

第 1 章

人工智能和机器学习基础

近年来，随着人工智能技术的飞速发展，机器学习和深度学习技术已经摆在了人们的面前，一时间成为程序员们的学习热点。在本章的内容中，将详细介绍人工智能的基础知识，讲解和人工智能、机器学习、深度学习相关的概念，为读者步入本书后面知识的学习打下基础。

1.1 人工智能的基础知识

本节将简要介绍人工智能技术的基本知识。



扫码观看本节视频讲解

1.1.1 人工智能介绍

自从机器诞生以来，聪明的人类就开始试图让机器具有智能，也就是人工智能。人工智能是一门极富挑战性的学科，从事这项工作的人必须懂得计算机知识、心理学和哲学。人工智能涉及不同的领域，如机器学习、计算机视觉等，总的说来，人工智能研究的一个主要目标是使机器能够胜任一些通常需要人类智能才能完成的复杂工作。

人工智能不是一个非常庞大的概念，单从字面上理解，应该理解为人类创造的智能。那么什么是智能呢？如果人类创造了一个机器人，这个机器人能有像人类一样甚至超过人类的推理、知识、学习、感知处理等这些能力，那么就可以将这个机器人称之为一个有智能的物体，也就是人工智能。

现在通常将人工智能分为弱人工智能和强人工智能。我们看到电影里的一些人工智能大部分都是强人工智能，它们能像人类一样思考如何处理问题，甚至能在一定程度上做出比人类更好的决定，它们能自适应周围的环境，解决一些程序中没有遇到的突发事件。但是在目前的现实世界中，大部分人工智能只是实现了弱人工智能，只能够让机器具备观察和感知的能力，在经过一定的训练后能计算一些人类不能计算的事情，但是它并没有自适应能力，也就是说，它不会处理突发的情况，只能处理程序中已经写好的、已经预测到的事情。

1.1.2 人工智能的发展历程

1950年，一位名叫马文·明斯基（人工智能之父）的大四学生与他的同学邓恩·埃德蒙一起，建造了世界上第一台神经网络计算机。同样是在1950年，被称为“计算机之父”的阿兰·图灵提出了一个举世瞩目的想法：图灵测试。按照图灵的设想：如果一台机器能够与人类开展对话而不能被辨别出机器身份，那么这台机器就具有智能。而就在这一年，图灵还大胆预言了真正具备智能的机器的可行性。

20世纪70年代，人工智能步入了一段艰难险阻的岁月。对于人工智能方面的研究，由于科研人员对于难度估量过低和缺乏经费的原因，导致与美国国防高级研究计划署的合作计划失败，社会舆论的压力也开始慢慢压向人工智能这边，导致很多研究经费被转移到其他项目上，这也让大家对人工智能的前景比较担忧。

人工智能产业面临衰落，但科技并不会因外界因素而停止发展，直至20世纪80年代初期，人工智能产业开始崛起。从20世纪90年代中期开始，随着AI技术尤其是神经网络技术的逐步发展，以及人们对AI开始抱有客观理性的认知，人工智能技术开始进入平稳发展时期。1997年5月11日，IBM的计算机系统“深蓝”战胜了国际象棋世界冠军卡斯帕罗夫，又一次在公众领域引发了现象级的AI话题讨论。这是人工智能发展的一个重要里程碑。

2006年，杰弗里·辛顿在神经网络的深度学习领域取得突破，人类又一次看到机器赶超人类的希望，

这也是标志性的技术进步。紧接着谷歌（Google）、微软、百度等互联网巨头，还有众多的初创科技公司，纷纷加入人工智能产品的战场，掀起又一轮的智能化狂潮。

2016年，谷歌公司的AlphaGo战胜韩国棋手李世石，再度引发AI热潮。

1.1.3 人工智能的两个重要发展阶段

1. 推理期

20世纪50年代，人工智能的发展经历了“推理期”，即通过赋予机器逻辑推理能力使机器获得智能。当时的AI程序能够证明一些著名的数学定理，但由于机器缺乏知识，远不能实现真正的智能。

2. 知识期

20世纪70年代，人工智能的发展进入“知识期”，即将人类的知识总结出来教给机器，使机器获得智能。在这一时期，大量的专家系统问世，在很多领域取得大量成果，但由于人类知识量巨大，故出现“知识工程瓶颈”。

1.1.4 和人工智能相关的几个重要概念

1. 监督学习

监督学习的任务是学习一个模型，这个模型可以处理任意的一个输入，并且针对每个输入都可以映射输出一个预测结果。这里的模型就相当于我们数学中的一个函数，输入就相当于数学中的 X ，而预测的结果就相当于数学中的 Y 。对于每一个 X ，我们都可以通过一个映射函数映射出一个结果。

2. 非监督学习

非监督学习是指直接对没有标记的训练数据进行建模学习，注意在这里的数据是没有标记的数据，与监督学习最基本的区别是建模的数据一个有标签，一个没有标签。例如，聚类（将物理或抽象对象的集合分成由类似的对象组成的多个类的过程被称为聚类）是一种典型的非监督学习，而分类是一种典型的监督学习。

3. 半监督学习

当我们拥有标记的数据很少，未被标记的数据很多，但是人工标注又比较费时的时候，可以根据一些条件（查询算法）查询（Query）一些数据，让专家进行标记。这是半监督学习与其他算法的本质区别。所以说对主动学习的研究主要是设计一种框架模型，运用新的查询算法查询需要专家来确认标注的数据。最后用查询到的样本训练分类模型提高模型的精确度。

4. 主动学习

当使用一些传统的监督学习方法做分类处理的时候，通常是训练样本的规模越大，分类的效果就越好。但是在现实生活的很多场景中，标记样本的获取是比较困难的，这需要领域内的专家进行人工标注，所花费的时间成本和经济成本都是很大的。而且，如果训练样本的规模过于庞大，训练的时间花费也会比较多。那么问题来了：有没有一种有效办法，能够使用较少的训练样本来获得性能较好的分类器呢？主动学习（Active Learning）为我们提供了这种可能。主动学习通过一定的算法查询最有用的未标记样本，并交由专家进行标记，然后用查询到的样本训练分类模型来提高模型的精确度。

在人类的学习过程中，通常利用已有的经验来学习新的知识，又依靠获得的知识来总结和积累经验，经验与知识不断交互。同样，机器学习模拟人类学习的过程，利用已有的知识训练出模型去获取新的知识，并通过不断积累的信息去修正模型，以得到更加准确、有用的新模型。不同于被动学习被动地接受知识，主动学习能够选择性地获取知识。

1.2 机器学习的基础知识

在人工智能的两个发展阶段中，无论是“推理期”还是“知识期”，都会存在如下两个缺点。

(1) 机器都是按照人类设定的规则和总结的知识运作，永远无法超越其创造者：人类。

(2) 人力成本太高，需要专业人才进行具体实现。

基于上述两个缺点，人工智能技术的发展出现了一个瓶颈期。为了突破这个瓶颈期，一些权威学者就想到，如果机器能够自我学习的话，问题不就迎刃而解了吗？此时机器学习（Machine Learning, ML）技术便应运而生，人工智能开始进入“机器学习”时代。在本节的内容中，将简要介绍机器学习的基本知识。



扫码观看本节视频讲解

1.2.1 机器学习介绍

机器学习是一门多领域交叉学科，涉及概率论、统计学、逼近论、凸分析、算法复杂度理论等多门学科。机器学习专门研究计算机怎样模拟或实现人类的学习行为，以获取新的知识或技能，重新组织已有的知识结构使之不断改善自身的性能。

机器学习是一类算法的总称，这些算法企图从大量历史数据中挖掘出其中隐含的规律，并用于预测或者分类，更具体地说，机器学习可以看作是寻找一个函数，输入是样本数据，输出是期望的结果，只是这个函数过于复杂，以至于不太方便形式化表达。需要注意的是，机器学习的目标是使学到的函数很好地适用于“新样本”，而不仅仅是在训练样本上表现很好。学到的函数适用于新样本的能力，称为泛化（Generalization）能力。

机器学习有一个显著的特点，也是机器学习最基本的做法，就是使用一个算法从大量的数据中解析并得到有用的信息，并从中学学习，然后对之后真实世界中会发生的事情进行预测或作出判断。机器学习需要海量的数据来进行训练，并从这些数据中得到有用的信息，然后反馈到真实世界的用户中。

我们可以用一个简单的例子来说明机器学习，假设在淘宝或京东购物的时候，天猫和京东会向我们推送商品信息，这些推荐的商品往往是我们很感兴趣的东西，这个过程是通过机器学习完成的。其实这些推送商品是京东和天猫根据我们以前的购物订单和经常浏览的商品记录而得出的结论。

1.2.2 机器学习的三个发展阶段

机器学习是人工智能的核心，是使计算机具有智能的根本途径，其应用遍及人工智能的各个领域，

它主要使用归纳、综合，而不是演绎。机器学习的发展分为如下三个阶段。

- 20世纪80年代，连接主义较为流行，代表工作有感知机（Perceptron）和神经网络（Neural Network）。
- 20世纪90年代，统计学习方法开始占据主流舞台，代表性方法有支持向量机（Support Vector Machine）。
- 21世纪初，深度神经网络技术被提出，连接主义卷土重来，随着数据量和计算能力的不断提升，以深度学习（Deep Learning）为基础的诸多AI应用逐渐成熟。

1.2.3 机器学习的分类

根据不同的划分角度，可以将机器学习划分为多种不同的类型。

1. 按任务类型划分

机器学习模型可以分为回归模型、分类模型和结构化学习模型，具体说明如下。

- 回归模型：又叫预测模型，输出是一个不能枚举的数值。
- 分类模型：又分为二分类模型和多分类模型，常见的二分类问题有垃圾邮件过滤，常见的多分类问题有文档自动归类。
- 结构化学习模型：此类型的输出不再是一个固定长度的值，如图片语义分析，其输出是图片的文字描述。

2. 从方法的角度划分

机器学习可以分为线性模型和非线性模型，具体说明如下。

- 线性模型：虽然比较简单，但是其作用不可忽视，线性模型是非线性模型的基础，很多非线性模型都是在线性模型的基础上变换而来的。
- 非线性模型：又可以分为传统机器学习模型（如SVM、KNN、决策树等）和深度学习模型。

3. 按照学习理论划分

机器学习模型可以分为有监督学习、半监督学习、无监督学习、迁移学习和强化学习，具体说明如下。

- 当训练样本带有标签时是有监督学习。
- 训练样本部分有标签，部分无标签时是半监督学习。
- 训练样本全部无标签时是无监督学习。
- 迁移学习就是把已经训练好的模型参数迁移到新的模型上以帮助新模型训练。
- 强化学习是一个学习最优策略（Policy），可以让本体（Agent）在特定环境（Environment）中，根据当前状态（State）做出行动（Action），从而获得最大回报（Reward）。强化学习和有监督学习最大的不同是，每次的决定没有对与错，而是希望获得最多的累计奖励。

1.2.4 深度学习和机器学习的对比

前面介绍的机器学习是一种实现人工智能的方法，深度学习是一种实现机器学习的技术。深度学习本来并不是一种独立的学习方法，其本身也会用到有监督和无监督的学习方法来训练深度神经网络。但

由于近几年该领域发展迅猛，一些特有的学习手段相继被提出（如残差网络），因此越来越多的人将其单独看作一种学习的方法。

假设我们需要识别某个照片是狗还是猫，如果是传统机器学习的方法，会首先定义一些特征，如没有胡须，耳朵、鼻子、嘴巴的模样等。总之，我们首先要确定相应的“面部特征”作为我们的机器学习的特征，以此来对我们的对象进行分类识别。深度学习的方法则更进一步，它会自动地找出这个分类问题所需要的重要特征，而传统机器学习则需要我们人工地给出特征。那么，深度学习是如何做到这一点的呢？继续以猫、狗识别的例子进行说明，按照以下步骤操作。

- (1) 确定出有哪些边和角跟识别出猫狗关系最大。
- (2) 根据上一步找出的很多小元素（边、角等）构建层级网络，找出它们之间的各种组合。
- (3) 在构建层级网络之后，就可以确定哪些组合可以识别出猫和狗。

注意 其实深度学习并不是一个独立的算法，在训练神经网络的时候也通常会用到监督学习和无监督学习。但是由于一些独特的学习方法被提出，把它看成是单独的一种学习的算法也没什么问题。深度学习可以大致理解成包含多个隐含层的神经网络结构，深度学习中的“深”字指的就是隐藏层的深度。

在机器学习方法中，几乎所有的特征都需要通过行业专家来确定，然后人工就特征进行编码，而深度学习算法会自己从数据中学习特征。这也是深度学习十分引人注目的一点，毕竟特征工程是一项十分烦琐、耗费很多人力物力的工作，深度学习的出现大大减少了发现特征的成本。

在解决问题时，传统机器学习算法通常先把问题分成几块，一个个地解决好之后，再重新组合起来。但是深度学习则是一次性地、端到端地解决。假如存在一个任务：识别出在某图片中有哪些物体，并找出它们的位置。

传统机器学习的做法是把问题分为两步：发现物体和识别物体。首先，我们有几个物体边缘的盒型检测算法，把所有可能的物体都框出来。然后，再使用物体识别算法，识别出这些物体中分别是什么。图 1-1 是一个机器学习识别的例子。

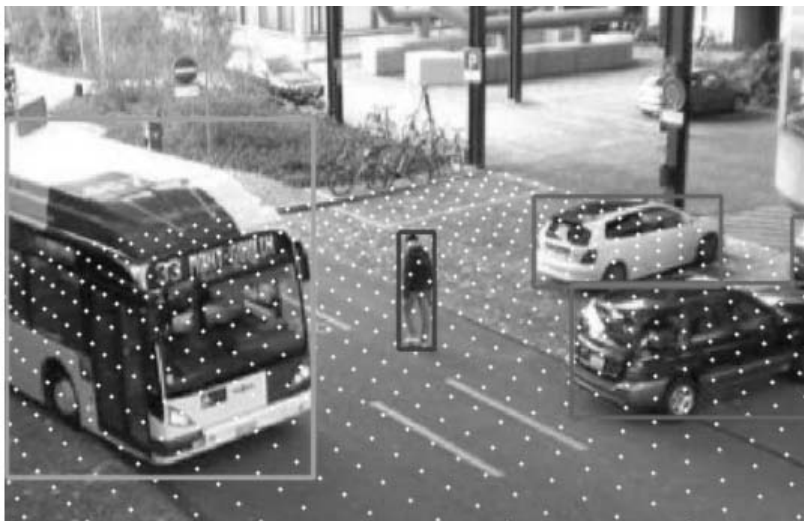


图 1-1 机器学习的识别

但是深度学习不同，它会直接在图片中把对应的物体识别出来，同时还能标明对应物体的名字。这样就可以做到实时的物体识别，例如 YOLO net 可以在视频中实时识别物体，图 1-2 是 YOLO 在视频中实现深度学习识别的例子。



图 1-2 深度学习的识别

注意 人工智能、机器学习、深度学习三者的关系

机器学习是实现人工智能的方法，深度学习是一种机器学习算法，也是一种实现机器学习的技术和学习方法。

1.3 人工智能的研究领域和应用场景

在本节的内容中，将对人工智能的研究领域和应用场景进行讲解，为读者步入本书后面知识的学习打下基础。



扫码观看本节视频讲解

1.3.1 人工智能的研究领域

人工智能的研究领域主要有五层，具体如图 1-3 所示。

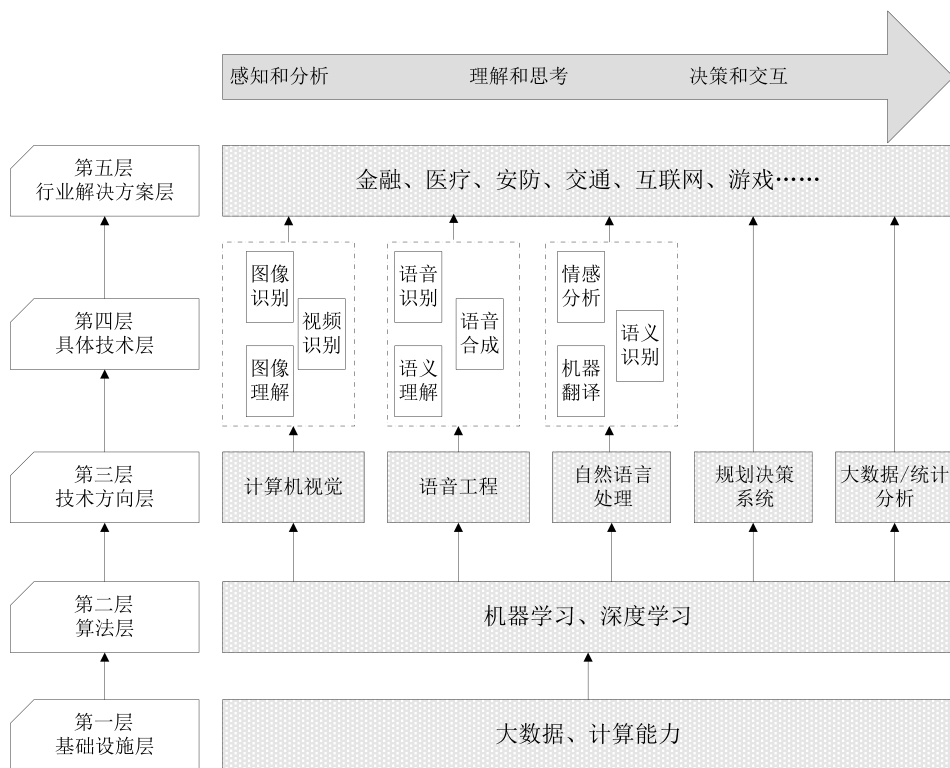


图 1-3 人工智能的研究领域

在图 1-3 所示的分层中，从下往上的具体说明如下。

第一层：基础设施层，包含大数据和计算能力（硬件配置）两部分，数据越多，人工智能的能力越强。

第二层：算法层，例如卷积神经网络、LSTM 序列学习、Q-Learning 和深度学习等都是机器学习的算法。

第三层：技术方向层，例如计算机视觉、语音工程和自然语言处理等。另外，还有规划决策系统，例如 Reinforcement Learning（增强学习）和类似于大数据分析的统计系统，这些都能在机器学习算法上产生。

第四层：具体技术层，例如图像识别、语音识别、语义理解、视频识别、机器翻译等。

第五层：行业解决方案层，例如人工智能在金融、医疗、互联网、安防、交通和游戏等领域的应用。

1.3.2 人工智能的应用场景

1. 计算机视觉

在 2000 年左右，人们通过机器学习用人工特征实现了较好的计算机视觉系统，如车牌识别、安防、人脸识别等技术。而深度学习则逐渐运用机器代替人工来学习特征，扩大了其应用场景，例如，无人汽车和电商服务等领域。

2. 语音技术

在 2010 年后，深度学习的广泛应用使语音识别的准确率大幅提升，像 Siri、Voice Search 和 Echo 等，可以实现不同语言间的交流，录一段语音，随之将其翻译为另一种文字。再如智能助手，我们可以对手机说一段话，它能帮助你完成一些任务。与图像识别相比，自然语言更难、更复杂，不仅需要认知，而且还需要理解。

3. 自然语言处理

目前人工智能一个比较重大的突破是机器翻译，这大大提高了原来的机器翻译水平。举个例子，谷歌的 Translation 系统，是人工智能的一个标杆性事件。在 2010 年左右，IBM 的 Watson 系统在一档综艺节目上，和人类冠军进行自然语言的问答并获胜，代表了计算机能力的显著提高。

4. 决策系统

决策系统的发展随着棋类问题的解决而不断提升，从 20 世纪 80 年代西洋跳棋开始，到 90 年代的国际象棋对弈，机器的胜利标志着科技的进步，决策系统可以在自动化、量化投资等系统上广泛应用。

5. 大数据应用

人工智能可以通过你之前看到的文章，理解你所喜欢的内容而进行更精准的推荐；也可以分析各个股票的行情，进行量化交易；还可以分析所有客户的一些喜好而进行精准的营销等。机器能够通过一系列的数据进行判别，找出最适合的一些策略后反馈给我们。

1.3.3 人工智能的未来目标

按照具体应用领域进行分析，人工智能的未来目标如下。

1. 计算机视觉领域

未来的人工智能应更加注重效果的优化，加强计算机视觉在不同场景、问题上的应用。

2. 语音识别领域

当前的语音识别虽然在特定的场景（安静的环境）下，已经能够得到和人类相似的水平。但在噪声情景下仍有挑战，如原场识别、口语、方言等长尾内容。未来需增强计算能力、提高数据量和提升算法等来解决这个问题。

3. 自然语言处理领域

机器的优势在于拥有更多的记忆能力，但欠缺语义理解能力，包括对口语不规范的用语识别和认知等。人说话时，是与物理事件相联系的，比如一个人说电脑，人知道这个电脑意味着什么，或者它能够干什么，而在自然语言里，它仅仅将“电脑”作为一个孤立的词，不会产生类似的联想，自然语言的联想只是通过在文本上和其他所共现的一些词的联想，并不是物理事件里的联想。所以要真正解决自然语言的问题，将来需要建立从文本到物理事件的一个映射，但目前仍没有很好的解决方法。因此，这是未来着重考虑的一个研究方向。

4. 决策系统领域

在决策系统领域存在两个问题：一是不通用，即学习知识的不可迁移性，如用一个方法学了下围棋，不能直接将该方法转移到下象棋中；二是大量模拟数据。所以它有两个目标，一个是算法的提升，如何解决数据稀少或怎么自动产生模拟数据的问题，另一个是自适应能力，当数据发生变化的时候，它能够

去适应变化，而不是能力有所下降。

注意 上述未来目标的实现时间不确定，也许是下一个3年、5年或10年才能解决。

1.4 学习机器学习的步骤

对于开发人员来说，学习机器学习主要是学习一类算法的过程，这些算法的目的是从大量历史数据中挖掘出其中隐含的规律，并用于预测或者分类。更具体一点说，可以将机器学习看作是寻找一个合适函数的过程，我们对这个函数的描述是：

我们输入的是样本数据，通过这个函数输出期望的结果。

只是这个函数的事项过于复杂，不太容易形式化地将这个函数的具体语法格式表示出来。机器学习要想学好一个合适的函数，主要通过以下4个步骤来实现。

(1) 选择一个合适的模型，这通常需要依据实际问题而定，针对不同的问题和任务需要选取恰当的模型（模型就是一组函数的集合）。

(2) 判断一个函数的好坏，这需要确定一个衡量标准，也就是我们通常所说的损失函数（Loss Function），损失函数的确定也需要依据具体问题而定，如回归问题一般采用欧式距离，分类问题一般采用交叉熵代价函数。

(3) 找出“最好”的函数，这一步最大的难点是如何从众多函数中最快地找出“最好”的那一个。

(4) 在得到并学习完“最好”的函数后，需要在新样本上进行测试，只有在新样本上表现很好，才能算作是一个“好”的函数。



扫码观看本节视频讲解

1.5 使用 Python 学习人工智能开发

Python 语言和人工智能的发展是相辅相成的。近几年人工智能的快速发展，也促进了 Python 语言的飞速发展，使其成为世界上使用最多的开发语言之一。



扫码观看本节视频讲解

1.5.1 Python 在人工智能方面的优势

在开发人工智能程序方面，Python 语言拥有如下优势。

1. 更加人性化的设计

Python 的设计更加人性化，具有快速、坚固、可移植、可扩展等特点，这些特点十分适合人工智能。并且内置了很多强大的库，可以轻松实现更强大的功能。

2. 拥有很多 AI（人工智能）库，包括机器学习库

Python 可以使用很多已经存在的人工智能库，也同样拥有很多可用的机器学习库，这些库的功能非常强大，可以提高开发效率。Python 语言可用的人工智能库的数量最多，这是其他语言所不能比拟的。

3. 强大的自然语言和文本处理库

Python 具有丰富而强大的自然语言处理库和文本处理库，能够将其他语言制作的各种模块很轻松地联结在一起，Python 编程对人工智能是一门非常有用的语言。可以说人工智能和 Python 是紧密相连的。

4. 可移植、可扩展

Python 语言的设计非常好，具有快速、坚固、可移植和可扩展等特点。很明显，这些特点对于人工智能应用来说都是非常重要的因素。

1.5.2 常用的 Python 库

在使用 Python 语言开发人工智能程序时，可以使用人工智能库快速实现我们需要的功能。这些库的功能十分强大，大大提高了开发效率。

1. 数据处理库

1) Numpy 库

Numpy 是构建科学计算代码集的最基础的库，提供了许多用 Python 进行 N 维数组和矩阵操作的功能。Numpy 库提供了 Numpy 数组类型的数学运算向量化，可以改善性能，从而加快执行速度。

2) Scipy 库

Scipy 包含致力于科学计算中常见问题的各个工具箱，其不同子模块实现不同的应用，例如插值、积分、优化、图像处理、统计、特殊函数等。因为 Scipy 的主要功能是建立在 Numpy 基础之上的，所以它使用了大量的 Numpy 数组结构。Scipy 库通过其特定的子模块提供高效的数学运算功能，如数值积分、优化等。

3) Pandas 库

Pandas 是一个简单直观的应用于“带标记的”和“关系型的”数据的 Python 库，可以快速地数据进行数据操作、聚合和可视化操作。

4) Matplotlib 库

Matplotlib 是一个可以实现数据可视化图表的库。与之功能相似的库是 seaborn，并且 seaborn 是建立在 Matplotlib 基础之上的。

2. 机器学习库

1) PyBrain 库

PyBrain 是一个灵活、简单而有效的针对机器学习任务的算法库，是模块化的 Python 机器学习库，并且提供了多种预定义好的环境来测试和比较算法。

2) PyML 库

PyML 库是一个用 Python 编写的双边框架，重点研究 SVM 和其他内核方法，支持 Linux 和 Mac OS。

3) Scikit-Learn 库

Scikit-Learn 库旨在提供简单而强大的解决方案，可以在不同的上下文中重用。机器学习作为科学和工程的一个多功能工具，Scikit-Learn 是 Python 的一个模块，集成了经典的机器学习的算法，这些算法是和 Python 科学包紧密联系在一起。

4) MDP-Toolkit 库

MDP-Toolkit 是一个 Python 数据处理的框架，可以很容易地在其基础上进行扩展。MDP-Toolkit 还收集了监管学习和非监管学习算法和其他数据处理单元，可以组合成数据处理序列或者更复杂的前馈网络结构。在 MDP-Toolkit 中提供的算法是在不断地增加和升级的，包括信号处理方法（主成分分析、独立成分分析、慢特征分析）、流形学习方法（局部线性嵌入）、集中分类、概率方法（因子分析、RBM）、数据预处理方法等。

5) Crab 库

Crab 是基于 Python 语言开发的推荐库，实现了 item 和 user 的协同过滤功能，可以快速开发出 Python 推荐系统。

6) TensorFlow 库

TensorFlow 是当今最流行的机器学习库，是科技巨头谷歌推出的一个开源库，是目前市场上占有率最高的机器学习库。本书将以 TensorFlow 库为主题，来详细讲解使用 TensorFlow 库的知识。

1.6 TensorFlow 基础

TensorFlow 可以帮助我们开发和训练机器学习模型。



扫码观看本节视频讲解

1.6.1 TensorFlow 介绍

TensorFlow 是一个端到端开源机器学习平台。它拥有一个全面而灵活的生态系统，其中包含各种工具、库和社区资源，可助力研究人员推动先进机器学习技术的发展，并使开发者能够轻松地构建和部署由机器学习提供支持的应用。

TensorFlow 由谷歌人工智能团队谷歌大脑（Google Brain）负责开发和维护，拥有包括 TensorFlow Hub、TensorFlow Lite、TensorFlow Research Cloud 在内的多个项目以及各类应用程序接口（application programming interface, API）。自 2015 年 11 月 9 日起，TensorFlow 依据 Apache 2.0 协议开放源代码。

在机器学习框架领域，PyTorch、TensorFlow 已分别成为目前学术界和工业界使用最广泛的两大实力玩家，而紧随其后的 Keras、MXNet 等框架也由于其自身的独特性受到开发者的喜爱。截至 2020 年 8 月，主流机器学习库在 Github 网站的活跃度如图 1-4 所示。由此可见，在众多机器学习库中，本书将要讲解的 TensorFlow 最受开发者的欢迎，是当之无愧的机器学习第一库。

	TensorFlow	Keras	MXNet	PyTorch
star	148k	49.4k	18.9k	41.3k
folk	82.5k	18.5k	6.7k	10.8k
contributors	2692	864	828	1540

图 1-4 主流机器学习库的活跃度

1.6.2 TensorFlow 的优势

TensorFlow 是当前最受开发者欢迎的机器学习库，之所以能有现在的地位，主要原因有如下两点。

(1) “背靠大树好乘凉”，谷歌几乎在所有应用程序中都使用 TensorFlow 来实现机器学习。得益于谷歌在深度学习领域的影响力和强大的推广能力，TensorFlow 一经推出关注度就居高不下。

(2) TensorFlow 本身设计宏大，不仅可以为深度学习提供强力支持，而且灵活的数值计算核心也能广泛应用于其他涉及大量数学运算的科学领域。

除了上述两点之外，TensorFlow 的主要优点如下：

- 支持Python、JavaScript、C++、Java和Go、C# 和Julia等多种编程语言；
- 灵活的架构支持多GPU、分布式训练，跨平台运行能力强；
- 自带TensorBoard组件，便于用户通过可视化技术实时监控观察训练过程；
- 官方文档非常详尽，可供开发者查询的资料众多；
- 开发者社区庞大，大量开发者活跃于此，可以共同学习，互相帮助，一起解决学习过程中的问题。

1.6.3 TensorFlow Lite 介绍

TensorFlow Lite 是一组工具，可帮助开发者在移动设备、嵌入式设备和 IoT 设备上运行 TensorFlow 模型。TensorFlow Lite 支持设备端机器学习推断，延迟较低，并且二进制文件很小。

TensorFlow Lite 允许开发者在多种设备上运行 TensorFlow 模型。TensorFlow 模型是一种数据结构，这种数据结构包含了在解决一个特定问题时，训练得到的机器学习网络的逻辑和知识。

在实际开发过程中，可以通过多种方式获得 TensorFlow 模型，从使用预训练模型（pre-trained models）到训练自己的模型。为了在 TensorFlow Lite 中使用模型，模型必须转换成一种特殊格式。

第 2 章

搭建开发环境

经过第 1 章内容的学习，已经了解了人工智能和机器学习的基本概念，并且对 TensorFlow 库有了一个大概的了解。在使用 TensorFlow 库之前，必须在电脑中安装 TensorFlow。在本章的内容中，将和大家一起来探讨学习搭建 TensorFlow 开发环境的知识，为读者步入本书后面知识的学习打下基础。

2.1 安装环境要求

在安装并使用 TensorFlow 库之前，需要确保自己所用电脑的硬件配置和软件环境满足要求，在本节的内容中，将详细讲解学习 TensorFlow 开发所需要的环境要求。



扫码观看本节视频讲解

2.1.1 硬件要求

1. GPU（图形处理器，显卡）

机器学习和深度学习对电脑硬件的要求比较高，现在主流的深度学习都是通过多显卡计算来提升系统的计算能力。因为机器学习的核心计算都需要依托 GPU 进行，所以硬件要求的核心是显卡（GPU）。建议在采购电脑时，GPU 越大越好，尽量使用 12GB 以上的 GPU。

2. 内存

内存的大小要根据 CPU 的配置来购买，建议使用 16GB 以上的内存。

3. 硬盘

建议 SSD（固态硬盘）和 HDD（机械硬盘）结合使用，SSD 用作系统盘，HDD 用作仓储，SSD 至少 250GB，机械硬盘至少 1TB。

4. CPU

对 CPU 的主频要求比较高，对核心数的要求并不高。但是为了提高效率，还是越高越好。

注意 上面只是列出了对主要硬件的建议，大家可以根据自己的预算进行采购。如果要编写的机器学习项目太大，对硬件要求极高，可以考虑使用云服务器，例如阿里云等。

2.1.2 软件要求

在安装 TensorFlow 库之前，必须在电脑中安装好 Python。不同版本的 Python，对应需要下载安装的 TensorFlow 的版本也不同。在下载 TensorFlow 时，一定要下载正确的版本。例如，如果电脑中安装的是 Python 3.8，并且是 64 位的 Windows 10 操作系统，则只能安装适应于 Python 3.8 的并同时适应于 64 位 Windows 系统的 TensorFlow。

2.2 安装 TensorFlow

准备好硬件环境和软件环境后，接下来开始正式安装 TensorFlow。在本节的内容中，将详细讲解常用的几种安装 TensorFlow 的方法。

2.2.1 使用 pip 安装 TensorFlow

安装 TensorFlow 最简单的方法是使用 pip 命令，在使用这种安装方式时，无须考虑当前所使用的 Python 版本和操作系统的版本，pip 会自动安装适合当前 Python 版本和操作系统版本的 TensorFlow。在安装 Python 后，会自动安装 pip。


(1) 在 Windows 系统中单击桌面左下角的  图标，在弹出的界面中找到“命令提示符”图标，鼠标右键单击，在弹出的快捷菜单中依次选择“更多”→“以管理员身份运行”命令，如图 2-1 所示。



图 2-1 选择“以管理员身份运行”命令

(2) 在弹出的“命令提示符”界面中输入如下命令即可安装 TensorFlow 库：

```
pip install TensorFlow
```

在输入上述 pip 安装命令后，会弹出下载并安装 TensorFlow 的界面，如图 2-2 所示。因为 TensorFlow 库的容量比较大，并且还需要安装相关的其他库，所以整个下载安装过程会比较慢，需要大家耐心等待，确保 TensorFlow 能够正确安装。

```
管理员: 命令提示符 - pip install TensorFlow
Microsoft Windows [版本 10.0.18363.1198]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\WINDOWS\system32>pip install TensorFlow
Collecting TensorFlow
  Downloading tensorflow-2.3.1-cp38-cp38-win_amd64.whl (342.5 MB)
    |                                     | 337 kB 38 kB/s eta 2:27:56
```

图 2-2 下载、安装 TensorFlow 界面

注意 使用 pip 命令安装的另外一大好处是，会自动安装适合的、当前最新版本的 TensorFlow。此处是在“计算机”中安装的是 Python 3.8，并且是 64 位的 Windows 10 操作系统。通过图 2-2 可知，这时适合电脑的最新版安装文件是 tensorflow-2.3.1-cp38-cp38-win_amd64.whl。这个安装文件的名称中，各个字段的含义如下。

- tensorflow-2.3.1：表示 TensorFlow 的版本号是 2.3.1。
- cp38：表示适用于 Python 3.8 版本。

- win_amd64: 表示适用于64位的Windows操作系统。

在使用前面介绍的 pip 方式下载安装 TensorFlow 时, 能够安装成功的一个关键因素是网速。如果网速过慢, 可以考虑在百度中搜索 TensorFlow 下载包。目前最新版本的安装文件是 tensorflow-2.3.1-cp38-cp38-win_amd64.whl, 因此可以在百度中搜索这个文件, 然后下载。下载完成后保存到本地硬盘中, 例如保存位置是 D:\tensorflow-2.3.1-cp38-cp38-win_amd64.whl, 那么在“命令提示符”界面中定位到 D 盘根目录, 然后运行如下命令就可以安装 TensorFlow, 具体安装过程如图 2-3 所示。

```
pip install tensorflow-2.3.1-cp38-cp38-win_amd64.whl
```



```


管理员: 命令提示符
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26.>=1.21.1 in c:\users\apple\appdata\local\programs\python\python38\lib\site-packages (from requests<3,>=2.21.0->tensorboard<3,>=2.3.0->TensorFlow) (1.25.9)
Requirement already satisfied: idna<3,>=2.5 in c:\users\apple\appdata\local\programs\python\python38\lib\site-packages (from requests<3,>=2.21.0->tensorboard<3,>=2.3.0->TensorFlow) (2.9)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in c:\users\apple\appdata\local\programs\python\python38\lib\site-packages (from google-auth<2,>=1.6.3->tensorboard<3,>=2.3.0->TensorFlow) (4.1.1)
Requirement already satisfied: pyasn1-modules<=0.2.1 in c:\users\apple\appdata\local\programs\python\python38\lib\site-packages (from google-auth<2,>=1.6.3->tensorboard<3,>=2.3.0->TensorFlow) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4, python_version >= "3.5" in c:\users\apple\appdata\local\programs\python\python38\lib\site-packages (from google-auth<2,>=1.6.3->tensorboard<3,>=2.3.0->TensorFlow) (4.6)
Requirement already satisfied: oauthlib<=3.0.0 in c:\users\apple\appdata\local\programs\python\python38\lib\site-packages (from requests-oauthlib<=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<3,>=2.3.0->TensorFlow) (3.1.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\apple\appdata\local\programs\python\python38\lib\site-packages (from pyasn1-modules<=0.2.1->google-auth<2,>=1.6.3->tensorboard<3,>=2.3.0->TensorFlow) (0.4.8)
Building wheels for collected packages: wrapt, termcolor
  Building wheel for wrapt (setup.py) ... done
  Created wheel for wrapt: filename=wrapt-1.12.1-py3-none-any.whl size=19558 sha256=9a77ae4bfd4fd234ca41bdc438cb8d71dc167a84e1ce7f58752f0b6eac3fcd
  Stored in directory: c:\users\apple\appdata\local\pip\cache\wheels\4\4\76\49\969a24d4cecb94b1c96a2d37dc4e45d96a3ec0e2e485cd11a
  Building wheel for termcolor (setup.py) ... done
  Created wheel for termcolor: filename=termcolor-1.1.0-py3-none-any.whl size=4835 sha256=9ae63efb612bde7307f05c57a2cb49e4a0335f791f1155039f9fbclacda4165b
  Stored in directory: c:\users\apple\appdata\local\pip\cache\wheels\76\d4\7e\16cb0b889c0f014a026d097fa782ea32bb4dlfe4b731bc3b7a
Successfully built wrapt termcolor
Installing collected packages: keras-preprocessing, opt-einsum, astunparse, tensorflow-estimator, h5py, gast, google-pasta, wrapt, termcolor, TensorFlow
Successfully installed TensorFlow-2.3.1 astunparse-1.6.3 gast-0.3.3 google-pasta-0.2.0 h5py-2.10.0 keras-preprocessing-1.1.2 opt-einsum-3.3.0 tensorflow-estimator-2.3.0 termcolor-1.1.0 wrapt-1.12.1

```

图 2-3 在 Windows 10 的“命令提示符”界面中安装 TensorFlow

2.2.2 使用 Anaconda 安装 TensorFlow

使用 Anaconda 安装 TensorFlow 的方法和上面介绍的 pip 方式相似, 具体流程如下。

(1) 在 Windows 系统中单击桌面左下角的  图标, 在弹出界面中找到 Anaconda Powershell Prompt 选项, 鼠标右键单击, 在弹出的快捷菜单中依次选择“更多”→“以管理员身份运行”命令, 如图 2-4 所示。

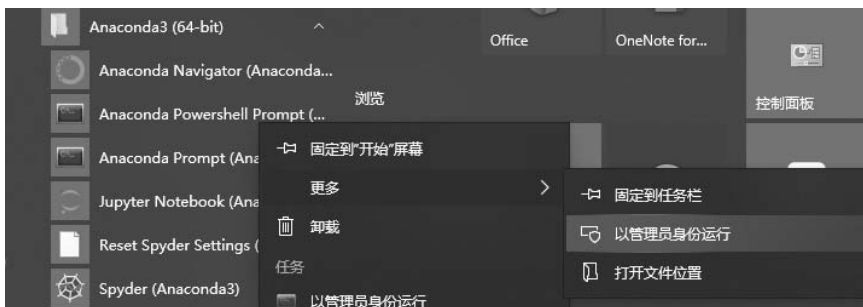


图 2-4 以管理员身份运行 Anaconda Powershell Prompt

(2) 在弹出的“命令提示符”界面中输入如下命令即可安装 TensorFlow 库：

```
pip install TensorFlow
```

在输入上述 pip 安装命令后，会弹出下载并安装 TensorFlow 的界面，安装成功后的界面效果如图 2-5 所示。

```

管理员: Anaconda Powershell Prompt (Anaconda3)
oard<3,>=2.3.0->TensorFlow) (1.7.0)
Requirement already satisfied: requests<3,>=2.21.0 in c:\programdata\anaconda3\lib\site-packages (from tensorboard<3,>=2.3.0->TensorFlow) (2.22.0)
Requirement already satisfied: markdown>=2.6.8 in c:\programdata\anaconda3\lib\site-packages (from tensorboard<3,>=2.3.0->TensorFlow) (3.2.2)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\programdata\anaconda3\lib\site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<3,>=2.3.0->TensorFlow) (1.3.0)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from google-auth<2,>=1.6.3->tensorboard<3,>=2.3.0->TensorFlow) (4.1.1)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\programdata\anaconda3\lib\site-packages (from google-auth<2,>=1.6.3->tensorboard<3,>=2.3.0->TensorFlow) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4; python_version >= "3.5" in c:\programdata\anaconda3\lib\site-packages (from google-auth<2,>=1.6.3->tensorboard<3,>=2.3.0->TensorFlow) (4.6)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<3,>=2.3.0->TensorFlow) (1.25.8)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<3,>=2.3.0->TensorFlow) (2019.11.28)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<3,>=2.3.0->TensorFlow) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<3,>=2.3.0->TensorFlow) (2.8)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in c:\programdata\anaconda3\lib\site-packages (from markdown>=2.6.8->tensorboard<3,>=2.3.0->TensorFlow) (1.5.0)
Requirement already satisfied: oauthlib>=3.0.0 in c:\programdata\anaconda3\lib\site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<3,>=2.3.0->TensorFlow) (3.1.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\programdata\anaconda3\lib\site-packages (from pyasn1-modules>=0.2.1->google-auth<2,>=1.6.3->tensorboard<3,>=2.3.0->TensorFlow) (0.4.8)
Requirement already satisfied: zipp>=0.5 in c:\programdata\anaconda3\lib\site-packages (from importlib-metadata; python_version < "3.8"->markdown>=2.6.8->tensorboard<3,>=2.3.0->TensorFlow) (2.2.0)
(base) PS C:\WINDOWS\system32>

```

图 2-5 安装成功后的界面

2.2.3 安装 TensorFlow Lite 解释器

要想使用 Python 快速运行 TensorFlow Lite 模型，需要先安装 TensorFlow Lite 解释器，而无须安装本书 2.2.1 小节和 2.2.2 小节中介绍的所有 TensorFlow 软件包。

只包含 TensorFlow Lite 解释器的软件包是完整的 TensorFlow 软件包的一小部分，其中只包含使用 TensorFlow Lite 运行所需要的最少代码：仅包含 Python 类 `tf.lite.Interpreter`。如果只想执行 `.tflite` 模型，而不希望庞大的 TensorFlow 库占用磁盘空间，那么只安装这个小软件包是最理想的选择。

注意 如果需要访问其他 Python API（如 TensorFlow Lite 转换器），则必须安装完整的 TensorFlow 软件包。

在电脑中可以使用 `pip install` 命令安装 TensorFlow Lite，假如 Python 版本是 3.9，则可以使用以下命令安装 TensorFlow Lite：

```
pip install https://dl.google.com/coral/python/tflite_runtime-2.1.0.post1-cp39-cp39m-linux_armv7l
```

2.2.4 解决速度过慢的问题

在使用前面介绍的 `pip` 方式安装 TensorFlow 库时，经常会遇到因为网速过慢而安装失败的问题。这是因为 TensorFlow 库的安装包保存在国外的服务器中，所以国内用户在下载时会遇到网速过慢导致安装失败的问题。为了解决这个问题，国内很多网站也为开发者提供了常用的 Python 库的安装包，例如清华大学和豆瓣网等。

(1) 使用清华源安装 Python 库的语法格式如下：

```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple 库的名字
```

例如，在 Windows 10 系统的“命令提示符”界面中，输入下面的命令即可安装 TensorFlow：

```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple TensorFlow
```

(2) 使用豆瓣源安装 Python 库的语法格式如下：

```
pip install 库的名字 -i http://pypi.douban.com/simple/ --trusted-host pypi.douban.com
```

例如，在 Windows 10 系统的“命令提示符”界面中，输入下面的命令即可安装 TensorFlow：

```
pip install TensorFlow -i http://pypi.douban.com/simple/ --trusted-host pypi.douban.com
```

2.3 准备开发工具

对于 Python 开发者来说，建议使用 PyCharm 开发并调试运行 TensorFlow 程序。另外，为了提高开发效率，谷歌为开发者提供了 Google Colaboratory 工具，这样可以在谷歌浏览器中调试运行 TensorFlow 程序，非常方便。



扫码观看本节视频讲解