

认证测试工程师 人工智能测试 大纲

版本：CT-AI-1.0-CN-1.1

发布日期：2023 年 6 月 6 日

国际软件测试认证委员会



提供者

A4Q 联盟、人工智能联盟、中国软件测试认证委员会与韩国软件测试认证委员会

A4Q



中文版的翻译、编辑和出版统一由 ISTQB®授权的 CSTQB®负责



版权申明

英文版权声明

版权声明©国际软件测试认证委员会（后文称 ISTQB®）。

ISTQB®是国际软件测试认证委员会的一个注册商标。

版权声明©2021，作者 Klaudia Dussa-Zieger（主席）、Werner Henschelchen、Vipul Kocher、Qin Liu、Stuart Reid、Kyle Siemens 和 Adam Leon Smith。

保留所有权利。作者们在此转移版权至 ISTQB®。作者们（作为当前的版权所有者）与 ISTQB®（作为未来的版权所有者）对以下使用条件达成一致：

对于非商业性质用途，从本文档中提取出的信息在注明信息源的情况下，可以被复制。在注明本大纲的信息来源以及版权所有人为作者们和 ISTQB®的前提下，任何获得认证的培训提供者，均可使用本大纲作为培训课程的依据；并且仅允许在培训课程材料获得 ISTQB®认可的成员委员会认证时，才可以在任何该课程相关广告中提及本大纲。

在注明本大纲的信息来源以及版权所有人为作者们和 ISTQB®的前提下，任何个人以及由个人组成的团体均可以允许使用本大纲作为编写文献和书籍的依据。

在未事先获得 ISTQB®的书面同意时，不允许将本大纲用作任何其他用途。

任意 ISTQB®认可的成员委员会均可以翻译本大纲，前提是需要翻译版本的大纲中复现上述版权声明。

中文版权声明

未经许可，不得复制或抄录本中文版文档内容。

版权标志©国际软件测试认证委员会中国分会（以下简称“CSTQB®”）。

修订历史

版本	日期	备注
1.0（英文版）	2021/10/01	GA 大会发布
CT-AI-1.0-CN-1.0	2023/02/13	英文大纲 1.0 本地化完成
CT-AI-1.0-CN-1.1	2023/06/06	中文翻译内容更正

目录

版权申明.....	2
修订历史.....	3
目录.....	4
致谢.....	7
0. 大纲简介.....	8
0.1 大纲的目的.....	8
0.2 人工智能软件测试工程师认证.....	8
0.3 可考核的学习目标和知识认知水平.....	8
0.4 实际操作的能力水平.....	9
0.5 人工智能软件测试工程师认证考试.....	10
0.6 资格认证.....	10
0.7 细节程度.....	10
0.8 本大纲是如何编排的.....	10
1. 人工智能介绍 - 105 分钟.....	12
1.1 人工智能与人工智能效应的定义.....	13
1.2 有限人工智能、通用人工智能与超级人工智能.....	13
1.3 基于人工智能的系统与常规系统.....	13
1.4 人工智能技术.....	14
1.5 人工智能开发框架.....	15
1.6 基于人工智能的系统的硬件.....	15
1.7 人工智能即服务（AIaaS）.....	16
1.7.1 人工智能即服务的合约.....	17
1.7.2 人工智能即服务举例.....	17
1.8 预训练模型.....	17
1.8.1 预训练模型的介绍.....	17
1.8.2 迁移学习.....	18
1.8.3 使用预训练模型与迁移学习的风险.....	18
1.9 标准、规章制度与人工智能.....	19
2. 基于人工智能的系统的质量特征 - 106 分钟.....	20
2.1 灵活性与适应性.....	21
2.2 自治.....	21
2.3 进化.....	22
2.4 偏差.....	22
2.5 行为准则.....	22
2.6 副作用与奖励黑客.....	23
2.7 透明度、整体可解释性与单一可解释性.....	24
2.8 安全性与人工智能.....	24
3. 机器学习（ML）-总览-145 分钟.....	26
3.1 机器学习形式.....	27
3.1.1 有监督学习.....	27
3.1.2 无监督学习.....	27
3.1.3 强化学习.....	28
3.2 机器学习 workflow.....	28
3.3 选择一种机器学习的形式.....	31
3.4 选择机器学习算法涉及的因素.....	31
3.5 过度拟合与欠拟合.....	32
3.5.1 过度拟合.....	32
3.5.2 欠拟合.....	32

3.5.3 动手练习：演示过度拟合与欠拟合	32
4. 机器学习-数据-230 分钟	33
4.1 数据准备是机器学习工作流的一部分	34
4.1.1 数据准备中的挑战	35
4.1.2 动手练习:机器学习的数据准备	35
4.2 工作流中的训练、验证和测试集	36
4.2.1 动手练习:识别训练和测试数据，并创建机器学习模型	36
4.3 数据集质量问题	37
4.4 数据质量及其对机器学习模型的影响	38
4.5 有监督学习的数据标注	38
4.5.1 数据标注方式	38
4.5.2 数据集中的错误标注数据	39
5. 机器学习功能表现度量-120 分钟	40
5.1 混淆矩阵	41
5.2 附加机器学习功能表现度量的分类、回归和聚类	42
5.3 机器学习功能表现度量的局限性	42
5.4 选择机器学习功能表现度量	43
5.4.1 动手练习:评估创建的机器学习模型	44
5.5 基准套件	44
6. 机器学习-神经网络和测试-65 分钟	45
6.1 神经网络	46
6.1.1 动手练习：实现简单的感知器	47
6.2 神经网络覆盖度量	47
7. 基于人工智能系统的测试概述-115 分钟	49
7.1 基于人工智能系统的规范	50
7.2 基于人工智能系统的测试级别	50
7.2.1 输入数据测试	51
7.2.2 机器学习模型测试	51
7.2.3 组件测试	51
7.2.4 组件集成测试	51
7.2.5 系统测试	52
7.2.6 验收测试	52
7.3 测试基于人工智能系统的测试数据	52
7.4 人工智能系统的自动化偏差测试	53
7.5 记录人工智能组件	53
7.6 概念漂移的测试	54
7.7 为机器学习系统选择测试方法	54
8. 测试人工智能特定的质量特征-150 分钟	56
8.1 测试自学习系统时的挑战	57
8.2 测试基于人工智能的自治系统	58
8.3 测试算法、样本和不适当的偏差	58
8.4 测试基于概率的和非确定性的人工智能系统所面临的挑战	59
8.5 测试基于人工智能的复杂系统所面临的挑战	59
8.6 测试基于人工智能的系统的透明性、整体可解释性和单一可解释性	60
8.6.1 实践练习：模型的可解释性	60
8.7 基于人工智能的系统的测试结果参照物	61
8.8 测试目标和验收准则	61
9. 基于人工智能的系统测试的方法和技术-245 分钟	63
9.1 对抗攻击与数据中毒	64
9.1.1. 对抗攻击	64
9.1.2. 数据中毒	64

9.2	结对测试	65
9.2.1	实践练习：结对测试	65
9.3	背靠背测试	65
9.4	A/B 测试	66
9.5	蜕变测试	66
9.5.1	实践练习：蜕变测试	68
9.6	基于经验的人工智能系统测试	68
9.6.1	实践练习。探索性测试和探索性数据分析（EDA）	70
9.7	为基于人工智能的系统选择测试技术	70
10.	基于人工智能的系统的测试环境-30 分钟	72
10.1	基于人工智能的系统的测试环境	73
10.2	用于测试基于人工智能系统的虚拟测试环境	73
11.	使用人工智能进行测试-195 分钟	75
11.1	用于测试的 AI 技术	76
11.1.1	实践练习：AI 在测试中的应用	76
11.2	使用人工智能分析报告的缺陷	76
11.3	使用人工智能生成测试用例	77
11.4	使用人工智能优化回归测试套件	77
11.5	使用人工智能进行缺陷预测	77
11.5.1	实践练习。建立一个缺陷预测系统	78
11.6	使用人工智能测试用户界面	78
11.6.1	通过图形用户界面（GUI）使用 AI 进行测试	78
11.6.2	使用人工智能测试用户图形界面	79
12.	参考文献	80
12.1	标准	80
12.2	ISTQB® 文档	80
12.3	书籍和文献	80
12.4	其他参考资料	83
13.	附录 A-缩写	85
14.	附录 B-人工智能专用或其他术语	86

致谢

本文档在 2021 年 10 月 1 日由 ISTQB® 大会正式发布。

本文由以下国际软件测试认证委员会成员组成的团队完成：Klaudia Dussa-Zieger（主席）、Werner Henschelchen、Vipul Kocher、Qin Liu、Stuart Reid、Kyle Siemens 和 Adam Leon Smith。

团队向以下三个参与方的作者们表示感谢；

- A4Q: Rex Black、Bruno Legeard、Jeremias Rößler、Adam Leon Smith、Stephan Goericke、Werner Henschelchen
- AiU: 主作者 Vipul Kocher、Saurabh Bansal、Srinivas Padmanabhuni 与 Sonika Bengani 和合著者 Rik Marselis、José M. Diaz Delgado
- CSTQB®/KSTQB: Qin Liu、Stuart Reid

团队感谢考试组、术语组以及市场组对编写本大纲提供的协助，感谢 Graham Bath 的技术上的改动，感谢成员委员会提供的建议。

以下人员参与了本大纲的评审与评论工作：

Laura Albert、Reto Armuzzi、Árpád Beszedes、Armin Born、Géza Bujdosó、Renzo Cerquozzi、Sudeep Chatterjee、Seunghye Choi、Young-jae Choi、Piet de Roo、Myriam Christener、Jean-Baptiste Crouigneau、Guofu Ding、Erwin Engelsma、Hongfei Fan、Péter Földházi Jr.、Tamás Gergely、Ferdinand Gramsamer、Attila Gyúri、Matthias Hamburg、Tobias Horn、Jarosław Hryszko、Beata Karpinska、Joan Killeen、Rik Kochuyt、Thomas Letzkus、Chunhui Li、Haiying Liu、Gary Mogyorodi、Rik Marselis、Imre Mészáros、Tetsu Nagata、Ingvar Nordström、Gábor Péterffy、Tal Pe'er、Ralph Pichler、Nishan Portoyan、Meile Posthuma、Adam Roman、Gerhard Runze、Andrew Rutz、Klaus Skaftø、Mike Smith、Payal Sobti、Péter Sótér、Michael Stahl、Chris van Bael、Stephanie van Dijck、Robert Werkhoven、Paul Weymouth、Dong Xin、Ester Zabar、Claude Zhang。

人工智能测试文档中文版本 1.0 翻译参与者（按姓氏拼音排序）：

曹栋、陈嘉诚、陈希、陈智迪、高方原、高蕊、冷炜、刘惠、刘佳钰、任亮、商超博、张希婷

人工智能测试文档中文版本 1.0 QA 评审参与者（按姓氏拼音排序）：

陈晟、丁国富、董昕、范鸿飞、梁静、刘琴、吴洁

致谢企业：

上海均瑜管理咨询有限公司



0. 大纲简介

0.1 大纲的目的

本大纲为 ISTQB® 认证人工智能软件测试工程师的基础。ISTQB® 为下述人员提供本大纲：

1. 提供给成员委员会，用来翻译为当地语言并认证培训提供者。成员委员会可以使大纲满足特定的语言需求，并可以修改引用文献来适应当地出版商。
2. 提供给认证机构，通过本大纲的学习目标来编写当地语言的考试题。
3. 提供给培训提供者，用来制作课程并且确定合适的教学方法。
4. 提供给认证考试考生，用来准备认证考试（无论是参与培训或者自学）。
5. 提供给国际软件与系统工程师社区，用来推动软件与系统测试职业发展，并作为书籍以及文章的依据。

0.2 人工智能软件测试工程师认证

人工智能软件测试工程师的目标群体为，所有参与针对基于人工智能的系统或者/与参与针对人工智能本身的测试的人员。这包括了测试人员、测试分析师、数据分析师、测试工程师、测试咨询师、测试管理员、用户验收测试员与软件开发等职位。这份认证也适合对所有想要参与针对基于人工智能的系统或者/与参与针对人工智能本身的测试有一定理解的人员，例如产品经理、质量经理、软件开发经理、商业分析师、运维团队成员、信息技术总监和管理咨询师。

人工智能软件测试工程师认证总览[I03]是一份另外的文档，包括了以下信息：

- 大纲的业务成果。
- 业务成果表及其与学习目标的联系。
- 大纲总结。
- 大纲中的关系。

0.3 可考核的学习目标和知识认知水平

学习目标支撑了业务成果，并用来构建人工智能软件测试工程师认证考试。

考生可能会被要求认识、记忆或者回忆十一章里任意一章所提到的一个关键词或者概念。具体的学习目标展示在每一章的开头，并使用如下分类：

- K1: 记忆。
- K2: 理解。
- K3: 应用。
- K4: 分析。

章节标题下所有列为关键词的用语，就算没有专门在学习目标中提及，也全部需要记忆（K1）。

0.4 实际操作的能力水平

人工智能软件测试工程师认证包含了实际操作目标，这些目标关注实际操作技能以及能力。

以下等级适用于（所展示的）实际操作目标：

- H0: 对练习进行实时演示，或者进行视频录像。
- H1: 受指引的练习。学生仿照老师进行的一系列步骤进行操作练习。
- H2: 有提示的练习。给学生一个练习，并给予相关提示，使学生可以在给定的时间内完成练习；或者让学生们参与讨论。

通过进行实际练习获得能力，如下所示：

- 演示欠拟合与过度拟合（H0）。
- 为构建机器学习模型进行数据准备（H2）。
- 识别训练数据集与测试数据集，并创建一个机器学习模型（H2）。
- 使用所选的机器学习功能表现度量来评估所创建的机器学习模型（H2）。
- 有编写一个感知机的经验（H1）。
- 使用工具来展示测试人员可以如何使用可解释性（H2）。
- 对一个基于人工智能的系统，使用结对测试来编写并执行测试用例（H2）。
- 对一个给定的情景，使用蜕变测试来编写并执行测试用例（H2）。
- 对一个基于人工智能的系统进行探索性测试（H2）。
- 举例讨论哪一些测试中的行为很少会用到人工智能（H2）。
- 编写一个简单的基于人工智能的缺陷预测系统（H2）。

0.5 人工智能软件测试工程师认证考试

人工智能软件测试工程师认证考试将会基于本大纲。考试问题的答案可能会需要使用到大纲中多于一个章节所提供的材料。除去介绍与附录，大纲中所有的章节都在考试范围内。大纲中包含了标准和书籍作为参考，但是这些标准与书籍中超出大纲中所概括的部分以外的内容，不在考试范围内。

更多细节请参看人工智能软件测试工程师“总览”文档，“考试结构”一章。

参与要求：需要获得 ISTQB® 基础级认证后才可以参与人工智能软件测试工程师认证考试。

0.6 资格认证

ISTQB® 成员委员会可以对按照本大纲编制课程材料的培训提供者进行认证。培训提供者需要从成员委员会或者资格认证机构获取资格认证指导。受认证的课程被认为遵循本大纲，并允许将一次 ISTQB® 考试作为课程的一部分。

本大纲的资格认证指导遵循由进程管理与合规工作小组所发布的通用资格认证指导。

0.7 细节程度

本大纲的细节程度允许国际上的课程与考试保持一致性。为了达成这个目标，大纲由以下部分组成：

- 使用通用的指导性目标来描述人工智能软件测试工程师的目的。
- 一个学生必须能够回忆起来的术语表。
- 为每一个知识领域提供学习与实际操作目标，以此描述应该达成的学习成果。
- 对于关键概念的描述，包括对其来源的引用，例如受认可的文章或标准。

大纲所描述的内容，并未涵盖有关针对测试基于人工智能系统的全部知识领域；它只反映了需要被人工智能测试认证专业测试工程师课程覆盖到的细节程度。大纲专注于介绍人工智能（AI）与机器学习，尤其是如何对基于这些技术的系统进行测试的基本概念。

0.8 本大纲是如何编排的

考试范围总共有十一个章节。每一章的大标题具体给出了学习该章所需要的时间；章节中的内容未给出所需时间。对于获得资格认证的培训课程来说，大纲要求以下十一章的内容，至少需要总共 25.1 小时的授课时间。

- 第一章：105 分钟 人工智能介绍。

- 第二章：105 分钟 基于人工智能的系统的质量特征。
- 第三章：145 分钟 机器学习（ML）-总览。
- 第四章：230 分钟 机器学习-数据。
- 第五章：120 分钟 机器学习功能表现度量。
- 第六章：65 分钟 机器学习-神经网络和测试。
- 第七章：115 分钟 测试基于人工智能的系统总览。
- 第八章：150 分钟 测试人工智能专属质量特征。
- 第九章：245 分钟 测试基于人工智能的系统的方法与技术。
- 第十章：30 分钟 基于人工智能的系统的测试环境。
- 第十一章：195 分钟 使用人工智能进行测试。

1. 人工智能介绍 - 105 分钟

测试关键词

无

人工智能专用关键词

人工智能即服务（AIaaS）、人工智能开发框架、人工智能效应、基于人工智能的系统、人工智能（AI）、神经网络、深度学习（DL）、深度神经网络、通用人工智能、通用数据保护条例（GDPR）、机器学习（ML）、有限人工智能、预训练模型、超级人工智能、技术奇点、迁移学习

第一章学习目标：

1.1 人工智能与人工智能效应的定义

AI-1.1.1 (K2) 描述人工智能效应与其如何影响人工智能的定义。

1.2 有限人工智能、通用人工智能与超级人工智能

AI-1.2.1 (K2) 区分有限人工智能、通用人工智能与超级人工智能。

1.3 基于人工智能的系统与常规系统

AI-1.3.1 (K2) 区分基于人工智能的系统与常规系统。

1.4 人工智能技术

AI-1.4.1 (K1) 识别编写人工智能使用到的不同技术。

1.5 人工智能开发框架

AI-1.5.1 (K1) 识别流行的人工智能开发框架。

1.6 基于人工智能的系统硬件

AI-1.6.1 (K2) 比较搭建基于人工智能的系统时可选择的硬件。

1.7 人工智能即服务（AIaaS）

AI-1.7.1 (K2) 解释人工智能即服务（AIaaS）的概念。

1.8 预训练模型

AI-1.8.1 (K2) 解释预训练模型的用途及其相关的风险。

1.9 标准、规章制度与人工智能

AI-1.9.1 (K2) 描述标准是如何应用于基于人工智能的系统的。

1.1 人工智能与人工智能效应的定义

人工智能（AI）这个词可以追溯到 20 世纪 50 年代，用来指代建造与编写拥有模拟人类能力的“智能”机器的目的。现如今人工智能的概念已经有了很大程度的进化，如以下定义所描述[S01]：

一个设计好的系统获取、处理、新增以及应用知识与技能的能力。

人们理解人工智能意义的方法取决于他们当前的认知。在 20 世纪 70 年代，大部分人都认为计算机系统未来能够在国际象棋上战胜人类，这属于人工智能。现在，在计算机系统 Deep Blue 战胜世界国际象棋冠军 Garry Kasparov 的二十多年后，很多人已经不再认为编写当时那个系统使用的暴力破解方法是真正的人工智能（这个系统并非从数据中学习，并且也不能够进行自主学习）。与之相似的还有 20 世纪 70 年代和 80 年代的专家系统，这些系统使用人类的专业知识作为规则，并可以在专家不在场的情况下重复执行。这些系统曾经也被认为属于人工智能，但是现如今已不再被如此认为。

这种对于什么构成了人工智能的认知上的持续变化，被称作“人工智能效应”[R01]。当社会对于人工智能的认知变化时，人工智能的定义就会跟着改变。结果就是，现在对其做出的任何定义，在未来都很可能被改变，并且与过去的定义也不一致。

1.2 有限人工智能、通用人工智能与超级人工智能

在高级层面上，人工智能可以被分成三类：

- 有限人工智能（也被称为弱人工智能）系统是被用来进行一项特定的工作，并且工作内容也受到限制。现在这种形式人工智能随处可见。比如游戏系统、垃圾信息过滤、测试用例生成和语音助手。
- 通用人工智能（也被称为强人工智能）系统拥有与人类相似的通用（广泛的）认知能力。这些基于人工智能的系统可以像人类一样对于他们所在的环境进行推理与理解，并且依此做出行动。到 2021 年为止，没有已经被实现的通用人工智能系统。
- 超级人工智能系统能够复制人类认知（通用人工智能），并且还拥有庞大的处理能力、几乎无限的内存以及能获取到全部的人类知识（例如通过互联网）。人们认为超级人工智能会很快变得比人类更聪明。基于人工智能的系统从通用人工智能转变为超级人工智能的那一刻，通常被称为科技奇点[B01]。

1.3 基于人工智能的系统与常规系统

在一个典型的常规计算机系统中，软件是人类使用命令式的语言编写的，这些软件包括了诸如判断和循环的结构。人类可以相对容易的理解这种系统的输入值是如何转化为输出值的。在一个使用了机器学习（ML）的基于人工智能的系统中，系统使用数据中的模式，来判断如何对未来获得的新数据

做出反应（对与机器学习的详细解释见第三章）。举个例子，设计一个基于人工智能的图像处理器来识别猫的图片时，使用了一系列已知包含了猫的图片对处理器进行训练。人工智能自己来判断这些数据中有哪些模式或者特征可以用来识别猫。之后使用这些模式与规则来识别一些新的图片中是否包含了猫。在很多基于人工智能的系统中，这些通过预测步骤得出的结果就没有那么容易被人类所理解了（见 2.7 章）。

实际使用中，基于人工智能的系统可以使用很多种不同的科技来实现（见 1.4 章），并且“人工智能效应”（见 1.1 章）可以决定现在哪些系统被认为是基于人工智能的系统，而哪些被认为是常规系统。

1.4 人工智能技术

人工智能可以由广泛的技术来实现，比如：

- 模糊逻辑。
- 搜索算法。
- 推理技巧。
 - 规则引擎。
 - 演绎分类器。
 - 基于用例的推理。
 - 程序推理。
- 机器学习方法。
 - 神经网络。
 - 贝叶斯模型。
 - 判定树。
 - 随机森林。
 - 线性回归。
 - 逻辑回归。
 - 聚类算法。
 - 遗传算法。
 - 支持向量机（SVM）。

基于人工智能的系统通常使用了这些方法里的一种或多种方法。

1.5 人工智能开发框架

有很多人工智能开发框架可以使用，其中一些是关注特定领域的。这些框架支持许多的行为，比如数据准备、算法选取和在多种处理器上编译模型，例如中央处理器（CPU）、图形处理器（GPU）和云量子处理器（TPU）。选用特定的开发框架也取决于一些特定因素，比如开发语言与使用难度。以下是（到 2021 年四月为止）最流行的一些框架：

- **Apache MxNet**：一种开源的深度学习框架，亚马逊在亚马逊云服务（AWS）中使用了这种框架[R02]。
- **CNTK**：微软认知工具包（CNTK）是一个开源的深度学习工具包[R03]。
- **IBM Watson Studio**：一个用来支持人工智能解决方案开发的工具套件[R04]。
- **Keras**：一个高阶开源接口，使用 Python 编写，可以在 TensorFlow 和 CNTK 之上运行[R06]。
- **PyTorch**：Facebook 的一个开源机器学习库，软件用来进行图像处理和自然语言处理（NLP）。为 Python 以及 C++ 接口提供支持[R07]。
- **Scikit-learn**：一个 Python 的开源机器学习库[R08]。
- **TensorFlow**：一个为了可扩展的机器学习设计的基于数据流图形的开源机器学习框架，该框架由谷歌提供[R05]。

请注意，这些开发框架一直在进化，有时会被合并，有时也会被新的框架所取代。

1.6 基于人工智能的系统的硬件

有多种硬件被用来进行机器学习模型训练（见第三章）和编写模型的工作。举个例子，一个用来进行语言识别的模型能在一个低端智能手机上运行，尽管可能需要使用云计算来训练它。当主机设备无法连接互联网时，一个通用的方法就是在云中训练模型，再部署到主机设备上。

机器学习通常在支持以下功能的硬件中受益：

- **低精度算法**：这种算法使用更少的比特位进行计算。（比如使用 8 比特位而不是 32 比特位，8 比特位对于机器学习来说通常已经够用了）。
- **处理大型数据结构的能力**（比如能够支持矩阵乘法）。
- **大规模并行（并发）处理**。

通用的中央处理器提供了许多机器学习应用不特别需要的复杂运算的支持，而且只能够为机器学习

习提供几个内核。结果就是中央处理器的构架与图形处理器相比，在训练与运行机器学习模型时效率更低，图像处理器拥有上千个内核，而且这些内核被设计用来进行大规模的并行且相对简单的图像处理。所以图形处理器运行机器学习应用通常都比中央处理器表现的要好，尽管中央处理器通常有更高的时钟速度。对于小规模机器学习工作来说，图形处理器通常都是最好的选择。

一些硬件是人工智能专用的，比如专用的特定应用集成电路（ASIC）和片上系统（SoC）。这些人工智能专用的解决方法，拥有的特征比如多内核、特殊数据管理和进行内存中处理的能力。它们最适合边缘计算，机器学习模型的训练则在云端进行。

使用了人工智能专用构架的硬件现在（截至 2021 年 4 月）正在开发中。这种硬件包含了神经形态处理器[B03]，这种处理器不使用传统的冯诺伊曼构架，而是使用了一种松散的模仿人脑神经元的构架。

人工智能硬件提供商与他们的处理器（截至 2021 年 4 月）示例：

- 英伟达：他们提供了一系列的图形处理器与人工智能专用处理器，比如 Volta[R09]。
- 谷歌：他们为训练与干预开发了特定应用集成电路。用户们可以通过谷歌云使用谷歌的 TPU（云张量处理器）[R10]，边缘云张量处理器[R11]是一个专用的特定应用集成电路，设计用来在个人设备上运行人工智能。
- 英特尔：他们为深度学习（训练与干预）提供了 Nervana 神经网络处理器[R12]和 Movidius Myriad 视觉处理器，用来干涉计算机视觉与神经网络应用。
- Mobileye：他们生产了 EyeQ 家族的片上系统设备[R13]，为复杂与高强度计算的视觉处理提供支持。这些设备使用在汽车中有更低的耗能。
- 苹果：他们为 iPhone 的机载人工智能生产了仿生芯片[B04]。
- 华为：他们的智能手机麒麟 970 芯片拥有为人工智能提供的内置神经网络处理器[B05]。

1.7 人工智能即服务（AIaaS）

人工智能组件，例如机器学习模型，能够在组织内被创建，通过第三方下载，或者用来作为一个网上的服务（AIaaS）。混合的方法也是可行的，这种方法里一部分人工智能功能从系统内部提供，另一部分作为服务提供。

当机器学习被作为一种服务使用时，通过网页提供一个机器学习模型的接入口，并可以为数据准备和存储、模型训练、评估、调整、测试和部署提供协助。

第三方提供商（比如亚马逊云服务、微软）提供专门的人工智能服务，比如人脸与语言识别。允许个人或团体使用基于云的服务编写人工智能，即使这些个人与团体的资源与技术不足以搭建他们自己的人工智能服务。在此之上，机器学习模型作为第三方服务的一部分进行提供，这些模型比起已经

提供给利益相关者的模型，更可能被大量的、更加多元的训练集所训练，比如那些近期转入人工智能市场的提供商。

1.7.1 人工智能即服务的合约

这些人工智能服务在提供时通常都会有一份与非人工智能基于云的软件即服务相似的合同。人工智能即服务的合同都包含了一个服务等级协议（SLA），这份协议定义了对可用性和安全性的承诺。这类服务等级协议通常包含了服务的正常运行时间（比如 99.9% 的正常运行时间）和一个处理缺陷的响应时间，但是很少会用同样的方式（见第五章）定义机器学习功能表现度量（比如正确性）。人工智能即服务经常使用订阅制进行付费，当合同约定的可用性与/或响应时间不达标时，服务提供商通常会为未来的服务提供额度。除了这些额度以外，大部分的人工智能即服务合同提供了有限的责任（在已经付费的部分外），意味着依赖于人工智能即服务的基于人工智能的系统，通常仅有相对低风险的应用会被使用，当这些应用丢失服务不会有特别大的损失。

这些服务经常会有一个初始的免费期，用来作为验收阶段。在此阶段期间，人工智能即服务的消费者应该测试服务提供商是否满足他们对于功能性和绩效方面（例如正确性）的需求。为了弥补所提供服务的透明度，这一步通常都是必要的（见 7.5 章）。

1.7.2 人工智能即服务举例

以下是一些人工智能即服务的例子（截至 2021 年 4 月）：

- IBM Watson 助手：这是一个人工智能聊天机器人，根据月活跃用户数定价。
- 谷歌云人工智能与机器学习产品：这些提供了基于文档的人工智能，包括了表格分析器与文档识别器。定价基于需要处理的文件页数。
- 亚马逊 CodeGuru：提供了一份人工智能 Java 代码的评测，给开发者提供提升代码质量的建议。定价基于所需分析的代码行数。
- 微软 Azure 意识搜索：提供了人工智能云搜索。定价基于搜索单元（取决于使用的储存和吞吐量）。

1.8 预训练模型

1.8.1 预训练模型的介绍

训练人工智能模型可能变得花销很大（见第三章）。首先要有准备好的数据，然后要有训练好的模型。第一项活动可能消耗大量的人力资源，后一项活动则会消耗大量的计算资源。很多组织没有这

种程度的资源。

一种更便宜，而且通常更高效的替代方法是使用预训练模型。预训练模型提供了与所需模型相似的功能性，并且通过扩展和/或关注预训练模型的功能性作为一个新模型的基础。仅有很有限的技术可以使用这类模型，比如神经网络和随机森林。

如果需要一个图片分类器，它可以使用公开的 ImageNet 数据集进行训练，这个数据集包含了已经超过 1000 个分类的超过 1400 百万张图片。这降低了耗费大量资源但又不能保证成功的风险。另外也可以使用已经拿这个数据集训练好的模型。通过使用预训练模型，省下了训练成本，并且排除了模型无法使用的大部分风险。

当不经过修改就使用预训练模型时，这个模型可以很简单地嵌入基于人工智能的系统，或者也可以用作一项服务（见 1.7 章）。

1.8.2 迁移学习

把一个预训练模型拿来修改并去执行另一个不同的需求也是可能的。这被称为迁移学习，被用于深度神经网络中，神经网络的外层（见第六章）通常用来执行一些最基本的任务（比如在图像分类器里面识别直线与曲线的区别）。在这个例子里，图像识别的除了最里层以外都可以被重复利用，消除了训练外层的需求。之后再重新训练内层去处理新分类器的特殊需求。实际操作中，可以使用有针对性的问题额外训练预训练模型，以对其进行优化。

这种方法的效率很大程度上取决于原始模型与新模型在所需功能上的相似度。打个比方，修改识别猫的图像分类器来识别狗，要远比修改去识别人类口音有效得多。

现在有很多现成的预训练模型，特别是那些来自于学术研究的模型。一些这类预训练模型是 ImageNet 模型[R14]，比如用来分类图像的 Inception、VGG、AlexNet 和 MobileNet，和预训练自然语言处理模型，比如谷歌的 BERT[R15]。

1.8.3 使用预训练模型与迁移学习的风险

使用预训练模型与迁移学习在建造基于人工智能的系统时都是很常见的方法，但是两种方法都有其风险。这些风险包括：

- 比起内部生成的模型，预训练模型可能会缺少透明度。
- 预训练模型执行的功能与需要的功能之间的相似度可能会不足。而且数据科学家可能还没有理解这些区别。
- 预训练模型在开发时数据准备阶段（见 4.1 章），与新系统使用这个模型时数据准备阶段的区别，有可能会对功能绩效结果产生影响。

- 一个预训练模型的短处可能会被使用它的系统所继承，而且这些短处可能也没写在文档里。举个例子，在训练模型使用的数据缺少文档时，继承偏差（见 2.4 章）可能会不明显。而且，如果这个预训练模型没有被大规模使用的话，这个模型可能还会有更多未知（或者文档未说明）的缺陷，可能会需要更多严密的测试才能减轻这种风险。
- 对于预训练模型有的缺陷，基于这些模型进行迁移学习所构建的模型很有可能会暴露于对同样的缺陷（比如 9.1.1 章中的对抗攻击）。在此之上，如果已知一个基于人工智能的系统使用了某一个预训练模型（或者基于某个预训练模型），那这些模型所导致的缺陷可能已经被潜在攻击者知道了。

注意上述风险的严重程度可以很容易地通过对预训练模型编写详细的文档来减轻（见 7.5 章）。

1.9 标准、规章制度与人工智能

国际电工委员会联合技术委员会和信息技术国际标准化组织（ISO/IEC JTC1）为人工智能准备了国际标准。举个例子，一个人工智能领域的分会（ISO/IEC JTC1/SC42）在 2017 年成立。在此之上，涵盖了软件与系统工程的 ISO/IEC JTC1/SC7 也针对“测试基于人工智能的系统”发布了一份技术报告[S01]。

人工智能标准也在区域范围内（比如欧盟标准）和国家范围内进行发布。

欧盟范围内的通用数据保护条例（GDPR）在 2018 年五月生效，为数据控制者在私人数据和自动化决策方面制定了规则[B06]。它包括了对评测以及提升人工智能系统功能绩效的要求，包括了对潜在歧视的减轻方法，以及确保了个人不受自动化决策系统管制的权利。从测试角度来说，通用数据保护条例最重要的一点是个人信息（包括预测）需要准确。这不代表系统做出的每一个预测都需要是准确的，但是系统用来达到的目的需要足够的准确。

德国国家标准机构（DIN）也开发了一套人工智能质量元模型（[S02] [S03]）。

行业机构也会发布人工智能的标准。举个例子，电气与电子工程师协会（IEEE）正在编写很多关于伦理与人工智能的标准（IEEE 关于人工智能与自动化系统的伦理考量的全球倡议）。很多这样的标准在编写本文时正在编写中。

在有关安全的系统中使用人工智能时，相关的规章标准也会生效，比如对汽车系统的 ISO 26262[S04]和 ISO/PAS 21448（SOTIF）[S05]标准。这种规章标准通常由政府组织强制执行，如果汽车内置的软件不符合 ISO 26262 标准，那在很多国家销售这种汽车都会是非法的。标准独立来看都是自愿性的文档，但是他它通常都会被法规或合同强制使用。不过，很多这些标准的使用者遵循标准是为了从作者的专业知识中有所收获，并且能制造出高品质的产品。

2. 基于人工智能的系统的质量特征 - 106 分钟

关键词

无

人工智能专属关键词

适应性、算法偏差、自治、偏差、演变、可解释性、可解释人工智能（XAI）、灵活性、不恰当偏差、可理解性、机器学习系统、机器学习、奖励黑客、鲁棒性、样本偏差、自学习系统、副作用、透明性

第二章学习目标

2.1 灵活性与适应性

AI-2.1.1 (K2) 解释灵活性与适应性作为基于人工智能的系统的特征的重要性。

2.2 自治

AI-2.2.1 (K2) 解释自治与基于人工智能的系统之间的关系。

2.3 进化

AI-2.3.1 (K2) 解释管理基于人工智能的系统中的演变的重要性。

2.4 偏差

AI-2.4.1 (K2) 描述基于人工智能的系统中的不同类型的偏差，与不同的发生原因。

2.5 行为准则

AI-2.5.1 (K2) 讨论在开发、部署以及使用基于人工智能的系统时应该遵循的伦理条例。

2.6 副作用与奖励黑客

AI-2.6.1 (K2) 解释副作用的产生与基于人工智能的系统中的奖励黑客。

2.7 透明度、整体可解释性与单一可解释性

AI-2.7.1 (K2) 解释透明性、可理解性与可解释性如何作用在基于人工智能的系统中。

2.8 安全性与人工智能

AI-2.8.1 (K1) 回忆导致基于人工智能的系统难以使用在安全相关的应用中的特征。

2.1 灵活性与适应性

灵活性与适应性是关系密切的质量特征。这本大纲中，我们认为灵活性是系统能够使用在非原有需求的场景下的能力，而适应性是系统能够为新的场景进行修改的容易程度，比如不同的硬件与更换操作系统。

灵活性与适应性两者都有用，如果：

- 在系统部署时不完全了解操作环境。
- 期望系统能够适应新的操作环境。
- 期望系统能够适应新的场景。
- 系统必须能够决定它何时切换行为模式。

我们期望自学习的基于人工智能的系统能够实现以上所有特征。因此，这类系统必须有适应能力并且拥有灵活性的潜质。

一个基于人工智能的系统对于灵活性与适应性的需求，应该包括这个系统期望适应的环境转变的细节。这些需求也应该指出系统进行适应所用的时间与资源上的限制（比如它需要适应多长时间才能识别一个新类别的物体）。

2.2 自治

在定义自治时，我们首先需要认识到，一个完全自治的系统将会完全独立于人类的监督与控制。实践中，我们通常不希望达到完全自治。比如说，人们喜欢用自治来形容完全自动驾驶汽车，这种汽车的官方定义是拥有“完全的驾驶自动化”[B07]。

很多人认为自治系统“聪明”或者“智能”，从中推断出它们包括了基于人工智能的组件来执行一些特定的功能。比方说，需要感知环境的自动驾驶汽车通常使用了很多的感知器和图像处理器来收集车辆周遭的实时环境信息。机器学习，尤其是深度学习（见 6.1 章），被认为是实现这种功能最有效的方式。自治系统可能还包括了决策和控制功能。这两种功能都可以使用基于人工智能的组件有效实现。

虽然一些基于人工智能的系统被认为属于自治，但是这不代表所有基于人工智能的系统都可以称之为自治。在本大纲中，我们认为自治时系统具备在独立于人类监督与控制的情况下长时间运行的能力。这能够帮助识别出自治系统需要被明确和测试的特征。比方说，我们需要知道在没有人类干预的情况下，自治系统能够令人满意运行的期望时长。在此之上，识别出自治系统必须将控制权交还给人类的情况也十分重要。

2.3 进化

本大纲中，我们认为进化是系统自我提升以应对变化的外部限制的能力。一些人工智能系统可以被形容为自学习系统，成功的基于人工智能的自学习系统需要包含这种形式的演变。

基于人工智能的系统经常在不断演变的环境中运行。像其他形式的信息技术系统一样，一个基于人工智能的系统需要有足够的灵活性和适应性，来应对其操作环境的变化。

自学习的基于人工智能的系统通常需要管理两种形式的变化：

- 一种形式是当系统从自己的决策及自己与环境的交互中学习。
- 另一种形式是当系统从对系统的操作环境做出的改变中学习。

在两种情况下系统都可以进行理想化的进化，来增强有效性和效率。但是，这种演变必须要被控制住以防止系统发展出我们不想要的特征。任何进化都需要继续满足原系统的需求与限制。当缺少这些需求与限制时，系统必须要确保任何的进化都在其能力限制内，并且始终与人类的价值相符。2.6 章给出了关于自学习的基于人工智能的系统的副作用与奖励黑客的相关例子。

2.4 偏差

对于基于人工智能的系统来说，偏差是一种统计度量，用来衡量系统输出的结果与我们认为的不存在偏好的正确结果之间的距离。不恰当偏差跟比如性别、人种、种族、性取向、收入水平和年龄有关。报告也指出了在基于人工智能的系统中产生不恰当偏差的原因，举个例子，向银行提供借款推荐的系统、招聘系统和司法监控系统。

偏差在很多种基于人工智能的系统中都能见到。比方说，很难防止专家自己的偏见被专家系统内置的规则所用。然而，机器学习系统的流行意味着与偏见有关的大部分讨论都是在这些系统的背景下进行的。

机器学习系统用来做决策和预测，使用由收集的数据产出的算法，这两种组件会导致结果有以下偏差：

- 算法偏差在当算法配置错误时会发生，比如说，当它会把某些数据看的与其他数据更有价值。这种偏差可以由机器学习算法的超参数调整引发。（见第 3.2 节）
- 样本偏差在当训练数据不能完全代表机器学习所应用到的数据空间时会产生。

不恰当偏差通常由样本偏差产生，但是偶尔也会由算法偏差产生。

2.5 行为准则

在剑桥字典中伦理有如下定义：

一个由公认的理念控制行为的系统，尤其是当这个系统基于道德时

拥有增强能力的基于人工智能的系统，对人们的生活的影响大部分都是正面的。当这些系统应用越来越广泛时，关于这些系统的使用是否符合伦理的担心就出现了。

什么是被认为符合伦理的，是会随着时间的推移所改变的，而且也会由地域与文化间的变化所改变。当把一个基于人工智能的系统从一个地方部署到另一个地方时，一定要关注这两方利益相关方价值观之间的区别。

很多国家与地区都能找到国家或国际上对于人工智能伦理的政策。经济合作与发展组织在 2019 年发表了他们对于人工智能的原则，这是第一个由政府认可的关于负责任的开发人工智能的国际标准【B08】。这些原则在发布时被 42 个国家采纳，并且欧洲委员会也为其背书。它包括了实际政策推荐，也包括了对于“负责任的管理可信任的人工智能”的基于价值观的原则。概括如下：

- 人工智能应该通过推动包容性增长、可持续发展和福祉来造福人类和地球。
- 人工智能系统应该遵守法律、人权、民主价值观和多样性，并且应当为确保社会公平包含恰当的防范措施。
- 为了确保人们理解人工智能的产出，并且能挑战它们，人工智能应该有一定的透明度。
- 人工智能系统在其生命周期中，必须要能够用坚固、可靠和安全的方法运行，并且需要持续的评估其风险。
- 开发、部署与运营人工智能系统的组织与个人需要为系统负责。

2.6 副作用与奖励黑客

副作用与奖励黑客能使基于人工智能的系统在尝试达成目标时产生非预期的、甚至有害的结果【B09】。

当基于 AI 的系统的设计者指定的目标“专注于完成环境中的某些特定任务但忽略（可能非常大）环境的其他方面，因此隐含地表达了对环境变量的漠不关心，这可能会产生负面影响，这实际上可能对改变有害”【B09】。比如说，一个自动驾驶汽车如果目标设定为前往目的地时“尽可能的省油，并尽可能的注意安全”可能可以达成目标，但是可能会产生使乘客对于旅程所用的过长时间这种极度厌烦的副作用。

当基于人工智能的系统明确目标时使用了“聪明”或者“简单”的解决方法，但是这些方法“违背了设计者的意图”时，就会导致出现奖励黑客现象。这类目标可能被有效博弈。奖励黑客一个运用广泛的例子是，基于人工智能的系统在教自己玩一款街机电脑游戏。给它的目标是获得“最高分”，为了达成这个目标，系统并没有通过玩游戏达成目标，而是直接黑了储存最高分的数据。

2.7 透明度、整体可解释性与单一可解释性

基于人工智能的系统通常应用在那些用户需要相信这些系统的领域。这些领域有的是出于安全原因，也有可能是为了保护隐私，也可能是因为它们可能会做出改变人生的预测与决策。

大部分客户接触基于人工智能的系统都是以“黑盒”的形式，并且不清楚这些系统是如何得出结果的。一些情况下，甚至建造系统的数据科学家也会有这种无知的情况。偶尔有些用户甚至都不知道他们在跟基于人工智能的系统进行互动。

基于人工智能的系统的固有的复杂性，导致了“可解释性人工智能”（XAI）领域的出现。可解释性人工智能的目的是让用户能够理解基于人工智能的系统是如何得出最终结果的，从而增加用户对它们的信任。

据皇家学会的说法【B10】，有很多想要使用可解释的人工智能的理由，包括：

- 让用户对系统有信心。
- 防止偏差。
- 达到规章标准或政策需求。
- 提升系统设计。
- 评估风险、鲁棒性与弱点。
- 理解并验证一个系统的输出值。
- 自治、代理（使用户感到自己有权限）、并满足社会价值观。

从利益相关方的角度来看，这引出了以下三个基本需要的在基于人工智能的系统中的可解释性人工智能的特征（同时请见 8.6 章）：

- 可公开性：这被认为是用来确定生成模型的算法和训练数据的容易程度。
- 可理解性：人工智能技术被不同利益相关方，包括用户，所理解的能力。
- 可解释性：用户是否能够容易地确定基于人工智能的系统是怎么得出结果的。

2.8 安全性与人工智能

这份大纲中，我们认为安全性是我们期望基于人工智能的系统不会对人员、财产或环境造成伤害。基于人工智能的系统可能被用来对影响安全的事情做出决策。比如，基于人工智能的系统用于医药、生产、防卫、保卫和运输领域都会产生潜在的安全性影响。

使得基于人工智能的系统更难保障其安全性（比如不伤害人类）的特征如下：

- 复杂度。

- 不确定性。
- 概率性质。
- 自学习。
- 缺少透明度、可理解性与可解释性。
- 缺少鲁棒性。

第八章中涵盖了测试这里面很多特征会遇到的挑战。

中国软件测试认证委员会 (CSTQB®)

3. 机器学习（ML）-总览 - 145 分钟

关键词

无

人工智能专属关键词

关联、分类、聚类、数据准备、机器学习算法、机器学习框架、机器学习功能表现准则、机器学习模型、机器学习训练数据、机器学习 workflow、模型评价、模型调优、异常值、过拟合、回归、增强学习、有监督学习、欠拟合、无监督学习

第三章学习目标

3.1 机器学习形式

- | | | |
|----------|------|---------------------|
| AI-3.1.1 | (K2) | 描述作为有监督学习一部分的分类与回归。 |
| AI-3.1.2 | (K2) | 描述作为无监督学习一部分的聚类与关联。 |
| AI-3.1.3 | (K2) | 描述增强学习。 |

3.2 机器学习 workflow

- | | | |
|----------|------|-----------------|
| AI-3.2.1 | (K2) | 总结创建机器学习系统的工作流。 |
|----------|------|-----------------|

3.3 选择一种机器学习的形式

- | | | |
|----------|------|--|
| AI-3.3.1 | (K3) | 给出一个项目场景，（从分类、回归、聚类、关联和增强学习中）识别出适合的机器学习形式。 |
|----------|------|--|

3.4 选择机器学习算法涉及的因素

- | | | |
|----------|------|-----------------|
| AI-3.4.1 | (K2) | 解释选择机器学习算法时的因素。 |
|----------|------|-----------------|

3.5 过度拟合与欠拟合

- | | | |
|----------|------|---------------|
| AI-3.5.1 | (K2) | 总结过拟合与欠拟合的概念。 |
| HO-3.5.1 | (H0) | 演示过拟合与欠拟合。 |

3.1 机器学习形式

机器学习算法可以分为：

- 有监督学习，
- 无监督学习，
- 强化学习。

3.1.1 有监督学习

在这种类型的学习中，该算法在训练阶段从标注的数据来创建机器学习模型。标注的数据通常包括一对输入（例如，狗的图像和标注为“狗”），在训练期间该算法用于推断输入数据（例如狗的图像）与输出标注（例如“狗”和“猫”）之间的关系。在机器学习模型测试阶段，一组新的看不见的数据给到被训练的模型来预测输出。一旦输出准确度等级令人满意，就可以部署该模型。

有监督学习解决的问题分为两类：

- 分类：当需要解决一个输入被归类为一些预定义的类时，可以使用分类。图像中的人脸识别或目标检测就是使用分类解决问题的示例。
- 回归：这适用于当问题需要机器学习模型使用回归来预测数字输出时。根据输入的习惯数据预测一个人的年龄或预测未来的股票价格都是使用回归解决问题的示例。

请注意，术语“回归”在机器学习问题的上下文中使用，不同于它在其他 ISTQB®教学大纲中的使用，例如 [I01]，在 ISTQB®中回归用于描述变更相关缺陷导致的软件修改问题。

3.1.2 无监督学习

在这种学习中，该算法在训练阶段从未标注的数据创建机器学习模型。在训练期间该算法使用未标注数据推断输入数据中的模式，并根据其共性将输入分配给不同的类别。在测试阶段，一组新的看不见的数据给到被训练的模型来预测输出，用以预测输入数据应分配给哪些类别。一旦输出准确度等级令人满意，就可以部署该模型。

无监督学习解决的问题分为两类：

- 聚类：这是指当问题需要识别输入数据点的相似性，以便根据共同的特征或属性对它们进行分组。例如，为了营销的目的，聚类被用来对不同类型的客户进行分类。
- 关联：这是指当问题需要在数据属性中识别利益关系或依赖关系。例如，产品推荐系统可以根据客户的购物行为识别关联。

3.1.3 强化学习

强化学习是系统（“智能体”）通过与上下文交互的迭代方式学习，从而在经验中学习的方法。强化学习不使用训练数据。智能体在做出正确决定时会得到奖励，在做出错误决定时会得到惩罚。

设置环境、为智能体选择正确的策略以满足预期目标以及设计奖励函数，是实行强化学习的关键挑战。机器人、自动驾驶车辆和聊天机器人都是使用强化学习应用程序的示例。

3.2 机器学习 workflow

机器学习 workflow 中的活动包括：

理解目标

要部署的机器学习模型的目的需要得到利益相关方的理解和同意，以确保与业务优先级保持一致。应该为开发的模型定义验收准则（包括机器学习功能表现度量——参见第 5 章）。

选择框架

应根据目标、验收准则和业务优先级（参见 1.5 节）选择合适的人工智能开发框架。

选择并建立算法

机器学习算法的选择基于各种因素，包括目标、验收准则和可用数据（参见 3.4 节）。该算法可能是手动编码，但它通常从预先编写的代码库中检索。如果需要，那么编译算法为训练模型做准备。

准备并测试数据

数据准备（参见 4.1 节）包括数据采集、数据预处理和特征工程。探索性数据分析（EDA）可与这些活动同时执行。

算法和模型使用的数据将根据目标，并被图 1 所示的“模型生成与测试”活动中的所有活动使用。例如，如果系统是实时交易系统，数据将来自交易市场。

用于训练、调优和测试模型的数据必须代表模型将使用的操作数据。在某些情况下，可以使用预先收集的数据集对模型进行初始训练（例如，参见 Kaggle 数据集[R16]）。否则，原始数据通常需要一些预处理和特征工程。

需要执行测试数据以及任何自动化的数据准备步骤。有关输入数据测试的更多详细信息，请参阅 7.2.1 节。

训练模型

选定的机器学习算法使用训练数据来训练模型。

某些算法，例如生成神经网络的算法，会多次读取训练集。训练集上的每一次训练迭代都被称为

一个学习周期。

定义模型结构的参数(例如,神经网络的层数或决策树的深度)被传递给算法。这些参数被称为模型超参数。

控制训练的参数(例如,在训练神经网络时使用多少时期)也传递给算法,这些参数称为超参数。

评估模型

模型根据意见一致的机器学习功能表现度量进行评估,使用验证数据集,然后使用结果来改进(调优)模型。模型评估和调优应该类似科学实验,需要在受控条件下仔细地进行,并有明确的文档。在实践中,通常使用不同的算法创建和训练多个模型(如随机森林法、支持向量机和神经网络),然后根据评估和调优的结果选择最佳模型。

调优模型

根据意见一致的机器学习功能表现度量评估模型的结果,用于修改模型设置以适应数据,从而提高其性能。模型可以通过超参数调优方法进行调优,其中训练活动被修改(例如通过改变训练步骤的数量或改变用于训练的数据量),或更新模型的属性(例如神经网络中的神经元数量或判定树的深度)。

训练、评估和调优这三项活动可视为模型产生的组成步骤,如图 1 所示。

测试模型

生成模型后(即:已对模型进行训练、评估和调优),应对它进行独立的测试数据集测试,以确保符合意见一致的机器学习功能表现准则(参见 7.2.2 节)。还需要将测试中的功能绩效指标与评估中的指标进行比较,如果具有独立数据的模型的绩效指标明显低于评估期间的绩效指标,则可能需要选择不同的模型。

除了功能的绩效测试之外,还需要执行非功能测试,例如训练模型的时间,以及提供预测所需的时间和资源使用情况。通常,这些测试由数据工程师/科学家执行,但具有足够领域知识和访问相关资源的测试人员也可以执行这些测试。

部署模型

一旦模型开发完成,如图 1 所示,调优后的模型通常需要重新设计,以便与其相关资源一起部署,包括相关的数据管道。这通常是通过框架实现的。目标可能包括嵌入式系统和云,其中模型可以通过 web API 访问。

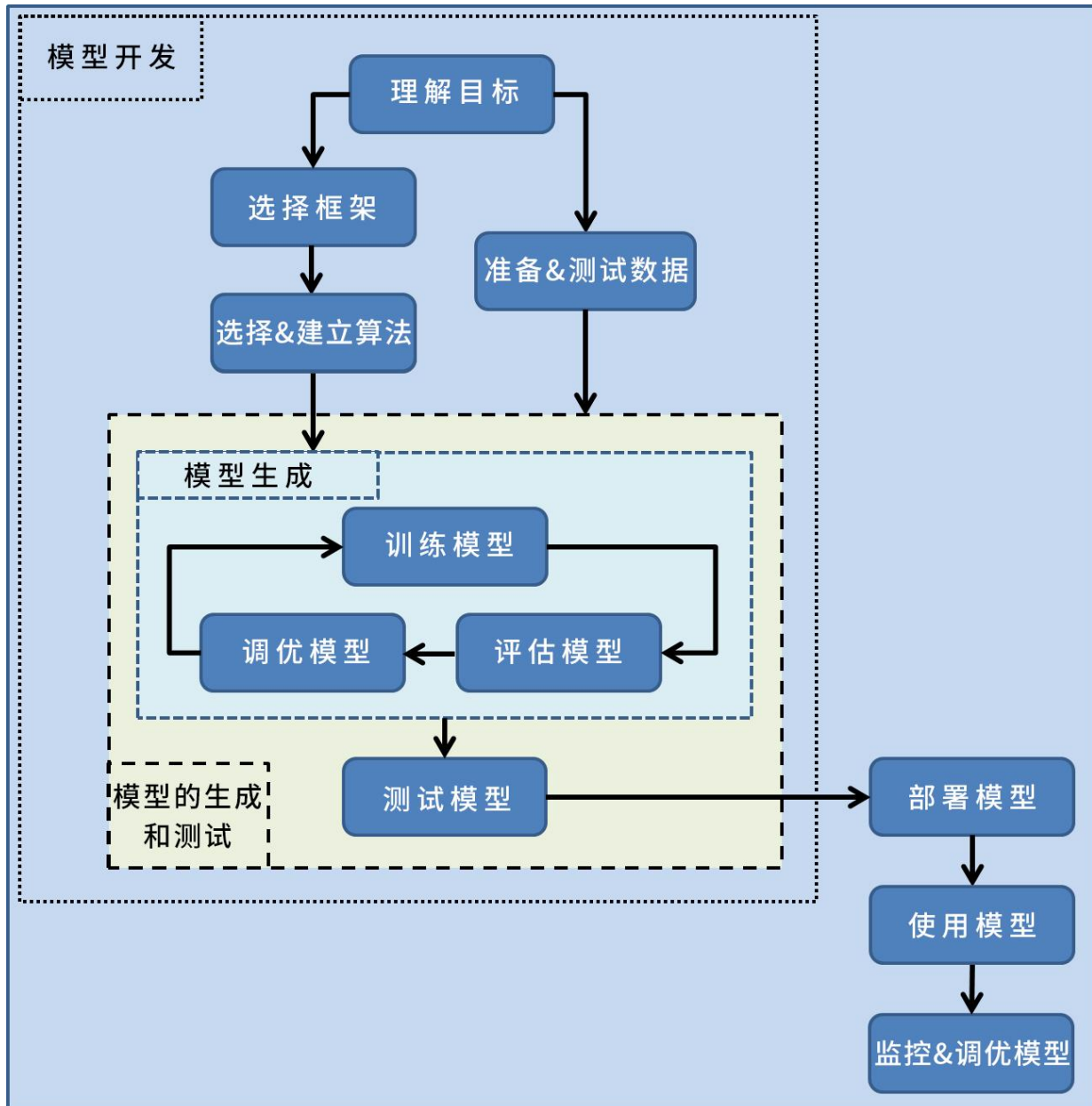


图 1: workflow

使用模型

一旦部署，该模型通常是会成为更大的人工智能系统的一部分，并用于操作。模型可以按设定的时间间隔执行预定的批量预测，也可以根据请求实时运行。

监控和调优模型

当模型正在被使用时，它的情况可能发生变化、模型可能偏移其预期绩效(参见 2.3 和 7.6 节)。为了确保任何漂移都被识别和管理，应该根据其接受准则定期评估操作模型。

可能被认为有必要更新模型设置以解决漂移问题，或者可能决定需要使用新数据进行重新训练，以创建更准确或更稳健的模型。在这种情况下，可以创建一个新模型并用更新的训练数据进行训练。

然后可以使用 A/B 测试的形式将新模型与现有模型进行比较(参见 9.4 节)。

图 1 所示的机器学习工作流程是一个逻辑序列。在实践中, 工作流是以一种重复迭代的方式应用(例如, 当模型被评估时, 经常需要返回到训练步骤, 有时还需要返回到数据准备)。

图 1 所示的步骤不包括机器学习模型与整个系统的非机器学习部分的集成。通常机器学习模型不能单独部署, 需要与非机器学习部件集成。例如, 在视觉应用程序中, 有一个数据管道, 它在将数据提交到机器学习模型之前清理和修改数据。如果模型是成为更大的基于人工智能的系统的一部分, 则需要在部署之前将其集成到该系统中。在这种情况下, 可以执行集成、系统和验收测试级别, 如 7.2 节所述。

3.3 选择一种机器学习的形式

当需要选择合适的机器学习方法时, 以下指南可以应用:

- 对于所选的机器学习方法, 应该有足够的训练和测试数据可用。
- 对于有监督学习, 必须有适当的标注数据。
- 如果有输出标注, 可能是有监督学习。
- 如果输出是离散的和分类的, 可能是分类。
- 本质上如果输出是数值并且是持续的, 可能是回归。
- 如果在给定的数据集条件下没有输出, 可能是无监督学习。
- 如果问题涉及到对类似数据进行分组, 可能是聚类。
- 如果问题涉及查找共存的数据项, 可能是关联。
- 强化学习更好地适用于有与环境互动的上下文中。
- 如果问题涉及多个状态的概念, 并涉及在每个状态下的决策, 那么强化学习可能是适用的。

3.4 选择机器学习算法涉及的因素

没有确定的办法来选择最佳的机器学习算法、机器学习模型设置和机器学习模型超参数。在实践中, 这一组是基于以下因素的混合选择:

- 必须功能(例如功能是分类或者离散值的预测)。
- 需求的质量特性, 比如:
 - 准确度(例如一些模型可能更准确, 但较慢)。

- 可用内存的约束(例如对于嵌入式系统)。
- 训练(和再训练) 模型的速度。
- 预测的速度(例如对于实时操作系统)。
- 透明性、可理解性和可解释性需求。
- 训练模型的可用数据类型(例如一些模型可能只用于图像数据工作)。
- 训练和测试模型的可用数据量(例如有些模型可能倾向于更大程度对有限的数据量进行过拟合)。
- 模型预期使用的输入数据中的特征数量(其他因素，例如速度和准确度，可能直接受到特征数量的影响)。
- 用于聚类的预期类数（例如，某些模型可能不适合具有多个类的问题）。
- 以往的经验。
- 反复试验。

3.5 过度拟合与欠拟合

3.5.1 过度拟合

当模型太接近于一组数据点而不能正确地概括时，就会发生过拟合。这种模型与用于训练它的数据配合得很好，但很难为新数据提供准确的预测。当模型试图拟合每一个数据点时，包括那些可能被描述为噪声或异常值的数据点，就会发生过拟合。当训练集中提供的数据不足时，也可能发生这种情况。

3.5.2 欠拟合

欠拟合发生在模型不够复杂，不能准确地拟合训练数据中的模式。欠拟合模型往往过于简单，难以以为新数据与训练数据非常相似的数据提供准确的预测。欠拟合的一个原因可能是训练集不包含反映输入和输出之间重要关系的特征。当算法不能正确地匹配数据时也会发生这种情况(例如，为非线性数据创建线性模型)。

3.5.3 动手练习：演示过度拟合与欠拟合

演示模型的过拟合和欠拟合的概念。这可以通过使用包含非常少的数据集(过拟合)和特征相关性较差的数据集(欠拟合)来证明。

4. 机器学习-数据-230 分钟

关键词

无

人工智能专用关键词

注释、增强、分类模型、标注数据、数据准备、机器学习训练数据、有监督学习、测试集、验证集

第四章 学习目标：

4.1 数据准备是机器学习 workflow 的一部分

- AI-4.1.1 (K2) 描述与数据准备相关的活动和挑战。
- HO-4.1.1 (H2) 执行数据准备以支持创建机器学习模型。

4.2 workflow 中的训练、验证和测试集

- AI-4.2.1 (K2) 对比训练集、验证集和测试集在开发机器学习模式中的使用。
- HO-4.2.1 (H2) 识别训练集和测试集，并创建一个机器学习模型。

4.3 数据集质量问题

- AI-4.3.1 (K2) 描述典型的数据集质量问题。

4.4 数据质量对机器学习模型的影响

- AI-4.4.1 (K2) 认识差的数据质量会如何导致由此产生的机器学习模型出现问题。

4.5 有监督学习的数据标注

- AI-4.5.1 (K1) 回顾在有监督学习中对数据集中的数据进行标注的不同方法。
- AI-4.5.2 (K1) 回顾数据集中的数据被错误标注的原因。

4.1 数据准备是机器学习 workflow 的一部分

数据准备工作量平均占机器学习 workflow 工作量的 43%，并且可能是机器学习 workflow 中资源密集程度最高的活动。相比之下，模型选择和建立仅使用 17% [R17]。数据准备是数据管道的一部分，它接收原始数据并以一种既可用于训练机器学习模型又可用于由训练过的机器学习模型进行预测的形式输出数据。

数据准备可被认为包括下列活动：

数据收集

- **识别：**识别用于训练和预测的数据类型。例如对于自动驾驶汽车，它可以包括识别对雷达、视频和激光成像、探测和测距（激光雷达）数据的需求。
- **聚集：**确定数据的来源和收集数据的方法。例如这可能包括确定国际货币基金组织 (IMF) 作为金融数据的来源，以及将用于将数据提交到基于人工智能的系统的渠道。
- **标注：**参见 4.5。

获取的数据可以是多种形式（例如数字、分类、图像、表格、文本、时间序列、传感器、地理空间、视频和音频）。

数据预处理

- **清除：**当发现不正确的数据、重复的数据或异常值时，删除或纠正它们。此外，数据缺失处理可以用估计或猜测的值替换缺失的数据值（例如，使用均值、中值和模式值）。个人信息的删除或匿名化也可能被执行。
- **转换：**改变已知数据的格式（例如，将保存为一串字符的地址分解为其组成部分，删除包含随机标识符的字段，将分类数据转换为数值型数据，更改图像格式）。应用于数值型数据的一些转换包括缩放，以确保使用相同的范围。例如，重新调优数据以确保其均值为 0，标准偏差为 1。这种标准化确保数据的范围在 0 到 1 之间。
- **增强：**这用于增加数据集中的样本数量。增强也可以用于在训练数据中包含对抗的示例，提供对抗攻击的稳健性（见 9.1）。
- **抽样：**这涉及到选择整个可用数据集的某些部分，以便观察较大数据集中的模式。这样做通常是为了降低成本和创建机器学习模型所需的时间。

注意所有预处理会带来一种风险，它有可能改变有用的有效数据或添加无效数据。

特征工程

- **特征选择：**特征是反映在数据中的属性/性质。特征选择涉及到选择那些最有可能有助于模型训练和预测的特征。在实践中，它经常包括那些不期望（或不希望）对最终模型有任何影响的

移除特征。通过去除不相关信息(噪声)，特征选择可以减少整体训练次数，防止过拟合(参见 3.5.1 节)，增强精度，使模型更加可归纳。

- 特征提取：这涉及从现有特征中提取信息性和不重复特征。生成的数据集通常较小，可用于以更便宜和更快速的方式产生具有同等准确度的机器学习模型。

在这些数据准备活动的同时，通常还进行探索性数据分析(EDA)，以支持整个数据准备任务。这包括执行数据分析以发现数据中固有的趋势，并通过在数据中绘制趋势，使用数据可视化以可视化格式表示数据。

虽然上面的数据准备活动和子活动已经按逻辑顺序显示，但不同的项目可能会重新排序它们或只使用它们的一个子集。有些数据准备步骤，如识别数据源，只执行一次和可以手动执行，其他步骤可能是成为数据管道的一部分，通常影响实时数据，这些任务应该被自动化实施。

4.1.1 数据准备中的挑战

与数据准备相关的一些挑战包括：

- 对知识的需要：
 - 应用程序域。
 - 数据及其属性。
 - 与数据准备相关的各种技术。
- 从多个来源获得高质量数据的困难。
- 自动化数据管道的难度，以及确保生产数据管道既可扩展又具有合理的性能效率(例如完成数据项处理所需的时间)。
- 与数据准备相关的成本。
- 没有对在数据准备期间引入到数据管道中的缺陷进行足够优先的检查。
- 引入样本偏差(参见 2.4)。

4.1.2 动手练习：机器学习的数据准备

对于一组给定的原始数据，执行 4.1 节中概述适用的数据准备步骤，生成一个数据集，该数据集将用于使用有监督学习创建分类模型。

这个活动是创建机器学习模型的第一步，该模型将用于以后的练习。

为了完成这项活动，学生们将被提供适当的(和特定的语言)材料，包括：

- 库。
- 机器学习框架。
- 工具。
- 开发环境。

4.2 workflows 中的训练、验证和测试集

逻辑上，开发机器学习模型需要三组等效数据（即从单个初始数据集中随机选择的数据）：

- 训练数据集，用于训练模型。
- 验证数据集，用于评估和随后调优模型。
- 测试数据集（也称为保持数据集），用于测试调优的模型。

如果有无限可用的适宜数据，机器学习 workflow 中用于训练、评估和测试的数据量通常取决于以下因素：

- 用于训练模型的算法。
- 资源的可用性，如内存、磁盘空间、计算能力、网络带宽和可用时间。

在实践中，由于获取足够的合适数据的挑战，训练和验证数据集通常来自单一的组合数据集。测试集是分开的，在训练期间不使用。这是为了确保所开发的模型不受测试数据的影响，从而测试结果能够真实地反映模型的质量。

对于将合并的数据集分割成三个单独的数据集，没有最佳的比率，但可以作为指导原则的典型比率范围从 60:20:20 到 80:10:10（训练:验证:测试）。除非数据集很小，或者有生成的数据集不能代表预期的操作数据的风险，将数据分解为这些数据集通常是随机进行的。

如果可用的数据有限，再将可用的数据分成三个数据集，那么可能会导致没有足够的数据来进行有效的训练。为了克服这个问题，训练和验证数据集可以合并（保持测试数据集独立），然后用该数据集创建多个分割组合（例如，80%的训练/ 20%的验证）。再将数据随机分配到训练集和验证集。训练、验证和调优使用这些多个分组合来创建多个调优模型，并且可以将整个模型的性能计算为所有运行的平均值。有各种的方法用于创建多个分割组合，包括分割测试、bootstrap、K-折交叉验证和保留一个交叉验证（参见[B02]了解更多细节）。

4.2.1 动手练习：识别训练和测试数据，并创建机器学习模型

将之前准备好的数据（参见 4.1.2）分解为训练、验证和测试数据集。

应用这些数据通过有监督学习来训练和测试一个分类模型。

通过比较验证和测试数据集获得的精确性，解释评估/调优和测试之间的差异。

4.3 数据集质量问题

在数据集中数据相关的典型质量问题，包括但不限于下表所示：

质量方面	说明
错误数据	捕获的数据不正确(例如，通过错误的传感器)或输入错误(例如，复制-粘贴错误)。
不完整数据	数据值可能会丢失(例如，记录中的某个字段可能是空的，或者某个特定时间间隔的数据可能被省略)。导致数据不完整的原因有很多，包括安全问题、硬件问题和人为错误。
错误标注数据	有几个可能导致数据标注错误的原因(见 4.5.2 节)。
不充分的数据	可供使用的学习算法识别数据模式不充分(注意，所需的最小数据量会因不同的算法而不同)。
数据没有预处理	数据应该经过预处理，以确保数据是干净的，格式一致，不包含不需要的异常值(见 4.1 节)。
过时数据	用于学习和预测的数据应尽可能是最新的(例如，使用几年前的财务数据很可能导致不准确的结果)。
不均衡数据	均衡的数据可能是由不适当的偏差造成的(例如，基于种族、性别或民族)、传感器放置不当(例如，将面部识别摄像机放置在天花板高度)、数据集可用性的差异以及数据供应商的不同动机可能导致数据不平衡。
不公平数据	公平是一种主观但可被识别的质量特性。例如，为了支持多样性或性别平衡，选择的数据可能对少数群体或弱势群体有积极的偏向(注意，这些数据可能被认为是公平的，但有可能不平衡)。
重复数据	重复的数据记录可能会对生成的机器学习模型产生不适当的影响。
无关数据	与所处理的问题无关的数据可能对结果产生不利影响，并可能导致资源浪费。
隐私问题	使用任何数据都应尊重相关的数据隐私法律(例如，欧盟关于个人信息的通用数据保护条例)。
安全问题	将欺骗性数据或误导性数据故意插入训练数据，可能会导致训练模型的不准确性。

4.4 数据质量及其对机器学习模型的影响

机器学习模型的质量高度依赖于创建它的数据集的质量，低质量的数据会导致产生有缺陷的模型和有缺陷的预测。

以下几类缺陷是由数据质量问题导致的：

- 准确性降低：这些缺陷是由数据错误、数据不完整、数据错误标注、数据不充分、数据过时、不相关数据和未进行预处理的数据造成的。例如如果这些数据被用来建立一个预期房价的模型，但是训练数据很少或没有包含带阳光房的独栋房屋的数据，那么对于这个特定类型的住宅的预测可能是不准确的。
- 偏差模型：这些缺陷是由于数据不完整、数据不平衡、数据不公平、数据缺乏多样性或重复等原因。例如，如果缺失来自于一个特性的数据（例如所有用于疾病预测的医学数据都是从特定性别的受试者那里收集的），那么这很可能对合成模型有不利影响（除非该模型仅仅是用于该性别预测操作）。
- 受损模型：这些缺陷是由于数据隐私和安全限制造成的。例如，数据中的隐私问题可能导致安全漏洞，这将使攻击者能够通过反向工程从模型中获得信息并可能导致个人信息的泄漏。

4.5 有监督学习的数据标注

数据标注是通过添加标注丰富未标注（或不良标注）数据，因此它适合于有监督学习。数据标注是一种资源密集型活动，根据报告，在机器学习项目中平均 25% 的时间用于数据标注上 [B11]。

在最简单的形式中，数据标注是根据他们的种类将包括图像或文本文件放在不同的文件夹中。例如，将所有产品正面评论的文本文件放在一个文件夹中，所有负面评论的文本文件放在另一个文件夹中。通过在图像周围绘制矩形来标注图像中的对象是另一种常见的标注技术，通常称为注释。可能需要更复杂的注释来标注 3D 对象或在不规则对象周围绘制边界框。数据标注和注释通常由相关工具支持。

4.5.1 数据标注方式

可以通过多种方式进行标注：

- 内部：标注工作由开发人员、测试人员或组织内为标注而建立的团队执行。
- 外包：标注由外部专家组织完成。
- 众包：标注由众多的独立个体完成。由于管理标注质量困难，可能会要求若干个注释人员对相同的数据进行标注，然后对要使用的标注做出决定。

- 人工智能辅助：基于人工智能的工具用于识别和注释数据或聚类类似的数据。然后，作为两步过程的一部分，是由人工来确认或补充(例如，修改边界框)结果的。
- 混合：可以使用上述标注方法的组合。例如，众包标注通常由外部组织管理，该组织可以使用专门基于人工智能的众包管理工具。

在适用的情况下，可以重用预先标注的数据集，因此完全避免了数据标注的需要。许多这样的数据集是公开的，例如，来自 Kaggle[R16]。

4.5.2 数据集中的错误标注数据

有监督学习时假设数据被数据注解者正确标注。然而，在实践中很难正确标注数据集中的所有项。数据标错的原因如下：

- 注释员的随机错误(例如，按错按钮)。
- 可能会出现系统性错误(例如，标注员得到了错误的指示或缺乏训练)。
- 恶意数据注释员可能会故意出错。
- 翻译错误可能将在某一种语言中原来正确标注，在使用另一种语言时标注错误。
- 如果对于选项的解释是开放性的，数据注释者的主观判断可能会导致来自不同注释者的数据标注发生冲突。
- 缺乏必需的领域知识可能导致错误的标注。
- 复杂的分类任务可能会导致更多的错误。
- 用于支持数据标注的工具具有缺陷，导致错误的标注。
- 基于机器学习的标注方法是概率性的，这可能导致一些不正确的标注。

5. 机器学习功能表现度量-120 分钟

关键词

无

人工智能特定关键词

正确性，曲线下面积，混淆矩阵，F1-分数，簇间度量，簇内度量，均方误差，机器学习基准套件，机器学习功能表现度量，精准度，受试者工作特定曲线，回归模型，拟合度，轮廓系数

第五章学习目标

5.1 混淆矩阵

AI-5.1.1 (K3) 从给定的混淆矩阵数据集中计算机器学习函数功能表现度量。

5.2 用于分类、回归和聚类的附加机器学习功能表现度量

AI-5.2.1 (K2) 对比和比较在分类、回归和聚类方法的机器学习功能表现度量背后的概念。

5.3 机器学习功能表现度量的局限性

AI-5.3.1 (K2) 总结使用机器学习功能表现度量来确定机器学习系统质量的局限性。

5.4 选择机器学习功能表现度量

AI-5.4.1 (K4) 为给定的机器学习模型和场景选择合适的机器学习功能表现度量和/或它们的值。

HO-5.4.1 (H2) 使用选定的机器学习功能表现度量评估创建的机器学习模型。

5.5 机器模型基准套件

AI-5.5.1 (K2) 机器学习上下文中基准测试套件的使用解释。

5.1 混淆矩阵

在分类问题中，模型很少能一直正确地预测结果。对于任何这样的问题，可以用下列可能性创建一个混淆矩阵：

		实际结果	
		阳性	阴性
预测结果	阳性	真阳性	假阳性
	阴性	假阴性 (FN)	真阴性 (TN)

图 2 混淆矩阵

请注意，图 2 中所示的混淆矩阵可能以不同的方式呈现，但总是会生成四种可能情况的值：真阳性 (TP)、真阴性 (TN)、假阳性 (FP) 和假阴性 (FN)。

根据混淆矩阵，定义了以下度量：

- 正确性。

$$\text{正确性} = (\text{真阳性} + \text{真阴性}) / (\text{真阳性} + \text{真阴性} + \text{假阳性} + \text{假阴性}) * 100\%。$$

正确性测量所有正确分类的百分比。

- 精准率（查准率）。

$$\text{精准率} = \text{真阳性} / (\text{真阳性} + \text{假阳性}) * 100\%。$$

精准率测量正确预测阳性的比例。它衡量的是一种对阳性预测的确信程度。

- 调用召回率（查全率）。

$$\text{调用召回率} = \text{真阳性} / (\text{真阳性} + \text{假阴性}) * 100\%。$$

调用召回率（也称为灵敏度）测量的是预测正确的真阳性的比例。这是一种衡量有多确信不会错过任阳性的指标。

- F1-分数。

$$\text{F1-分数} = 2 * (\text{精准度} * \text{召回率}) / (\text{精准度} + \text{召回率})。$$

F1 分数作为精准率和召回率的调和平均值。它的值在 0 到 1 之间。接近 1 的分数表示假数据对结果的影响很小。F1 分数低表示该模型检测阳性的能力较差。

5.2 附加机器学习功能表现度量的分类、回归和聚类

对于不同类型的机器学习问题，有许多度量(除了 5.1 节中描述的分类相关指标之外)。下面描述了一些最常用的度量。

监督分类度量

- 接受者操作特性曲线（ROC）是一个图示，说明了当二值分类器的鉴别阈值变化时的能力。这种方法最初是为军事雷达而开发的，这就是它被命名的原因。接受者操作特性曲线以真阳性率(TPR) (也称为召回率)与假阳性率($FPR = FP / (TN + FP)$)绘制，真阳性率在 y 轴，假阳性率在 x 轴。
- 曲线下面积(AUC)是接受者操作特性曲线（ROC）下的面积。它表示分类器的可分离程度，显示模型区分类的良好程度。曲线下面积越大，模型的预测越好。

监督回归度量

对于监督回归模型，度量表示回归线与实际数据点的吻合程度。

- 均方误差是实际值与预测值之间的平方差的平均值。均方误差的值总是正的，越接近于 0 表示回归模型越好。通过取差的平方，它确保了正误差和负误差不会相互抵消。
- R 方(也称为决定系数)是衡量回归模型与被解释变量吻合程度的指标。

无监督聚类度量

对于无监督的聚类，有几个度量表示不同聚类之间的距离和给定聚类中数据点的紧致度。

- 簇内指标度量簇内数据点的相似性。
- 簇间指标度量不同簇中数据点的相似性。
- 轮廓系数(Silhouette Coefficient，也称为轮廓分数)是基于平均簇间和簇内距离的一种度量(在-1 和+1 之间)。得分为+1 说明聚类分离良好，得分为 0 表示随机聚类，得分为-1 表示聚类分配错误。

5.3 机器学习功能表现度量的局限性

机器学习功能表现度量仅限于度量模型的功能，如正确性、精准率、召回率、均方误差、曲线下面积(AUC)和轮廓系数(Silhouette Coefficient)。它们不测量其他非功能性质量特征，如 ISO 25010 [S06]中定义的特征(例如，性能效率)和第 2 章中描述的特征(例如，可解释性、灵活性和自主性)。在本教学大纲中，使用术语“机器学习功能表现度量”是因为广泛使用术语“表现度量”来指代这些功能度量。添加“机器学习函数”强调这些度量是特定于机器学习的，与性能效率度量没有关系。

机器学习功能表现度量受到其他几个因素的限制：

- 对于有监督学习，机器学习函数功能表现度量是在标注数据的基础上计算的，结果度量的准确性取决于正确的标注(参见 4.5 节)。
- 用于测量的数据可能不具有代表性(例如，数据存在偏差)，生成的机器学习功能表现度量依赖于该数据(参见第 2.4 节)。
- 系统可能包含多个组件，但机器学习功能表现度量仅适用于机器学习模型。例如，机器学习功能表现度量不会考虑数据管道来评估模型。
- 大多数机器学习功能表现度量只能在工具支持下进行测量。

5.4 选择机器学习功能表现度量

通常不可能构建一个机器学习模型来获得由混淆矩阵生成所有机器学习功能表现度量的最高分。相反，根据模型的预期使用情况，选择最合适的机器学习功能表现度量作为接受准则(例如，为了最小化假阳性的数量，需要较高的精确率，而为了最小化假阴性的数量，召回率需要较高)。在选择第 5.1 节和 5.2 节中描述的机器学习功能表现度量时，可以使用以下标准：

- 正确性：如果数据集是对称的(例如，假阳性和假阴性计数和成本相似)，则该度量可能适用。如果一类数据优于其他数据，那么这个度量就会成为一个糟糕的选择，在这种情况下就应该考虑 F1 分数。
- 精准率：当假阳性的成本很高，对阳性结果的信任度也需要很高时，这可能是一个合适的度量指标。垃圾邮件过滤器(将电子邮件分类为垃圾邮件被认为是积极的)是一个需要高精准率的例子，因为将太多并非垃圾邮件的电子邮件放入垃圾邮件文件夹对大多数用户来说是不可接受的。当分类器处理的情况中有很高比例的情况是阳性的，那么仅使用精准度不太可能是一个好的选择。
- 召回率：当不被漏掉阳性结果非常重要时，高分数的召回率就很重要。例如，在癌症检测中遗漏任何真正的阳性结果并将其标注为阴性(即未检测到癌症)可能是不可接受的。
- F1-分数 - F1-分数最有用的是当预期的分类不均衡的时候，与当精准率和召回率是同等重要的时候。

除了上述度量之外，在第 5.2 节中还描述了几个度量。这些可能适用于给定的机器学习问题，例如：

- 接受者操作特定曲线的曲线（ROC）下面积可用于监督分类问题。
- 均方误差和决定系数可用于监督回归问题。
- 簇间度量、簇内度量和轮廓系数（Silhouette Coefficient）可用于无监督聚类问题。

5.4.1 动手练习:评估创建的机器学习模型

使用前面练习中训练的分类模型，计算并显示正确性、精准率、召回率和 F1 分数的值。在合适的情况下，使用开发框架提供的库函数来执行计算。

5.5 基准套件

新的人工智能技术，如新的数据集、算法、模型和硬件，会定期发布，因此很难确定每项新技术的相对有效性。

为了提供这些不同技术之间的客观比较，可以使用行业标准的机器学习基准套件。这些涵盖了广泛的应用领域，并为人工智能和机器学习性能提供了评估硬件平台、软件框架和云平台的工具。

机器学习基准套件可以提供各种度量，包括训练时间（例如，框架使用定义的训练数据集训练 1 模型到指定的目标质量度量的速度，例如 75% 的正确率）和推理时间（例如，训练的机器学习模型执行推理的速度）。

机器学习基准测试套件由几个不同的组织提供，如：

- MLCommons [R18]：这是一个成立于 2020 年的非营利组织，之前命名为机器学习绩效，为软件框架、人工智能特定处理器和机器学习云平台提供基准。
- DAWNBench [R19]：这是一个来自斯坦福大学的机器学习基准测试套件。
- MLMark [R20]：这是一个机器学习基准测试套件，旨在度量来自嵌入式微处理器的基准测试联盟的嵌入式推理的性能和准确性。

6. 机器学习-神经网络和测试-65 分钟

关键词

无

人工智能特定关键词

激活值，深度神经网络（DNN），机器学习训练数据，多层感知器，神经网络，神经元覆盖率，感知器，符号变更覆盖率，符号-符号覆盖率，有监督学习，阈值覆盖率，训练数据，值变更覆盖率

第 6 章 学习目标：

6.1 神经网络

- | | | |
|----------|------|----------------------|
| AI-6.1.1 | (K2) | 解释包括 DNN 神经网络的结构和功能。 |
| HO-6.1.1 | (H1) | 体验感知器的实现。 |

6.2 神经网络覆盖度量

- | | | |
|----------|------|-----------------|
| AI-6.2.1 | (K2) | 描述对神经网络不同的覆盖度量。 |
|----------|------|-----------------|

6.1 神经网络

人工神经网络最初旨在模仿人脑的功能。我们考虑人脑，它被认为可以想到尽可能多的相连的生物神经元。单层感知器是实施人工神经网络的第一类示例之一，它由一层神经网络即一个神经元组成。它可用于对分类器进行监督训练，从而决定输入是否属于一种特定类。

现在大多数的神经网络被认为是深度神经网络，因为它们由多个层组成，可以被视为多层感知器（见图 3）。

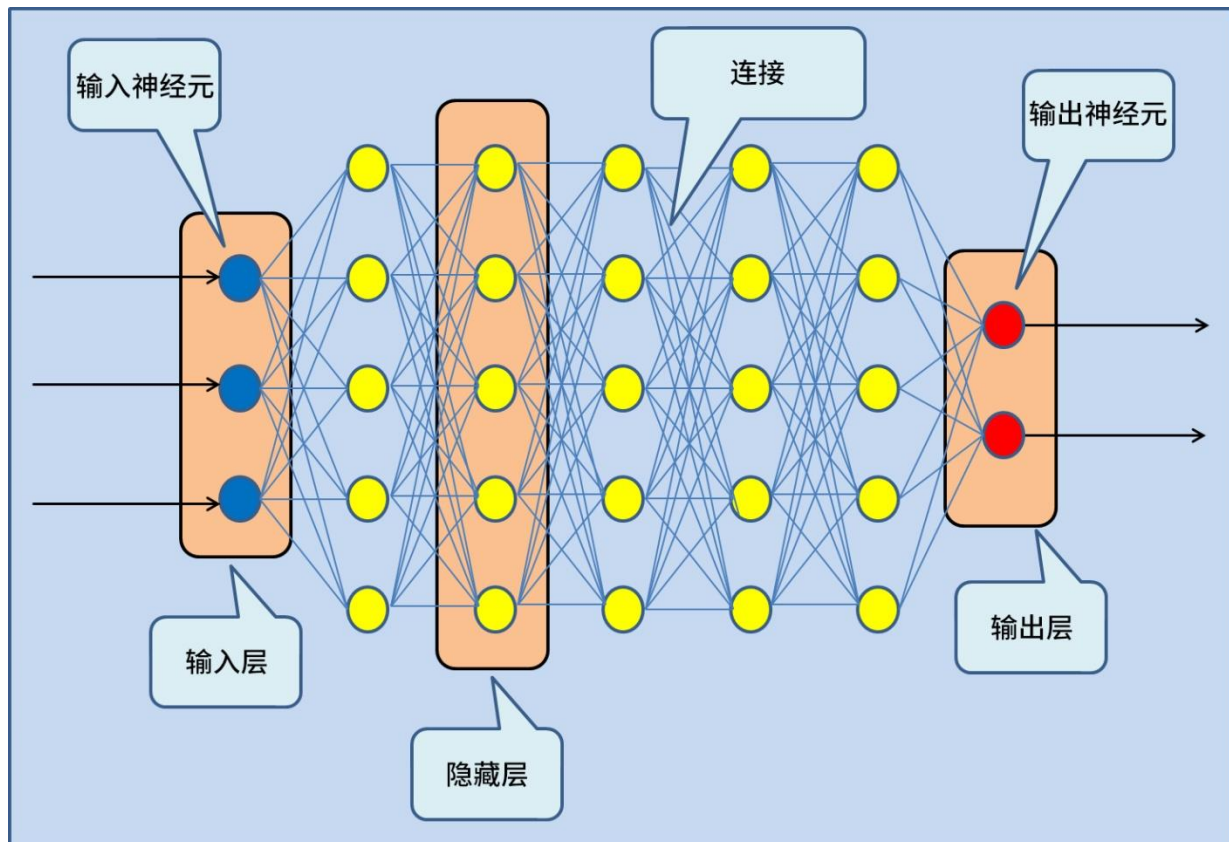


图 3 深度神经网络结构

一个深度神经网络由三种类型的层组成。输入层接收输入，例如来自摄像机的像素值。输出层为外部世界提供结果。例如输出可能表示输入图像是猫的可能性的值。输入层和输出层之间是隐藏层，由人工神经元组成，也被称为节点。一层中的神经元与下一层的每个神经元相连，每个连续层的神经元数量可能不同。神经元执行计算并将信息从输入神经元传递到输出神经元。

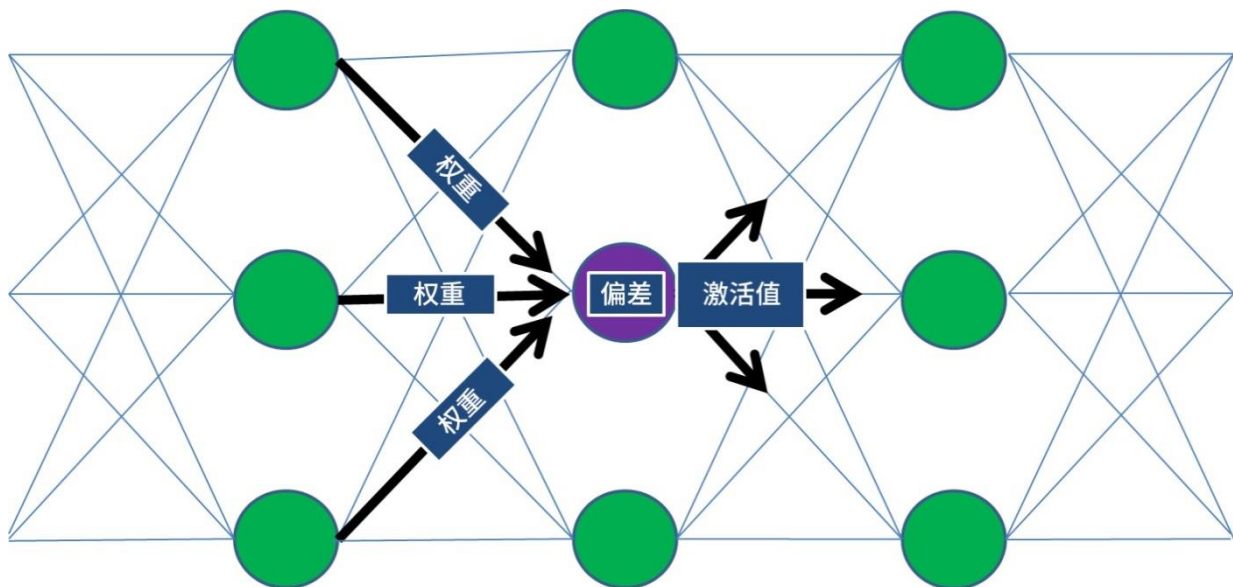


图 4 每个神经元的计算

如图 4 所示，每个神经元（输入层中的那些除外）执行的计算生成激活值。此值的计算方法是运行一个公式（激活函数），该公式从上一层的所有神经元接收激活值、分配给神经元之间的连接权重（这些权重随着网络学习而变化）和每个神经元的个体偏差。请注意，此偏差是预设的常数值，与第 2.4 节早期考虑的偏差无关。运行不同的激活函数可能导致计算不同的激励值。这些值通常以零为中心，范围介于 -1（意味着神经元“不感兴趣”）和 +1（意味着神经元“非常感兴趣”）之间。

训练神经网络时，每个神经元预设偏差值，训练数据通过网络传递，每个神经元运行激活函数，最终产生输出。然后将生成的输出与已知的正确结果进行比较（在有监督学习的示例中使用标注数据）。然后，通过网络反馈实际输出和已知正确结果之间的差值，以修改神经元连接上的权重值，以最大限度地减少这种差异。随着更多的训练数据通过网络输入，权重会随着网络学习而逐渐调优。最终，产出被认为足够好，已经可以结束训练。

6.1.1 动手练习：实现简单的感知器

学生将通过一个练习来演示感知器学习一个简单的函数，如“And”函数。

练习应该涵盖感知器如何通过修改多个时期的权重来学习，直到误差降为零。此活动可用各种机制（例如电子表格，模拟）。

6.2 神经网络覆盖度量

实现白盒测试覆盖准则（例如语句，分支，改进的条件/判定覆盖（MC/DC）[I01]）是必需的，以符合一些安全相关标准[S07]使用传统的必修源代码，并建议许多测试从业者用于其他关键应用。监测和提高覆盖范围支持新测试案例的设计，从而增强了对测试对象的信心。

使用此类度量项测量神经网络的覆盖几乎没有什么价值，因为每次执行神经网络时，都会运行相同的代码。相反，根据神经网络本身的结构覆盖来实施覆盖度量，更具体地说，根据神经网络内的神经元。大多数度量项基于神经元的激励值。

神经网络覆盖是一个新的研究领域。学术论文自 2017 年才发表，因此，几乎没有客观证据（例如，可重复的研究结果）表明建议的措施是有效的。然而，应当指出，尽管声明和决策覆盖已经使用了 50 多年，但几乎没有客观证据证明其相对有效性，尽管它们已被授权用于测量与安全有关的应用（如医疗设备和航空电子设备系统）软件的覆盖。

研究人员已提出以下神经网络覆盖准则，并应用于各种应用：

- **神经元覆盖率：**全神经元覆盖率要求神经网络中的每个神经元实现大于零 [B12] 的激励值。这在实践中很容易实现，研究表明，在各种深度神经网络上，几乎 100% 的覆盖率是通过很少的测试用例来实现的。当无法实现此覆盖措施时，该覆盖度量可能最有用，因为它可以作为报警信号。
- **阈值覆盖率：**全阈值覆盖率要求神经网络中的每个神经元实现大于指定阈值的激活值。创建 DeepXplore 框架的研究人员实际上建议，神经元覆盖范围应该根据超过阈值的激活值来度量，而阈值会根据情况而改变。当他们报告使用这种白盒方法有效地发现（的结果）数千种不正确的极端情况行为时，他们以 0.75 作为阈值进行研究。这种类型的覆盖已经被重命名，以更容易地区分它与阈值为零的神经元覆盖率，因为其他一些研究人员在使用术语“神经元覆盖率”的时候，它的意思是神经元覆盖率的阈值为零。
- **符号变更覆盖率：**要实现完整的符号变更覆盖率，测试案例需要使每个神经元同时实现正激活值和负激活值 [B13]。
- **值变更覆盖率：**要实现完整的值变更覆盖率，测试案例需要使每个神经元实现两个激活值，其中两个值之间的差值超过某些选定的值 [B13]。
- **符号-符号覆盖率：**此覆盖范围考虑相邻层中的神经元对及其激活值所采取的标志。要考虑覆盖一对神经元，测试案例需要表明，在第一层更改神经元符号会导致第二层的神经元改变其符号，而第二层所有其他神经元信号保持不变 [B13]。这是一个强制源代码条件/判定覆盖的概念。

研究人员报告了基于层的进一步覆盖度量（尽管比信号-信号覆盖简单），在 TensorFuzz 工具 [B14] 中实现了使用最近邻算法识别相邻神经元组有意义变化的成功方法。

7. 基于人工智能系统的测试概述-115 分钟

关键词

输入数据测试，机器学习模型测试

人工智能特定关键词

人工智能组件、自动化偏差、大数据、概念漂移、数据管道、机器学习功能表现度量、训练数据

第七章学习目标：

7.1 基于人工智能系统的规范

AI-7.1.1 (K2) 解释基于人工智能系统的系统规范如何在测试中带来挑战。

7.2 基于人工智能系统的测试级别

AI-7.2.1 (K2) 描述如何在每个测试级别测试基于人工智能的系统。

7.3 测试基于人工智能系统的测试数据

AI-7.3.1 (K1) 回顾那些与测试数据相关的因素，这些因素可能会使基于人工智能系统的测试变得困难。

7.4 基于人工智能系统的自动化偏差测试

AI-7.4.1 (K2) 解释自动化偏差以及其如何影响测试。

7.5 记录机器学习模型

AI-7.5.1 (K2) 描述人工智能组件的文档，理解文档如何支持基于人工智能系统的测试。

7.6 概念漂移的测试

AI-7.6.1 (K2) 解释频繁地测试训练模型用以处理概念漂移的必要性。

7.7 为机器学习系统选择测试方法

AI-7.7.1 (K4) 对于给定的场景，确定开发机器学习系统时要遵循的测试方法。

7.1 基于人工智能系统的规范

系统需求和设计规范对基于人工智能系统和传统系统一样重要。这些规范为测试人员检查实际的系统行为是否与指定的需求一致提供了基础。然而，如果规范不完整并且缺乏可测试性，这就会引入一个预期测试结果的问题(参见 8.7 节)。

基于人工智能系统的规范之所以特别具有挑战性，有以下几个原因：

- 在许多基于人工智能系统的项目中，只根据高级业务目标和所需的预测来指定需求。原因之一是基于人工智能系统开发具有探索性。通常，基于人工智能的系统项目从数据集开始，目标是确定可以从该数据中获得哪些预测。这与在常规项目开始时指定所需的逻辑相反。
- 在独立测试的结果出来之前，基于人工智能系统的准确性通常是未知的。伴随着探索性开发方法，这通常会导致规范不充分，因为在确定所需的验收准则时，实施已经在进行中。
- 许多基于人工智能系统的概率特性使得有必要为某些预期的质量要求指定公差，例如预测的正确性。
- 如果系统目标要求复制人的行为，而不是提供特定的功能，这通常会导致基于系统与它所替换的人类活动几乎一样，或者比它所替换的人类活动更好的情况下，对人的行为需求的规定很差。这可能会使定义预期测试结果变得困难，尤其是当它所取代的人员的能力差异很大时。
- 人工智能用于实现用户界面，如通过自然语言识别、计算机视觉或人类与物理交互，系统需要演示更大的灵活性。然而，这种灵活性也会在识别和记录所有可能发生这种交互的不同方式方面带来挑战。
- 基于人工智能系统特有的质量特性，如适应性、灵活性、演化性和自主性，需要考虑并定义为需求规范的一部分(见第 2 章)。这些特性的新颖性可能使它们难以定义和测试。

7.2 基于人工智能系统的测试级别

基于人工智能系统通常包括人工智能和非人工智能组件。非人工智能组件可以使用常规方法进行测试[I01]，而人工智能组件和包含人工智能组件的系统可能需要在某些方面进行不同的测试，如下所述，对于包括人工智能组件测试在内的所有测试级别，得到数据工程师/科学家和领域专家的密切支持是非常重要的。

与传统软件使用测试级别的一个主要区别是，它包含了两个新的专门测试级别，以明确地处理基于人工智能系统中使用的输入数据测试和模型测试[B15]。本节的大部分内容适用于所有基于人工智能的系统，尽管部分内容专门针对机器学习。

7.2.1 输入数据测试

输入数据测试的目的是确保系统用于训练数据和预测数据具有最高质量（见第 4.3 节）。它包括以下内容：

- 评审。
- 统计技术（例如：测试数据的偏差）。
- 训练数据的探索性数据分析。
- 数据管道的静态和动态测试。

数据管道通常包括几个执行数据准备的组件（见第 4.1 节），这些组件的测试包括组件测试和集成测试。用于训练的数据管道可能与用于支持操作预测的数据管道大不相同。对于训练而言，数据管道可以被视为一个原型，而不是操作上使用的完全工程化、自动化版本。由于这个原因，这两个版本的数据管道测试可能非常不同。但是，还应考虑测试两个版本的功能等效性。

7.2.2 机器学习模型测试

机器学习模型测试的目标是确保所选择的模型满足可指定的任何绩效准则。这包括：

- 机器学习功能表现准则（见章节 5.1 和 5.2）。
- 单独适用于模型机器学习非功能接受准则，如训练速度、预测速度、使用的计算资源、适应性和透明度。

机器学习模型测试目的是决定机器学习框架、算法、模型、模型设置和超参数的选择尽可能最优。在适当的情况下，机器学习模型测试还可以包括实现白盒覆盖准则的测试（参见 6.2 节）。所选择的模型随后与包括人工智能和非人工智能的其他组件集成。

7.2.3 组件测试

组件测试是一种传统的测试级别，它适用于任何非模型组件，如用户界面和通信组件。

7.2.4 组件集成测试

组件集成测试是一种传统测试级别，用于确保系统组件（包括人工智能和非人工智能）正确交互。它测试数据管道的输入是否如模型所期望的那样被接收，模型产生的任何预测是否与相关的系统组件（例如，用户界面）交换，并被它们正确地使用。在人工智能作为服务提供的情况下（见 1.7 节），通常将所提供服务的 API 测试作为组件集成测试的一部分。

7.2.5 系统测试

系统测试是一种传统测试级别，用于确保集成组件（包括人工智能和非人工智能）的完整系统在与操作环境密切相关的测试环境中，功能和非功能的角度按预期执行。根据系统的不同，这种测试可能采取在预期的操作环境中进行现场试验或在模拟器中进行测试的形式（例如，如果测试场景在操作环境中是危险的或难以复制的）。

在系统测试期间，将重新测试机器学习功能性性能准则，以确保当模型嵌入到一个完整的系统中时，最初机器学习模型测试的测试结果不会受到负面影响。这种测试在人工智能组件被故意改变的情况下尤其重要（例如，压缩深度神经网络以减小其尺寸）。

系统测试也是测试了系统的许多非功能性需求的测试级别。例如，对抗性测试可以用来测试稳定性，系统可以用来测试可解释性。在适当的情况下，与硬件组件（如传感器）的接口可以作为系统测试的一部分进行测试。

7.2.6 验收测试

验收测试是一种常规的测试级别，用于确定整个系统是否被客户所接受。对于基于人工智能的系统，验收准则的定义是具有挑战性的（见 8.8 节）。如果人工智能是作为服务提供的（见 1.7 节），则可能需要进行验收测试，以确定服务对预期系统的适用性，以及是否已充分达到机器学习功能表现准则。

7.3 测试基于人工智能系统的测试数据

由于依赖于（具体的）情况和被测系统（自身特性），测试数据的获取可能是一个挑战。在为基于人工智能系统处理测试数据时，有几个潜在的挑战，包括：

- 大数据（大容量、高增长率和多样化的数据）难以创建和管理。例如，为一个高速消耗大量图像和音频的系统创建有代表性的测试数据可能是困难的。
- 输入数据可能需要随着时间的推移而改变，特别是当它表示真实世界中的事件时。例如，测试面部识别系统的记录照片可能需要“老化”，以代表人们在现实生活中几年的老化。
- 个人或其他机密数据可能需要特殊的技术进行脱敏、加密或编辑，使用时也可能需要法律批准。
- 当测试人员使用与数据科学家进行数据采集和数据预处理时同样的实施步骤，这些步骤中的缺陷可能会被掩盖。

7.4 人工智能系统的自动化偏差测试

基于人工智能系统的一个类别是帮助人类做决策。然而，人类偶尔会倾向过于相信这些系统，这种错位的信任可以称为自动化偏差或自满偏差，有两种形式。

- 自动化/自满偏差的第一种形式是人类接受系统提供的建议，而没有考虑其他来源(包括他们自己)的输入。例如，在一个过程中，通过使用机器学习预先填充表单，可以改进将关键数据输入表单的人，然后由人验证该数据。事实证明，这种形式的自动化偏差通常会降低决策质量的 5%，也可能会更大，这取决于系统上下文[B16]。同样地，自动更正输入的文本(例如手机短信)也经常出错，可能会改变意思。用户通常不会注意到这一点，也不会修正这个错误。
- 自动化/自满偏差的第二种形式是，由于没有充分监控系统，人类错过了系统故障。例如，半自动驾驶汽车的自动驾驶能力越来越强，但在即将发生事故的情况下，仍要依靠人类来接管。通常情况下，人类车辆乘员会逐渐变得过于相信系统控制车辆的能力，他们开始不太关注，这可能导致他们无法在需要的时候做出适当的反应。

在这两种情况下，测试人员都应该理解人类的决策是如何被妥协的，并且同时测试系统的建议质量和由有代表性的用户提供的相应人工输入的质量。

7.5 记录人工智能组件

人工智能组件文档的典型内容包括：

- 通用：标识符、描述、开发人员详细信息、硬件要求、许可证详细信息、版本、日期和联系点。
- 设计：假设和技术决策。
- 用途：主要和次要用例，典型用户，自学习方法，已知的偏差，道德问题，安全问题，透明度，决策阈值，平台和概念漂移。
- 数据集：功能，收集，可用性，预处理要求，使用，内容，标注，大小，隐私，安全，偏见/公平和限制/约束。
- 测试：测试集数据(描述和可用性)，测试的独立性，测试结果，测试方法的稳定性，可解释性，概念漂移和可移植性。
- 训练和机器学习功能表现：机器学习算法，权重，验证集，机器学习功能表现度量的选择，机器学习功能表现度量的阈值，以及实际的机器学习功能表现度量。

清晰的文档通过提供基于人工智能系统实施的透明度帮助改进测试。文档中对测试非常重要的关

键领域是：

- 系统的目标，以及功能和非功能需求的规格说明。这些类型的文档通常构成测试基础的一部分。
- 架构和设计信息，概述不同的人工智能和非人工智能组件如何交互。这支持了集成测试目标的识别，并为系统结构的白盒测试提供了基础。
- 操作环境的规范。这是测试系统的自主性、灵活性和适应性所需要的。
- 任何输入数据的来源，包括相关的元数据。在测试以下几个方面时需要清楚地了解这一点：
 - 不可靠输入的功能正确性。
 - 显性或隐性样本偏差。
 - 灵活性，包括对自学习系统的不良数据输入的错误学习。
- 系统预计如何适应其运行环境的变化。在进行适应性测试时，需要将其作为测试依据。
- 预期系统用户的详细说明，这需要确保测试具有代表性。

7.6 概念漂移的测试

操作环境可以随着时间的推移而改变，而经过训练的模型不会相应地改变，这种现象被称为概念漂移。通常会导致模型的输出变得越来越不准确和不那么有用。例如，营销活动的影响可能会在一段时间内导致潜在客户行为的变化。这种变化可能是由于系统外部的文化、道德或社会变化而产生的季节性或突然变化。这种突然变化的一个例子是：COVID-19 疾病大流行的影响，及其对用于销售预测和股票市场模型准确性的影响。

可能容易出现概念漂移的系统，应该根据他们一致同意的机器学习功能表现准则定期进行测试，以确保任何概念漂移的发生都能及时被检测到，从而减轻问题。典型的缓解措施可能包括使系统退役或重新训练系统。在重新训练的情况下，将执行最新的训练数据，然后是确认测试，回归测试，可能还有一种形式的 A / B 测试(见第 9.4 节)，其中更新的 B 系统必须优于原来的 A 系统。

7.7 为机器学习系统选择测试方法

基于人工智能的系统通常包括人工智能和非人工智能组件。测试方法基于这样一个系统的风险分析，将包括传统测试和更专业的测试，以解决特定于人工智能组件和基于人工智能的系统的那些因素。

下面的列表提供了一些典型的风险和相应的缓解措施，具体针对机器学习系统。请注意，此列表仅提供了一组有限的示例，并且有许多特定于机器学习系统的风险需要通过测试来缓解。

风险方面	描述和可能的缓解方法
数据质量可能低于预期。	这种风险可能在几个方面成为一个问题，其中每一种可以通过不同的方式加以预防（见 4.4 节）。常见的缓解措施包括使用审查、探索性数据分析和动态测试。
业务数据管道可能故障。	通过对单个管道组件进行动态测试并且对整个管道进行集成测试，可以在一定程度上减轻这种风险。
用于开发模型的机器学习工作流可能不是最佳的。（见 3.2 节）	这种风险可能是由于以下原因： <ul style="list-style-type: none"> ● 缺乏对机器学习工作流的预先共识。 ● 工作流选择不佳。 ● 数据工程师未能遵循工作流。 专家的评审可以减少选择错误工作流的可能性，而更多的实践管理或审计可以解决关于工作流的协议和执行的问题。
机器学习框架、算法、模型、模型设置和/或超参数的选择可能不是最优的。	这种风险可能是由于决策者缺乏专业知识，或者是由于机器学习工作流的评估和调优步骤（或测试步骤）的错误实施。由专家进行评审可以减少做出错误决策的可能，更好的管理可以确保遵循工作流的评估和调优（和测试）步骤。
尽管机器学习组件单独满足了这些标准，但可能无法实现所需的机器学习功能表现准则。	这种风险可能是由于单独用于训练和测试模型的数据集不能代表操作中遇到的数据。专家（或用户）对所选数据集的评审可以减少所选数据不具有代表性的可能性。
满足了预期的机器学习功能表现准则，但是用户可能对交付的结果不满意。	这种风险可能是由于选择了错误的绩效准则（例如，当需要高精度时选择了高召回率）。专家的评审可以减少选择错误的机器学习功能表现度量的可能性，或者基于经验的测试也可以识别不适当的标准。风险也可能是由于概念漂移，在这种情况下，更频繁的测试运行中的系统可以减少风险。
满足了预期的机器学习功能表现准则，但是用户可能对交付的服务不满意。	这种风险可能是由于缺乏对系统非功能需求的关注。注意，基于人工智能系统的质量特性范围超出了国际标准化组织/国际电工委员会 25010 中列出的范围（见第 2 章）。使用基于风险的方法来划分质量特征的优先级，并执行相关的非功能测试，可以减少这种风险。或者，问题可能是由于多种因素的组合造成的，这些因素可以通过基于经验的测试（作为系统测试的一部分）来识别。第 8 章提供了如何测试这些特性的指导。
自学系统可能无法提供用户期望的服务。	这种风险可能是由多种原因造成的，例如： <ul style="list-style-type: none"> ● 系统用于自主学习的数据可能不恰当。在这种情况下，专家的检查可以识别出有问题的数据。 ● 系统可能会失败，因为新的自学功能是不可接受的。这可以通过自动化的回归测试（包括与先前功能的性能比较）来缓解。 ● 系统可能在以用户不期望的方式学习，这可以通过基于经验的测试发现。
用户可能会因为不理解系统如何决定其决策而受到阻挠。	这种风险可能是由于缺乏可理解性、可解释性和/或透明度。关于如何测试这些特性的详细信息，请参见 8.6 节。
用户可能会发现，当数据与训练数据相似时，该模型提供了优秀的预测，但在其他时候提供了较差的结果。	这种风险可能是由于过拟合（见第 3.5.1 节），这可以通过使用完全独立于训练数据集的数据集测试模型或执行基于经验的测试来检测。

8. 测试人工智能特定的质量特征-150 分钟

关键词

测试结果参照物

人工智能特定的关键词

算法偏差，自主系统，自主性，专家系统，可解释性，不适当的偏差，可解释性，LIME 方法，机器学习训练数据，非决定性系统，概率性系统，样本偏差，自学系统，透明度

第八章的学习目标：

8.1 测试自学习系统时的挑战

AI-8.1.1 (K2) 解释基于人工智能的系统的自学习所带来的测试方面的挑战。

8.2 测试基于人工智能的自治系统

AI-8.2.1 (K2) 描述如何测试基于人工智能的自主系统。

8.3 测试算法、样本和不适当的偏见

AI-8.3.1 (K2) 解释如何测试基于人工智能的系统中的偏差。

8.4 测试基于概率和非确定性的人工智能系统所面临的挑战

AI-8.4.1 (K2) 解释基于人工智能的系统的概率性和非确定性所带来的测试挑战。

8.5 测试基于人工智能的复杂系统的挑战

AI-8.5.1 (K2) 解释基于人工智能的系统的复杂性在测试中带来的挑战。

8.6 测试基于人工智能的系统的透明度、整体可解释性和单一可解释性

AI-8.6.1 (K2) 描述如何测试基于人工智能的系统的透明度、可解释性和可解释性。

HQ-8.6.1 (H2) 使用一个工具来说明测试人员如何使用可解释性。

8.7 基于人工智能的系统的测试结果参照物

AI-8.7.1 (K2) 解释由于基于人工智能的系统的特殊性，在创建测试结果参照物方面的挑战。

8.8 测试目标和验收准则

AI-8.8.1 (K4) 针对特定的基于人工智能的系统的特定质量特征，选择适当的测试目标和验收标准。

8.1 测试自学习系统时的挑战

在测试自学系统时，有几个潜在的挑战需要克服（关于这些系统的更多细节见第二章），包括：

- 意外的变化。系统工作的原始要求和约束条件通常是已知的，但是关于系统本身的变化可能很少或没有信息。通常可以根据原始需求和设计（以及任何指定的约束条件）进行测试，但是如果系统已经设计了一个创新的实施方案，或者操纵了一个解决方案（其实施方案不能被看到），可能很难设计适合这个新实施方案的测试。此外，当系统改变了自己（和他们的输出），以前通过的测试的结果也会改变。这是一个测试设计的挑战。它可以通过设计适当的测试来解决，这些测试在系统改变其行为时仍然是相关的，因此可以防止潜在的回归测试问题。然而，它也可能需要根据观察到的新的系统行为来设计新的测试。
- 复杂的验收标准。可能需要定义系统在自学习时的改进预期。例如，可以假设，如果系统自我改变，它的整体功能性能应该得到改善。此外，除了简单的“改进”之外，指定其他任何东西都会很快变得复杂。例如，可能期望有一个最小的改进（而不是简单的任何改进），或者要求的改进可能与环境因素有关（例如，如果环境因素 F 变化超过 Y，则要求功能 X 至少有 10% 的改进）。这些问题可以通过规范和针对更复杂的验收标准的测试来解决，并通过保持当前系统基线功能性能的持续记录。
- 测试时间不足。可能需要知道在不同的情况下，系统学习和适应的速度有多快。这些验收标准可能很难指定和获得。如果一个系统适应得很快，可能没有足够的时间在每次变化后手动执行新的测试，所以可能有必要编写测试，当系统本身发生变化时可以自动运行。这些挑战可以通过指定适当的验收标准（见第 8.8 节）和自动持续测试来解决。
- 资源要求。系统要求可能包括系统在进行自学习或适应时允许使用的资源的验收标准。这可能包括，例如，允许用于改进的处理时间和内存的数量。此外，还需要考虑这种资源的使用是否应该与功能或准确性的可衡量的改进相联系。这一挑战影响到验收标准的规范。
- 操作环境的规格不充分。如果一个自学系统收到的环境输入超出了预期范围，或者没有反映在训练数据中，它就可能发生变化。这些输入可能是以数据中毒的形式进行攻击（见第 9.1.2 节）。要预测全部的操作环境和环境变化可能很困难，因此要确定全部的代表性测试案例和环境要求。理想的情况是，系统所要应对的运行环境的全部可能变化范围将被定义为验收标准。
- 复杂的测试环境。管理测试环境以确保它能模仿所有潜在的高风险操作环境的变化是一个挑战，可能需要使用测试工具（例如，故障注入工具）。根据操作环境的性质，可以通过操纵输入和传感器，或通过获得不同的物理环境来测试，在这些环境中可以测试系统。
- 不理想的行为修改。一个自学系统会根据其输入修改其行为，测试人员可能无法阻止这种情况的发生。例如，如果正在使用第三方系统，或者正在测试生产系统，就可能出现这种情况。

通过重复相同的测试，自学系统可能在应对这些测试时变得更加有效，然后可能影响系统的长期行为。因此，重要的是要防止出现测试导致自学系统的行为发生不利变化的情况。这是测试案例设计和测试管理的一个挑战。

8.2 测试基于人工智能的自治系统

自主系统必须能够确定何时需要人类干预，何时不需要。因此，测试基于人工智能的系统的自主性需要为系统行使这种决策创造条件。

测试自主性可能需要：

- 测试在系统应该放弃控制的特定场景下，系统是否要求人类干预。这种情况可能包括操作环境的变化，或系统超过其自主性的极限。
- 测试当系统在特定时间段后应该放弃控制时，系统是否要求人类干预。
- 测试当系统仍应自主工作时，是否不必要地要求人类干预。

使用适用于操作环境的边界值分析来产生这种测试的必要条件可能是有帮助的。界定操作环境中表明决定自主性的参数，以及创建取决于自主性性质的测试场景，这可能是一个挑战。

8.3 测试算法、样本和不适当的偏差

机器学习系统应该针对不同的偏差进行评估，并采取行动消除不适当的偏差。这可能涉及到故意引入积极的偏见以对抗不适当的偏见。

用独立的数据集进行测试通常可以发现偏差。然而，由于机器学习算法可以使用看似不相关的特征的组合来创造不需要的偏差，因此很难识别所有导致偏差的数据。

基于人工智能的系统应测试算法偏差、样本偏差和不适当的偏差（见 2.4 节）。这可能涉及：

- 在模型的训练、评估和调整活动中进行分析，以确定是否存在算法偏差。
- 审查训练数据的来源和获取数据的过程，以确定是否存在样本偏差。
- 审查作为机器学习工作流程一部分的数据预处理，以确定数据是否受到了可能导致样本偏差的影响。
- 在大量的互动中测量系统输入的变化如何影响系统输出，并根据系统可能不适当地偏向或反对的人或物的群体来检查结果。这类似于 8.6 中讨论的 LIME (Local Interpretable Model-Agnostic Explanations) 方法，可以在生产环境中进行，也可是发布前测试的一部分。

- 获得可能与偏差有关的输入数据属性的额外信息，并将其与结果联系起来。例如，这可能与人口统计学数据有关，在测试影响群体的不适当偏差时，这可能是适当的，因为群体的成员资格与评估偏差有关，但不是模型的输入。这是因为偏差可能是基于“隐藏”的变量，这些变量没有明确地出现在输入数据中，而是由算法推断出来的。

8.4 测试基于概率的和非确定性的人工智能系统所面临的挑战

大多数概率系统也是非确定性的，因此下面列出的测试挑战通常适用于具有任何这些属性的基于人工智能的系统：

- 在一组相同的前提条件和输入的情况下，测试可能有多个有效的输出结果。这使得期望结果的定义更具挑战性，并可能造成困难：
 - 当测试被重复使用于确认测试时。
 - 当测试被重复用于回归测试时。
 - 测试的可重复性是很重要的。
 - 当测试是自动化的时候。
- 测试人员通常需要对所需的系统行为有更深入的了解，以便他们能合理的检查测试是否通过，而不是简单地说明预期测试结果的准确值。例如，与传统系统相比，测试人员可能需要定义更复杂的期望结果。这些预期测试结果可能包括公差（例如，“实际结果是否在最优解的2%以内？”）。
- 如果由于系统的概率性质，无法从测试中获得单一确定输出，测试人员往往需要多次运行测试，以产生一个统计上有效的测试结果。

8.5 测试基于人工智能的复杂系统所面临的挑战

基于人工智能的系统经常被用来实现人类无法完成的太过复杂的任务。这可能会导致测试结果参照物问题，因为测试人员无法像通常那样确定期望结果（见第8.7节）。例如，基于人工智能的系统经常被用来识别大量数据的模式。之所以使用这样的系统，是因为它们可以找到人类即使经过大量分析也根本无法找到的模式。充分深入地理解这类系统的所需行为，以便能够产生期望结果，这可能是一个挑战。

当基于人工智能的系统的内部结构是由软件生成的，也会出现类似的问题，会使其过于复杂，人类无法理解。这就导致了基于人工智能的系统只能作为一个黑盒进行测试。即使内部结构是可见的也没有提供额外的有用信息来帮助测试。

当基于人工智能的系统提供概率性的结果且本质上非确定性时，其复杂性就会增加（见第8.4节）。

当一个基于人工智能的系统由几个相互作用的组件组成，每个组件都提供概率性的结果时，非确定性系统的问题就会加剧。例如，一个面部识别系统很可能使用一个模型来识别图像中的人脸，另一个模型来识别已识别的人脸。人工智能组件之间的相互作用可能很复杂，难以理解，因此很难确定所有的风险，也很难设计出能充分验证系统的测试。

8.6 测试基于人工智能的系统的透明性、整体可解释性和单一可解释性

关于系统如何实现的信息可以由系统开发者提供。这可能包括训练数据的来源、如何进行标注、以及系统组件是如何设计的。当这些信息不可用时，会使测试的设计具有挑战性。例如，如果训练数据信息不可用，那么确定这些数据的潜在差距和测试这些差距的影响，就变得很困难。这种情况可以与黑盒和白盒测试相比较，有类似的优点和缺点。透明性可以通过比较数据和算法上记录的信息与实际执行情况来测试，并确定它们的匹配程度。

与传统系统相比，使用机器学习时，通常更难解释特定输入和特定输出之间的联系。这种低水平的可解释性主要是因为产生输出的模型本身是由代码(算法)产生的，并不反映人类思考问题的方式。不同的机器学习模型提供不同级别的可解释性，应该根据对系统的要求来选择，其中包括可解释性和可测试性。

理解可解释性的一种方法是通过对测试数据施加扰动对机器学习模型进行动态测试。有一些方法可以通过这种方式量化可解释性，并对其进行可视化解释。其中一些方法是与模型无关的，而另一些则是于某一特定类型的模型有效，并需要访问它。探索性测试也可以用来更好地理解一个模型的输入和输出之间的关系。

LIME 方法与模型无关，使用动态注入的输入扰动和对输出的分析来为测试人员提供输入和输出之间关系的视图。这可以成为提供模型可解释性的有效方法。然而，它仅限于为输出提供可能的原因，而不是一个确定的原因，并且不适用于所有类型的算法。

基于人工智能的系统的可解释性在很大程度上取决于它的应用对象。在需要掌握基础技术的程度上，不同的利益相关者可能有不同的要求。

测量和测试对可解释性和可解释性的理解程度是具有挑战性的，因为利益相关者的能力水平各不相同，而且可能不一致。此外，对于许多类型的系统来说，确定典型的利益相关者的情况可能是困难的。在进行测试时，该测试通常采用用户调查和/或问卷的形式。

8.6.1 实践练习：模型的可解释性

使用适当的工具来提供基于先前创建的模型的可解释性。例如，对于图像分类模型或文本分类模型，可能适合使用 LIME 这样一个与模型无关的方法。

学生应该使用该工具来产生对模型决策的解释；特别是输入中的特征是如何影响输出的。

8.7 基于人工智能的系统的测试结果参照物

基于人工智能的系统测试的一个主要问题可能是对期望结果的说明。测试结果参照物是用来确定测试的期望结果的来源[I01]。确定期望结果的一个挑战被称为测试结果参照物问题。

对于复杂的、非确定性的或概率性的系统，如果不知道“真实数据”（即基于人工智能的系统试图预测的现实世界中的实际结果），就很难建立一个测试结果参照物。这个“真实数据”与测试结果参照物不同，因为测试结果参照物不一定提供一个预期值，而只是提供一个机制来确定系统是否运行正确。

基于人工智能的系统可以进化（见第 2.3 节），自学习系统的测试（见第 8.1 节）也会受到测试结果参照物问题的困扰，因为它们会自我修改，从而有必要经常更新系统的功能预期。

导致难以获得有效测试结果参照物的另一个原因是，在许多情况下，软件行为的正确性是主观的。虚拟助手（如 Siri 和 Alexa）就是这个问题的一个例子，因为不同的用户往往有完全不同的期望，并可能根据他们选的词和说话的清晰度而经历不同的结果。

在某些情况下，可以用限制或公差来定义期望结果。例如，自动驾驶汽车的停车点可以被定义为某个特定点的最大距离内。在专家系统的背景下，期望结果的确定可以通过咨询专家来实现（注意专家的意见仍然可能是错误的）。在这种情况下，有几个重要因素需要考虑：

- 人类专家的能力水平各不相同。所涉及的专家至少要和该系统要取代的专家一样有能力。
- 专家们可能不同意对方的意见，即使是在得到相同的信息时。
- 人类专家可能不赞成他们判断的自动化。在这种情况下，他们对潜在产出的评级应该是双盲的（也就是说，专家和产出的评估者都不应该知道哪些评级是自动）。
- 人类更有可能对回答提出警告（例如，用“我不确定，但是……”这样的短语）。如果基于人工智能的系统没有这种警告，那么在比较反应的时候就应该考虑到这一点。

有一些测试技术可以缓解测试结果参照物问题，比如 A/B 测试（见第 9.4 节）、背靠背测试（见第 9.3 节）和蜕变测试（见第 9.5 节）。

8.8 测试目标和验收准则

系统的测试目标和验收标准需要建立在感知到的产品风险之上。这些风险通常可以通过对所需的质量特性的分析来确定。基于人工智能的系统的特征包括那些在 ISO/IEC 25010[S06]中传统考虑的质量特征（即功能适用性、性能效率、兼容性、可用性、可靠性、安全性、可维护性和可移植性），但也应该包括对以下方面的考虑：

方面	验收准则
适应性	<ul style="list-style-type: none">● 当系统在适应环境的变化时，检查系统是否仍能正常运行并满足非功能要求。这可以作为自动回归测试的一种形式来实施。● 检查系统适应其环境变化所需的时间。● 检查系统在适应环境变化时使用的资源。
灵活性	<ul style="list-style-type: none">● 考虑系统在初始规范之外的环境中是如何应对的。这可以作为一种自动回归测试的形式，在改变后的操作环境中执行。● 检查系统改变自己以管理新上下文所花费的时间和/或资源。
演进	<ul style="list-style-type: none">● 检查系统从其自身经验中学习的程度。● 检查系统在数据剖面发生变化（即概念漂移）时的应对能力。
自主性	<ul style="list-style-type: none">● 检查当系统被迫脱离预期完全自主的操作范围时，如何反应。● 检查系统是否可以被“说服”，在它应该完全自主时要求人类干预。
透明性、整体可解释性和单个可解释性	<ul style="list-style-type: none">● 通过审查访问算法和数据集的难易程度来检查透明性。● 通过询问系统用户来检查整体可解释性和单个可解释性，如果没有实际的系统用户，可以询问具有类似背景的人。
不受不适当的偏差影响	<ul style="list-style-type: none">● 如果系统有可能受到偏差的影响，那么可以通过使用独立的无偏差测试套件，或者使用专家评审员来进行测试。● 使用外部数据（如人口普查数据）对测试结果进行比较，以检查推断变量上是否存在不必要的偏差（外部有效性测试）。
伦理道德	<ul style="list-style-type: none">● 根据合适的检查表检查系统，如欧共体可信人工智能评估清单[R21]，该清单支持《可信人工智能伦理准则》[R22]列出的关键要求。
概率系统和非确定性系统	<ul style="list-style-type: none">● 这不能用精确的验收标准来评估。当工作正常时，系统对相同的测试可能返回略有不同的结果。
副作用	<ul style="list-style-type: none">● 识别潜在的有害的副作用，并尝试生成导致系统表现出这些副作用的测试。
奖励黑客	<ul style="list-style-type: none">● 当这些测试与被测试的智能代理相比使用不同的测量成功的手段时，独立测试可以识别奖励黑客行为。
安全问题	<ul style="list-style-type: none">● 这需要仔细评估，也许在一个虚拟测试环境中（见第 10.2 节）。这可能包括试图强迫一个系统对自己造成伤害。

对于机器学习系统，应该规定机器学习模型所需的机器功能表现度量（见第五章）。

9. 基于人工智能的系统测试的方法和技术 -245 分钟

关键词

A/B 测试，对抗测试，背靠背测试，错误猜测，基于经验的测试，探索性测试，蜕变关系(MR)，蜕变测试(MT)，结对测试，伪测试结果参照物，测试结果参照物问题，向导法

人工智能特定的关键词

对抗性攻击、对抗样本、数据中毒、机器学习系统、训练模型

第九章的学习目标。

9.1 对抗性攻击和数据中毒

AI-9.1.1 (K2) 解释机器学习系统的测试如何有助于防止对抗性攻击和数据中毒。

9.2 结对测试

AI-9.2.1 (K2) 解释如何将结对测试用于基于 AI 的系统。
LO-9.2.1 (H2) 应用结对测试，为基于人工智能的系统推导和执行测试用例。

9.3 背靠背测试

AI-9.3.1 (K2) 解释如何将背靠背测试用于基于 AI 的系统。

9.4 A/B 测试

AI-9.4.1 (K2) 解释 A/B 测试如何应用于基于 AI 的系统的测试。

9.5 蜕变测试

AI-9.5.1 (K3) 在基于人工智能的系统测试中应用蜕变测试。
HO-9.5.1 (H2) 应用蜕变测试，为给定的场景推导出测试用例，并执行它们。

9.6 对基于人工智能的系统的基于经验的测试

AI-9.6.1 (K2) 解释基于经验的测试如何应用于基于人工智能的系统的测试。
HO-9.6.1 (H2) 将探索性测试应用于基于人工智能的系统。

9.7 为基于人工智能的系统选择测试技术

AI-9.7.1 (K4) 对于一个给定的场景，在测试一个基于人工智能的系统时，选择适当的测试技术。

9.1 对抗攻击与数据中毒

9.1.1. 对抗攻击

对抗性攻击是指攻击者巧妙地扰乱传递给训练好的模型的有效输入，使其提供错误的预测。这些被扰乱的输入，被称为对抗性样本，首先被注意到的是垃圾邮件过滤器，它可以通过轻微修改垃圾邮件而不失去可读性来欺骗。最近，它们与图像分类器变得更加相关。通过简单地改变几个人眼不可见的像素，就有可能说服神经网络将其图像分类改为非常不同的对象，并且有很高的置信度。

对抗性样本通常是可转移的[B17]，这意味着导致一个机器学习系统失败的对抗样子往往会导致另一个被训练来执行相同任务的机器学习系统失败。即使第二个机器学习系统是用不同的数据训练出来的，并且是基于不同的架构，它仍然经常容易在相同的对抗性例子中失败。

白盒对抗性攻击是指攻击者知道哪种算法被用来训练模型，也知道哪些模型设置和参数被使用（有合理的透明性）。攻击者利用这些知识来产生对抗性的例子，例如，对输入进行小扰动，并监测哪些扰动会对模型输出造成大的变化。

黑盒式对抗性攻击包括攻击者探索模型以确定其功能，然后建立一个提供类似功能的复制模型。然后，攻击者使用白盒方法为这个复制的模型确定对抗性例子。由于对抗性例子通常是可转移的，同样的对抗性例子通常也会对原始模型起作用。

如果不可能创建一个复制的模型，也许可以使用大批量的自动测试来发现不同的对抗性例子并观察结果。

对抗测试只是涉及执行对抗性攻击，目的是为了识别漏洞，以便采取预防措施防止未来的失败。确定的对抗性例子被添加到训练数据中，以便模型被训练成能够正确识别它们。

9.1.2. 数据中毒

数据中毒攻击是指攻击者操纵训练数据以达到两种结果之一。攻击者可能会插入后门或神经网络木马以促进未来的入侵，或者更常见的是，他们会使用被破坏的训练数据（例如，错误标注的数据）来诱导训练模型提供不正确的预测。

中毒攻击可能是有针对性的，目的是使机器学习系统在特定情况下发生错误分类。它们也可能是恣意的，例如拒绝服务攻击。一个著名的中毒攻击的例子是微软 Tay 聊天机器人的堕落，通过相对较少的有害 Twitter 对话训练系统在未来提供有污点的对话。一种常用的数据中毒攻击形式是将数以百万计的垃圾邮件错误地报告为非垃圾邮件，以试图影响垃圾邮件过滤软件的准确性。数据中毒的一个关注领域是公共广泛使用的人工智能数据集更可能中毒。

使用 EDA 检测数据中毒是可能的，因为中毒的数据可能显示为异常值。此外，可以审查数据采集策略，以确保训练数据的来源。如果一个运行中的机器学习系统可能通过输入中毒数据而受到攻击，

可以使用 A/B 测试（见第 9.4 节）来检查该系统的更新版本是否仍然与之前的版本严密地一致。另外，使用可信的测试套件对更新的系统进行回归测试，也可以确定系统是否中毒。

9.2 结对测试

基于人工智能的系统所关注的参数数量可能非常多，特别是当系统使用大数据或与外部世界互动时，如自动驾驶汽车。详尽测试需要将这些参数的所有可能值组合设置进行测试。然而，由于这将导致实际上无限多的测试，测试技术被用来选择一个可以在有限的时间内运行的子集。

当有可能组合许多参数时，每个参数有许多离散的值，组合测试可以显著减少所需测试用例的数量，理想情况下不应影响测试套件的缺陷检测能力。有几种组合测试技术（见[I02]和[S08]）。在实践中，结对测试是最广泛使用的技术，因为它易于理解，有充足的工具支持。此外，研究表明，大多数缺陷是由涉及少数参数的相互作用引起的[B33]。

在实践中，即使使用结对测试也会导致一些系统产生大量的测试套件，而使用自动化和虚拟测试环境（见第 10.2 节）往往成为必要，以允许运行足够数量的测试。例如，当考虑自动驾驶汽车时，系统测试的高级测试方案需要同时测试汽车预期运行的不同环境和各种车辆功能。因此，参数需要包括环境约束的范围（例如，道路类型和路面，天气和交通状况以及能见度）和各种自动驾驶功能（例如，自适应巡航控制，车道保持辅助，以及变道辅助）。除了这些参数外，来自传感器的输入可以考虑不同的有效性水平（例如，来自摄像头的输入将随着旅程的进展变得更脏而降低）。

目前的研究还不清楚对基于人工智能安全至关重要的系统（如自动驾驶汽车）使用组合测试所需的必要严格程度。即使结对测试可能还不够，但众所周知，该方法在发现（的结果）缺陷方面是有效的。

9.2.1 实践练习：结对测试

对于一个至少有五个参数和至少五百种可能组合的基于人工智能的系统，使用结对测试工具来确定一个缩小的结对组合集，并对这些组合进行测试。将测试的结对组合的数量与理论上所有可能的组合都要测试的数量进行比较。

9.3 背靠背测试

在测试基于人工智能的系统时，测试结果参照物问题的潜在解决方案之一（见第 8.7 节）是使用背靠背测试。这也被称为差异测试。在背靠背测试中，系统的另一个版本被用作伪测试结果参照物（参照物），其输出与 SUT 产生的测试结果进行比较。伪测试结果参照物可以是一个现有的系统，也可以是由不同的团队，可能是在不同的平台上，采用不同的结构和不同的编程语言开发的。当测试功能适用性（相对于非功能需求）时，作为伪测试结果参照物的系统在达到与 SUT 相同的非功能验收标准上

不受限制。例如，它可能不需要执行得那么快，在这种情况下，它的构建成本会低得多。

在机器学习的背景下，有可能使用不同的框架、算法和模型设置来创建一个机器学习伪测试结果参照物。在某些情况下，也有可能使用传统的、非人工智能的软件来创建一个伪测试结果参照物。

为了使伪测试结果参照物能有效地检测出缺陷，在伪测试结果参照物和 SUT 中不应该有共同的软件。否则，当两者都有缺陷时，两者中的相同缺陷就有可能导致两个测试结果的匹配。由于有这么多不成熟的、可重复使用的、开源的人工智能软件被用来开发基于人工智能的系统，在伪测试结果参照物和 SUT 之间重复使用代码会损害伪测试结果参照物。可重用的人工智能解决方案的糟糕文档也可能使测试人员难以认识到这个问题的发生。

9.4 A/B 测试

A/B 测试是一种方法，对程序的两个变体（A 和 B）对相同输入的反应进行比较，目的是确定这两个变体中哪个更好。它是一种统计测试方法，通常需要比较多个测试运行的测试结果，以确定程序之间的差异。

这种方法的一个简单例子是，将两个促销优惠通过电子邮件发送给一个营销名单，分为两组。一半的名单得到优惠 A，一半得到优惠 B，每个成功的优惠有助于决定将来使用哪一个。许多电子商务和基于 web 的公司在生产中使用 A/B 测试，将不同的消费者引向不同的功能，以帮助确定消费者的偏好。

A/B 测试是解决测试结果参照物问题的一种方法，现有系统被用作部分结果参照物。A/B 测试不产生测试用例，也不提供关于如何设计测试的指导，尽管测试中经常使用该操作输入。

A/B 测试可以用来测试基于人工智能的系统的更新，其中有商定的验收标准，如第五章所述的机器学习工作功能表现度量。每当系统被更新时，A/B 测试被用来检查更新后的变体是否和以前的变体一样好，或者更好。这种方法可以用于简单的分类器，但也可以用于测试复杂得多的系统。例如，为提高智能城市交通路线系统的有效性而进行的更新也可以使用 A/B 测试（例如，比较系统的两个变体在连续几周的平均通勤时间）。

A/B 测试也可以用来测试自学习系统。当系统做出变更时，自动测试被运行，结果特性与改变前的特性进行比较。如果系统得到了改善，那么该变更就被接受，否则系统就会恢复到以前的状态。

A/B 测试和背靠背测试之间的一个主要区别在于使用 A/B 测试来比较同一系统的两个变体，而使用背靠背测试来检测缺陷。

9.5 蜕变测试

蜕变测试[B18]是一种旨在基于已经通过的源测试用例生成新测试用例的技术。通过蜕变关系

（MR）改变（蜕变）源测试用例生成一个或多个后续测试用例。MR 是基于测试对象所需功能的属性，描述测试用例的测试输入变化如何反映在同一测试用例的期望结果中。

例如，考虑一个确定一组数字的平均值的程序。一个源测试用例被生成，包括一组数字和一个预期的平均值，测试用例被运行以确认它通过。现在可以根据对程序的平均功能的了解来生成后续测试用例。最初，被平均的数字的顺序可能被简单地改变。考虑到平均函数的属性，可以预测期望结果将保持不变。因此，可以生成一个数字顺序不同的后续测试用例，而不需要计算期望结果。对于一个大的数字集，这可能会导致生成大量不同的数字集，其中相同的数字以不同的顺序使用，每个数字都可以用来创建一个单独的后续测试用例。所有这些测试用例将基于相同的源测试用例，并有相同的期望结果。

通常情况下，MR 和后续测试用例的期望结果与源测试用例的原始期望结果不同。例如，使用相同的平均函数可以得出一个 MR：输入集的每个元素都乘以 2，这样一个集合的期望结果是原始期望结果乘以 2。同样，任何其他值都可以作为乘数，在这个 MR 的基础上有可能产生无限多的后续测试用例。

MT 可用于大多数测试对象，并可应用于功能和非功能测试（例如，可安装性测试涵盖不同的目标配置，其中安装参数可按不同的顺序选择）。在由于缺乏廉价的测试结果参照物而导致期望结果的生成有问题时，它特别有用。一些基于大数据分析的人工智能系统就是这种情况，或者测试人员不清楚机器学习算法如何得出其预测结果时。在人工智能领域，MT 已被用于测试图像识别、搜索引擎、路线优化和语音识别等。

如上所述，MT 可以基于通过的源测试用例，但如果不能验证任何源测试用例是否正确时它也是有用的。例如，程序实现的功能过于复杂，人类测试人员无法复制并用作测试结果参照物参照，如一些基于人工智能的系统。在这种情况下，MT 可以用来生成一个或多个测试用例，这些用例在运行时将创建一组输出，然后可以检查输出之间的关系是否有效。有了这种形式的 MT，单个测试不知道是否正确，但它们之间的关系必须是成立的，因此提高了对程序改进的信心。例如一个基于人工智能的精算程序，该程序根据大量数据预测死亡年龄，如果吸烟数增加，预测的死亡年龄应该减少（或至少保持不变）。

MT 是一种相对较新的测试技术，于 1998 年首次提出。它与传统的测试技术不同，后续测试用例的期望结果不是用绝对值来描述，而是相对于源测试用例的期望结果。它基于一个容易理解的概念，可以由缺乏应用技术经验但了解应用领域的测试人员来使用，而且与传统技术相比具有类似的成本。它在揭示缺陷方面也很有效，研究表明，只要三到六个不同的 MRs 可以揭示超过 90% 的缺陷，这些缺陷可以通过基于传统测试结果参照物的技术来检测[B19]。有可能从指定的 MRs 和源测试用例中自动生成后续测试用例。然而，目前还没有商用级工具，尽管 Google 已经在使用 GraphicsFuzz 工具测试 Android 图形驱动，该工具是开源的（见[R23]）。

9.5.1 实践练习：蜕变测试

在这个练习中，学生将获得以下的实际经验：

- 为一个给定的基于人工智能的应用或程序推导出几个蜕变关系（MRs）。这些 MRs 应该包括一些源测试用例和后续测试用例的期望结果相同的地方，以及一些不同的地方。
- 为基于人工智能的应用或程序生成源测试用例。这些测试不必保证通过，但应提醒学生在没有这种“黄金标准”的情况下 MT 的局限性。
- 使用派生的 MR 和生成的源测试用例来导出后续测试用例。
- 运行后续测试用例。

9.6 基于经验的人工智能系统测试

基于经验的测试包括错误猜测、探索性测试和基于检查表的测试[I01]，所有这些都可以应用于基于 AI 的系统的测试。

错误猜测通常是基于测试人员的知识、典型的开发人员的错误、以及类似系统（或以前的版本）的失效。应用于基于人工智能的系统的错误猜测的一个例子可以是使用关于机器学习系统在过去由于使用有系统偏差的训练数据而失败的知识。

在探索性测试中，测试是以迭代的方式设计、生成和执行的，根据早期测试的结果，有机会得出后来的测试。探索性测试在糟糕的规范或测试结果参照物问题时特别有用，基于人工智能的系统往往就是这样。因此，探索性测试经常在这种情况下使用，并用来补充基于技术的更系统的测试，如蜕变测试（见 9.5 节）。

向导法（tour）是一种隐喻，用于测试人员进行探索性测试时，围绕一个特别关注点组织的一系列策略和目标[B20]。基于人工智能的系统的探索性测试的典型向导法可能会关注机器学习系统中的偏差、欠拟合和过拟合的概念。例如，一个数据向导法可能被应用于测试模型。在这个向导法中，测试者可以确定用于训练的不同类型的数据、它们的分布、变化、格式和范围等等，然后用这些数据类型来测试模型。

机器学习系统高度依赖于训练数据的质量，而现有的 EDA 领域与探索性测试方法密切相关。EDA 是检查数据的模式、关系、趋势和异常值的地方。它涉及到对数据的交互、假设驱动的探索，在[B21]中被描述为“我们带着期望探索数据。我们根据我们在数据中看到的内容来修改我们的期望。而且我们反复进行这个过程”。EDA 通常需要两个方面的工具支持：一是与数据的交互，让分析师更好地理解复杂的数据；二是数据的可视化，让他们轻松显示分析结果。主要由数据可视化驱动的探索性技术的使用，可以帮助验证正在使用的机器学习算法，识别导致高效模型的变化，并利用领域的专业知识[B22]。

谷歌有一套写成断言的 28 个 ML 测试，涉及数据、模型开发、基础设施和监控等领域，在谷歌内部被用作机器学习系统的测试清单[B23]。这里介绍的谷歌“机器学习测试清单”由谷歌发布如下：

机器学习数据：

1. 特征期望被捕获在一个模式中。
2. 所有特征都是有益的。
3. 没有任何特征的成本是过高的。
4. 特征遵守元级别（metalevel）要求。
5. 数据管道有适当的隐私控制。
6. 可以快速添加新的特征。
7. 所有输入的特征代码都经过测试。

模型开发：

1. 模型规格经过审查并提交。
2. 线下和线上的指标相互关联。
3. 所有的超参数都已调好。
4. 模型过时的影响是已知的。
5. 一个更简单的模型并不是更好。
6. 在重要的数据片上，模型质量是足够的。
7. 该模型经过测试，考虑到包容性。

机器学习基础设施：

1. 训练是可重复的。
2. 模型规格经过单元测试。
3. 机器学习的管道经过了集成测试。
4. 模型质量在服务前进行验证。
5. 模型是可调试的。
6. 模型在提供前都经过了验证。
7. 服务的模型可以回退。

监控测试：

1. 依赖关系的改变会导致通知。
2. 数据不变量适用于输入。
3. 训练和服务是不偏不倚的。
4. 模型不会太陈旧。
5. 模型在数值上是稳定的。
6. 计算性能没有退步。
7. 预测质量没有退步。

9.6.1 实践练习。探索性测试和探索性数据分析（EDA）

对于选定的模型和数据集，学生将进行数据向导法，考虑各种类型的数据和它们在各种参数上的分布。学生将对数据进行 EDA，以识别数据中的缺失和/或潜在的偏差。

9.7 为基于人工智能的系统选择测试技术

一个基于人工智能的系统通常包括人工智能和非人工智能组件。测试非人工智能组件的测试技术的选择通常与常规测试相同。对于基于人工智能的组件，选择可能更受限制。例如，如果察觉到测试结果参照物问题（即产生期望结果是困难的），那么，基于察觉到的风险，可以通过使用以下方式缓解这个问题。

- **背靠背测试：**需要有测试用例或生成测试用例，并有一个等效的系统作为伪测试结果参照物，对于回归测试来说，伪测试结果参照物可以是系统的前一个版本。为了有效检测缺陷，可能需要一个独立开发的系统。
- **A/B 测试：**通常使用操作输入作为测试用例，使用统计分析来比较同一系统的两个变体。A/B 测试可用于检查新变体的数据中毒，或用于自学习系统的自动回归测试。
- **蜕变测试：**可以被没有经验的测试人员用来低成本地发现缺陷，但他们需要了解应用领域。MT 不适合提供明确的结果，因为期望结果不是绝对的，而是相对于源测试用例而言。目前还没有商业工具支持，但许多测试可以手动生成。

对抗测试通常适用于错误处理对抗性样本可能会产生重大影响或者系统可能被攻击的机器学习模型。同样地，数据中毒测试适合于系统可能被攻击的机器学习系统。

如果基于人工智能的系统很复杂，有多个参数，成对测试往往是合适的。

基于经验的测试通常适用于测试基于人工智能的系统，特别是考虑到用于训练和操作数据的数据。探索性数据分析可用于验证所使用的机器学习算法，确定效率改进，并利用领域专业知识。谷歌发现

他们的机器学习测试清单是机器学习系统的有效方法。

在神经网络的特定领域，网络的覆盖率通常适用于关键任务系统，有些覆盖标准需要比其他标准更严格。

中国软件测试认证委员会 (CSTQB®)

10. 基于人工智能的系统的测试环境-30 分钟

关键词

虚拟测试环境

人工智能特定关键词

人工智能专用处理器、自治系统、大数据、可解释性、多代理系统、自学习系统

第 10 章的学习目标：

10.1 基于人工智能系统的测试环境

AI-10.1.1 (K2) 描述基于人工智能的系统的测试环境区别于传统系统所需的主要因素。

10.2 用于测试基于人工智能的系统的虚拟测试环境

AI-10.2.1 (K2) 描述虚拟测试环境在基于人工智能的系统测试中提供的好处。

10.1 基于人工智能的系统的测试环境

基于人工智能的系统可用于各种操作环境，这意味着测试环境也同样多样化。基于人工智能的系统的特点会导致测试环境与传统系统的不同，这些特点包括：

- **自学习：**自学习系统和一些自治系统要适应不断变化的操作环境，这些环境在系统最初部署时可能还没有完全确定（见 2.1 节）。因此，定义能够模仿这些未定义的环境变化的测试环境本身就很困难，可能需要测试人员的想象力和测试环境中内置一定程度的随机性。
- **自主性：**自治系统应在没有人类干预的情况下对环境的变化做出反应，同时也要认识到在哪些情况下应将自主权交还给人类操作者（见第 2.2 节）。对于一些系统来说，识别并模仿让渡自主权的情况可能需要测试环境将系统推向极端。对于一些自治系统来说，它们的目的是在危险的环境中工作，而建立有代表性的、危险的测试环境可能是具有挑战性的。
- **多代理：**当基于人工智能的多代理系统被期望与其他基于人工智能的系统协同工作时，测试环境可能需要纳入一定程度的非确定性，以便它能模仿与 SUT 交互的基于人工智能的系统的非确定性。
- **可解释性：**一些基于人工智能的系统的性质可能使人难以确定系统是如何做出决策的（见第 2.7 节）。如果在部署前了解这一点很重要，测试环境可能需要纳入一些工具，作为解释决策是如何做出的手段。
- **硬件：**一些用于承载基于人工智能的系统的硬件是专门为此目的而设计的，如人工智能专用处理器（见 1.6 节）。在测试环境中包括这种硬件应该作为相关测试规划的一部分来考虑。
- **大数据。**如果一个基于人工智能的系统预计会消费大数据（例如，高容量、高速度和/或高变化的数据），那么需要仔细规划实施将其设置为测试环境的一部分（见 7.3 节）。

10.2 用于测试基于人工智能系统的虚拟测试环境

在测试基于人工智能的系统时，使用虚拟测试环境会带来以下好处：

- **危险的场景：**这些可以在不危及 SUT、其他交互系统（包括人类）或操作环境（如树木、建筑物）的情况下进行测试。
- **不寻常的场景：**当为实际操作设置这些场景非常耗时或昂贵时，虚拟环境就可以对这些场景进行测试（例如罕见的事件，如日全食或四辆公共汽车同时进入同一个道路交叉口）。同样，在现实世界中很难创建的边缘用例，可以在虚拟测试环境中更容易地、反复地、可重复地创建。
- **极端情景：**当在现实中设置这些东西很昂贵或不可能时，可以进行测试（例如核灾难或深空探索）。

- **时间密集型场景:**这些可以在虚拟环境中以更短的时间尺度（如每秒数次）进行测试。相比之下，这些原本可能需要几个小时或几天的时间来设置和实时运行。另一个优势是，多个虚拟测试环境可以并行运行。通常在云端允许许多场景同时运行，这在使用实际系统硬件时是不可能的。
- **可观察性和可控制性:**虚拟测试环境提供了更大的测试环境可控性。例如，它们可以确保一组不寻常的金融交易条件被复制出来。此外，它们提供了更好的可观察性，因为环境的所有数字部分都可以被持续监控和记录。
- **可用性:**通过虚拟测试环境对硬件进行模拟，可以用（模拟的）硬件组件对系统进行测试，因为这些硬件可能没有被开发出来或者太昂贵而不可用。

虚拟测试环境可以是专门为一个特定的系统构建的，可以是通用的，也可以是为支持特定应用领域而开发的。商业和开源的虚拟测试环境都可以用来支持基于人工智能的系统的测试。例如：

- **Morse:** 模块化开放机器人仿真引擎，是一个基于 Blender 游戏引擎[R24]的通用移动机器人仿真器，用于单人或多人机器人的仿真。
- **AI Habitat:** 这是一个由 Facebook AI 创建的模拟平台，目的是在照片般逼真的 3D 环境中训练形体代理（如虚拟机器人）[R25]。
- **DRIVE Constellation:** 这是英伟达为自动驾驶汽车提供的一个开放和可扩展的平台。它基于一个云平台，能够产生数十亿英里的自动车辆测试[R26]。
- **MATLAB 和 Simulink:** 提供了准备训练数据、制作机器学习模型和模拟执行基于 AI 的系统的能力，包括使用合成数据的模型[R27]。

11. 使用人工智能进行测试-195 分钟

关键词

视觉测试

人工智能特定的关键词

贝叶斯技术、分类、聚类算法、缺陷预测、图形用户界面（GUI）。

第 11 章的学习目标：

11.1 用于测试的 AI 技术

- | | | |
|-----------|------|-------------------------|
| AI-11.1.1 | (K2) | 对用于软件测试的人工智能技术进行分类。 |
| HO-11.1.1 | (H2) | 举例讨论测试中那些不太可能使用人工智能的活动。 |

11.2 使用人工智能来分析报告的缺陷

- | | | |
|-----------|------|---------------------|
| AI-11.2.1 | (K2) | 解释 AI 如何协助支持新缺陷的分析。 |
|-----------|------|---------------------|

11.3 使用人工智能生成测试用例

- | | | |
|-----------|------|--------------------|
| AI-11.3.1 | (K2) | 解释人工智能如何协助测试用例的生成。 |
|-----------|------|--------------------|

11.4 使用人工智能来优化回归测试套件

- | | | |
|-----------|------|---------------------|
| AI-11.4.1 | (K2) | 解释 AI 如何协助优化回归测试套件。 |
|-----------|------|---------------------|

11.5 使用人工智能进行缺陷预测

- | | | |
|-----------|------|-----------------------|
| AI-11.5.1 | (K2) | 解释人工智能如何协助进行缺陷预测。 |
| HO-11.5.1 | (H2) | 实现一个简单的基于人工智能的缺陷预测系统。 |

11.6 使用人工智能测试用户界面

- | | | |
|-----------|------|--------------------|
| AI-11.6.1 | (K2) | 解释 AI 在测试用户界面中的应用。 |
|-----------|------|--------------------|

11.1 用于测试的 AI 技术

第 1.4 节中列出了几种人工智能技术，所有这些技术都可以用来支持软件测试的某些特定方面。根据 Harman[B24]的说法，软件工程使用了三大领域的人工智能技术：

- **模糊逻辑和概率方法：**这些方法涉及使用人工智能技术来处理现实世界的问题，这些问题本身就是概率性的。例如，人工智能可以使用贝叶斯技术分析和预测可能的系统故障。估计部件或功能失效的可能性，或反映人类与系统交互的潜在随机性。
- **分类、学习和预测：**适用于各种使用案例，如作为项目规划的一部分预测成本或预测缺陷。正如机器学习所体现的，这一领域被用于许多软件测试任务，包括缺陷管理（见第 11.2 节）、缺陷预测（见第 11.5 节）和用户界面测试（见第 11.6 节）。
- **计算性搜索和优化技术：**通过计算搜索潜在的大而复杂的搜索空间来解决优化问题（使用搜索算法）。例如，生成测试用例（见第 11.3 节），确定达到给定覆盖标准的最小数量的测试用例，以及优化回归测试用例（见第 11.4 节）。

上述分类必然是宽泛的，因为可以由人工智能实现的测试任务和不同的人工智能技术之间存在相当大的重叠。这只是一种分类，其他的分类也可能同样有效。

11.1.1 实践练习：AI 在测试中的应用

作为讨论的一部分，学生将确定目前不适用于人工智能实施的测试活动和任务。这些可能包括：

- 指定测试预期结果。
- 与利益相关者沟通，以澄清歧义并检索缺失的信息。
- 对用户体验提出改进建议。
- 挑战利益相关者的假设并提出棘手的问题。
- 了解用户需求。

应该区分弱人工智能和通用人工智能，前者可用于一些有限的任务，后者目前还不能使用（见第 1.2 节）。

11.2 使用人工智能分析报告的缺陷

报告的缺陷通常会被分类、优先级排序，并识别出任何重复的缺陷。这项活动通常被称为缺陷分类或分析，目的是优化解决缺陷所花费的时间。人工智能可用于以各种方式支持这项活动，例如。

- **分类：**NLP[B25]可用于分析缺陷报告中的文本并提取主题，如受影响的功能区域，然后将其与其他元数据一起提供给聚类算法，如 K 近邻或支持向量机。这些算法可以识别合适的缺

陷类别，并突出显示类似或重复的缺陷。基于人工智能的分类对于自动缺陷报告系统（如微软 Windows 和火狐浏览器）和有许多软件工程师的大型项目特别有用。

- 关键性：根据最关键缺陷的特征训练的机器学习模型可以用来识别那些最可能导致系统故障的缺陷，这些缺陷在报告的缺陷中占很大比例[B26]。
- 指派：机器学习模型可以根据缺陷内容和以前的开发人员分配，建议哪些开发人员最适合修复特定缺陷。

11.3 使用人工智能生成测试用例

使用人工智能生成测试是一种非常有效的技术，可以快速创建测试资产并最大化覆盖率（例如，代码或需求覆盖）。生成这些测试的基础包括源代码、用户界面和机器可读的测试模型。一些工具还将测试建立在通过仪器或通过日志文件观察系统的低级行为的基础上[B27]。

然而，除非定义了所需行为的测试模型被用作测试的基础，否则这种形式的测试生成通常会受到测试预期结果问题的影响，因为基于人工智能的工具不知道对于一组给定的测试数据，期望结果应该是什么。一个解决方案是，如果有一个合适的系统可以作为伪测试结果使用，就使用背靠背测试（见第 9.3 节）。另外，测试的期望结果可以是既不发生“应用程序无响应”，也不发生系统崩溃，或其他类似的简单故障指标。

将基于人工智能的测试生成工具与类似的非人工智能模糊测试工具进行比较的研究表明，基于人工智能的工具可以实现同等水平的覆盖率，并发现更多的缺陷，同时将导致故障的平均步骤序列从平均约 15,000 步减少到约 100 步。这使得调试工作变得更加容易[B27]。

11.4 使用人工智能优化回归测试套件

随着系统的变化，新的测试被创建、执行并成为回归测试套件的候选者。为了防止回归测试套件过于庞大，应该经常对其进行优化，以选择、确定优先次序，甚至增加测试用例，以创建一个更有效和更高效的回归测试套件。

基于人工智能的工具可以对回归测试套件进行优化，例如，通过分析以前的测试结果、相关的缺陷和最新的变化，如哪些功能更频繁地被破坏，哪些测试运行的代码受到最近变化的影响来进行优化。

研究表明，回归测试套件的规模可以减少 50% 仍然可以检测到大多数缺陷[B28]，在持续集成测试中，测试执行时间可以减少 40%，而故障检测也不会明显减少[B29]。

11.5 使用人工智能进行缺陷预测

缺陷预测可以用来预测是否存在缺陷、有多少缺陷或者是否能找到缺陷。这种能力取决于所用工

具的先进性。结果通常被用来确定测试的优先次序（例如，对那些预测有更多缺陷的部件进行更多测试）。

缺陷预测通常是基于源代码度量、流程度量和/或人员和组织度量。由于有如此多的潜在因素需要考虑，确定这些因素和缺陷的可能性之间的关系超出了人类的能力。因此，使用基于人工智能的方法（通常用机器学习）是必要的。当基于类似情况下的先前经验（例如，相同的代码库和/或相同的开发人员），缺陷预测是最有效的。

使用 ML 的缺陷预测已经成功地应用于几种不同的情况（例如，[B30]和[B31]）。已发现最佳的预测因素是人和组织度量，而不是更广泛使用的源代码度量，如代码行和圈复杂度[B32]。

11.5.1 实践练习。建立一个缺陷预测系统

学生将使用一个合适的数据集（例如，包括源代码衡量标准和相应的缺陷数据）来建立一个简单的缺陷预测模型，并使用它来预测使用类似代码的源代码衡量标准的缺陷的可能性。

该模型应该使用数据集中的至少四个特征，并且该课程应该使用几个不同的特征来探索结果，以突出结果是如何根据所选特征而变化的。

11.6 使用人工智能测试用户界面

11.6.1 通过图形用户界面（GUI）使用 AI 进行测试

通过 GUI 进行测试是手动测试的典型方法（除组件测试外），通常是测试自动化计划的出发点。由此产生的测试模拟了人类与测试对象的互动。这种脚本化的测试自动化可以通过应用捕获/回放的方法来实现，用到了用户界面元素的实际坐标，或界面的软件定义对象/部件。然而，这种方法在对象识别方面存在一些缺点，包括对界面变化、代码变化和平台变化的敏感性。

人工智能可以用来减少这种方法的脆弱性，通过采用基于人工智能的工具，使用各种准则（例如 XPath、标注、id、类、X/Y 坐标）来识别正确的对象，并选择历史上最稳定的识别准则。例如，应用程序中某个特定区域的按钮的 ID 可能会随着每个版本的发布而改变，因此基于人工智能的工具可能会随着时间的推移对这个 ID 赋予较低的重要性，而过多地依赖其他准则。这种方法将用户界面中的对象分类为匹配测试，或不匹配测试。

另外，视觉测试使用与实际用户相同的界面通过图像识别与 GUI 对象互动，因此不需要访问底层代码和界面定义。这使得它完全不具侵入性，并独立于底层技术。脚本只需要通过可见的用户界面工作。这种方法允许测试人员创建脚本，直接与屏幕上的图像、按钮和文本字段进行交互，与人类用户的交互方式相同，不受整个屏幕布局的影响。在测试自动化中使用图像识别会受到所需计算资源的限制。然而支持复杂图像识别的廉价人工智能的出现，使这种方法有可能成为主流。

11.6.2 使用人工智能测试用户图形界面

机器学习模型可以用来确定用户界面屏幕的可接受性（例如，通过使用启发式步骤和有监督学习）。基于这些模型的工具可以识别不正确的渲染元素，确定一些对象是否无法访问或难以检测，并检测 GUI 的视觉外观的各种其他问题。

虽然图像识别是计算机视觉算法的一种形式，但其他形式的基于人工智能的计算机视觉可以用来比较图像（例如屏幕截图），以确定对布局、对象的大小、位置、颜色、字体或其他可见属性的意外更改。这些比较的结果可以用来支持回归测试，以检查对测试对象的改变没有对用户界面产生不利影响。

检查屏幕可接受性的技术可以与比较工具相结合，以创建更复杂的基于人工智能的回归测试工具，能够建议检测到的用户界面变化是否有可能被用户接受，或者这些变化是否应该被标注为由人类检查。这种基于人工智能的工具也可以用来支持在不同的浏览器、设备或平台上的兼容性测试，旨在检查同一应用程序的用户界面在不同的浏览器/设备/平台上是否正常工作。

12. 参考文献

12.1 标准

- [S01] ISO/IEC TR 29119-11:2020, 软件和系统工程-软件测试-第 11 部分基于人工智能系统的测试指南。
- [S02] DIN SPEC 92001-1, 人工智能--生命周期过程和质量要求--第 1 部分：质量元模型, <https://www.din.de/en/wdc-beuth:din21:303650673> (2021 年 5 月查阅)。
- [S03] DIN SPEC 92001-2, 人工智能-生命周期过程和质量要求-第 2 部分：技术和组织要求, <https://www.din.de/en/innovation-and-research/din-spec-en/projects/wdc-proj:din21:298702628> (2021 年 5 月查阅)。
- [S04] ISO 26262-<https://www.iso.org/standard/68383.html> (2021 年 5 月查阅)。
- [S05] ISO/PAS 21448:2019, 道路车辆--预期功能的安全 (SOTIF) --<https://www.iso.org/standard/70939.html> (2021 年 5 月访问)。
- [S06] ISO/IEC 25010:2011, 系统和软件工程-系统和软件质量要求和评估 (SQuARE) -系统和软件质量模型, 2011。
- [S07] ISO 26262-6:2018 - 道路车辆 - 功能安全 - 第 6 部分：软件层面的产品开发。
- [S08] ISO/IEC/IEEE 29119-4:2015, 软件和系统工程-软件测试-第 4 部分：测试技术。

12.2 ISTQB® 文档

- [I01] ISTQB® 认证测试工程师基础级课程大纲, 2018 年版 V3.1。
<https://www.istqb.org/downloads/category/2-foundation-level-documents.html> (2021 年 5 月查阅)。
- [I02] ISTQB® 认证测试工程师高级测试分析师大纲, 3.1 版, 3.2.6 节
<https://www.istqb.org/downloads/category/75-advanced-level-test-analyst-v3-1.html> (2021 年 8 月查阅)。
- [I03] ISTQB® 认证测试工程师 AI 测试, 大纲概述, 1.0 版。

12.3 书籍和文献

- [B01] Cadwalladr, Carole (2014)。“机器人要崛起了吗？谷歌的新任工程总监这样认为……”《卫报》，卫报新闻传媒有限公司。
<https://www.theguardian.com/technology/2014/feb/22/robots-google-ray-kurzweil-terminator-singularity-artificial-intelligence> (2021 年 5 月查阅)。
- [B02] Stuart Russell 和 Peter Norvig, 《人工智能：现代方法》，第四版，皮尔逊出版社，2020。

- [B03] M. Davies 等人,《用 Loihi 推进神经形态计算: 结果和展望的调查》, IEEE 会议录, 第 109 卷, 第 5 期, 第 911-934 页, 2021 年 5 月, doi: 10.1109/JPROC.2021.3067593。
- [B04] Chris Wiltz, 苹果公司能否将其最新的人工智能芯片用于照片以外的用途?, 电子与测试, 人工智能,
<https://www.designnews.com/electronics-test/can-apple-use-its-latest-ai-chip-more-photos/153617253461497> (2021 年 5 月访问)。
- [B05] HUAWEI 在 IFA 2017 上揭示了移动 AI 的未来, 华为新闻稿,
<https://consumer.huawei.com/en/press/news/2017/ifa2017-kirin970/> (2021 年 5 月查阅)。
- [B06] 欧洲议会和理事会发布关于在处理个人数据方面保护自然人和此类数据自由流动的 (EU) 第 2016/679 号条例, 并废除第 95/46/EC 号指令 (一般数据保护条例), 2016 年 4 月, <https://eur-lex.europa.eu/eli/reg/2016/679/oj> (2021 年 5 月访问)。
- [B07] 路面机动车驾驶自动化系统相关术语的分类和定义 J3016_201806, SAE, ,
https://www.sae.org/standards/content/j3016_201806/ (2021 年 5 月查阅访问)。
- [B08] G20 关于贸易和数字经济的部长级声明:
<https://www.mofa.go.jp/files/000486596.pdf> (2021 年 5 月查阅)。
- [B09] 人工智能安全的具体问题, Dario Amodei (谷歌大脑), Chris Olah (谷歌大脑), Jacob Steinhardt (斯坦福大学), Paul Christiano (加州大学伯克利分校), John Schulman (OpenAI), Dan Man'el (谷歌大脑), 2016 年 3 月。
<https://arxiv.org/pdf/1606.06565> (2021 年 5 月查阅)。
- [B10] 可解释的人工智能: 基础知识, 政策简报, 发布日期: 2019 年 11 月 DES6051, ISBN: 978-1-78252-433-5, 英国皇家学会。
- [B11] 机器学习数据标注的终极指南 www.cloudfactory.com/data-labeling-guide (2021 年 5 月查阅)。
- [B12] Pei 等人, 深度探索: 深度学习系统的自动白盒测试, ACM 操作系统原理研讨会论文集 (SOSP' 17), 2017 年 1 月份。
- [B13] Sun 等人, 测试深度神经网络,
https://www.researchgate.net/publication/323747173_Testing_Deep_Neural_Networks, (2021 年 5 月查阅)。
- [B14] A. Odena 和 I. Goodfello, TensorFuzz: 用覆盖率引导的 Fuzzing 来调试神经网络 ArXiv 电子版, 2018 年 7 月, <https://arxiv.org/pdf/1807.10875> (2021 年 5 月查阅)。
- [B15] Riccio, V 等人, 测试基于机器学习的系统: 系统隐射。实证软件工程,
<https://link.springer.com/article/10.1007/s10664-020-09881-0> (2021 年 5 月查阅)。
- [B16] Baudel, Thomas 等人, 解决增强型商业决策系统中的认知偏差,
<https://arxiv.org/abs/2009.08127> (2021 年 5 月查阅)。
- [B17] Papernot, N. 等人, 机器学习中的可转移性: 使用对抗性样本从现象到黑箱攻击, arXiv 预印本 arXiv:1605.07277, 2016。 <https://arxiv.org/pdf/1605.07277> (2021 年 5 月查阅)。
- [B18] 陈 等, 蜕变测试: 挑战和机遇的回顾, ACM 计算机. Surv. 51, 1, Article 4, 2018 年 4 月

- https://www.researchgate.net/publication/322261865_Metamorphic_Testing_A_Review_of_Challenges_and_Opportunities (2021 年 5 月查阅)。
- [B19] Huai Liu, Fei-Ching Kuo, Dave Towey, 和 Tsong Yueh Chen。蜕变测试如何有效地缓解 oracle 问题?, 软件工程学报, 29 (4), 422 - 427, 2014。
- [B20] James Whittaker, 探索性软件测试:指导测试设计的技巧、花招、向导法和技术, 1。艾迪森-韦斯利专业出版社, 2009。
- [B21] L. Wilkinson, A. Anand, and R. Grossman。高维视觉分析:由成对的点分布视图引导的交互式探索。计算机图形学与可视化, IEEE Transactions on, 12(6):1363 - 1372, 2006, <https://www.cs.uic.edu/~wilkinson/Publications/sorting.pdf> (2021 年 5 月查阅)。
- [B22] Ryan Hafen 和 Terence Critchlow, EDA 和 ML -大规模数据分析的完美组合, IEEE 第 27 届并行和分布式处理国际研讨会 2013, <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6651091> (2021 年 5 月通过)。
- [B23] Breck, Eric, Shanqing Cai, Eric Nielsen, Michael Salib, 和 D. Sculley, 机器学习测试评分:机器学习生产准备和技术债务减少的标准, 计算机工程与应用, 2017, <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8258038> (于 2021 年 5 月查阅)。
- [B24] Harman, 人工智能在软件工程中的作用, 第一届软件工程中的人工智能协同实现国际研讨会, 第 1-6 页。IEEE, 2012 年 6 月, <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6227961> (2021 年 5 月查阅)。
- [B25] Nilambri 等人, Bug 库自动重复检测方法研究, 工程研究与技术国际期刊, 2014, <https://www.ijert.org/research/a-survey-on-automated-duplicate-detection-in-a-bug-repository-IJERTV3IS041769.pdf> (2021 年 5 月查阅)。
- [B26] Kim, D.; Wang, X.; Kim, S.; Zeller, A.; Cheung, S.C.; Park, S. (2011)。 “我应该先修复哪些崩溃? 在早期阶段预测最重要的崩溃, 以确定调试工作的优先次序”, 载于《IEEE 软件工程专题报告》第 37 卷 <https://ieeexplore.ieee.org/document/5711013> (2021 年 5 月查阅)。
- [B27] Mao 等人, Sapienz: Android 应用程序的多目标自动测试, 第 25 届软件测试与分析国际研讨会论文集, 2016 年 7 月, http://www0.cs.ucl.ac.uk/staff/K.Mao/archive/p_issta16_sapienz.pdf (2021 年 5 月查阅)。
- [B28] Rai 等人, 使用带有模糊规则库的蜜蜂配对优化算法进行回归测试用例优化, 世界应用科学杂志 31 (4): 654-662, 2014, https://www.researchgate.net/publication/336133351_Regression_Test_Case_Optimization_Using_Honey_Bee_Mating_Optimization_Algorithm_with_Fuzzy_Rule_Base (2021 年 5 月查阅)。
- [B29] Dusica Marijan, Arnaud Gotlieb, Marius Liaaen。一种优化持续集成开发和测试实践的学习算法, 软件学报, 2018 年 11 月
- [B30] Tosun 等, 基于人工智能的软件缺陷预测方法:应用案例研究, 第 22 届人工智能创新应用大会论文集 (IAAI-10), 2010。
- [B31] Kim 等, 从缓存历史中预测故障, 第 29 届国际软件工程会议 (ICSE'07), 2007。

- [B32] Nagappan 等人, 组织结构对软件质量的影响: 实证案例研究, 第 30 届软件工程国际会议论文集 (ICSE'08), 2008 年 5 月。
- [B33] Kuhn 等人, 软件测试中的软件故障交互及其影响, 软件工程学报, vol. 30, no. 4. 6, (2004 年 6 月) 第 418-421 页。

12.4 其他参考资料

以下参考资料指向互联网上的信息。尽管这些参考资料在出版时已被检查过, 但如果这些参考资料已不再可用, ISTQB® 也不承担任何责任。

- [R01] 维基百科贡献者, “AI 效应”, 维基百科, https://en.wikipedia.org/wiki/AI_effect (2021 年 5 月查阅)。
- [R02] <https://mxnet.apache.org/> (2021 年 5 月查阅)。
- [R03] <https://docs.microsoft.com/en-us/cognitive-toolkit/> (2021 年 5 月查阅)。
- [R04] IBM 沃森, <https://www.ibm.com/watson/ai-services>。
- [R05] <https://www.tensorflow.org/> (2021 年 5 月查阅)。
- [R06] <https://keras.io/> (2021 年 5 月查阅)。
- [R07] <https://pytorch.org/> (2021 年 5 月查阅)。
- [R08] https://scikit-learn.org/stable/whats_new/v0.23.html (2021 年 5 月查阅)。
- [R09] 英伟达公司简介, <https://www.nvidia.com/en-us/data-center/volta-gpu-architecture/> (2021 年 5 月查阅)。
- [R10] 云计算 TPU, <https://cloud.google.com/tpu/> (2021 年 5 月查阅)。
- [R11] 边缘 TPU, <https://cloud.google.com/edge-tpu/> (2021 年 5 月查阅)。
- [R12] 英特尔® Nervana™ 神经网络处理器提供了深度学习模型演化所需的规模和效率, <https://www.intel.ai/nervana-nnp/> (2021 年 5 月查阅)。
- [R13] EyeQ 的演变 <https://www.mobileye.com/our-technology/evolution-eyeq-chip/> (2021 年 5 月查阅)。
- [R14] 图像网络 - <http://www.image-net.org/> (2021 年 5 月查阅)。
- [R15] 谷歌的 BERT - <https://github.com/google-research/bert> (2021 年 5 月查阅)。
- [R16] <https://www.kaggle.com/datasets> (2021 年 5 月查阅)。
- [R17] <https://www.kaggle.com/paultimothymooney/2018-kaggle-machine-learning-data-science-survey> (2021 年 5 月查阅)。
- [R18] MLCommons - <https://mlcommons.org/> (2021 年 5 月查阅)。
- [R19] DAWNBench - <https://dawn.cs.stanford.edu/benchmark> (2021 年 5 月查阅)。

- [R20] MLMark - <https://www.eembc.org/mlmark> (2021 年 5 月查阅)。
- [R21] <https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment> | Shaping Europe's digital future (europa.eu) (2021 年 8 月查阅)。
- [R22] <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai> (2021 年 8 月查阅)。
- [R23] 谷歌的 GraphicsFuzz, <https://github.com/google/graphicsfuzz> (2021 年 5 月查阅)。
- [R24] <http://www.openrobots.org/morse/doc/0.2.1/morse.html> (2021 年 5 月查阅)。
- [R25] <https://ai.facebook.com/blog/open-sourcing-ai-habitat-a-simulation-platform-for-embodied-ai-research/> (2021 年 5 月查阅)。
- [R26] <https://www.nvidia.com/en-gb/self-driving-cars/drive-constellation/> (2021 年 5 月查阅)。
- [R27] <https://uk.mathworks.com/discovery/artificial-intelligence.html#ai-with-matlab> (2021 年 5 月查阅)。

13. 附录 A - 缩写

缩写	描述
AI	人工智能
AIaaS	人工智能即服务
API	应用程序编程接口
AUC	曲线下面积
DL	深度学习
DNN	深度神经网络
EDA	探索性数据分析
EU	欧盟
FN	假阴性结果
FP	假阳性结果
GDPR	通用数据保护条例
GPU	图形处理器
GUI	图形用户界面
LIME	对于局部可理解的与模型无关的解释
MC/DC	改进的条件 判定覆盖率
ML	机器学习
MR	蜕变关系
MSE	均方误差
MT	蜕变测试
NLP	自然语言处理
ROC	接受者操作特性曲线
SUT	被测系统
SVM	支持向量机
TN	真阴性结果
TP	真阳性结果
XAI	可解释的人工智能

14. 附录 B - 人工智能专用或其他术语

术语名称	解释
正确性	通过测量正确预测的占比来评估分类器的一个机器学习功能绩效度量（依照 ISO/IEC TR 29119-11）。
激活函数	在神经网络的神经元中，根据输入信息决定输出信息的方程式。
激活值	神经网络的神经元中激活函数的输出值。
对抗攻击	故意使用对抗样本使机器学习模型失败。
人工智能即服务（AIaaS）	以人工智能以及人工智能开发服务为中心的软件授权和交付模式。
人工智能组件	提供人工智能功能的组件。
人工智能效应	跟随科技的发展，曾经被标记为人工智能的系统已经不再被认为是人工智能的情况（ISO/IEC TR 29119-11）。
基于人工智能的系统	一个集合了一个或多个人工智能组件的系统。
人工智能专用处理器	一种经专门设计，用来加速人工智能应用的硬件。
算法偏差	一种由机器学习算法引起的偏差。
注解	一种通过识别有边框图像中的物体，并为分类器提供做好标记的数据的行为。
曲线下面积（AUC）	用来衡量分类器能否能够很好的区分两个类别的度量。
人工智能（AI）	设计好的系统获取、处理、新增以及应用知识与技能的能力（ISO/IEC TR 29119-11）。
关联	一种分辨样本间关系与依赖的无监督式学习技术。
增强	基于已知数据集创建新的数据点的行为。
自动化偏差	由于一个人比起其他来源，更喜欢自动化决策系统做出的推荐，从而产生的一种偏差。 同义词：自满偏差。
自治系统	一种无人干预的情况下能够长期运行的系统。
自治	系统在无人干预的情况下可以长期运行的能力（ISO/IEC TR 29119-11）。
贝叶斯模型	一种使用概率表示模型输入与输出的不确定性的统计模型。
贝叶斯方法	一种考虑使用先验概率分布与后验概率分布作为参数代入统计模型的方法。
偏差	进行比较时，对其中特定对象、人或者群体的系统性待遇差距。
大数据	大量的数据集，这些数据的特征在体积、种类、速率与/或者可变性方面需要使用特殊的科技或方法进行处理。
基于用例的推论	根据过去相似问题的解决办法，来解决新问题的方法。
聊天机器人	一种使用文本或者文本转语音来进行对话的应用程序。
分类	一类给出输入值并以此预测输出值的机器学习方法（依照 ISO/IEC TR 29119-11）。
分类器	一种用来分类的机器学习模型。 同义词：分类模型。
聚类	一种将相似数据点聚合在一起的机器学习方法。
聚类算法	一种将相似对象聚合成簇的机器学习算法。
概念漂移	由于用户期望、行为以及操作环境的改变，导致机器学习模型预测结果的感知正确性在一段时间后发生的改变。
混淆矩阵	一种用来总结分类算法的机器学习功能表现的方法。
数据采集	一种为了使用机器学习模型来解决商业问题，而获取其相关数据的行为。
数据标注	一种为原始数据添加有意义的标签，从而协助机器学习分类的行为。

术语名称	解释
数据管道	一种做数据准备的方法，为支持机器学习算法训练或机器学习模型预测提供输入数据。
数据点	将一组由一次观察构成的一个或多个测量值作为数据集的一部分。
数据投毒	对机器学习模型的训练或者输入数据进行人为恶意的操纵。
数据准备	在机器学习流程中的数据采集、数据预处理以及特征工程行为。
数据预处理	在机器学习流程中的数据清洗、数据转换、数据增强以及数据采样行为。
数据可视化	一种使用图像表达数据关系、数据趋势以及数据模式的方法。
数据集	一组在机器学习中用来训练、评估、测试以及预测的数据的集合。
判定阈值	将预测的结果转换为是否高于或低于该值的二元结果。 同义词：歧视阈值。
判定树	一种树状机器学习模型，该模型的节点代表判定，分支代表可能的结果。
演绎分类器	一种输入值基于对推理与逻辑的应用的分类器。
深度学习（DL）	使用多层神经网络的机器学习。
深度神经网络	一种包含多层神经元的神经网络。 同义词：多层感知器。
缺陷预测	一种预测测试对象内将出现缺陷的区域或存在缺陷数量的技术。
确定性系统	可以从一组给定的输入和开始状态产生一组相同的输出和最终状态的系统。
边缘计算	分布式体系架构的一部分，在该部分中，信息处理在接近信息使用的地方进行。
周期	机器学习训练的迭代在整个训练数据集上。
演变	从较低、较简单或较差的状态向较高、较复杂或较好的状态连续变更的过程。
专家系统	一种基于人工智能的系统，它通过从人类专业知识发展的知识库中推理来解决特定领域或应用领域的问题。
可解释性	基于人工智能系统对于给定结果的理解程度 (ISO/IEC TR 29119-11)。
可解释人工智能（XAI）	该领域的研究涉及理解影响人工智能系统输出的因素。
探索性数据分析（EDA）	交互式、假设驱动和可视化探索的数据用于支持特征工程。
F1-分数	一种机器学习函数性能度量，它用于评估在检索率和精度之间提供平衡的分类器。
假阴性（FN）	一种机器学习预测模型，该模型错误地预测了反向类。
假阳性（FP）	一种机器学习预测模型，该模型错误地预测了正向类。
特征	一种用于机器学习算法训练和机器学习模型预测的输入数据的独立测量属性。
特征工程	在原始数据中最能代表机器学习模型中应显示的基本关系的属性，这些属性被识别用来训练数据（ISO/IEC TR 29119-11）。
弹性	依照 ISO/IEC TR 29119-11 系统在其初始规范之外的上下文中的工作能力 (根据 ISO/IEC TR 29119-11 标准)。
模糊逻辑	一种基于部分真值概念的逻辑，用 0 到 1 之间的确定性因素来表示。
通用人工智能	在所有认知能力方面表现出可与人类相媲美的智能行为的人工智能。 同义词：强人工智能。
通用数据保护条例（GDPR）	适用于欧盟和欧洲经济区公民数据保护和隐私的欧盟 (EU) 规则。
图形处理器（GPU）	一种特定应用的集成电路，它被设计用于操纵和改变存储器，以加速在用于输出到显示设备的帧缓冲区中生成图像。
真实数据	由直接观察和测量所提供的已知真实或正确的信息。
超参数	一个用于控制机器学习模型的训练或设置机器学习模型的配置参数。
超参调优	基于特定目标确定最优超参数的活动。

术语名称	解释
不恰当偏差	一种导致系统产生对特定群体不利影响的结果的偏差。
智能代理	通过观察和行动指导其活动实现目标的自主程序。
簇间度量	度量不同簇中数据点的相似性指标。
可解释性	对基本的人工智能技术工作原理的理解程度 (ISO/IEC TR 29119-11)。
簇内度量	度量簇内数据点的相似性指标。
K 最近邻算法	一种分类方法，根据最接近数据点的组成员关系来估计数据点的组成员可能性。
学习算法	基于训练数据集的属性生成机器学习模型的程序。
LIME 法	用于解释机器学习模型预测的本地化可解释模型的不可知论解释程序。
线性回归	一种统计技术，当目标变量为数值时，通过对观测数据拟合线性方程来模拟变量之间的关系。
逻辑回归	一种统计技术，当目标变量是分类变量而不是数字变量时，对变量之间的关系进行建模。
机器学习 (ML)	使用计算技术使系统从数据或经验中学习的过程 (ISO/IEC TR 29119-11)。
均方误差 (MSE)	估计值与实际值之间的平均平方差的统计度量。
机器学习算法	一种用于从训练数据集创建机器学习模型的算法。
机器学习基准套件	一个用于比较机器学习模型和机器学习算法的评估度量范围的数据集。
机器学习框架	一种支持创建机器学习模式的工具或库。
机器学习功能	由机器学习模型实现的功能，如分类、回归或群集。
机器学习模型评价	将实现的机器学习功能特征度量与所需的标准和其他机器学习模型的标准进行比较的过程。
机器学习模型训练	将机器学习算法应用于训练数据集以创建机器学习模型的过程。
机器学习模型调优	测试超参数以实现最佳性能的过程。
机器学习系统	集成一个或多个机器学习模型的系统。
机器学习 workflow	用于管理机器学习模型的开发和部署的一系列活动。
多智能体系统	由多个智能体组成的系统。
狭义人工智能	人工智能专注于单个明确定义的任务来解决特定问题 (ISO/IEC TR 29119-11)。 同义词：弱人工智能。
自然语言处理 (NLP)	一种计算领域，提供了阅读、理解和从自然语言中获取意义的能力。
神经网络	一种原始处理元件网络，通过加权链路连接，权值可调，其中每个元件通过对其输入值应用非线性函数产生一个值，并将其传输给其他元件或作为输出值显示 (ISO/IEC 2382)。 同义词：人工神经网络。
神经网络木马	利用数据中毒攻击，将漏洞注入神经网络，以便日后加以利用。
神经形态处理器	一种用来模仿人类大脑的生物神经元的集成电路。
神经元	神经网络中的一个节点，通常接收多个输入值并产生一个激活值。
声音	数据的失真或损坏。
非确定性系统	给定一组特定的输入和启动状态，系统并不总是产生相同的输出和最终状态。
离群值	在数据分布的整体模式之外的观察。
过拟合	产生与训练数据集过于紧密对应的机器学习模型，导致模型难以推广到新数据依照 ISO/IEC TR 29119-11。
感知器	只有单层和一个神经元的神经网络。
精确度	用于评估分类器的机器学习功能特性度量，它测量正确的预测阳性的比例依照 ISO/IEC TR 29119-11。

术语名称	解释
预训练模型	机器学习模型在获得时已经经过训练。
概率性系统	用概率描述其行为的系统；因此，它的产出无法被完美预测。
程序推理	一种可以用来构造在动态条件下可以执行复杂任务的实时论证系统的人工智能技术。
随机森林	通过建立和运行许多决定树来确定是否产生可以用来预测平均个体树的用来分类、回归其他任务机器学习技术效果。
推理技术	从运用了逻辑技术的可获得的信息中产生结论的人工智能（依照 ISO/IEC TR 29119-11）。
召回	用于评估一个分类的机器学习函数表现，用来衡量被正确预测的实际阳性值的比例，依照 ISO/IEC TR 29119-11。 同义词：灵敏度。
接收者操作特征曲线	一个表现二元分类器能力的图形，它的区分阈值是可变的。
回归	给定输入情况下输出值是可数值模拟或者连续的机器学习函数类型依照 ISO/IEC TR 29119-11。
回归模型	在给定的数字输入可预测输出是连续值的机器学习模型（在 ISO/IEC TR 29119-11 之后）。
强化学习	通过一个测试和反馈过程来实现目标的用来构建机器学习模型的活动依照 ISO/IEC TR 29119-11。
回报函数	一种描述强化学习成功的函数。
黑客奖励	人工智能代理为最大化奖励功能而执行的活动，这些活动不利于实现预想的目标。，依照 ISO/IEC TR 29119-11。
拟合度	一种用来描述数据点与拟合回归直线接近程度的度量模型。 同义词：相关性。
规则引擎	一系列当特定条件满足情况下决定可能发生行为的规则集。
安全性	在特定条件下，一个系统不趋向人类生命、健康、财富或者环境濒危环境的期望。
样本偏差	一种偏差类型，其中的数据集不能完全代表机器学习被应用 的数据空间。
搜索算法	一种系统浏览一系列所有可能状态或者结构直到目标状态或者结构达到的算法依照 ISO/IEC TR 29119-11
自我学习系统	一种基于试验和错误学习来校正表现的自适应系统依照 ISO/IEC TR 29119-11
轮廓系数	一组基于集群内和集群间差别范围在-1 和+1 之间的度量数据 同义词：轮廓分数
超级人工智能	一种远超过人类能力的人工智能技术
监督学习	通过输入数据和它的相关特性来进行测试一种机器学习模型。
支持向量机	被一个超平面分离的多维空间下数据点被视为矢量的一种机器学习技术。
技术奇点	当技术进步不再受人为控制情况下未来可能达到的一个点依照 ISO/IEC TR 29119-11。
预取测试结果问题	用来判断在指定输入和状态下测试是否通过的挑战。
训练数据集（也叫做训练集）	用于训练一个机器学习模型的数据集。
转移学习	一种用于改进实现不同相关任务预训练机器学习模型的技术。

术语名称	解释
透明性	人工智能基础系统上算法和数据集的可视化程度依照 ISO/IEC TR 29119-11。
真阴性	一种按照与模型指示方向相反趋势发展的预测。
真阳性	一种按照与模型指示方向相同趋势发展的预测。
欠拟合	生成的 ML 模型不能反映训练数据集的潜在趋势，导致模型难以做出准确的预测 (ISO/IEC TR 29119-11)。
无监督学习	使用未标记的数据集从输入数据训练 ML 模型。
验证数据集（也叫做验证集）	用于评估训练过的 ML 模型的数据集，用来调整模型。
冯诺伊曼架构	一种计算机结构，由五个主要部件组成：存储器、中央处理单元、控制单元、输入和输出。
权重	神经网络中神经元之间连接的内部变量，影响其计算输出的方式，并且随着神经网络被训练而改变。