$T$上取得较好的效果。等价地，也可以说有监督学习是要在训练集D上学习一个条件概率分布$p(y|\boldsymbol{x})$。后面这种基于概率的形式化方式有一个显著的优点，考虑到了不确定性——做新的类似的题目，一般也不会有 100%的把握做对。实际上，机器学习也常被称为“统计机器学习”或“统计学习方法”就是这个原因。

将学习到的函数$y=f(\boldsymbol{x})$或条件概率分布$p(y|\boldsymbol{x})$称为机器学习模型，有了这个模型，给

注 1：半监督也是比较常见的一类，本书限于篇幅不予涉及。简单来说，半监督就是针对部分样本有标签、部分样本无标签的数据集的学习。其介于有监督和无监督之间。

定一个输入$\boldsymbol{x_i}$，就可以给出预测值$\hat{y}_i$或$p(\hat{y}_i)$。特别地，对于训练过程，需要定义一个损失函数$L(\hat{y}_i, y_i)$或$L(p(\hat{y}_i), p(y_i))$，用来度量预测值偏离真实值的程度。这样，训练目标就是让L的值尽可能小。比如，对于一个单选题，可以定义一个简单的损失函数：只有答对了损失才为 0，其它情况损失都为 1。这就是所谓的“0-1”损失函数。

有监督学习根据标签y_i（也称为目标变量）是否连续可以进一步分为分类（y_i是离散变量）和回归（y_i是连续变量）[1]。第 2 章和第 3 章将分别讨论这两种情况。这里先给出一些简单的例子。

图 1.1 给出了一个二分类任务的例子。如图 1.1 所示，共有 30 个训练样本，每个样本$\boldsymbol{x}$有两个特征(x_1, x_2)，即特征向量$\boldsymbol{x}$为二维。30 个样本中，15 个样本为正类（图中用带“+”号的圆圈表示），15 个样本为负类（图中用带“-”号的圆圈表示）。注意，标签y_i是离散变量，只能取两个值，分别对应正类和负类。目标是通过有监督学习得出一个合理的规则，即找到一个决策边界（图中虚线所示），能够将两类样本完全分开。然后，对于新的测试样本，也能对其进行正确分类。细心的读者一定注意到了一点，图 1.1 中刻意让正负样本数刚好相同，这其实是机器学习对于数据的一个基本要求——“样本数类间平衡”。实际中，不一定能做到各个类别的样本数刚好相同，但是应该努力做到基本相同。如果确实存在一些类别的样本数差异较大，就需要从样本难度、损失权重等角度加以补偿，比如样本数多的类别丢弃一些简单样本、加大样本数少的类别对应的损失权重等。

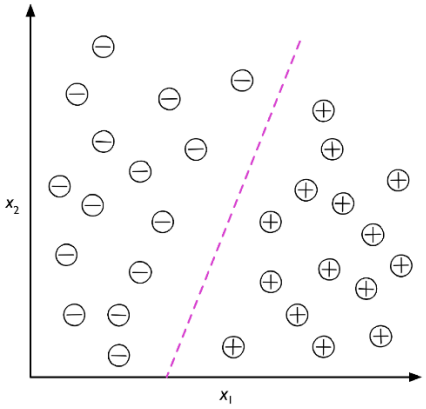


图 1.1 一个二分类任务

多分类任务的一个典型例子是手写数字识别。首先，收集包含 0~9 所有数字（共 10 个类别）的手写样本，构建“类别平衡的”训练集。然后，在训练集上通过有监督学习得到一个模型$y = f(\boldsymbol{x})$或$p(y|\boldsymbol{x})$。当输入新的手写数字时，该模型能够以较高的正确率将其识别为正确的数字（即 0~9 中的某一个）。

图 1.2 是线性回归的一个例子。如图 1.2 所示，特征向量$\boldsymbol{x}$为横轴，标签$y \in R$为纵轴，目标是拟合出一条直线（图中虚线所示）使得所有训练样本点（图中用带“+”号的圆圈表示）与其对应的直线上样本点的距离（图中用短实线标出了两个距离）在总体上达到最小。之后，对于测试样本$\boldsymbol{x}$，就能给出预测值$\hat{y}$。读者可以思考下，为何不采用样本点到该直线的垂直距离呢？（习题 1.7）

注 1：离散变量是取有限个值的变量，比如表示 10 个数字（0~9）的变量就是一个例子。而连续变量是一个实数，比如今天的气温就是一个例子。

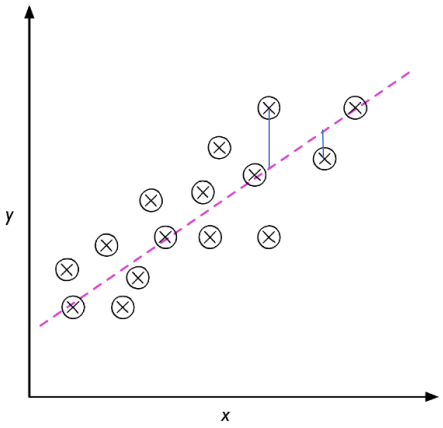


图 1.2 一个线性回归任务

1.1.2 无监督学习

类似地，没有老师指导下的学习或者说未给定“正确答案”的学习，就被称为“无监督学习”。这种情况下，训练集$D = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$，其中第$i$个元素$\boldsymbol{x}_i, i = 1, \ldots, N$表示第$i$个训练样本。注意，不再有$\boldsymbol{x}_i$对应的标签$y_i$！测试集$T = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$类似。

尽管没有了标签y_i，仍然要学习关于目标变量y的函数$y = f(\boldsymbol{x})$或条件概率分布$p(y|\boldsymbol{x})$，从而探索数据蕴含的结构及信息，实现对未知的预测。降维与聚类是常见的两种无监督学习任务，本书将分别在第 4 章和第 5 章进行讨论。这里先看一个关于聚类的简单例子。如图 1.3 所示，目标是将一些无标签数据（图中的实心圆点所示）根据特征x_1和x_2的相似性分成三个“簇”（图中的虚线圆所示）。读者会发现，这里的“簇”比较类似于有监督学习中“类”的概念（所以聚类也常被称为“无监督分类”）。但是，要特别注意，“簇”和“类”有一个关键区别，前者事先并不知道，而后者则是事先给定的。比如，图 1.3 中，找到每个“簇”具体是什么就是需要完成的任务。对比 1.1.1 给出的二类和多类任务，这些“类”都是由类别标签事先给定的。

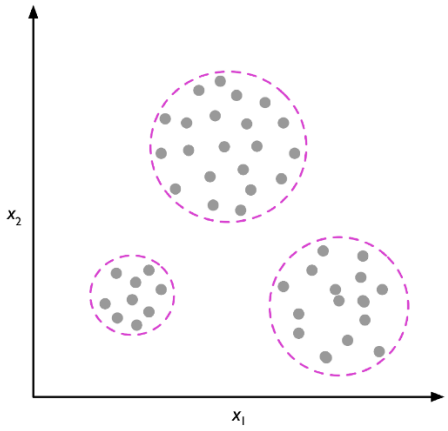


图 1.3 一个聚类任务

说到这里，聪明的读者马上会问一个问题：没有了“正确答案”，那如何来度量模型的预测值$\hat{y}_i$或$p(\hat{y}_i)$是好还是坏呢？确实，这种情况下，不可能再像有监督学习一样，定义一个损失函数L。能做的是，提供一些客观的或主观的评估标准，然后基于这些标准对模型性能进行度量和评估。以聚类为例，可以提供一个参考模型（相当于给一个“参考答案”），比如

领域专家给出的“簇”划分，这可以认为是一个客观的标准；也可以定义一些来自于几何直观的度量，比如每个聚类“簇”的平均直径、不同聚类“簇”之间的最小距离等，这可以认为是一些主观的标准。具体细节，第 5 章再来详谈。

通过上面的介绍，读者一定体会到了无监督学习的核心特点：好与坏的标准自己来定！这就给了模型设计相当大的自由发挥空间，当然是一个显著的优势了。由此启发，可以进一步追问两个问题：无监督与有监督一定是“井水不犯河水，各自独善其身”吗？无监督与有监督可以相互结合甚至融合，形成“你中有我，我中有你”的局面吗？这些问题同样放在第 5 章来详谈。

1.1.3 自监督学习

对于互联网上的海量数据，有监督学习面临一个基本的困境——这些数据没有标签，一般而言也不太可能给所有这些数据人工打上标签，因此“先人工后智能”的方式行不通。那么，无监督学习呢？读者会说，无监督学习不需要打标签嘛。确实如此，这正是我们思考的方向。

所谓“自监督学习”其实可以认为是无监督学习的一种，只不过随着其近些年的快速发展，人们就将其单独列出来了，作为一种独立的重要的学习方式进行研究。这样，我们在 1.1.2 中探讨的无监督学习就特指传统的无监督学习方法，比如降维和聚类这些。

简单来说，自监督学习就是利用数据自身包含的信息进行学习。这样说还是比较笼统，举一个自然语言处理的例子来进行说明。比如“今天的气温比昨天的气温高出 2 度”这句话，为了充分利用句子本身包含的信息，我们可以将句子中的“字词”（称为 token）进行随机遮挡，然后让模型来预测这些被遮挡的字词。这样，被遮挡的句子本身就提供了标签，我们就可以类似有监督学习，定义一个损失函数L，用来度量预测值偏离真实值的程度。由此就形成了一种典型的自监督学习机制——遮挡语言建模。

对于遮挡语言建模，可以给出一个形式化的描述。训练集$D = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$包含$N$个句子，其中第$i$个句子$x_i = \{w_1, w_2, \dots, w_M\}$，$w_j, j = 1, \dots, M$表示该句子的第$j$个字词。目标是预测句子$\boldsymbol{x}_i$中被遮挡的第$j$个字词是某个字词的条件概率，即$p(w_j|\boldsymbol{x}_i)$。损失函数$L(p(\widehat{w}_j), p(w_j))$用来度量预测值偏离真实值的程度。

关于自监督学习，本书第 7 章将进一步讨论。

1.1.4 环境监督与强化学习

环境监督与强化学习被认为是生物适应环境的基本方式，所以其重要性不言而喻。举一个读者都熟悉的例子，小孩学习走路。刚开始，小孩很容易摔倒，经过一段时间反复的试错，小孩慢慢地就越走越稳了。这里“环境”起到了教师的作用，所以可以称之为“环境监督”，对应的学习方式称为“强化学习”。

更正式地，假设模型在时刻t处于状态$\boldsymbol{s}_t \in S$（S称为“状态空间”），接下来模型将根据策略$\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$选择一个动作$\boldsymbol{a}_t \in A$（$A$称为“动作空间”）。由于采取了动作$\boldsymbol{a}_t$，模型将立即接收到来自环境的反馈$r_t = r(s_t, a_t)$，并且到达时刻$t+1$的状态$\boldsymbol{s}_{t+1}$。这里，$r(s, a)$被称为“奖励函数”。重复这个过程，直到到达某个结束状态，就称为完成了一轮训练。根据需要，可以进行多轮训练。训练的最终目标是最大化整个过程中的“总奖励”。注意，总奖励可能即时获得，也可能延后获得，这恰恰是强化学习的难点所在。图 1.4 给出了一个原理框图。

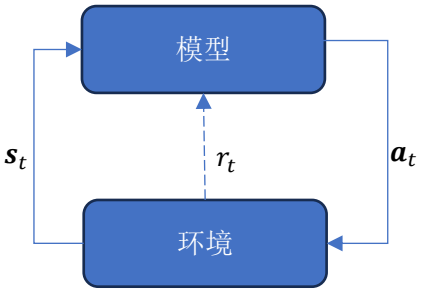


图 1.4 强化学习

对应小孩学习走路的例子。小孩的身体状态构成状态空间S，当前小孩的身体状态是$\boldsymbol{s}_t$，根据他自身的判断（即策略$\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$）选择了动作$\boldsymbol{a}_t$（比如左脚迈出一小步）。如果动作$\boldsymbol{a}_t$（即左脚迈出一小步）比较恰当，那么环境就会立即给予一个正向的反馈——对的，继续走；如果不恰当，可能摔倒在地就是环境的立即反馈，从而先得爬起来才能继续往前走。可见，环境的反馈$r_t = r(s_t, a_t)$ 将使得小孩到达新的状态$\boldsymbol{s}_{t+1}$。正是这样一个试错的过程，使得小孩不断调整自己并学习到如何走稳的策略。

本书第 8 章将对环境监督与强化学习进行讨论。

1.2 机器学习的三个重要方面

通过 1.1 节的学习，相信读者对机器学习的基本流程已经有了一个初步的认识：第一步，准备数据（包括训练集和测试集）；第二步，在训练集上训练模型；第三步，在测试集上评估模型。其实，这三步恰恰反映了机器学习的三个重要方面：数据的表示（见 1.2.1 节）、模型的最优化（见 1.2.2 节）、模型的评估（见 1.2.3 节）。本节就来进一步探讨这三个方面，目的是借此介绍相关的基本概念，完善机器学习的基本知识体系，从而加深对机器学习的认识。

1.2.1 数据的表示

数据及其表示是机器学习基本流程的第一环。不同的数据表示，往往意味着不同的机器学习范式。为什么这么说呢？读者可以想一想，机器学习应用在各种模态（数值、文本、语音、图像、视频等）和各行各业（农业、工业、服务业、教育等），其数据的形态千差万别，可能都用一种统一的方式来对其进行表示吗？笔者认为答案是否定的，因为数据的表示方式取决于具体的应用。举个有代表性的例子，语音一般是经过采样、量化、编码的过程（称为 PCM 编码）存储为.wav 文件，这是语音数据的原始表示。然而，为了提高语音识别的精度，需要对其进行傅里叶变换，得到频率域的表示（其实就是基于傅里叶级数的表示）。

根据数据是否需要先进行变换，可以将机器学习的范式划分为两阶段方式和端对端方式。两阶段方式是经典的机器学习范式：先对原始数据（比如语音的 PCM 编码）进行变换（比如傅里叶变换）得到特征表示（比如傅里叶级数表示），然后将特征表示作为机器学习模型的输入，进而完成模型的训练。可见，所谓两阶段就是“特征提取”+“模型训练”。这种学习范式的成败很大程度上取决于“特征提取”的好坏。好的特征表示配合一个相对简单的模型往往也能取得较好的效果，而坏的特征表示既使把模型搞得很复杂可能也无济于事。由于“特征提取”强烈依赖于人的专业知识和经验，所以研究者们也将其称为“特征工程”或“手工特征”，这是非常恰当的。

端对端方式就是针对两阶段方式存在的关键问题而提出的。既然“特征提取”强烈依赖

于人的专业知识和经验，对于人的要求很高，那么为什么不让机器去干这件事呢？给机器原始数据，让机器自己去学习应该采用哪种特征表示。这就是所谓的端对端方式，“特征提取”也成为了“模型训练”的一部分。输入原始数据（比如语音的 PCM 编码），输出最终结果（比如语音对应的文本）。这就很好的解决了两阶段方式存在的关键问题——“特征提取”强烈依赖于人的专业知识和经验。由此，大幅拉低了应用机器学习解决实际问题的门槛，只要你有数据有需求，就可以训练模型并将其应用到实际中。

话说回来，两阶段方式真的就只有被扔进历史的垃圾堆里了？！事实并非如此！对于一些常见问题或者一些已经解决得比较好的问题，没理由不采用两阶段方式啊——模型更容易训练、更容易解释等，都是优点。一方面，模型更简单，所以模型更容易训练、更容易解释。另一方面，“手工特征”就是人设计的特征，当然更容易解释了。

无论是两阶段方式还是端对端方式，数据在变换或送入模型之前，一般都要经历一个预处理阶段。这个阶段可能会涉及数据清洗（去掉无效数据和明显错误的数据）、数据缺失值填充（填充什么值颇有讲究）、数据归一化或标准化、数据降维与可视化等。这些方面将在后面的章节中结合具体的问题和具体的机器学习模型来详谈。

1.2.2 模型的最优化

从 1.1 节读者一定注意到了一点，机器学习模型的训练最终都归结为一个最优化问题，比如有监督学习的损失函数最小化、强化学习的奖励函数最大化等。不同的机器学习模型需要解决不同的最优化问题，有些问题很简单、有些问题就要复杂得多，有些问题能够找到全局最优解、有些问题却只能找到局部最优解或近似解。这些具体情况，后面结合具体的机器学习模型来详谈，此处不作展开。这里重点讲一讲两个带有全局性、普遍性的方面。

第一，根据机器学习模型是否含有需要优化的模型参数，可以分为含参模型和非参模型。回顾 1.1.1 节中用函数$y = f(\boldsymbol{x})$或条件概率分布$p(y|\boldsymbol{x})$来表示机器学习模型，那么对于含参模型，应该将其改写为$y = f(\boldsymbol{x};\boldsymbol{\theta})$或$p(y|\boldsymbol{x};\boldsymbol{\theta})$，其中，$\boldsymbol{\theta}$就是需要优化的模型参数。模型学习就是要找到$\boldsymbol{\theta}^*$，使得$L(\boldsymbol{\theta}^*)$取得最小值。大多数的机器学习模型都属于含参模型，因此本书也会将其作为重点进行介绍。非参模型的典型例子是 k-近邻（见本书 2.1 节），其基本特点是不（显式的）含有需要优化的模型参数。为什么要加上“（显式的）”？原因如下。k-近邻模型有一个超参数k，这个正整数k指定了要找几个相邻样本点，决定了模型的复杂程度，k取 1 则模型最复杂。因此，如果假设训练集有N个样本点，则可以视为 k-近邻模型有N/k个参数。这些参数并不需要显式的去优化，所以称为“不（显式的）含有需要优化的模型参数”。

读者要注意区分“模型参数”和“超参数”这两个概念。“模型参数”是需要优化的目标，而“超参数”更多是根据经验来设定。当然，也有所谓的“自动机器学习”，会根据一些规则或算法进一步去搜索超参数，这也是很有意义的工作。

第二，不要把机器学习和最优化混为一谈。最优化本质上就是求解一个数学问题（比如找到函数的最小值），求解出来了，问题就解决了。机器学习则不同，最优化只是其解决问题的一个环节，最优化求解出来的结果好不好，还得在测试集上进行测试和评估。具体来说，既然是在训练集上求解的最优化问题，那么这个求解的结果只能保证对于训练集最优。而我们更关心的是对于训练集之外的数据，这个求解的结果究竟表现如何。训练集之外的数据一般是无穷多的，所以一般选择其中一部分作为测试集，并在其上对最优化求解的结果进行测试和评估。如果测试集上的表现也比较好，才能说最优化求解的结果（即训练得到的模型）比较好。

这就自然引出了机器学习的三个核心概念：训练误差、推广误差和测试误差。训练误差和测试误差好理解，分别对应训练集和测试集上的误差。那么，推广误差指的是什么误差呢？

为了说清楚这一点，首先要回顾一下概率与统计里讲到的“独立同分布”概念。

所谓“独立同分布”指的是，对于一个概率分布P（比如伯努利分布），对其进行独立采样，每采样一次就得到一个样本。由于随机性，显然这些采样得到的样本不尽相同（比如扔硬币的实验，采样得到的结果是有些为正面、有些为反面）。虽然采样得到的样本不尽相同，但如果将所有这些样本拿到一起来看（比如扔硬币的实验，计算一下总共有多少次正面、多少次反面），会发现采样的结果是符合概率分布P的（比如扔硬币的实验结果符合正反面概率均为 50%的伯努利分布），采样次数越多符合度就越高（由大数定律保证）。我们称这些采样得到的样本是“独立同分布”的（比如这里的伯努利分布）。

实际上，“独立同分布”是机器学习的基本假设之一。设数据的真实分布为P，对其进行独立采样，得到训练集D，然后在D上训练模型。训练得到的模型究竟怎么样，训练误差显然不足以作为标准。如同学习一门课程，把平时的习题反复做了很多遍，一道都不会出错，但这也不能代表这门课程就学得很好了。只有对所有没见过的题，都能做对，才能代表确实学得很好了。由此，可以说，“推广误差”指的就是模型在训练集D之外的所有数据上的误差。“推广误差”低才能真正说明模型性能好、精度高。

读到这里，聪明的读者一定会发问：“训练集D之外的所有数据”这个在实践中办不到啊？确实如此，所以“推广误差”一般只能是一个理论值。所以如上面所说，实践中只能从真实分布P中再采样一部分数据，将其作为测试集，这样通过测试误差来逼近推广误差。

尽管推广误差一般只能是一个理论值，真正理解它对于理解机器学习却是至关重要的。比如关于训练集D和测试集T的样本数比例问题，为了得到较低的推广误差，从而在测试集T上有良好表现，会让训练集D的样本数N倍于测试集T（N的一个典型值为 9）。为什么这么说呢？考虑极限情况，假设D穷尽了真实分布P的所有样本，那么这时候D上的训练误差就等于推广误差。这充分说明D的样本数确实越多越好。由此，还可以引申出一个问题供读者思考：除了数量上越多越好，质量上有没有要求呢？

实践中，为了在训练过程中对推广误差及时进行估计，还会从真实分布P中或训练集中采样一部分数据，将其作为验证集V。这样，对于迭代式的训练算法（比如 2.3 节将介绍的梯度下降），每迭代完一轮就可以在V上进行一次测试和评估，得到验证误差。关于这一点，紧接着的“模型的评估”还会进一步详谈。

总结一下第二点，机器学习关心的是模型对数据真实分布P的拟合情况。由于数据真实分布P并不知道（如果知道了，就不用再搞什么机器学习了），因此，在实践中，需要通过训练误差和测试误差来衡量模型的好与坏。如果训练误差和测试误差都高，说明模型处于“欠拟合”状态；如果训练误差和测试误差都低，说明模型处于“最佳拟合”状态；而如果训练误差低、测试误差高，则说明模型已经处于“过拟合”状态。

如图 1.5 所示，纵轴表示误差，分别画出了训练误差（图中实曲线所示）和测试误差（图中虚曲线所示）；横轴为训练轮数或模型参数量。如果横轴为训练轮数，则图 1.5 反映的是拟合情况随着模型的训练程度而变化的情况：以图中的垂直虚线为界，左边为训练不够（欠拟合）、垂直虚线为训练合适（最佳拟合）、右边为训练过度（过拟合）。如果横轴为模型参数量，则图 1.5 反映的是拟合情况随着模型的复杂度而变化的情况：同样以图中的垂直虚线为界，左边为模型过于简单（欠拟合）、垂直虚线为模型合适（最佳拟合）、右边为模型过于复杂（过拟合）。我们的目标是通过对模型性能的评估，进而选择合适的训练程度和模型复杂度。

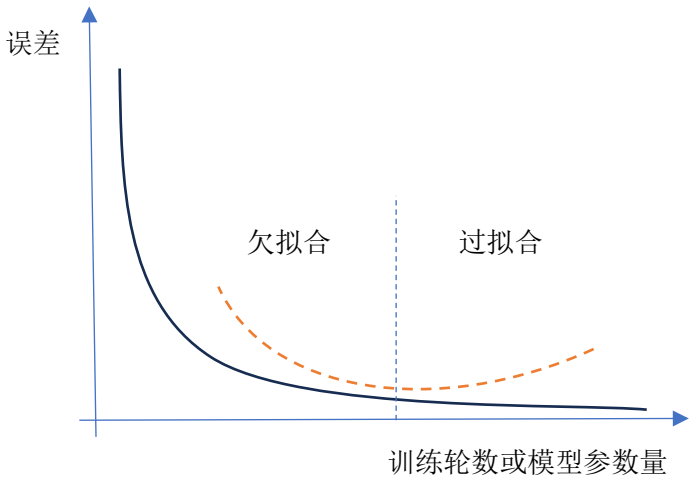


图 1.5 误差与训练轮数或模型参数量的变化关系

图 1.5 假定了训练集和测试集都是固定的。实际上，训练数据的多少和复杂程度也会影响到模型的拟合状态。一般而言，训练数据太少太简单，而模型过于复杂，则模型将对数据过拟合；相反，训练数据太多太复杂，而模型过于简单，则模型将对数据欠拟合。一些模型（比如线性模型）的主要问题是欠拟合；而另外一些模型（比如多层神经网络）的主要问题则是过拟合，需要海量的训练数据。

1.2.3 模型的评估

紧接 1.2.2 节，训练误差、验证误差和测试误差是模型性能评估的三个基本指标。以有监督学习为例，“误差”反映的是模型预测值$\hat{y}_i$和真实值（标签）y_i的差异。以分类为例，如果$\hat{y}_i$和y_i相同，则误差为 0；否则，记误差为 1，如式（1.1）所示：

$$Err = \sum_{i=1}^{N} I(\hat{y}_i, y_i) \tag{1.1}$$

其中，$I(\hat{y}_i, y_i)$为指示函数：$\hat{y}_i = y_i$，则值为 0；$\hat{y}_i \neq y_i$，则值为 1。所以，误差Err反映的就是有多少个样本被分错。将Err除以总样本数N，就得到错误率$E = Err/N$。而$1 - E$则被称为“精度”，显然精度反映的就是正确率，即样本被分对的比例。

如 1.2.2 节所说，实践中往往需要在训练过程中对模型的推广误差及时进行估计。虽然上面讲到是直接计算训练误差和验证误差，实际操作中更常见的是直接使用训练损失和验证损失。“误差”和“损失”关系紧密，但概念上不能混淆。1.1.1 节中讲到了损失函数$L(\hat{y}_i, y_i)$，其实式（1.1）定义的误差就是最简单的关于分类问题的损失函数——0-1 损失：分类正确，损失为 0；否则，损失为 1。损失函数往往还有额外的要求，比如需要“正则项”来提高模型推广能力，再比如需要“可导”以满足梯度下降优化算法的要求等，后面会跟具体的模型结合起来详谈。

有了评估标准，那么紧接着的一个问题就是：如何基于有限的数据对模型的性能进行可靠的评估呢？如果就是一个训练集、一个验证集（或测试集），那也就是一次训练一次评估（称为“留出法”），这在“统计”的意义上是不充分的。为了解决这个问题，人们提出了“交叉验证”的评估方法。即把训练集D划分为k个相同的子集，然后依次选择$k - 1$个子集进行训练，并用剩下的一个子集进行测试，从而得到一个评估结果。这样，在D上就能进行k次训

练和评估，故而称之为“k折”交叉验证。对这k个评估结果，既可以求“均值”作为最终评估结果，也可以进一步计算“方差”考察每次评估的稳定性如何等。这种统计分析使得我们对评估结果的客观性、可靠性更有把握。比如，图 1.6 给出了训练集 D 上的 10 折交叉验证。特别地，如果取$k = N$（N为D中样本个数），就得到一种特殊的评估方法——留一法。这种方法的优点是用于训练的样本比较充分，仅比D中样本数 N 少一个，因而往往能更好发掘出模型的能力。缺点是不适合 N 很大的场合，计算量太大。

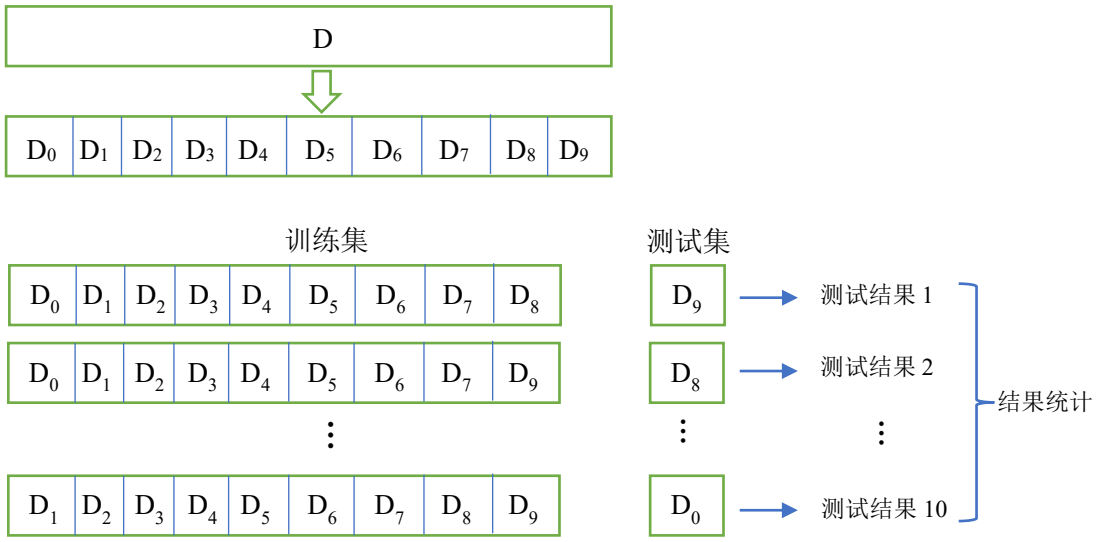


图 1.6 训练集 D 上的 10 折交叉验证

划分数据集的时候，还有一个问题需要引起特别注意：要保持原数据集中的类别比例。比如，训练集D中正负样本都为 500 个，那么从其划分出的 10 个子集里（每个子集 50 个样本），每个子集都应该有 25 个正样本和 25 个负样本。这种保持类别比例的划分方式，在统计学中被称为“分层采样”，目的是避免因类别分布的差异而导致的模型偏差。

误差或精度是对模型性能的有效评估方式，但还是比较粗，不能反映具体是什么样的错：是把 A 类分成了 B 类，还是反之。为了分清楚具体的错误类型，人们提出了“混淆矩阵”的概念。如表 1.1 所示，有 A、B、C、D、E 共 5 个类别，最左列表示真实值（标签），最顶行表示预测值。那么，真实值和预测值一致的情况记为TX（X对应具体的类别，即 A、B、C、D、E 中的任意一个），对应表 1.1 中的主对角线元素（表中用粗体标出）；否则记为FY（Y对应预测错误的类别，是除去正确类别之外的任意一个类别）。比如“类 A”被识别为“类 A”，则记为TA；“类 A”被识别为“类 B”，则记为FB。依此类推。混淆矩阵清楚地反映出了出错的具体情况：有哪些错误，每种错误有多少。这对于模型的进一步改进和设计的权衡都有着重要的指导意义。

表 1.1 混淆矩阵

真实值/预测值	类 A	类 B	类 C	类 D	类 E
类 A	**TA**	FB	FC	FD	FE
类 B	FA	**TB**	FC	FD	FE
类 C	FA	FB	**TC**	FD	FE
类 D	FA	FB	FC	**TD**	FE
类 E	FA	FB	FC	FD	**TE**

在具体的数据集上，可以统计出所有TX和FY的值，进而可以定义“查准率”（precision）和“查全率”（recall）。如式（1.2）所示，类X的“查准率”P_X定义为TX和$(TX + \sum FX)$的商，

反映的是被预测为类 X 的样本中有多少是准确的（故而命名为“查准率”）。比如$P_A = TA/(TA + \sum FA)$，其中TA和FA对应表 1.1 中“类 A”一列中各项。如式（1.3）所示，类 X 的“查全率”R_X定义为TX和$(TX + \sum FY)$的商，反映的是类 X 的所有样本有多少被找回来了（故而命名为“查全率”）。比如$R_A = TA/(TA + FB + FC + FD + FE)$，其中$TA$和$FB \sim FE$对应表 1.1 中“类 A”一行中各项。

$$P_X = \frac{TX}{TX + \sum FX} \tag{1.2}$$

$$R_X = \frac{TX}{TX + \sum FY} \tag{1.3}$$

对于某个具体的模型来讲，总是可以通过调整超参数（比如阈值）而使得P_X等于 1 或R_X等于 1。因此，使得P_X和R_X同时靠近 1 的模型才是真正好的模型。然而，一般来讲，查准率和查全率是存在矛盾的，提高了前者往往就会降低后者，反之亦然。P-R 曲线很好的反映了这一点。如图 1.7 所示，对模型选用一系列的不同超参，从而得到一系列的P_X和R_X，然后，以P_X为纵轴、以R_X为横轴，标出这些点，并将这些点连接成平滑的曲线，就得到了所谓的 P-R 曲线。显然，P-R 曲线与两个坐标轴所围的面积越大，模型整体性能越好，这就自然引出了“曲线下面积（AUC）”这个概念。曲线下面积一般不方便计算，所以为了方便比较，提出了“平衡点”的概念：即P等于R的那个值。这个值既方便（越大认为越好），又综合考虑了P与R。

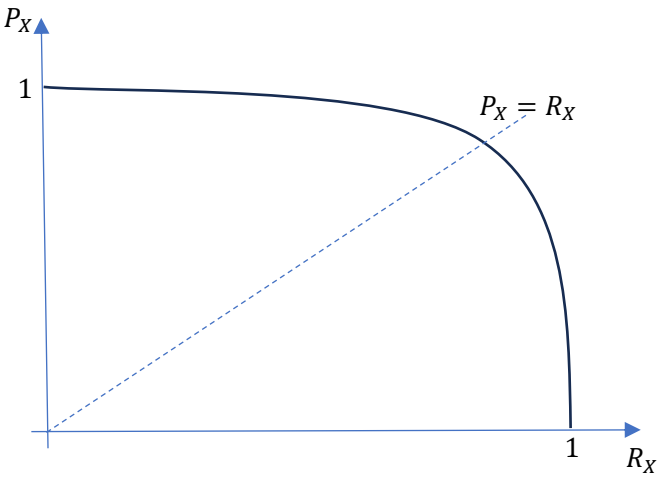


图 1.7 P-R 曲线

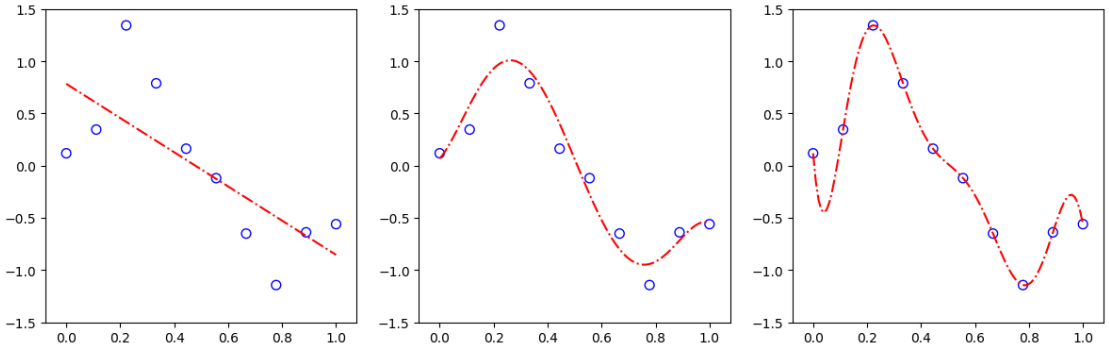
平衡点还是过于简单，因此为了更好地综合考虑P与R，进一步提出了$F1$度量，如式（1.4）所示。下面通过几种典型情况来分析下式（1.4）：如果$P = 1$而$R = 0.01$，得到$F1$约等于 0.02；如果$P = 0.01$而$R = 1$，得到$F1$也是约等于 0.02；如果$P = R$，则$F1 = P = R$。可见，$F1$确实较好的综合考虑了P与R，而且P与R是平等的，并不会偏向于哪一个。当然，如果在实践中，确实需要对P或R有所偏向，还可以使用更一般的F_β度量，这里就不再展开了。还有一点，以上都是针对一般的多分类而谈的，那么对于二分类这种特殊又常用的情况，有什么相同之处和不同之处呢（习题 1.3）？

$$F1 = \frac{2PR}{P + R} \tag{1.4}$$

基于以上这些评估标准，都可以类似画出图 1.5，从而选择出训练程度和复杂度都达到最佳的模型。特别地，对于一个复杂度确定的特定模型，我们关注训练程度，比如梯度下降

的训练轮数。一旦发现测试误差开始由低向高变化的时候，训练就应该停止了，这被称为“早期停止”，是常用的模型选择方法，简单而有效。

至于模型的复杂度，这里先举一个简单的例子。如图 1.8 所示，有 10 个数据点（图中空心圆所示），目标是拟合一条曲线C（图中点划线所示），使其对新的测试数据（此处未给出）也能表现良好。可以看到，最简单就是拟合一根直线（1 次多项式），使其从这些点的中间穿过，如图 1.8（a）所示。显然，这时候模型过于简单，处于欠拟合的状态。因此需要进一步增加模型的复杂度，如图 1.8（b）所示，曲线（5 次多项式）比较好的反映了数据点的整体趋势，可以认为复杂度合适，处于最佳拟合状态。如果进一步增加复杂度，如图 1.8（c）所示，曲线（9 次多项式）穿过了所有 10 个数据点，也就是说训练误差为 0。但是，一般而言，这种情况下模型过于复杂，处于过拟合状态。实际上，这个例子就是典型的回归问题，本书第 3 章将会谈到。



（a） （b） （c）

图 1.8 模型复杂度

既然谈到了回归，聪明的读者一定会问一个问题：前面只谈到了对于分类问题的评估标准，那么对于回归问题应该如何评估呢？图 1.7 可以给我们一些启发。对于图 1.7 中的某个数据点(x, y)，设拟合曲线对于x给出的值为$\hat{y}$，可以自然的用$|y-\hat{y}|$来度量拟合误差，称之为“绝对误差”。每个数据点都这样计算，把这些绝对误差加起来，再除以总的数据点数，就得到了常用的“平均绝对误差（MAE）”，如式（1.5）所示。如果考虑到损失函数一般要求可导，也可以对平均绝对误差稍做修改，定义“均方误差（MSE）”，以保持与损失函数的一致性，这个损失函数就是用得最多的“均方损失”。如式（1.6）所示。还有一系列衍生出来的回归度量方式，比如均方根误差（$RMSE$）等（习题 1.5）。

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{1.5}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{1.6}$$

1.3 机器学习的历史与现状

“机器学习”这个词是由 Arthur Samuel 于 1950 年代发明的，与“人工智能”的发展息息相关。本节简要地梳理一下机器学习的历史与现状，大致划分为感知机、神经网络、支持向量机、深度学习、大模型几个阶段。这样划分的基本考量是：第一，不求完备，但求便于读者形成一个比较简洁、清晰的发展脉络；第二，有所偏重，紧密结合当今的最新发展，由此回溯。

感知机由 Rosenblatt 于 1950 年代提出，基于 1943 年 McCulloc 和 Pitts 提出的“M-P 神

经元模型”。它只有输入和输出两层，可以从数据中进行自适应的端对端学习。虽然由于缺乏隐层和非线性变换函数（激活函数）而不能解决“非线性可分”问题，但是其历史意义是重大的、奠基性的。如今大行其道的端对端学习方式、深度学习、大模型都应该追溯到这里。

在感知机的基础上引入隐藏层（一个或多个）和激活函数，就形成了经典的“前馈神经网络”。而其学习算法则依赖于 1980 年代由 Rumelhart 等人重新发明的误差反向传播算法（BP 算法）。限于当时数据和计算能力的缺乏，对前馈神经网络（也包括其衍生出来的用于序列数据的循环神经网络等）无法进行有效的训练，从而无法有效解决实际问题，因此发展严重受阻。大概唯一的例外是 1990 年代由 Yan Lecun 发展并应用到手写数字识别上的卷积神经网络（CNN），这个网络源于日本科学家福岛邦彦提出的神经认知机。

统计学习理论的提出和支持向量机（SVM）的空前繁荣是 1990 年代的主旋律。SVM 以其严格的理论支撑、可靠的全局最优求解、避免过拟合等优点，被广泛研究和应用，更是发展出“核方法”，影响了众多的机器学习模型。

2006 年 Hinton 提出“深度学习”的概念，并带领学生在 2012 年的 ImageNet 图像识别竞赛中一骑绝尘、以巨大优势胜出。这个网络就是大名鼎鼎的 AlexNet。卷积神经网络、随机丢弃（Dropout）、GPU 并行计算等重要技术手段是其致胜的法宝。至此，神经网络所依赖的三个要素（算料、算法、算力）都已具备，接下来的十年就是“神经网络与深度学习”对各个技术领域（文本、语音、图像等）的彻底革命，一路狂飙，高歌猛进。其影响早已不限于技术领域，而是扩展到科学和社会的方方面面，有可能形成新的科学范式，而且被普遍认为将引领第四次工业革命，从而深刻影响人类社会。

“深度学习”的关键在于“深度”，而传统的神经网络受限于“梯度不稳定问题（表现为梯度消失或梯度爆炸等）”，网络的深度一旦加深，就很难通过误差反向传播算法成功训练出来。尽管人们从多方面（比如激活函数的选择、损失函数的选择等）尝试解决这个问题，但真正从根本上解决这个问题的是微软亚洲研究院何恺明等人提出的残差网络（ResNet），这种新型的网络结构可以使神经网络的深度达到数百层甚至上千层。事实上，残差网络已经成为了现代神经网络（包括近年来影响巨大的 Transformer）的标配，其影响具有基础性，也正由于此而获得 2023 年度未来科学大奖的数学与计算机科学奖，实至名归。

最近的大模型快速发展和巨大影响，则源于 2017 年谷歌（Google）公司提出的 Transformer 这种新型神经网络。基于此，谷歌和 OpenAI 分别发展出 Bert 和 GPT 两条主要技术路线。2022 年底，OpenAI 发布 ChatGPT，一时席卷互联网，成为有史以来增长最快的互联网应用。ChatGPT 基于 GPT 基础模型，综合运用了自监督、有监督、环境监督和强化学习等学习范式，具有高达 1750 亿的模型参数，其对互联网数据和 GPU 算力的需求也是空前的。ChatGPT 在自然语言方面所表现出来的前所未有的“理解和思考能力”让人们感到惊叹，因为无法清楚的解释，而被粗略的称为“智能涌现”（海量神经元之间的大规模相互作用导致智能的涌现）。人们普遍认为这是通用人工智能（AGI）的第一缕曙光。

当前，大模型正在席卷各行各业，专门化、多模态和内容生成（AIGC）也在进一步助推这个过程。比如，北京大学的一个研究团队发布了法律大模型，这是专门化大模型深入改变垂直应用领域的典型案例。

尽管“神经网络与深度学习”近十年来一路高歌猛进，然而对其质疑的声音也一直不绝于耳。“可解释性”就是被关注最多的一个问题，比如神经网络的每一个神经元究竟学到的是什么？“不可解释”意味着“不能被信任”或者“不能被充分信任”，这是“神经网络与深度学习”所面临的一个尴尬处境。以当前如火如荼的大语言模型为例，它会一本正经的胡说八道，编造不存在的事实等。尽管人们一直在尝试解决这些问题，但目前来看进展很有限，尚不能从根本上解决问题。

既然谈到可解释性，本书介绍的机器学习模型中，k-近邻、决策树、对率回归、基本线

性回归、朴素贝叶斯这些都是可解释的典型例子。而源于线性模型的 SVM、由对率回归单元堆叠而成的全连接神经网络都有其不可解释的一面。虽然如此，本书仍将深入其原理、详细剖析，从而充分把握其可解释的一面。从统计学习理论的角度来看，我们追求的是“概率近似正确（PAC）”，因而随机性和不确定性就是必须接受的现实，这或许也能从一个侧面帮助我们理解“不可解释性”。

在机器学习的发展过程中，人们也总结出一些带有哲学意味的“定律”。一个是著名的“没有免费的午餐定律”（NFL）。说的是，一个实用的模型或算法必有其针对性（称为归纳偏好）；而一个面面俱到的模型或算法则不可能具有实用性、因而只能存在于理论中。这个定律时时刻刻在提醒着我们，模型和算法是针对具体问题的，脱离具体问题泛泛而谈是没有意义的。另一个被广泛引用的定律是：奥卡姆剃刀。说的是，针对同一个任务，简单的机器学习模型更好。“简单”如何理解非常关键。笔者认为，这里的“简单”不能纵向地说（比如线性回归比大模型更好），而应该横向地说，即相同效果下简单的模型更好。因为通过前面的介绍，我们知道复杂的模型往往与过拟合相伴。图 1.7 给出了一个很好的例子，如果二阶多项式模型和三阶多项式模型在测试集上具有相同的性能表现，我们则更愿意选择简单一些的二阶多项式模型。

内容方面，本书根据教学的需要进行了内容的取舍，不包含集成学习，也不包含机器学习所涉及的法律、伦理等方面的问题，但包含了近年来发展迅速的自监督学习和环境监督学习。为了完整性，这里简单说一下集成学习。作为一种重要的“元学习”方式（“元”是超越的意思），集成学习的核心思想是“三个臭裨将抵一个诸葛亮”。即学习一组“弱学习器”，这些学习器“好而不同”，将他们集成起来（比如投票）得到性能更好的“强学习器”。典型的集成学习方法分为“串行化”和“并行化”两大类，前者的代表是 Boosting，后者的代表是 Bagging 和随机森林。至于机器学习所涉及的法律、伦理等方面的问题，首先我们要认识到技术是把“双刃剑”，技术应该“以人为本”，造福人类，而不是相反。因而，在应用机器学习的同时这些问题（比如人与机器的关系）也必须得到同步的妥善解决。在大模型快速发展的当下，这些问题尤其突出，必须引起高度重视。

当前，机器学习（特别是深度学习）已经成为了人工智能的主流方法，应用非常广泛，极大的方便了我们的日常生活。比如，翻译软件、语音识别软件、聊天机器人、智能垃圾邮件过滤器、可靠的网络搜索引擎、智能下棋程序等。再比如，在医疗领域，深度学习模型检测皮肤癌的准确率与专业医生的检测结果已经很相近。

这里，重点谈一下“自动驾驶”这个社会关注度高、影响面较广的重要应用。这里谈的“自动驾驶”特指在道路上行驶的机动车的驾驶。国际汽车工程师协会将机动车自动驾驶分为 L0~L5 共 6 级。其中，L0 级是无自动化，L1 级是辅助驾驶，L2 级是部分自动化，L3 级是有条件自动化，L4 级是高度自动化，而 L5 级就是最终的完全自动化（任何场景下都不再需要人的干预）。L1 级的典型代表是“定速巡航”，基本上已经成为标配，车辆并不需要感知周围环境的能力。L2 级的典型代表是“自适应巡航”，这在大多数中高端智能车上也已经成为标配，车辆已经具备了对周围环境的感知能力，比如与前车的距离、道路标线等，因此可以自动控制速度、自动转向等。L3 以上的级别目前并不多见，主要还限于特定道路和特定场景。可以看到，从 L2 级开始，车辆已经具备了对周围环境的感知能力，机器学习（特别是深度学习）在其中扮演着核心角色，目标是建立一个车辆周围环境的 3D 模型，这就涉及到从多模态数据（图像、视频、测距、语音等）中进行学习。端对端的深度学习已经成为这一应用领域的主流技术方案。而当下快速发展的大模型也必将会进一步优化和提升这一技术方案。当然，由于深度神经网络的“黑盒”特性，对端对端自动驾驶方案的质疑声也一直不绝于耳，值得引起充分重视。毕竟人命关天的事，无法给出清楚、合理的解释，是无论如何也站不住脚的。

小故事：机器学习的由来

1956 年 2 月 24 日，来自 IBM 的科学家 Arthur Samuel 在 IBM 701 计算机上，通过电视节目向公众展示了他的跳棋程序。这是公认的第一个 AI 程序，也是 AI 的第一次公开展示。这个电视节目使公众第一次认识到，电脑不仅可以用来作复杂的数学计算，也能用来玩游戏！人们首次了解到：计算机的确可以具有“智能”！

其实，早在 1949 年，Samuel 就有了开发跳棋程序的设想。因为跳棋相对简单，而下跳棋的策略又具有一定的思考深度。从 1952 年首次为 IBM 701 编写跳棋程序，到 1956 年 2 月 24 日的公众展示，在这个研发跳棋程序的过程中，Samuel 首次提出了“**机器学习**”的概念，并将其定义为“不显式编程地赋予计算机能力的研究领域”。

Samuel 为他的跳棋设计的学习方法，叫做“时间差分学习”方法。从今天机器学习的分类来看，是属于强化学习。学习过程中，塞缪尔跳棋程序会从随机位置开始，自我对战多局。每一步，程序都会选择能够最大化获胜机会的走步，并根据当前状态的价值函数进行决策。随着游戏的进行，该程序会使用一个公式来更新状态价值函数，这个更新被称为时间差分，因为它测量了当前状态的价值估计和下一个状态的价值估计之间的差异。通过反复进行这个过程，并不断更新状态的价值函数，程序逐渐改善了其下棋能力。

这是人工智能和机器学习领域的一项重大成就。为强化学习领域带来了重要的突破，得到了极其广泛的应用。也对现代机器学习产生了深远的影响。

启迪：

（1）敢于首创

（2）理论和实践的紧密结合

（3）做科研就是做人

习题 1

1.1 试比较有监督和无监督学习，并举出生活中的一些例子加以说明。

1.2 用生活中的例子说明强化学习中的“总奖励”可能即时获得，也可能延后获得。

1.3 在 1.2.3 节中，我们都是针对一般的多分类而谈的，那么对于二分类这种特殊又常用的情

况，有什么相同之处和不同之处呢？

1.4 对于二分类，除了 P-R 评估标准，与其紧密相关且比较类似的 ROC 也比较常用。请查阅相关文献资料，比较两者的异同。

1.5 除了 1.2.3 节中谈到的，回归问题还有哪些评估标准呢？这些评估标准又会如何影响损失函数的设计呢？

1.6 谈谈你对自动驾驶的认识。

1.7 在图 1.2 给出的线性回归例子中，为何不采用样本点到回归直线的垂直距离呢？