

普通高中



教学参考资料

信息技术

选择性必修 4 人工智能初步



华东师范大学出版社

普通高中
教学参考资料

信息技术

选择性必修 4
人工智能初步

总主编：李晓明

副总主编：赵健 李锋

本册主编：于晓雅

本册副主编：熊雪亭

编写人员(按姓氏笔画排序)：

马静 李威 李娟 郭培培 董玥
韩冬兵 熊雪亭 薛征

责任编辑：王健

美术设计：卢晓红 储平

普通高中 信息技术 选择性必修4 人工智能初步 教学参考资料

上海市中小学(幼儿园)课程改革委员会组织编写

出版发行 华东师范大学出版社(上海市中山北路3663号)

印 刷 上海昌鑫龙印务有限公司

版 次 2022年8月第1版

印 次 2025年8月第7次

开 本 890毫米×1240毫米 1/16

印 张 13.25

字 数 318千字

书 号 ISBN 978-7-5760-2954-3

定 价 28.50元

版权所有·未经许可不得采用任何方式擅自复制或使用本产品任何部分·违者必究

如发现内容质量问题,请拨打电话 021-60821714

如发现印、装质量问题,影响阅读,请与华东师范大学出版社联系。电话: 021-60821711

全国物价举报电话: 12315

说 明

《普通高中 信息技术 选择性必修 4 人工智能初步 教学参考资料》根据教育部颁布的《普通高中信息技术课程标准(2017 年版 2020 年修订)》和高中信息技术教科书的内容和要求编写,与信息技术教科书配套,供高中一年级使用。

本书由华东师范大学、上海市信息技术教育教学研究基地(上海高校“立德树人”人文社会科学重点研究基地)主持编写,经上海市中小学教材审查委员会审查准予使用。

编写过程中,上海市中小学(幼儿园)课程改革委员会专家工作委员会、上海市教育委员会教学研究室、上海市课程方案教育教学研究基地、上海市心理教育教学研究基地、上海市基础教育教材建设研究基地等单位给予了大力支持。在此表示感谢!

欢迎广大师生来电来函指出书中的差错和不足,提出宝贵意见。出版社电话:021-60821711。

声明 按照《中华人民共和国著作权法》第二十五条有关规定,我们已尽量寻找著作
权人支付报酬。著作权人如有关于支付报酬事宜可及时与出版社联系。

目 录



第一章 人工智能初识

一、本章学科核心素养的渗透	1
二、本章知识结构	2
三、本章项目活动的设计思路与实施建议	2
四、本章教学课时分配建议	3

第一节 人工智能：从体验到思考 4

一、教学目标与重点	4
二、内容分析与建议	4
三、习题分析与提示	7
四、核心概念精解与技术工具介绍	10
五、教学参考资源	12
六、教学参考案例	12

第二节 人工智能的发展历程 16

一、教学目标与重点	16
二、内容分析与建议	16
三、习题分析与提示	18
四、核心概念精解与技术工具介绍	18
五、教学参考资源	20
六、教学参考案例	20

第二章 人工智能实现

一、本章学科核心素养的渗透	24
二、本章知识结构	25
三、本章项目活动的设计思路与实施建议	26
四、本章教学课时分配建议	27

第一节 启发式搜索 27

一、教学目标与重点	27
二、内容分析与建议	28
三、习题分析与提示	29
四、核心概念精解与技术工具介绍	31
五、教学参考资源	33
六、教学参考案例	34

第二节 专家系统 39

一、教学目标与重点	39
二、内容分析与建议	39
三、习题分析与提示	41
四、核心概念精解与技术工具介绍	43
五、教学参考资源	48
六、教学参考案例	52

第三节 机器学习 64

一、教学目标与重点	64
二、内容分析与建议	65
三、习题分析与提示	66
四、核心概念精解与技术工具介绍	73
五、教学参考资源	75
六、教学参考案例	76

第三章 漫游深度学习的世界

一、本章学科核心素养的渗透	80
二、本章知识结构	81
三、本章项目活动的设计思路与实施建议	82
四、本章教学课时分配建议	82

第一节 人工神经元与单层感知机 83

一、教学目标与重点	83
二、内容分析与建议	84
三、习题分析与提示	85
四、核心概念精解与技术工具介绍	89
五、教学参考资源	91
六、教学参考案例	92

第二节 多层感知机与人工神经网络 98

一、教学目标与重点	98
二、内容分析与建议	99
三、习题分析与提示	100
四、核心概念精解与技术工具介绍	101
五、教学参考资源	103
六、教学参考案例	103

第三节 深度学习 110

一、教学目标与重点	110
二、内容分析与建议	110
三、习题分析与提示	111
四、核心概念精解与技术工具介绍	114
五、教学参考资源	116
六、教学参考案例	116

第四节 卷积神经网络	122
一、教学目标与重点	122
二、内容分析与建议	123
三、习题分析与提示	126
四、核心概念精解与技术工具介绍	147
五、教学参考资源	162
六、教学参考案例	162

第四章 人工智能未来发展

一、本章学科核心素养的渗透	166
二、本章知识结构	167
三、本章项目活动的设计思路与实施建议	168
四、本章教学课时分配建议	169

第一节 新一代人工智能技术 169

一、教学目标与重点	169
二、内容分析与建议	170
三、习题分析与提示	171
四、核心概念精解与技术工具介绍	172
五、教学参考资源	173
六、教学参考案例	175

第二节 潜在的风险 180

一、教学目标与重点	180
二、内容分析与建议	180
三、习题分析与提示	182
四、核心概念精解与技术工具介绍	184
五、教学参考资源	184
六、教学参考案例	184

第三节 伦理规范	187
一、教学目标与重点	187
二、内容分析与建议	187
三、习题分析与提示	190
四、核心概念精解与技术工具介绍	191
五、教学参考资源	191
六、教学参考案例	194

人工智能初识

一、本章学科核心素养的渗透

人工智能是推动国家战略性新兴产业融合集群发展增长引擎之一，是引领新一轮科技革命和产业变革的关键技术。2017年7月，国务院印发《新一代人工智能发展规划》，明确指出人工智能将成为国际竞争的新焦点，应在中小学阶段设置人工智能相关课程、逐步推广编程教育、建设人工智能学科，培养复合型人才，形成我国人工智能人才高地。本章作为《人工智能初步》的第一章，将智能与人工智能的基本概念、图灵测试及人工智能的发展历程等内容融入到以“走进人工智能世界”为主题的创新科技周活动中，引导学生通过项目学习、小组讨论与合作、体验思考、编程实践等方式深入理解人工智能的内涵、发展历程及如何实现简单的人工智能应用。本章内容对应的学科核心素养要求体现在以下几个方面：

- 通过“走进人工智能世界”创新科技周活动的实践，选取恰当方式，如网络搜索等，获取有效信息，分析并提炼有价值的信息，完成项目活动。（信息意识）
- 在合作解决问题的过程中，与团队成员共享信息，分析讨论，实现信息的最大价值。（信息意识）
- 针对项目展示活动，搜集并了解人工智能的新进展、新应用，如自动翻译、语音识别等，并能适当运用在学习和生活中。（信息意识）
- 针对创新科技周活动，选取运用合适的数字化学习策略与工具管理学习过程与资源，完成任务，创作作品。（数字化学习与创新）
- 通过项目实践，能够利用开源人工智能应用框架，搭建简单智能应用，总结方法，并能迁移到其他问题的解决中。（计算思维）
- 通过创新科技周活动，能客观认识智能技术对社会生活的影响，认识人工智能在信息社会中的重要作用；具有一定的信息安全意识与能力，如了解验证码的作用及安全性；对于人工智能发展所产生的新观念和新事物，具有积极学习的态度、理性的判断和负责的应用能力。（信息社会责任）

二、本章知识结构

本章分为两节,主要内容是了解人工智能中“智能”的本质及人工智能的发展历程。每一节都通过项目学习、探究体验来引导学生逐步深入理解知识。

第一节从“什么是智能”这一问题入手,引出关于智能的各种思想与概念,并通过探究、讨论、理解各种概念的区别与联系,从而了解什么是人工智能。然后通过“搭建简易聊天机器人”活动引出问题——如何判定机器是否具有智能,从而引出图灵测试,通过设计自己的图灵测试和“中文屋”问题讨论等活动引导学生深入理解图灵测试的内涵,并补充介绍图灵测试的应用——验证码,加深学生对图灵测试的理解。最后通过介绍人工智能相关观念和三种流派,进一步阐述关于人工智能的各种观点及应用领域,帮助学生进一步认识人工智能的研究方向。

第二节直接介绍人工智能的发展历程,提出人工智能发展的三个阶段,通过符号编程体验以及思考讨论等活动理解每一阶段的具体内涵。最后通过在线翻译体验与编程实践语音识别项目,加深学生对人工智能的理解与感性认识,促进学生形成自己的人工智能观。

本章主要知识与概念的关系如图 1-1 所示。

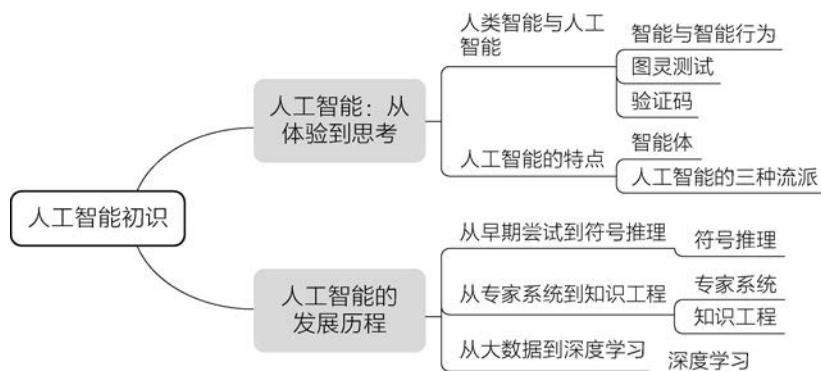


图 1-1 主要知识与概念的关系

三、本章项目活动的设计思路与实施建议

本章学生将围绕以“走进人工智能世界”为主题的创新科技周项目开展学习。让学生通过项目活动实践与小组合作,理解智能、人工智能相关概念,领会图灵测试的原理,了解人工智能发展历程,并通过实践搭建人工智能应用,形成对人工智能的初步认识。

本章的项目任务主要是针对“走进人工智能世界”创新科技周准备相关展板及体验活动。通过准备科技周活动,学生小组制订方案、搜集资料、讨论分析、制作作品,通过以上的项目实践活动,理解人工智能相关概念,了解人工智能发展历程。

本章的项目活动分为四个子任务,分别是“讨论人的智能与人工智能的关系”“揭秘图

灵测试”“制作人工智能发展历程图表”和“体验身边的人工智能”，引导学生层层递进，逐步进入对人工智能更深的理解层面。任务一主要是学生通过小组讨论、网络资料搜索等方式了解人的智能与人工智能的同与不同；任务二主要是通过阅读文献、观看视频等理解图灵测试及其应用准则以及验证码的基本原理；任务三主要通过小组协作，搜索并阅读材料，制作人工智能发展历程图表，了解人工智能的发展过程；任务四主要是在前三个任务的基础上，寻找身边具有智能的应用，并体验身边的人工智能应用。四个任务层层递进，清晰、系统地体现了人工智能的发展脉络，从理论过渡到生活中的实际应用，提高学生学习人工智能的兴趣。项目采用分组方式进行，项目结束后每组应形成自己的科技周展示活动相关展板、资料等。最后通过科技周展示分享各组成果。

本章的项目任务需要学生以小组合作方式完成。因此，在项目开始前，先要对学生进行分组（建议每组3~5人）。可结合任务需要与学生自主选择对学生进行分组，尽量遵循组间同质、组内异质的原则，以便不同能力的学生均能得到充分的发展。分组后，学生围绕项目开展讨论，制订项目实施方案并明确小组成员的分工。

在项目活动过程中，涉及到较多的资料搜集与作品制作工作，教师可提前提供一些相关文献、资料及作品制作方法介绍文档等，辅助学生实践项目活动。

项目活动的评价要结合过程性评价和总结性评价，评估学生发展，监控学习进展，检查学习效果，促进学生的学习，保障项目的完成。评价可以从学生的参与程度、贡献程度、协作精神及作品效果等多方面进行。项目完成后，还可以结合科技周的活动效果进行评价。可采用教师、学生、组内、组间的多元评价。特建议制订一份评价策略表，可参考表1-1。

表1-1 项目活动评价策略

		评价等级及分值			得分		
		优秀 (80~100分)	良好 (60~79分)	尚需努力 (60分以下)	组内 自评	组间 评价	教师 评价
小组协作(20%)							
科技周 展示活 动效果 (60%)	作品科学性(30%)						
	作品艺术性(15%)						
	作品创造性(15%)						
交流分享(10%)							
总结反思(10%)							
总分(100%)							

四、本章教学课时分配建议

本章以项目活动“走进人工智能世界”为导线，让学生在小组合作、实践探究与项目展

示的过程中了解人工智能的基本内涵、图灵测试的原理、人工智能的发展历程以及身边的人工智能应用,激发学生对于人工智能的兴趣,为后续章节的学习奠定基础。在教学中教师应激发学生的兴趣,让学生在自己的好奇心驱动下通过实践探究逐步去了解本章内容。本章教学课时分配建议如表 1-2 所示。

表 1-2 本章课时分配建议

节次	课时	具体 内 容
第一节	2 课时	第 1 课时:人类智能与人工智能
		第 2 课时:图灵测试
第二节	2 课时	第 3 课时:人工智能的发展历程
		第 4 课时:实践并体验人工智能应用

第一节 人工智能:从体验到思考

一、教学目标与重点

教学目标:

- 通过小组搜集整理资料、讨论分析,了解围绕智能、人工智能及相关概念的主要观点和争论;(信息意识、数字化学习与创新)
- 通过活动实践,了解图灵测试及“中文屋”问题等检验人工智能的设想和方案,理解其中的辩证关系、论据及其局限性。(信息意识)

教学重点:

- 智能与人工智能的内涵;
- 测试人工智能的方法,图灵测试的规则。

二、内容分析与建议

1. 内容分析

本节主要帮助学生理解智能及人工智能的相关概念,领会图灵测试的原理,体会人工智能的内涵。本节主要知识框图如图 1-2 所示。本节是本章学习的开篇也是全书的开篇,统领本章内容,需要在本节中引导学生形成自己的人工智能观。

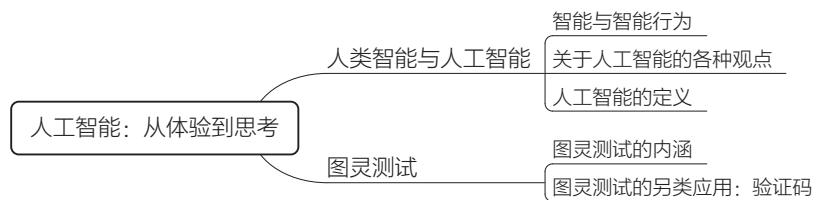


图 1-2 主要知识框图

2. 教学建议

(1) 教学活动建议

本节的重点内容包括智能的定义与图灵测试两个内容。理解智能的定义过程中出现的各种观念与思想,形成自己的智能观,了解如何用图灵测试验证机器是否具有智能,就基本实现了本节的目标。对于智能的定义可通过问题引导、列举说明各种定义的内涵,引导学生讨论分析各种思想的区别与联系。对于图灵测试,则应采用体验式的方式,让学生体验图灵测试的具体过程,并通过“中文屋”、验证码等问题让学生进一步了解图灵测试,激发学生的思考和对人工智能的兴趣。

本节内容围绕智能的定义和图灵测试展开,通过列举智能的定义、体验探索等方式了解智能的定义和图灵测试的内涵,围绕“走进人工智能世界”创新科技周活动展开,引导学生形成自己的智能定义。教师在教学中不必局限于教科书中的案例,可以引导学生结合自己的生活思考什么样的“机器”算是有智能的,也可以结合图灵针对各方对图灵测试的反对意见给出的驳斥做分析,还可以采取游戏的方式引导学生感受图灵测试的过程,设计自己的图灵测试,丰富教学形式,激发学生兴趣。另外,建议教师在课堂中多关注学生的学习反馈,通过提问及对小组活动的观察等及时了解学生对相关内容的掌握情况。

本节虽然以理论性知识的学习为主,但也涉及一些技术的体验和应用。例如,利用NLTK扩展库搭建简易聊天机器人,这些活动任务辅助理论的学习,激发学生兴趣,引导学生思考。并且教科书后续也有很多相似的体验活动,但这类体验活动对学生已有的Python基础有一定要求。因此,建议教师在教学中充分调研学生已有的Python基础,适度调节体验活动进行的节奏,通过这类活动,让学生了解搭建体验应用的基本过程,激发学生的兴趣与思考。另外,本节还涉及一些讨论分析等活动,建议教师组织学生进行小组讨论,把控讨论的节奏,促使学生形成小组讨论的结果,提升讨论效果。讨论时,教师可预设几个问题引导学生。

(2) 项目实施建议

本阶段,要让学生明确项目活动的宗旨、目标、过程和评价标准,制订项目活动的规划流程与分工安排,保证后续项目能够有序实施。

首先,教师可以从“什么是人工智能?”这一问题入手,提出项目主题,介绍“走进人工智能世界”创新科技周的具体安排和要求、相关评价标准和成果展示方法等,帮助学生整体理解和认识活动周的活动内容和相关要求。项目活动分组进行,教师可以向学生说明

项目任务大致安排,如做展板、设计游戏、体验设置等,引导学生依据自身情况结合组间同质、组内异质的方式进行分组,每组3~5人,确定好组长,为学生后续开展项目学习打好基础。

然后,组织学生针对问题“什么是智能?”进行讨论,收集学生的回答,同时向学生展示人工智能发展史上对智能的相关定义,引导小组学生进行分析、讨论,形成自己小组的人工智能定义,并进行分享。

接着,围绕“如何判断一个机器是否有智能?”这一问题展开,组织学生体验搭建简易聊天机器人,并结合小组对人工智能的定义,讨论分析该聊天机器人是否具有智能,从而引出图灵测试,引导学生理解并设计自己的图灵测试。在此过程中,小组学生可以搜集制作人工智能与自己的图灵测试游戏展板,并对设计的游戏进行体验分析,判断其能否测试被测者是否具有智能。接下来通过“中文屋”和验证码等内容引导学生进一步理解图灵测试,在此基础上进行展板设计、班级展示。

最后,学生可以在吸收其他组经验的基础上,进一步设计规划自己的图灵测试游戏,完善本组“人工智能与图灵测试”展板的设计思路,明确分工和任务。

(3) 评价建议

评价方式:过程性评价与作业评价相结合,教师评价与学生自评相结合。

评价量表如表1-3所示。

表1-3 小组协作及展板设计评价量表

		评价等级及分值			得分		
		优秀 (80~100分)	良好 (60~79分)	尚需努力 (60分以下)	组内 自评	组间 评价	教师 评价
小组协作(20%)		积极参与小组讨论分析、参与课堂体验活动、思考相关问题、踊跃交流表达自己的观点	参与小组讨论分析、参与课堂体验活动、思考相关问题、能够交流表达自己的观点	小组活动参与较少,不表达或很少表达自己的观点			
科技周展板、 资料制作效果 (60%)	作品科学性 (30%)	设计富有创造性的图灵测试,展板展示等材料的制作效果良好	能够设计出自己的图灵测试,制作相关展示材料	未能设计出自己的图灵测试,制作的展示材料质量差			
	作品艺术性 (15%)						
	作品创造性 (15%)						
交流分享(20%)		表达能力强,能够将小组成果和想法充分表达出来	能够说明小组成果和想法	对小组成果和想法说明得不清楚			
总分(100%)							

三、习题分析与提示

■ 习题 1 描述

见教科书第 5 页的探究活动“从数据到智慧”。

■ 习题 1 参考答案

1. 金字塔的含义：数据是对现实生活的理性描述，尽可能地从数量上来反映现实世界，数据是信息、知识和智慧的源泉，因而是金字塔中最基础的一个概念。信息是有一定含义的、经过加工处理的、对决策有价值的数据，信息 = 数据 + 处理。知识之所以在数据和信息之上，是因为它要接近行动，且与决策相关。信息虽是数据提炼出的意义，但它的价值往往会在时间效用失效后开始衰减，只有对信息运用归纳、演绎和比较手段进行挖掘，使有价值的部分沉淀下来，并与已存在的人类知识体系相结合，这部分有价值的信息才会转变成知识。智慧位于金字塔的最高一级，是人类解决问题的一种特有的能力，智慧的产生需要基于知识的应用。智慧是人类基于已有的知识，针对物质世界运动过程中产生的问题，根据获得的信息进行分析、对比、演绎，从而找出解决方案的能力。因此，数据、信息、知识、智慧这四者在认知程度上是逐层递进的。

2. 斯腾博格认为（个体）智能是个人从经验中学习、理性思考、记忆重要信息，以及应付日常生活需求的认知能力。斯腾博格提出了智力的三元理论，认为完备的智力理论必须说明智力的内在成分、智力成分与环境间的关系，以及智力成分的外部作用，这三方面构成了智力。斯腾博格的理论强调智力由分析性能力、创造性能力和应用性能力组成，与产生知识、运用知识及创造知识强调的三方面能力是一一对应的，只是斯腾博格的理论侧重在认知活动中需要的能力，产生知识、运用知识、创造知识侧重与知识的关系。

3. 言之合理即可。例如：我认为这一定义是合理的。因为人工智能研究的重点是如何表示知识以及怎样获得知识并使用知识，从而使得计算机能够表示知识、获得知识、使用知识，完成需要人的智力才能完成的工作，因此，“人工智能就是关于知识的科学”这一说法反映了人工智能学科的基本思想和基本内容，是合理的。

■ 习题 2 描述

见教科书第 7 页的探究活动“哪些 App 可以算作是智能的？”。

■ 习题 2 参考答案

1. 可从 App 使用到的技术分析，例如是否以主流人工智能技术为依托。目前人工智能领域的六大研究方向集中在计算机视觉、自然语言处理、机器学习、自动推理、知识表示和机器入学，以这六大方向为基础初步判断 App 是否智能。App 是否有较强的决策能力，也就是指能否进行合理的思考，并采取合理的行动。

2. 言之合理即可，如是否可以模拟人类智能，如听到、理解、表达等。

3. 目前的智能手机和智能 App 在智能方面更多地表现为特定领域的单一智能，如语音识别、图像识别等，往往还欠缺融合多种智能的交互渠道，并且目前的智能 App 主要

依托大数据和模型进行推测,欠缺对情境信息的了解,此外也缺乏对人类情感、情绪的理解以及对于上下文的理解。

4. 根据实际情况分析作答即可,如翻译软件、语音助手软件、购物软件等具备智能推荐、语音识别、自然语言理解等功能的 App 均可归为智能类,而计算器、时钟等可以归为非智能类,再根据情况计算比例即可。

5. 与语音识别、自然语言理解、图像识别、图像处理、大数据搜索和推荐相关的 App 智能水平均较高,不依托于数据和算法,只作为单纯的工具的 App 的智能水平较低,如计算器、指南针等。

■ 习题 3 描述

见教科书第 8 页的规划设计“规划自己的模仿游戏”。

■ 习题 3 参考答案

1. 图灵测试引入第一阶段是为了给第二阶段的说明提供一个基础,第一阶段中伪装成女性的男性被询问者是有智能的,因此如果机器可以替代男性被提问者,完成伪装成女性的任务,则也可以认为机器是有智能的。忽略第一阶段不会对结果有太大影响,第一阶段是作为有效性的基础,在使用时直接采用第二阶段也是可以的。

2. 设计合理即可。例如,游戏规则:询问者通过即时通讯工具与两个用户进行聊天,其中一个用户为人类用户,另一个用户为计算机聊天程序,通过询问问题清单中的问题,判断哪个用户是人,哪个用户是机器。判定胜负的标准:如果将聊天程序判定为人,则聊天程序通过了图灵测试,反之未通过。问题清单合理即可。

3. 不允许抛开问题清单直接提问,抛开问题清单提问可能会导致特意挑选人会犯的错误来进行提问,这样就违背了图灵测试探究机器是否能做到人可以做到的事情的初衷。

4. 略。

■ 习题 4 描述

见教科书第 9~10 页的探究活动“对图灵测试的质疑——塞尔的‘中文屋’问题”。

■ 习题 4 参考答案

1. 塞尔认为房间里的人因为拥有某些特定的工具,能够做出合适的回答,甚至可以让以中文为母语的人以为他能流利地说中文。但是,他其实是完全不懂中文的。塞尔认为,电脑就是这样工作的。它们无法真正地理解接收到的信息,但它们可以运行一个程序,处理信息,然后给出一个智能的印象。因而即便计算机通过了图灵测试,也不能说机器会思考。

2. 言之合理即可。例如:我认为该论据是不充分的,因为首先,房间里的人类、计算机程序、纸、笔构成了一个整体。可以说这个房间里的人类不会中文,但是这个房间“它”会中文。就像大脑的一个细胞不会思考,但整体就会思考。其次,能够让一个人回答出所有中文问题但却不懂中文的规则真的存在吗? 塞尔的“中文屋”测试存在的前提让人质疑。

■ 习题 5 描述

见教科书第 11 页的探究活动“CAPTCHA 的变体”。

■ 习题 5 参考答案

1. 如本题图片所示的是最传统的文字验证码，文字验证码目前已有很多算法可以实现自动识别破解，如 OCR 方法等，因而存在较大的安全隐患。
2. 如本题图片所示的方式能够在一定程度上提升安全性，因为，要实现算法自动识别该验证码，需要大量收集该网站所使用的图片数据，并进行建模自动识别，提高了破解的难度，在一定程度上提升了安全性能。
3. 与图形验证码相比，短信验证码首先能够验证身份，因为目前手机卡实名制，验证了手机验证码也就相当于验证了用户身份，在一定程度上提高了安全性。但短信验证码也有它的弱点，其一是来自智能平台上的短信木马，这类短信木马通过发送短链接短信，让用户在不知情的情况下下载安装木马，然后拦截短信验证码，重置支付软件账号密码；其二是无线电监听，简单来说就是通过伪基站对用户手机进行监听，获得短信内容然后进行盗窃活动；最后，手机丢失也是安全隐患之一。

■ 习题 6 描述

见教科书第 12~13 页的探究活动“关于人工智能的各种观点”。

■ 习题 6 参考答案

教科书第 12 页的思考题：

1. 智能体是指能够智能地感知环境，从环境中学习并与环境进行交互的系统。智能体的观点认为：人工智能的目标就是创建能够执行某类任务，并具备认知功能的智能体。机器人是一种能够半自主或全自主工作的智能机器，具有感知、决策、执行等基本特征，可以辅助甚至替代人类完成特定的工作。智能体和机器人都能够感知外部环境，决策并执行特定任务。但智能体更加强调环境，可以看作是驻留在环境中的实体，从环境中获得相关数据，并执行对环境产生影响的行动，既可以是硬件智能体（如智能机器人等），也可以是软件智能体。

2. “清扫机器人”的核心组成部分包括以下五个：传感部分，相当于人的五官，起到对外界的感知作用；控制部分，相当于人的大脑，连接并控制机器人各部分；驱动部分，相当于人的肢体，被控制完成动作；吸尘部分，嵌入机器人内部的真空吸尘器；电源部分，提供机器所需要的动力系统，也是至关重要的一部分。

3. 将智能手机看作多个智能体的组合更合适，其最主要的部分包括：重力感应器（支持摇晃切换相关功能）、手机距离传感器（支持远离手机时锁屏等）、手机电子罗盘（指示方向）、光电传感器（支持屏幕自动调节亮度）、全球定位功能（支持导航、定位）、陀螺仪（转动、偏转角度、游戏），以上用于感知，CPU 和 GPU 用于计算也就是决策，手机屏幕、扬声器等用于执行。

教科书第 13 页的思考题：

1. 关于人工智能的定义或界定，言之合理即可。如可以列举以下定义：
 - 人工智能是关于知识的科学，是怎样表示知识以及怎样获得知识并使用知识的科学。
 - 人工智能就是研究如何使计算机去做过去只有人才能做的智能工作的科学。

• 人工智能是研究、开发用于模拟、延伸和扩展人的智能理论、方法、技术及应用系统的一门新的技术科学。

• 人工智能是类人行为、类人思考、理性的思考、理性的行动。

(可从三种流派方面阐述其关联与区别,言之合理即可)

2. 关于讨论关于人工智能的每种观点,言之合理即可。例如,人工智能是关于知识的科学,是怎样表示知识以及怎样获得知识并使用知识的科学。这一定义强调知识的符号表示,认知就是处理符号,推理就是采用启发式知识及启发式搜索对问题求解的过程,而推理过程又可以用某种形式化的语言来描述。采用此种定义存在“常识”问题以及不确定性事物的表示和处理问题。

3. 略。

四、核心概念精解与技术工具介绍

1. 核心概念

(1) 智能

智能,是智慧和能力的总称,中国古代思想家一般把“智”与“能”看作是两个相对独立的概念。

《荀子·正名篇》:“所以知之在人者谓之知,知有所合谓之智。所以能之在人者谓之能,能有所合谓之能”。其中,“智”指进行认识活动的某些心理特点,“能”则指进行实际活动的某些心理特点。也有不少思想家把二者结合起来作为一个整体看待。

根据霍华德·加德纳的多元智能理论,人类的智能可以分成八个范畴:语言智能、数学逻辑智能、空间智能、身体运动智能、音乐智能、人际智能、自我认知智能、自然认知智能。

① 语言智能

语言智能是指人对语言的掌握和灵活运用的能力,表现为用词语思考,用语言和词语的多种不同方式来表达复杂意义。

② 数学逻辑智能

数学逻辑智能是指人对逻辑结果关系的理解、推理、思维表达能力,突出特征为用逻辑方法解决问题,有对数字和抽象模式的理解力。

③ 空间智能

空间智能是指人对色彩、形状空间位置的正确感受和表达能力,突出特征为对视觉世界有准确的感知,能产生思维图像,有三维空间的思维能力,能辨别感知空间物体之间的联系。

④ 身体运动智能

身体运动智能是指人的身体协调、平衡能力和运动的力量、速度、灵活性等,突出特征为利用身体交流和解决问题,熟练地进行物体操作以及需要良好动作技能的活动。

⑤ 音乐智能

音乐智能是指人感受、辨别、记忆、表达音乐的能力,突出特征为对环境中的非言语声音,包括韵律和曲调、节奏、音高音质的敏感。

⑥ 人际智能

人际智能是指对他人的表情、说话、手势动作的敏感程度以及对此作出有效反应的能力,表现为个人能觉察体验他人的情绪情感并作出适当的反应。

⑦ 自我认知智能

自我认知智能是指个体认识、洞察和反省自身的能力,突出特征为对自己的感觉和情绪敏感,了解自己的优缺点,用自己的知识来引导决策、设定目标。

⑧ 自然认知智能

自然认知智能是指观察自然的各种形态,对物体进行辨认和分类、能够洞察自然或人造系统的能力。

(2) 人工智能

人工智能指由人创造出来的,具有感知、认知、决策、学习、执行和社会协作能力,符合人类情感、伦理与道德观念的虚拟的或人工的系统。

(3) 图灵测试

“图灵测试”一词来源于英国数学家、逻辑学家、计算机科学家艾伦·图灵 1950 年发表的一篇论文——《计算机器与智能》。图灵测试通过观察机器在回答人们提问时的表现来判断其是否具有智能。假设让一个人与一台机器在同一个房间中,询问者待在另一个房间,且看不见两者中的任何一方。询问者通过一个终端对人和机器提问并接收两者的回答,如图 1-3 所示。如果经过对双方足够多次的问询和接收回答后,询问者仍无法分辨出哪个是人,哪个是机器,那么就认为机器是智能的。

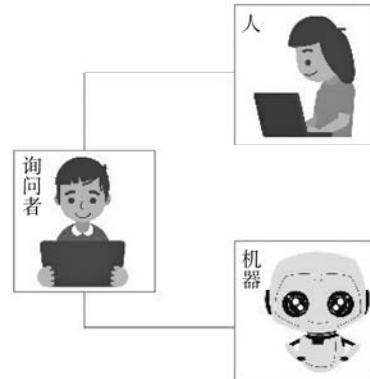


图 1-3 图灵测试

2. 技术工具

NLTK 全称为 natural language toolkit,自然语言处理工具包,是 NLP 研究领域常用的一个 Python 库,它是在 Python 的基础上开发的一个模块,至今已有超过十万行的代码。这是一个开源项目,包含数据集、Python 模块、教程等。

NLTK 是一个高效的 Python 构建的平台,用来处理人类自然语言数据。它提供了易于使用的接口,通过这些接口可以访问超过 50 个语料库和词汇资源(如 WordNet),还有一套用于分类、标记、词干分离、解析和语义推理的文本处理库,以及工业级 NLP 库的封装器和一个活跃的讨论论坛。

NLTK 的安装: `pip install nltk`

在使用 NLTK 库之前,要先安装需要的语料库。

安装语料库:`import nltk`

`nltk.download()`

NLTK 常见操作(NLTK 版本 3.6.2 Python 版本 3.7.5)

① 文本切分成语句

`import nltk`

```
text= "Don't hesitate to ask questions. Be positive."
from nltk.tokenize import sent_tokenize
print(sent_tokenize(text))
Out: ["Don't hesitate to ask questions.", 'Be positive.']
② 文本切分成语句(大批量句子切分、特定语言句子切分)
tokenizer= nltk.data.load("tokenizers/punkt/english.pickle")
print(tokenizer.tokenize(text))
Out: ["Don't hesitate to ask questions.", 'Be positive.']
```

五、教学参考资源

参考论文

1. 林婧. 关于人工智能的浅析[J]. 网络安全技术与应用, 2014(10).
2. 胡健, 柳青, 王海林. 验证码安全与验证码绕过技术[J]. 计算机应用, 2016(S1).
3. 王阳. 图灵测试六十五年——一种批判性的哲学概念分析[J]. 科学技术哲学研究, 2016(02).
4. 宋勇刚. 图灵测试: 哲学争论及历史地位[J]. 科学文化评论, 2011(06).
5. 胡宝洁, 赵忠文, 曾峦, 张永继. 图灵机和图灵测试[J]. 电脑知识与技术, 2006(23).

六、教学参考案例

参考案例 1

人工智能：从体验到思考

北京师范大学附属中学 李威

(1课时)

1. 教学目标

- 通过小组讨论分析, 以及搜集、整理、提炼资料, 理解智能及人工智能的基本概念。(信息意识)
- 通过与聊天机器人会话、评价聊天机器人的智能水平, 理解智能的不同定义方式。(信息意识)

2. 教学对象分析

本内容的教学对象是高一年级学生, 他们已经具备了一定的逻辑思维、分析与表达能力, 能够主动思考, 并通过讨论探究理解知识。大多数学生对前沿知识与技术有浓厚的兴趣, 但对理论学习感到枯燥, 对知识思考、归纳、总结存在畏难情绪。

3. 教学重难点

教学重点: 人工智能的基本概念。

教学难点:辩证分析人工智能的不同定义。

4. 教学方法

讲授法、任务驱动学习、小组协作等。

5. 教学工具

网络机房、投影。

6. 教学过程设计

教学过程设计如表 1-4 所示。

表 1-4 教学过程设计

教学环节	教学内容	学生活动	设计意图
情境引入	教师介绍:说到人工智能相信同学们都不会陌生,人机对弈、智能家居等在我们的生活中越来越常见。 你认为什么是智能呢?人工智能又是什么呢?	听讲,思考问题	激发学生的兴趣,提出问题
讲授新知	通过不同时期不同学者的不同的定义说明介绍什么是智能	听讲,思考	通过讲解,给出一些智能的概念,引导学生思考方向
分组讨论	1. 请学生分为 3~5 人的小组,进行资料搜集与讨论。 2. 同学们可以结合自己的理解,也可以辅助搜索网络上的相关文献,讨论什么是人类智能,什么是人工智能,以及二者的关系。 3. 最后请将结果发布在学习平台上进行分享交流	小组讨论,并将讨论结果发布在在线平台上,进行分享交流	通过小组讨论明确各组对于相关概念的认知,并且可以通过讨论结果了解学生对于概念理解存在的误区,有利于后续教学
交流评价	根据各组发布在平台上的结果进行交流讨论。分析异同点	思考讨论,交流反思	进一步明确各组形成的观点
分组体验探究	搭建简易聊天机器人并判定其是否具有智能。完成以下任务: 1. 按照课程资源的说明,下载和安装 NLTK,打开并运行配套程序 NLTK_chat_bot.py,创建一个简易聊天机器人。 2. 尝试与聊天机器人进行简单的英语会话,体验并初步评估机器人(以及你自己)的英语水平。 3. 你认为这样的聊天机器人是智能的吗? 为你的结论至少提供三条依据	进行体验活动,思考活动问题,与小组同学交流讨论	判定聊天机器人是否具有智能能够加深学生对智能的理解,辩证地思考自己提出的概念是否恰当
小组展示	介绍小组展示要求:展示与聊天机器人的会话内容,对聊天机器人及自己的英语水平进行评估,判断聊天机器人是否具有智能,并给出理由	小组进行探究成果展示	让学生了解各组对于智能的不同理解,思考分析是否合理
评价交流	选取一些展示中的亮点及各组共有的问题进行讲解与补充说明	反思探究结果中存在的问题,积极发言	让学生了解探究中存在的一些问题,通过讲解有针对性地再学习
归纳总结	总结本节的知识内容	总结归纳	加深印象,强化学生成对本节内容的理解

参考案例 2

走进图灵测试

北京师范大学附属中学 李 威

(1课时)

1. 教学目标

- 通过资料搜集、提炼有价值的信息等方式,了解图灵测试的基本原理。(信息意识、数字化学习与创新)
- 通过对“中文屋”问题的分析,辩证思考图灵测试的有效性。(信息意识)
- 通过体验不同类型的验证码,理解图灵测试在验证码中的应用。(信息意识)

2. 教学对象分析

本内容的教学对象是高一年级学生,他们已经具备了一定的自学能力,能够通过自主探究学习知识。学生对人工智能很有兴趣,喜欢了解新技术、新应用。但对理论学习感到枯燥,对思考、归纳、总结存在畏难情绪,对知识缺乏系统性了解。

3. 教学重难点

教学重点:图灵测试的基本原理。

教学难点:辩证思考图灵测试的有效性。

4. 教学方法

讲授法、任务驱动法、小组协作学习等。

5. 教学工具

网络机房、投影。

6. 教学过程设计

教学过程设计如表 1-5 所示。

表 1-5 教学过程设计

教学环节	教学内容	学生活动	设计意图
情境引入	教师介绍:在大家的手机和平板电脑上,有很多声称具有人工智能功能的应用。比如智能拍照、智能聊天应用等。这些应用真的如其所讲的那样智能吗?我们该如何判定一个应用、一个设备是否具有智能呢?	听讲,思考问题	激发学生的兴趣,提出问题
情境引入	提出问题:请列举你常用或熟悉的 5 个手机 App,将这些 App 按照各自的智能程度由高到低排序,并说明衡量标准,至少给出 3 条标准	思考并回答问题	由学生的实际应用出发,吸引学生注意,引导学生进行思考、梳理和总结

续表

教学环节	教学内容	学生活动	设计意图
新知讲解	教师讲解：图灵测试的由来与基本内涵。图灵在其 1950 年发表的论文《计算机器与智能》中开篇就提出了这个问题：“机器能思考吗？”图灵提出了一种“模仿游戏”用以判断计算机能否在智能行为上表现得和人“无法区别”，人们称之为图灵测试。图灵测试通过观察机器在回答人们提问时的表现来判断其是否具有智能。假设让一个人与一台机器同在一个房间中，询问者待在另一个房间中，且看不见两者中的任何一方。询问者通过一个终端对人和机器提问并接收两者的回答。如果经过对双方足够多次的问询和接叔回答后，询问者仍无法分辨出哪个是人，哪个是机器，那么就认为机器是智能的	听讲,思考	引导学生了解图灵测试的由来与内涵
分组探究	活动要求：请学生分为 5 人一组，分组探究。 1. 阅读教科书，搜索资料，了解什么是图灵测试，并尝试设计本组的图灵测试，列出问题清单。 2. 通过图灵测试就一定能证明机器具备智能吗？请阅读教科书、查找“中文屋”问题相关资料，通过分析讨论，提出你们组的观点，并说明理由	小组讨论，搜索资料，分析整理，设计本组的图灵测试，思考图灵测试的有效性	通过自主合作、搜索资料、分析整理、提取信息，培养学生的信息化意识，提高学生解决问题的能力。通过自主讨论，深化学生对图灵测试的理解，引导学生辩证地看待图灵测试的有效性，增强学生对理论学习的兴趣
交流评价	引导学生以小组为单位分享介绍本组设计的图灵测试，并介绍本组对图灵测试有效性的探讨结论	交流分享,总结归纳	通过交流分享，进一步加深学生对图灵测试的理解和认识。引导学生学会评价各组的成果
体验交流	组织学生体验不同类型的验证码，引导学生思考以下问题。 1. 验证码有什么作用，它和图灵测试有什么关系？ 2. 请举例说明验证码的不同类型，分析不同类型验证码的安全性，并进行说明	体验不同类型的验证码，思考并回答问题	通过对验证码的体验分析，引导学生理解应用验证码的目的以及其与图灵测试的关系，进一步理解图灵测试的内涵
课堂小结	结合各组的成果，总结本节的知识内容、图灵测试的内涵	总结归纳	加深印象，使知识系统化

第二节 人工智能的发展历程

一、教学目标与重点

教学目标：

- 通过小组资料搜集整理,了解人工智能的发展历程及其中的重要人物和事件。(信息意识、数字化学习与创新)
- 通过小组分析讨论,理解大数据、算法和计算力对人工智能复兴的共同推动作用。(信息意识)
- 通过活动实践,体验搭建身边的人工智能应用,进一步理解人工智能。(计算思维)

教学重点：

- 人工智能跌宕起伏的发展历程及其原因;
- 大数据、算法和计算力对人工智能发展的共同推动作用。

二、内容分析与建议

1. 内容分析

本节主要介绍人工智能的发展历程。在了解人工智能与图灵测试的相关概念与内涵后,通过讨论分析,了解人工智能跌宕起伏的发展历程及其原因,以及大数据、算法和计算力对人工智能发展的共同推动作用,为后续章节的学习奠定基础。本节主要知识框图如图 1-4 所示。

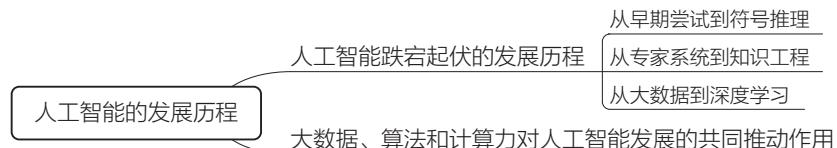


图 1-4 主要知识框图

2. 教学建议

(1) 教学活动建议

人工智能的发展过程跌宕起伏,其间涌现了很多的人物和事件,在教学中可以借助这些人物和事件,帮助学生了解人工智能的发展历程,同时通过体验、思考讨论、编程实践等活动让学生对于不同时期人工智能的主要特点有所了解。

在了解人工智能发展历程中的关键事件的同时,教师应引导学生思考分析大数据、算法和计算力对于人工智能发展所起到的作用。同时,应通过体验活动、探究讨论及编程实践等活动增加学生对理论知识的兴趣,帮助学生理解人工智能各个发展阶段的真正内涵,通过学生活动实践让学生生成自己对于人工智能的理解,由此培养学生自主思考探究的能力。

(2) 项目实施建议

本节的项目任务主要是制作人工智能发展历史大事记图表展板,搜集整理生活中、学习中常用的人工智能应用软件或者人工智能工具,为创新科技周展台布置做准备。

本节项目的一个任务是制作人工智能大事记图表展板,项目实施同第一节以小组为单位进行,在课堂学习中,小组成员通过教科书可以了解人工智能的发展阶段及关键人物和事件,但制作展板过程中,需要学生丰富发展过程的内容,制作具有本组风格的展板,因此学生需要借助网络搜集相关资料,讨论分析确定展板思路及各项任务分工,制作出本组自己的人工智能大事记展板。

本节项目的第二个任务是搜寻身边的人工智能应用,为后续创新科技周布置体验展台做准备,在完成该任务的过程中,小组成员应进行讨论、协作及分工,可以让每一名组员各自负责某一领域的日常生活应用,并安排一名组员负责总结补充、总体统筹等。小组成员可以依据课堂上介绍的机器翻译和语音识别联想、扩展,制作出本组的人工智能应用展示台。

(3) 评价建议

评价方式:教师评价与学生评价相结合,过程性评价与总结性评价相结合。

评价量表如表 1-6 所示。

表 1-6 小组协作及展板设计评价量表

	评价等级及分值			得分		
	优秀 (80~100 分)	良好 (60~79 分)	尚需努力 (60 分以下)	组内 自评	组间 评价	教师 评价
小组协作(20%)	小组合作分工合理、明确,积极参与小组讨论分析、参与课堂体验活动、思考相关问题、踊跃交流表达自己的观点	小组合作分工明确,参与小组讨论分析、参与课堂体验活动、思考相关问题、能够交流表达自己的观点	小组合作分工不明,小组活动参与较少,不表达或很少表达自己的观点			
科技周展板、 资料制作效果 (60%)	作品科学性(30%) 作品艺术性(15%) 作品创造性(15%)	能够结合网络资料收集,制作出内容丰富、展示效果良好的展板	能够制作相关展示材料	制作的展示材料质量差		

续表

	评价等级及分值			得分		
	优秀 (80~100 分)	良好 (60~79 分)	尚需努力 (60 分以下)	组内 自评	组间 评价	教师 评价
交流分享(20%)	表达能力强,能够将小组成果和想法充分表达出来	能够说明小组成果和想法	对小组成果和想法说明得不清楚			
总分(100%)						

三、习题分析与提示

■ 习题描述

见教科书第 18 页的探究活动“身边的专家系统”。

■ 参考答案

1. 专家系统大都针对某一特定领域,如个人理财专家系统、寻找油田的专家系统、贷款损失评估专家系统、学科教学专家系统等各种类型的专家系统。

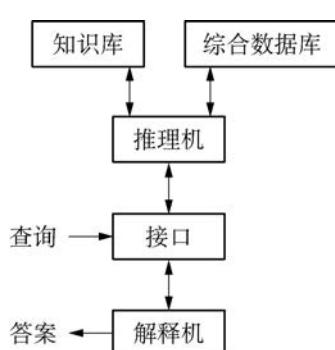


图 1-5 专家系统的构成

2. 专家系统 App 的基本结构是一样的,基本都是由知识库、综合数据库、推理机、接口、解释机五部分组成,如图 1-5 所示。但其中知识库中所存储的知识是各个领域的不同知识,推理的规则也不相同。

3. 知识库用来存放专家提供的知识。专家系统的问题求解过程是通过知识库中的知识来模拟专家的思维方式的,因此,知识库是专家系统质量是否优越的关键所在,即在知识库中知识的质量和数量决定着专家系统的质量水平。一般来说,专家系统中的知识库与专家系统程序是相互独立的,用户可以通过改变、完善知识库中的知识内容来提高专家系统的性能。如,农业专家系统需要动态更新其知识库。由于农作物生产的特性,要求专家系统中的基础数据不但是海量的,而且必须是动态的,故知识库、数据库、模型库必须要不断有新知识、新数据、新技术来更新扩充支撑,尽快解决农业生产中的实际问题。

四、核心概念精解与技术工具介绍

1. 核心概念

(1) 专家系统

专家系统(expert system)是一种在特定领域内具有专家解决问题能力水平的程序系统。它能够有效地运用专家多年积累的有效经验和专门知识,通过模拟专家的思维过程,解决原来需要专家才能解决的问题。

专家系统属于人工智能的一个发展分支,自 20 世纪 60 年代研究人员研制成功第一

个专家系统以来,专家系统获得了飞速的发展,并且运用于医疗、军事、地质勘探、教学、化工等领域,产生了巨大的经济效益和社会效益。

(2) 知识工程

研究知识信息处理的学科,提供研制基于知识的智能系统的一般方法和基本工具,是人工智能、数据库、数理逻辑、认知科学和心理学等学科交叉发展的结果。知识工程的概念于1977年首次被提出。知识工程的研究使人工智能发生了重大改变,从探索广泛普遍的思维规律转向研究利用知识解决特定问题。知识工程中所研究的最典型的智能系统就是专家系统,它是通过模仿专家的思维活动进行推理和判断,能像专家那样求解专门问题的计算机程序系统。其他智能系统还包括智能决策支持系统、知识库系统和自然语言理解系统等。

知识工程的研究内容主要包括知识表示、知识利用和知识获取。知识表示是设计一个智能系统的基础,常用的知识表示方法有逻辑表示(命题逻辑和一阶谓词逻辑等)产生式规则、框架、语义网和脚本等。知识利用的任务是利用已有知识解决问题,它涉及搜索和推理两个技术。搜索策略包括深度优先、广度优先、启发式、与或图等。推理包括精确推理和不精确推理等。知识获取是指将用于问题求解的专门知识从人类专家或数据等知识源中总结和抽取出来而转化为知识库的过程,它是知识处理中的一个瓶颈。从获取方式上可将知识获取分为人工获取、半自动获取和自动获取三种方式。

(3) 大数据

大数据(big data),IT行业术语,是指无法在一定时间范围内用常规软件工具进行捕捉、管理和处理的数据集合,是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率和多样化的信息资产。

大数据方法指不用随机分析法(抽样调查)这样的捷径,而采用对所有数据进行分析处理的分析方法。大数据具有五大特点:大量、高速、多样、低价值密度、真实性。

(4) 深度学习

深度学习(deep learning)是机器学习领域中一个新的研究方向。深度学习是学习样本数据的内在规律和表示层次,这些学习过程中获得的信息对诸如文字、图像和声音等数据的解释有很大的帮助。它的最终目标是让机器能够像人一样具有分析学习能力,能够识别文字、图像和声音等数据。

深度学习在搜索技术、数据挖掘、机器学习、机器翻译、自然语言处理、多媒体学习、语音、推荐和个性化技术,以及其他相关领域都取得了很多成果。深度学习使机器能模仿视听和思考等人类的活动,解决了很多复杂的模式识别难题,使得人工智能相关技术取得了很大进步。

2. 技术工具

SymPy是一个完全由Python写成的符号计算Python库,为许多数值分析、符号计算提供了重要的工具。SymPy的第一个版本于2007年开源,并且经历了十几个版本的迭代,在2019年开源了1.4版本。从1.4版本文档介绍中可以看到,SymPy支持很多初等数学及高等数学的符号计算,主要包括:基础计算、公式简化、微积分、解方程、矩阵、几何和级数等。此外,还可以支持范畴论、微分几何、常微分方程、偏微分方程、傅里叶变换、集合论、逻辑计算等。

五、教学参考资源

参考论文

- 董慧慧. 人工智能深度学习概念研究与综述[J]. 电脑编程技巧与维护, 2018(08).
- 陈盼盼. 人工智能技术的发展及伦理思考[J]. 青年记者, 2017(20).
- 邬贺铨. 大数据时代的机遇与挑战[J]. 求是, 2013(04).
- 邹蕾, 张先锋. 人工智能及其发展应用[J]. 信息网络安全, 2012(02).
- 黄荣怀, 李茂国, 沙景荣. 知识工程学: 一个新的重要研究领域[J]. 电化教育研究, 2004(10).

六、教学参考案例

参考案例 1

人工智能的发展历程

北京师范大学附属中学 李 威
(1课时)

1. 教学目标

- 通过小组合作、搜集整理资料、展板制作,了解人工智能的发展历程。(信息意识、数字化学习与创新)
- 通过对不同阶段人工智能的不同实现方式进行探究,理解大数据、算法和计算力对人工智能复兴的共同推动作用。(信息意识)

2. 教学对象分析

本内容的教学对象是高一年级学生,他们已经具备了一定的逻辑思维、分析与表达能力,能够主动思考并通过讨论探究理解知识。大多数学生对前沿知识与技术有很浓厚的兴趣,但对理论学习感到枯燥,对知识思考、归纳、总结存在畏难情绪。

3. 教学重难点

教学重点: 人工智能的发展历程。

教学难点: 大数据、算法、计算力对人工智能的共同推动作用。

4. 教学方法

讲授法、任务驱动法、小组协作等。

5. 教学工具

网络机房,投影,Python 运行环境,Python 集成开发环境,展板制作用的工具(纸、剪刀、胶水等)。

6. 教学过程设计

教学过程设计如表 1-7 所示。

表 1-7 教学过程设计

教学环节	教学内容	学生活动	设计意图
情境引入	教师介绍,提出问题: 从 20 世纪中叶人工智能的萌芽时期,到现今人工智能的蓬勃发展,人工智能的发展历史上发生了哪些激动人心的故事呢?你知道哪些人物和事件呢?	思考,回答问题	提出问题,激发学生的兴趣,引导学生主动思考。引出本节课的主题
讲授新知	介绍人工智能发展的三个阶段,从早期尝试到符号推理,从专家系统到知识工程,从大数据到深度学习。 提问:三个阶段都是如何实现人工智能的?区别是什么?典型应用有哪些?	听讲,思考,交流,回答问题	通过讲解,帮助学生清晰地认识人工智能的发展阶段。培养学 生归纳知识的能力
项目引领分组探究	1. 将学生分为 3~5 人的小组,布置以下 2 个体验探究活动: 活动 1:体验 Python 中的符号数学。 (1) 阅读教科书中的体验任务,通过 import 语句导入 SymPy 库; (2) 运行教科书中的程序片段,观察效果,尝试理解。 活动 2:探索身边的专家系统。 (1) 你使用过拍照辨别植物的 App 吗?你觉得它是专家系统吗?你还用过其他哪些专家系统 App? (2) 专家系统 App 有哪些共同点,又各自有哪些特点? 2. 提出问题:在体验的过程中,思考不同方法是如何实现智能的,智能可以达到什么程度,大数据、算法和计算力对人工智能的发展有什么影响。 3. 巡视学生实践情况,指导和帮助学生自主、协作探究	分组开展体验探究活动,思考提出的问题,与同学进行讨论交流	让学生通过实际的体验探究活动了解人工智能发展的不同阶段,对人工智能不同发展阶段的理解更加深入立体。引导学生进行归纳分析,培养学生分析总结的能力
制作展板	指导学生制作人工智能发展历程展板,并整理搜集到的人工智能相关应用材料	小组合作制作展板,整理材料	通过实践制作过程深化学生对人工智能发展阶段的了解
交流评价	组织学生分享各自小组的探究结果、展板及在探究中遇到的问题	思考讨论,分享交流	通过交流、探讨,加深学生对人工智能不同发展阶段的理解
归纳总结	总结本节的知识内容:人工智能发展的三个阶段,以及大数据、算法和计算力对人工智能发展的作用	总结归纳	使学生加深印象,强化对本节内容的理解

参考案例 2

实践智能工具解决问题

北京师范大学附属中学 李 威

(1课时)

1. 教学目标

- 了解生活中常见的人工智能应用,能在遇到相关问题时有意识地利用相关工具解决问题。(信息意识)

- 实践体验搭建身边的人工智能应用,并能够迁移到其他问题的解决中。(计算思维)

2. 教学对象分析

本内容的教学对象是高一年级学生,他们已经具备了一定的逻辑思维、分析与表达能力,能够主动思考并通过讨论探究理解知识。大多数学生对前沿知识与技术有浓厚的兴趣,但对理论学习感到枯燥,对知识思考、归纳、总结存在畏难情绪。

3. 教学重难点

教学重点:通过搭建人工智能应用深化对人工智能的认识。

教学难点:人工智能应用的搭建。

4. 教学方法

讲授法、任务驱动法、小组协作法等。

5. 教学工具

网络机房、投影、Python 运行环境、Python 集成开发环境。

6. 教学过程设计

教学过程设计如表 1-8 所示。

表 1-8 教学过程设计

教学环节	教学内容	学生活动	设计意图
情境引入	教师提问:你常用的人工智能应用有哪些呢?主要用来帮你解决什么问题?	思考,回答问题	从学生的生活实际入手,激发学生的兴趣
体验思考	体验介绍:随着人工智能和计算机技术的发展,机器翻译的水平不断提高。请大家打开在线翻译系统进行体验,分别输入文字、图片等不同类型的内容,观察翻译的结果是否准确。可以尝试先在机器翻译之前自己翻译,再对比自己和机器翻译的结果。完成体验后,请和同学交流:你认为机器的翻译水平如何,是否能达到甚至高于普通人的水平?	实践体验,思考问题,生生互动,回答问题	通过体验活动,激发学生的兴趣。引导学生了解人工智能在翻译方面的应用

续表

教学环节	教学内容	学生活动	设计意图
交流评价	1. 总结学生的体验收获。 2. 根据学生的回答,总结不同领域常用的人工智能应用	听讲,思考	引导学生了解生活中不同的人工智能应用
活动探究	分组活动: 将学生分成 5 人小组,组织学生自主分工,合作探究。 教师引导各组使用 Python 调用语音识别功能,对学生在探究过程中遇到的问题进行辅导。 1. 设计语音识别工具应具备的功能,规划所需要的素材,设计开发的工具要实现的功能目标。 2. 从课程资源包中下载本活动的文件。 3. 安装所需 AI 库,并在命令行环境下运行如下命令,体验语音识别的功能效果。 pip install baidu-aip python AipSpeech.py testxx.wav 4. 尝试观察程序内容,并进行修改、运行,再次体验其功能效果。 5. 思考语音识别工具的实现过程具体应包含哪些步骤	进行分组探究活动。 1. 小组讨论设计应用所需功能。 2. 下载课程资源,运行体验,观察效果。 3. 小组讨论,尝试修改,观察运行效果。 4. 思考语音识别工具的基本实现原理	通过自主探究、实践,向学生渗透编程思想。用体验与实践结合的方式让学生了解使用智能工具解决问题的过程,形成使用智能工具解决问题的意识
交流评价	组织各组进行展示交流,介绍各组的探究成果。引导学生发表各自看法	以小组为单位分享各组的语音识别工具探究过程,以及对语音识别原理的分析	引导学生分享、提出实践中遇到的问题,加深学生的体验
归纳总结	总结本节的知识内容。 1. 生活中常用的智能工具。 2. 智能工具能够解决的各类问题	总结归纳	使学生加深印象,强化对本节内容的理解

人工智能实现

一、本章学科核心素养的渗透

通过第一章的学习,学生已经理解了人工智能相关概念,了解了人工智能发展历程中的重要人物和大事件,感受到人工智能技术已经渗透到我们生活的方方面面,使我们的生活更加便利和美好,认识到人工智能在未来社会发展中的重要作用,初步形成了自己的人工智能观。本章在承继前文的基础上,把人工智能领域的基本概念、技术原理和典型算法有机地融入到问题求解中,引导学生通过项目实践来学习人工智能典型算法、认识人工智能关键技术、掌握并应用人工智能工具。本章内容对应的学科核心素养渗透在以下几个方面:

- 通过“创新园区自由行”项目,以贴近生活的案例,带领学生利用人工智能技术解决现实生活中的问题,培养学生主动使用新技术处理信息的意识。(信息意识)
- 通过多个项目问题的解决,带领学生深入理解经典人工智能算法及其实现原理,在实践探索中尝试搭建简单人工智能应用框架模块,培养并加深学生利用计算思维(特别是抽象、模块化及分解等问题求解方法)设计并实现解决方案的能力。(计算思维)
- 通过对决策树算法的决策过程与人类真实决策过程,领会机器学习对于人类智慧的模拟方式,能够采用计算机可以处理的方式界定问题、抽象特征、建立结构模型、合理组织数据。(计算思维)
- 通过项目实践,引导学生运用数字化学习策略管理学习过程与资源,利用人工智能的经典算法完成任务,形成解决方案。(数字化学习与创新)
- 通过体验思考、项目实践,让学生深入理解人工智能技术的原理及优缺点,客观认识技术的发展对社会生活的影响。(信息社会责任)
- 通过项目活动认识到人工智能技术的两面性,在带来积极作用的同时,对社会伦理、安全等方面的挑战形成清晰的认识,提升安全防护意识和社会责任感。(信息社会责任)

二、本章知识结构

本章分为三节,主要内容是关注人工智能的实现,了解人工智能技术中最为经典的三类技术:启发式搜索、专家系统和机器学习,通过主题项目开展经典算法的学习,体验人工智能技术实现的具体过程,理解人工智能的基本原理,掌握人工智能工具,感受人工智能处理实际问题的巨大作用。

第一节介绍了人工智能的基本技术之一:启发式搜索。通过引入路径规划,先将生活中的搜索问题用图的方式进行形象化表示,进而详细讲解了广度优先搜索和深度优先搜索这两种搜索策略。对比了解两种搜索策略的特点之后,介绍了启发式搜索的典型算法——A^{*}算法,利用额外的全局信息,即启发式规则来“指导”路径展开的次序,从而更快更好地找到问题的解。作业练习通过“8-数码游戏”问题加深学生对A^{*}算法的理解,知识延伸部分的旅行商问题则为学生更深入的学习提供了空间和方向。

第二节则关注专家系统,专家系统集成了某个领域专家水平的知识、经验和方法,能够协助人类解决该领域中的实际问题。教科书从专家系统的构成开始,介绍了知识库、综合数据库、推理机、接口和解释机等基本概念;接下来介绍了专家系统中的知识表示方法,重点介绍了基于规则的专家系统;再通过决策树算法自动从“事实”数据库中抽取规则,总结出规律和知识,即让专家系统能自行生成或更新知识库;最后以两种最常用的推理方法——正向推理和反向推理为例,介绍了运用知识库中的知识进行推断的机制和规则。

第三节从解密新一代人工智能的核心——机器学习开始,了解机器学习与人类学习方式的异同点,介绍了监督学习和无监督学习的概念。进而通过体验思考、项目实践等形式,理解k-均值聚类算法和k-近邻分类算法的基本原理,体验两种算法的典型应用场景。

本章主要知识与概念的关系如图 2-1 所示。

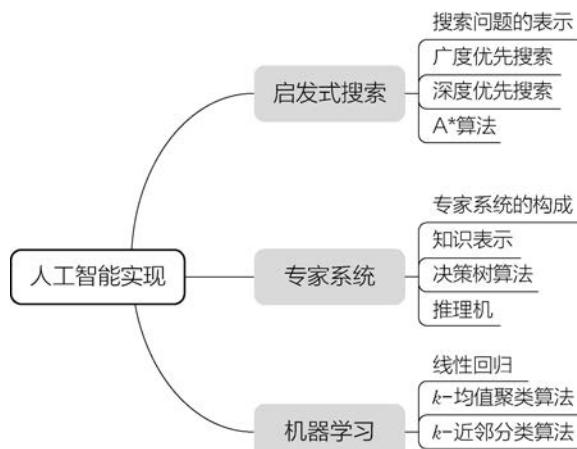


图 2-1 主要知识与概念的关系

三、本章项目活动的设计思路与实施建议

本章以“创新园区自由行”的情境为背景展开项目活动。让学生在真实的情境中理解人工智能技术的基本原理,掌握人工智能的经典算法,认识人工智能在社会生活中的重要作用,并学会应用人工智能技术解决实际问题。

本章的项目实践任务选取了参观中经常遇到的路径规划、主题展馆选择、集合地点选取等经典问题,将人工智能领域的基本理念和典型算法实现,有机地融入到具体问题的求解中,并使用数字化工具和编程方法实现问题求解的关键环节,使学生能够在算法理解和编程实践中学习人工智能方法,在应用中掌握人工智能工具。

本章的项目活动分三个子任务展开,每个子任务均以“体验情境—分析问题—尝试开发”的流程展开,教师在带领学生完成主题任务的同时,引领学生由表及里、逐步探索人工智能的经典算法,理解人工智能的基本原理,掌握人工智能工具。第一节作为项目的开始,进入园区参观后的首要任务是确定参观线路,因此学生的主要任务是利用启发式搜索算法来规划参观路径。第二节则关注主题展馆的个性化推荐,学生运用所学的专家系统知识,根据以往的用户反馈,计算各个特征的信息增益,设计并实现决策树算法来进行展馆推荐。第三节则需解决参观之后的集合地点选择问题和参观者行程安排问题,学生需要运用所学的 k -均值聚类算法找出适合的集合地点。项目实践方式可采用分组形式进行,项目结束后每位同学需提交个人项目总结,交流并分享在项目实施过程中的经验和教训。

本章的项目实施过程中会涉及到比较多的人工智能基本概念、数学表示以及经典算法,这些对学生来说都比较陌生,项目实践中还有一定量的代码编写和调试工作,因此需要教师尽可能地提前准备好人工智能的平台和环境,以及可提供案例的部分代码,帮助学生排除在课堂上进行大量代码输入的障碍,从而将学习的关注点更多地放在对人工智能经典算法原理的理解上。

项目活动的评价需要教师与学生一起参与,从每位同学的参与态度、协作精神、任务完成的数量与质量以及所提出的想法、代码编写等几个方面进行组内自评、组间互评和教师评价。因此建议制订一份任务评价量表,可参考表 2-1。

表 2-1 任务评价量表

	评价等级及分值			得分		
	优秀 (80~100 分)	良好 (60~79 分)	尚需努力 (60 分以下)	组内 自评	组间 评价	教师 评价
小组协作(20%)						
项目完成度(20%)						
项目数量(20%)						
交流分享(20%)						
总结反思(20%)						
总分(100%)						

四、本章教学课时分配建议

本章教学课时分配建议如表 2-2 所示。

表 2-2 本章课时分配建议

节次	课时	具体 内 容
第一节	2 课时	第 1 课时: 基本搜索方法
		第 2 课时: 启发式搜索
第二节	4 课时	第 3 课时: 知识表示
		第 4 课时: 专家系统
		第 5 课时: 决策树原理
		第 6 课时: 决策树构造与可视化
第三节	7 课时	第 7 课时: 认识机器学习和经典线性回归
		第 8 课时: k -均值聚类算法原理
		第 9 课时: k -均值聚类算法项目实践
		第 10 课时: k -近邻分类算法原理
		第 11 课时: k -近邻分类算法项目实践
		第 12 课时: 创新园区自由行项目实践
		第 13 课时: 创新园区自由行项目展示

第一节 启发式搜索

一、教学目标与重点

教学目标:

- 通过案例分析,了解搜索问题的形式化表示。(信息意识、计算思维)
- 通过观摩和实践,了解广度优先和深度优先搜索算法,理解启发式搜索的基本思想。(计算思维)
- 通过案例剖析和实践,理解 A* 搜索算法基本原理,体会启发函数的作用。(计算思维)

- 通过实例,了解启发式搜索在人工智能中的应用。(信息意识、数字化学习与创新)

教学重点:

- 启发式搜索的基本思想及其优点;
- A^{*} 算法基本原理及启发函数的作用;
- 启发式搜索在人工智能中的应用。

二、内容分析与建议

1. 内容分析

本节课主要介绍了人工智能最基本的一类问题的求解方法——搜索,包括搜索的基本概念、两种重要的搜索策略——盲目搜索和启发式搜索。接下来教科书结合本章项目主题“创新园区自由行”设计了“规划创新园区参观路径”的项目任务,先详细介绍了搜索问题的形式化表示,然后展开了广度优先搜索、深度优先搜索和启发式搜索(A^{*} 算法)的原理介绍和实践。在本小节的最后提供了拓展阅读资料,可以激发有兴趣有能力的学生将学习延伸到课后,引导学生进行深入思考与探索。本节主要知识框图如图 2-2 所示。

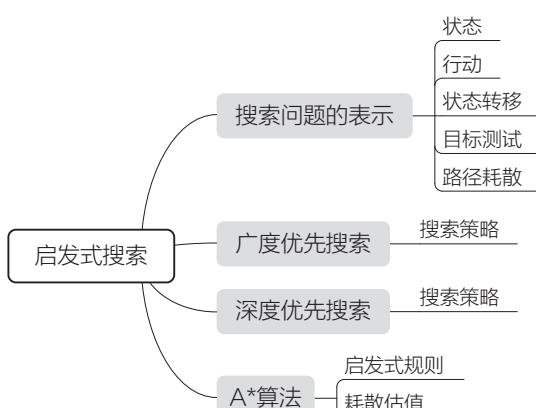


图 2-2 主要知识框图

2. 教学建议

(1) 教学活动建议

搜索是人工智能初期问题求解的重要方法,有着广泛应用。在课堂教学中不必拘泥于教科书给出的案例和活动,可以联系实际生活中搜索算法的应用场景,例如走迷宫、人机博弈等,设计相关演示或体验活动,引入搜索的概念,激发学生学习搜索算法的兴趣。

A^{*} 算法是一种典型的启发式搜索算法,了解 A^{*} 算法的基本原理和启发函数的作用是本节课的重点,同时也是本节课的难点。在算法实践过程中可以鼓励学生积极参与,引导学生尝试设计不同的启发函数,观察效果,从而让学生对启发函数的作用和启发式搜索的基本原理有更深刻的理解。

启发式搜索是人工智能的基本技术之一,应用广泛,例如游戏寻路、人机博弈等。在教学过程中可以开拓学生视野,让学生了解更多的搜索算法的应用。同时可以结合某个应用案例分析背后的原理,让学生对启发式搜索思想有更深刻的认识。

在该节内容的“知识延伸”部分提到了使用智能算法可以找到旅行商问题的近似最优

解。在教学过程中,可以将启发式搜索进行拓展,介绍与启发式算法类似的智能优化算法:模拟退火、遗传算法、蚁群算法等,让有兴趣有能力的同学课后利用数字化工具了解更多的启发式算法,提升学生的自主学习和数字化学习能力。

(2) 项目实施建议

教科书中布置了两个项目实践活动,在项目实施上建议将实践活动有机整合,层层递进,在解决问题的过程中渗透本节课的知识点。项目实践中编写搜索算法源代码部分需要一定的编程基础,所以教师需要在课前准备算法源代码等学习资料,在教学过程中可以根据学生的实际情况提供部分源代码,让学生补充算法核心代码或者重点讲解算法的核心代码,从而降低学习难度,突出教学重点。同时要对程序运行结果进行分析,引导学生进一步理解几种搜索算法的思想和过程。

项目实践其中包括三种搜索算法的实现:广度优先搜索、深度优先搜索和启发式搜索。由于算法比较抽象,理解起来有一定难度,所以在教学过程中可以通过形象直观的动画演示,使学生更好地理解三种算法的搜索原理、过程。对于三种算法的运行结果,可以给出效果对比图,使学生更好地了解每种算法的特点及算法之间的区别。

项目实践是难度较大的内容,建议采用学生分组合作的教学方式,让学生进行讨论交流,完成实践。实践后,教师组织各小组分享交流实践的成果,并进行拓展,进一步加深学生对搜索算法的理解。

(3) 评价建议

评价方式:将过程性评价和总结性评价相结合,以过程性评价为主。过程性评价主要考虑学生在课堂中的参与态度、互助合作精神、实践活动的结果等。总结性评价主要依据学生提交的个人项目报告。

评价要点:在课堂活动中,教师需要关注学生是否能积极参与到课堂活动中去,是否能积极主动思考、踊跃交流和表达观点;在分组活动中,是否能与小组成员有效合作,积极参与小组讨论,提出自己的想法,完成实践活动;是否按时提交项目总结报告及报告完成情况。

三、习题分析与提示

■ 习题 1 描述

见教科书第 30 页的项目实践“参观路径的形式化表示”。

■ 习题 1 参考答案

1. 从状态 $In(A)$ 执行行动序列 $\{Go(B), Go(F), Go(G)\}$, 过程如下:

$Result(In(A), Go(B)) = In(B)$,

$Result(In(B), Go(F)) = In(F)$,

$Result(In(F), Go(G)) = In(G)$ 。

执行完 $\{Go(B), Go(F), Go(G)\}$ 行动序列后, 状态为 $In(G)$, 是目标状态, 所以 $\{Go(B), Go(F), Go(G)\}$ 是一个能到达解状态的行动序列。

2. $c(\text{In}(A), \{\text{Go}(B), \text{Go}(F), \text{Go}(G)\}, \text{In}(G))$
 $= c(\text{In}(A), \{\text{Go}(B)\}, \text{In}(B)) + c(\text{In}(B), \{\text{Go}(F)\}, \text{In}(F)) + c(\text{In}(F), \{\text{Go}(G)\},$
 $\quad \text{In}(G))$
 $= 10 + 8 + 16$
 $= 34。$

习题 2 描述

见教科书第 31 页的体验思考“用广度优先搜索找出其他的解路径”。

习题 2 参考答案

广度优先搜索还可以找到解路径: $A \rightarrow B \rightarrow F \rightarrow G$, 表示为行动序列 $\{\text{Go}(B), \text{Go}(F), \text{Go}(G)\}$, 该路径的耗散值为 34。

习题 3 描述

见教科书第 33 页的体验思考“深度优先搜索的路径耗散”。

习题 3 参考答案

解路径 $A \rightarrow B \rightarrow C \rightarrow D \rightarrow G$ 的路径耗散值为 45。

习题 4 描述

见教科书第 36 页的项目实践。

习题 4 参考答案

1. 略。

2. 在本节的 A^* 算法案例中, 如果开始时的时间估计值与实际值相差较大(设实际值大于估计值), 则实际实施搜索时搜索的节点数多, 搜索范围大, 效率低, 但能保证找到最优解。

3. 从徐家汇乘坐地铁到陆家嘴, 途经车站数目最少为七站; 换乘次数最少为一次。
(具体乘坐方案可以让学生讨论交流)

习题 5 描述

见教科书第 36 页的作业练习“8—数码游戏”。

习题 5 参考答案

(1) 待解决的问题

在 3×3 组成的棋盘内有 9 个方块, 其中 8 个方块分别放置 1~8, 留下一个空格。游戏规则是: 空格周边的任何数码都可以滑动, 每滑动一次就产生一个状态。假设初始状态和目标状态如图 2-3 所示。

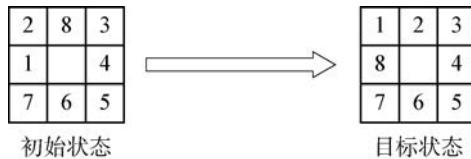


图 2-3 待解决问题的初始状态和目标状态

问:若用 A^{*} 算法来求解这个问题,该如何定义启发式信息呢?

(2) 问题的搜索形式表示

状态空间:8 个数字和空格在棋盘上的所有布局组成了问题的状态空间。

状态转移:通过滑动空格周边的数码到空格,到达新的状态。

目标测试:比较当前状态和目标状态是否一致。

路径耗散:每将数码滑入一次空格耗散值为 1,因此整个路径的耗散值是从初始状态到目标状态的数字滑动的总步数。

(3) 解题方法分析及启发式信息的定义

由于 8 - 数码问题状态空间共有 $9!$ 个状态,为减小搜索范围,提高效率,可以选择启发式搜索 A^{*} 算法来求解问题:在状态空间中的搜索对每一个搜索的位置进行评估,得到最好的位置,再从这个位置进行搜索直到目标。

我们可以定义评估函数: $f(n) = g(n) + h(n)$,其中 $g(n)$ 是从初始状态变换到节点 n 的状态所移动的步数,启发函数 $h(n)$ 可以有多种选择,常用的可以定义为节点 n 的状态与目标状态的差距,即节点 n 的状态与目标状态相比,不在目标位置的数码个数,也称错位数。 $h(n)$ 也可以定义为不在目标位置的数字与其该在位置的曼哈顿距离总和。

四、核心概念精解与技术工具介绍

1. 核心概念

(1) 搜索

搜索是人工智能最基本的一类问题求解方法,分为盲目搜索和启发式搜索。盲目搜索是逐一列举出所有可能性,直到找到问题的解。启发式搜索算法则是利用经验性法则帮助问题求解,达到缩小搜索范围,降低问题复杂度的目的。这些经验性法则没有定式,因此启发式搜索算法通常要根据不同的问题场景来进行针对性的设计。

(2) 广度优先搜索

广度优先搜索是在寻找从初始状态到目标状态的路径时,每一步都先向尽可能广的方向扩展,等这一步所有可能的方向都走遍了,再进入到下一步。换句话说就是先访问完同一层的节点,然后才继续访问下一层节点。举例说明:如图 2-4 所示,如果我们从 A 点

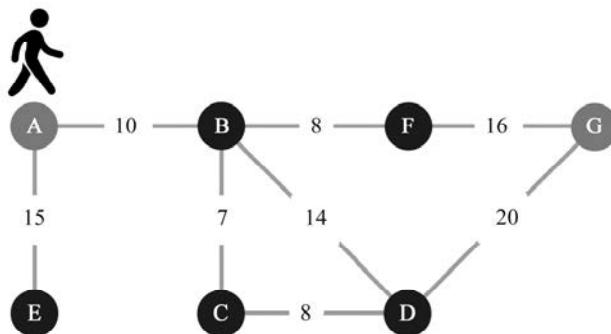


图 2-4 参观路径规划问题

发起广度优先搜索(以下的访问次序并不是唯一的,第二个点既可以是 B 也可以是 E),则我们可能得到访问次序:A→B→E(第一层节点访问完毕)→C→D→F(第二层节点访问完毕)→G(第三层节点访问完毕),所有节点都被访问过或者找到了问题的解,搜索结束。

广度优先搜索通常使用队列来实现。

(3) 深度优先搜索

深度优先搜索是在扩展根节点后面的子节点时,每次都选择距离根节点最远的未扩展节点,如果没有能继续扩展的节点了,则返回到根节点重新开始,直至到达目标状态或所有节点都已扩展完毕。如图 2-4 所示,通过深度优先搜索可以找到初始节点 A 到目标节点 G 的路径 A→B→C→D→G 或 A→B→D→G 或 A→B→F→G。

深度优先搜索可以使用递归函数或者栈来实现。

(4) 启发式搜索

启发式搜索又称为有信息搜索,它是利用问题拥有的启发信息来引导搜索,达到减少搜索范围、降低问题复杂度的目的。

A^{*} 算法是一种典型的启发式搜索算法,它需要一个评估函数: $f(n)=g(n)+h(n)$, $g(n)$ 为初始状态到状态 n 的实际代价, $h(n)$ 为状态 n 到目标状态的估计代价,被称为启发函数。

利用 A^{*} 算法求解教科书中的路径规划问题, $g(n)$ 定义为初始节点与 n 节点的实际值, $h(n)$ 定义为 n 节点到目标节点的直线步行时间值,然后根据后续节点与目标节点的 f 值来决定展开的次序,更快更好地找到问题的解。在路径规划图中加入启发式信息如图 2-5 所示。

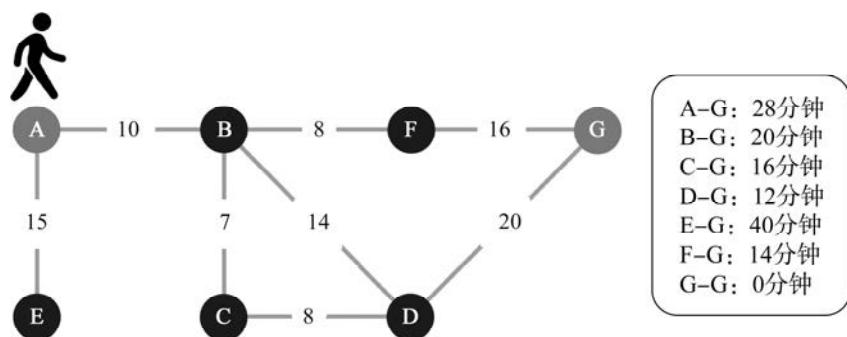


图 2-5 在路径规划图中加入启发式信息

A^{*} 算法的策略是每次都先扩展总耗散估值最小的节点,直到所有叶子节点的总耗散估值都大于目标节点的总耗散估值则搜索停止。

如图 2-6 所示,通过 A^{*} 算法可以找到一条从初始节点 A 到目标节点 G 的解路径 A→B→F→G。

2. 技术工具介绍

(1) 递归

所谓递归,就是一个过程或函数直接或间接调用自己。它通常把一个大型复杂的问

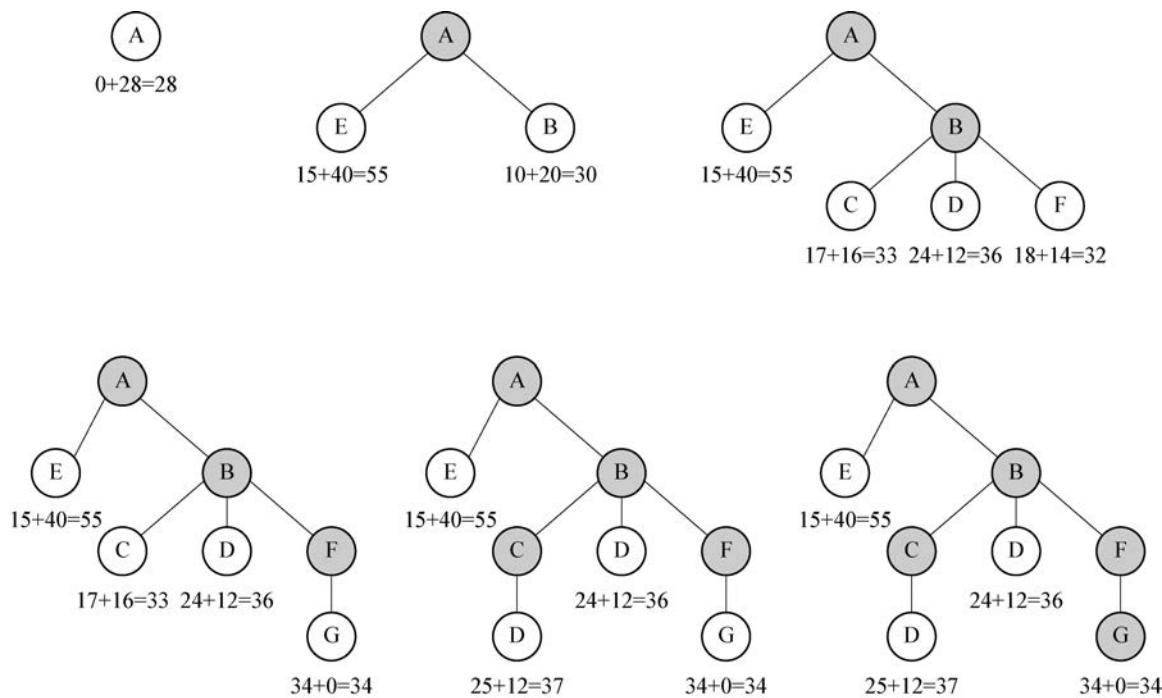


图 2-6 A^{*} 算法求解路径

题层层转化为一个与原问题相似的规模较小的问题来求解,递归策略只需少量的程序就可描述出解题过程所需要的多次重复计算,大大地减少了程序的代码量。一般来说,递归需要有边界条件、递归前进段和递归返回段。当边界条件不满足时,递归前进;当边界条件满足时,递归返回。

(2) 队列

队列是一种特殊的线性表,它只允许在表的前端(front)进行删除操作,而在表的后端(rear)进行插入操作。进行删除操作的端称为队头,进行插入操作的端称为队尾。队列具有先进先出的特点。

Python 语言中队列的用法:

```
import Queue
q = Queue.Queue()
q.put()      # 入队
q.get()      # 出队
```

五、教学参考资源

■ 参考书

- 史蒂芬·卢奇,丹尼·科佩克. 人工智能(第2版)[M]. 林赐,译. 北京:人民邮电出版社,2018.
- Stuart J. Russell, Peter Norvig. 人工智能:一种现代的方法(第3版)[M]. 殷建

平,祝恩,刘越,陈跃新,王挺,译.北京:清华大学出版社,2013.

六、教学参考案例

■ 参考案例 1

基本搜索方法

北京师范大学附属实验中学 郭培培

(1课时)

1. 教学目标

- 通过案例分析,了解搜索问题的形式化表示。(信息意识、计算思维)
- 通过观摩和实践,理解广度优先和深度优先搜索算法的基本思想及特点。(计算思维)
- 通过问题求解,加深对广度优先和深度优先搜索算法的理解,提高学习兴趣。(计算思维、数字化学习与创新)

2. 教学对象分析

本节课适用于高中选修“人工智能初步”模块的学生。在必修模块的课程中学生已经学习了 Python 语言编程基础知识,初步具备了从程序设计角度分析和解决简单问题的能力,但对于实现本节课的搜索算法还存在比较大的困难;在该模块第一章的学习中,学生对人工智能的应用场景、研究领域、发展历程有了一定的了解,对人工智能的学习充满了兴趣,但对人工智能中的算法以及如何实现还缺乏了解和认识。

搜索算法是人工智能最基本的一类问题的求解方法,也是人工智能中经常遇到的最重要的问题之一,它有着广泛的应用场景,与学生密切相关,因此学生对这部分内容的学习会比较感兴趣。为了帮助学生更好地学习搜索算法,在教学过程中通过直观的演示让学生理解算法的思想,通过细致讲解算法源代码帮助学生理解算法的实现过程,并通过引导学生调试代码以加深对算法实现的掌握。

3. 教学重难点

教学重点:搜索问题的形式化表示;广度优先搜索和深度优先搜索算法的基本思想和特点;用广度优先搜索和深度优先搜索算法求解简单的问题。

教学难点:广度优先搜索和深度优先搜索算法的编程实践。

4. 教学方法

讲授、演示、任务驱动、实践、讨论交流。

5. 教学工具

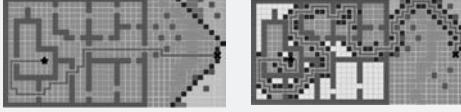
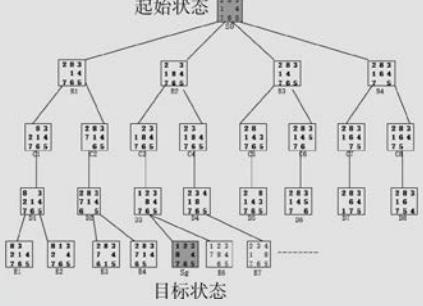
网络机房、教学课件、Python 编程环境。

6. 教学过程设计

教学过程设计如表 2-3 所示。

表 2-3 教学过程设计

教学环节	教学内容	学生活动	设计意图
情境引入	<p>1. 搜索是大部分人生活中的自然组成部分，例如找钥匙，我们可能检查口袋、抽屉，甚至翻箱倒柜。有时候，搜索可能在大脑中进行，例如跟朋友谈论到过哪些地方旅游；在早高峰时段选择走哪条路可以尽快到达学校等。</p> <p>2. 搜索是计算机科学中一种很重要的算法，也是人工智能中最基本的一类问题求解方法，它可以搜索走出迷宫的路径；搜索五子棋、象棋等游戏中下一步最佳移动方法；还可以帮助我们在导航中规划路径等。</p> <p>3. 搜索包括盲目搜索和启发式搜索等。盲目搜索包括广度优先和深度优先两种基本搜索方法</p>	<p>1. 倾听，积极思考生活中搜索的案例。</p> <p>2. 倾听，了解搜索算法的重要作用。</p> <p>3. 倾听，了解搜索算法的分类</p>	联系实际生活中的情境引出搜索的概念和分类，让学生了解搜索算法的重要性，同时激发学生学习搜索算法的兴趣
新知学习——搜索问题的表示	<p>引入问题：</p> <p>演示人工智能创新园区鸟瞰图，提出问题：如果已知有道路直接相连的展馆之间步行所需的时间，如何规划一条从 A 馆到 G 馆的可行路径呢？能否找到一条从 A 馆到 G 馆的最快步行路径呢？</p> <p>1. 将创新园区的路径规划问题形式化，将地点作为顶点，两点间的道路作为边，用图表示。</p> <p>2. 以路径规划问题为例，结合画出的图，给出搜索问题中几个重要的概念：状态、状态空间、行动、状态转移、目标测试和路径耗散</p>	<p>1. 倾听，思考。</p> <p>2. 倾听，思考</p>	通过分析路径规划案例，让学生了解搜索问题的形式化表示及其中一些重要的概念
新知学习——广度优先搜索	<p>1. 讲解广度优先搜索算法的基本思想。</p> <p>2. 基于路径规划问题建立的图，以广度优先搜索算法展开搜索，建立搜索树，找到一条 A 馆到 G 馆的解路径。</p> <p>3. 引导学生根据广度优先搜索算法的基本思想找出其他的解路径，并计算路径的耗散值。</p> <p>4. 打开广度优先搜索路径规划源程序，讲解算法的实现过程。</p> <p>5. 提示并引导学生调试程序，观察运行结果。</p> <p>(1) 可尝试修改园区路径图，自己分析广度优先搜索的解路径，运行程序验证。</p> <p>(2) 尝试找出所有可能的解路径</p>	<p>1. 倾听，思考。</p> <p>2. 倾听，思考。</p> <p>3. 期待答案： $A \rightarrow B \rightarrow F \rightarrow G$, 耗散值为 34。</p> <p>4. 倾听，理解源程序各部分的含义。</p> <p>5. 调试程序，观察并分析程序运行结果</p>	通过问题求解，理解广度优先搜索算法的基本原理和过程；了解搜索树的建立方法及如何应用搜索树求解问题；通过编程实践加深对广度优先搜索算法的理解和掌握
新知学习——深度优先搜索	<p>1. 讲解深度优先搜索算法的基本思想。</p> <p>2. 以深度优先搜索算法搜索 A 馆到 G 馆的解路径，建立搜索树。</p> <p>3. 引导学生根据深度优先搜索算法的过程找出其他的解路径，并计算路径的耗散值。</p> <p>4. 讲解和调试深度优先算法解决路径规划问题的程序，并演示搜索结果</p>	<p>1. 倾听，思考。</p> <p>2. 倾听，思考。</p> <p>3. 期待答案： $A \rightarrow B \rightarrow F \rightarrow G$, 耗散值为 34。</p> <p>4. 调试程序，观察并分析运行结果</p>	通过问题求解，理解深度优先搜索算法的基本原理和过程；通过编程实践加深对深度优先搜索算法的理解和掌握

教学环节	教学内容	学生活动	设计意图
归纳总结	<p>对本节课主要内容进行总结：</p> <ul style="list-style-type: none"> • 搜索问题的形式化表示。 • 广度优先和深度优先搜索算法的基本思想。 • 演示动画，对比广度优先和深度优先两种搜索算法，总结各自的特点。 	回顾，观看演示，将本节课的内容进行总结梳理	
拓展延伸	<p>以 8 - 数码游戏为例，引导学生分析如何表示该问题，如何用两种基本搜索方法解决问题。</p> <ol style="list-style-type: none"> 给出状态空间图及初始状态和目标状态，提问学生分别用广度优先搜索和深度优先搜索的结果是什么，并找出路径。  <ol style="list-style-type: none"> 分别演示广度优先搜索和深度优先搜索的过程。 布置课后任务：思考 8 - 数码游戏问题使用怎样的搜索策略效率会更高？可借助搜索引擎寻找问题的答案 	<ol style="list-style-type: none"> 倾听，思考，观看演示。 倾听，思考，观看演示。 课后积极探究思考 <p>通过剖析 8 - 数码游戏问题，让学生更深刻地了解两种基本搜索方法的基本原理、过程；通过布置课后任务让学生将学习延伸到课后，培养学生自主学习能力，同时为下节课启发式搜索做铺垫</p>	

参考案例 2

启发式搜索

北京师范大学附属实验中学 郭培培
(1 课时)

1. 教学目标

- 通过观摩和实践，理解启发式搜索的基本思想及其优点。（计算思维）
- 通过解决问题，理解 A* 搜索算法基本原理，体会启发函数的作用。（计算思维）
- 通过实例，了解启发式搜索在人工智能中的应用。（信息意识、数字化学习与创新）

2. 教学对象分析

本节课适用于高中选修“人工智能初步”模块的学生。通过上一节课的学习，学生对

搜索的概念和盲目搜索的基本思想和过程有了一定的了解。本节课继续沿用上节课的案例——人工智能创新园区路径规划和 8-数码游戏问题,通过用启发式搜索算法解决同一问题,让学生更好地理解启发式搜索的基本思想,体会其优点。

3. 教学重难点

教学重点:启发式搜索的基本思想;A* 算法的基本原理及启发函数的作用;启发式搜索在人工智能中的应用。

教学难点:用 A* 算法解问题。

4. 教学方法

演示、讲授、实践、对比、讨论交流。

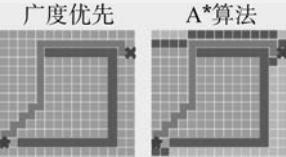
5. 教学工具

网络机房、投影、教学课件、Python 环境。

6. 教学过程设计

教学过程设计如表 2-4 所示。

表 2-4 教学过程设计

教学环节	教学内容	学生活动	设计意图
复习引入	<p>1. 复习广度优先搜索和深度优先搜索算法,提出其在解决规模比较大的问题时,例如在地图上搜索距离比较远的两点间的路径,将非常困难。</p> <p>2. 引出启发式搜索:利用经验性法则帮助问题求解,以减少搜索范围、降低问题复杂度。</p> <p>3. 演示启发式搜索和盲目搜索算法在解决同一个问题时不同效果的动画。</p> <p style="text-align: center;">广度优先 A*算法</p> 	<p>1. 倾听,思考。</p> <p>2. 倾听,思考。</p> <p>3. 观看启发式搜索效果演示动画,体会其优点</p>	通过回顾广度优先、深度优先搜索算法,引出本节课要学习的主要内容——启发式搜索;通过动画演示,激发学生学习兴趣,为讲授新知做铺垫
A* 算法 新知学习:	<p>以解决智能园区路径规划问题为例,学习 A* 启发式搜索算法的基本原理。</p> <p>1. 讲解 A* 算法: A* 算法是一种启发式搜索算法,它需要一个评估函数:</p> $f(n) = g(n) + h(n)$ <p>$g(n)$ 为起点到 n 点的实际代价; $h(n)$ 为 n 点到目标点的估计代价,又称为启发式函数。</p> <p>2. (1) 为路径规划图加入某节点到目标节点的直线步行时间启发式信息。 $g(n)$: 起点到 n 点花费的实际时间。</p>	<p>1. 倾听,理解 A* 启发式搜索算法的思想和过程。</p> <p>2. 倾听,思考,积极参与到 A* 算法求解路径的教学活动中。</p>	通过解决园区路径规划问题,让学生学习 A* 算法的基本思想、搜索过程及算法实现的要点

教学环节	教学内容	学生活动	设计意图
	<p>$h(n)$:n 点到目标节点的时间估值。</p> <p>(2) 以后续节点与目标节点的 f 值来决定展开的次序,引导学生一步步找到解路径,并计算 A 馆到 G 馆的时间。</p> <p>3. A^* 算法总结(思想、优点、算法实现要点)</p>		
编程实践	<p>1. 呈现 A^* 算法求解路径规划问题源代码,并讲解算法核心代码。</p> <p>2. 引导学生分组进行实践探究:</p> <p>(1) 如果启发函数估值与真实值差距比较大,结果怎么样?</p> <p>(2) 还可以选择其他的启发函数吗?</p> <p>3. 分享交流:</p> <p>组织学生分享实践成果</p>	<p>1. 倾听,理解代码含义。</p> <p>2. 积极参与到小组的实践探究活动。</p> <p>3. 积极表达自己的观点</p>	通过编程实践和分组探究,加强对 A^* 启发式搜索算法的理解
	<p>1. 引导学生分析作业练习中的 8 - 数码游戏问题,找出启发信息,设计合理的启发函数。</p> <p>$h(n)$: 节点 n 与目标状态的错位数。</p>	<p>1. 思考,回答问题。 期待答案: 节点 n 的状态与目标状态的错位数,或者不在目标位置的数字与其该在位置的曼哈顿距离和。</p>	通过用 A^* 算法解决 8 - 数码游戏问题,让学生对启发式搜索思想有更深刻的认识;
拓展延伸	<p>1. 不在目标位置的数字与其该在位置的曼哈顿距离和。</p> <p>2. 演示用 A^* 算法解决 8 - 数码游戏问题的过程。</p> <p>3. 启发式搜索算法的应用。</p> <p>启发式搜索是人工智能的基本技术之一,应用广泛。例如自动导航和游戏寻路等</p>	<p>2. 观看演示。</p> <p>3. 认真倾听</p>	通过介绍启发式搜索算法的应用,开拓学生视野,让学生了解更多的启发式搜索算法应用
课堂小结	<p>1. 启发式搜索算法的基本思想。</p> <p>2. 启发式搜索算法的优点。</p> <p>3. 启发式搜索算法的应用。</p> <p>4. 布置课后拓展阅读。</p> <ul style="list-style-type: none"> • 对抗搜索; • 蒙特卡洛树搜索; • 受到自然启发的搜索 	<p>回顾本节课学习的主要内容,认真倾听总结</p>	将本节课的内容进行总结梳理,让学生巩固新知;布置课后拓展阅读资料,让学生将学习延伸到课后,培养学生自主学习能力和数字化学习能力

第二节 专家系统

一、教学目标与重点

教学目标:

- 通过实例分析理解专家系统的组成。(信息意识)
- 通过项目实践了解基于规则的知识表示方法。(计算思维)
- 通过项目实践理解生成决策树的经典算法,能解释算法的关键环节。(计算思维、数字化学习与创新)
- 通过探究活动“由决策树产生推理规则”,了解知识推理的方法及其在专家系统中的应用。(信息意识、信息社会责任)

教学重点:

- 专家系统的组成;
- 生成决策树的基本过程;
- 专家系统推理的过程。

二、内容分析与建议

1. 内容分析

本节从构建个性化的推荐系统——“创新园区自动驾驶展馆参观体验”入手,引导学生思考微型专家系统如何根据以往经验数据和每个学生的特征,预测学生的参观体验,从而给出推荐参观的建议,进而引出本节主题——专家系统。接下来,向学生介绍专家系统的构成,包括知识库、综合数据库、推理机、接口、解释机。然后通过项目实践,让学生了解基于规则的知识表示方法,并构建专家系统的知识库。最后,通过项目实践,让学生理解生成决策树的经典算法及其关键环节,再以探究活动“由决策树产生推理规则”,让学生了解知识推理的方法及其在专家系统中的应用。本节主要知识框图如图 2-7 所示。

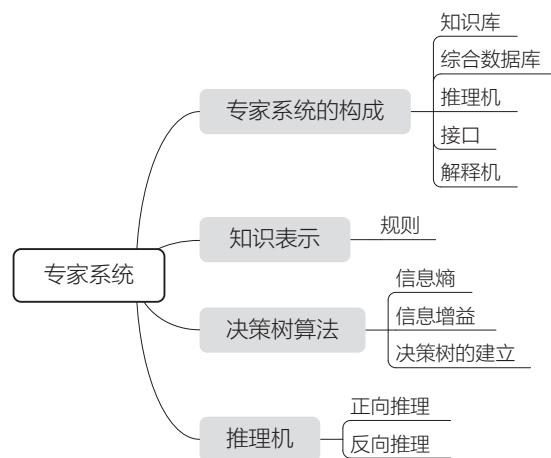


图 2-7 主要知识框图

2. 教学建议

(1) 教学活动建议

本节项目主题为“构建个性化的推荐系统”，这是在章节项目主题“创新园区自由行”下的一个案例，教学过程中，教师既可以让学生以设计者和参与者的身份沉浸在项目活动创设的情境中，也可以结合学生生活和学习的实际情境，提出其他类似问题。例如：你在脑海中想某个事物，机器向你提出 20 个问题，只能用“对”或“错”来回答，机器通过你的回答进行推断分析，逐步缩小待猜测事物的范围，最后猜出你想的事物；银行希望能够通过一个人的职业、年龄、收入、学历等信息，去判断他是否有贷款的意向；地产中介希望使用回归决策树进行房屋租赁或出售价格预测等。

本节所介绍的专家系统，本身也属于信息系统，可以让学生结合必修模块 2“信息系统与社会”中信息系统的相关知识，分析专家系统的结构组成。同时也可以让学生体会专家系统与传统信息系统之间的差异，感受新一代人工智能中基于数据和学习的方法给专家系统带来的影响。

本节内容中涉及到了特征、特征值、标签等机器学习的概念，教师要适当强化这些基本概念，为后续章节做好铺垫。此外还有信息熵、信息增益、基尼不纯度等数学概念，教师要尽可能避免单方面的讲述和灌输，要善于引导学生进行积极的思考和讨论，不但要学会计算，还要明白公式背后的原理。鼓励学生勇于动手实践，可以根据实际情况让学生手动计算信息熵、信息增益和基尼不纯度，再使用 Python 编写程序进行计算，通过师生交流、讨论和实践，帮助学生逐步理解和内化这些数学概念；鼓励学生在编写和调试程序的过程中进行主动思考，切身感受知识在实践中的应用，从而提高学生学习人工智能的积极性。

教学过程中可以设计导学案，通过导学案引导学生独立思考，实现掌握知识和发展能力的统一，同时也能检测学生知识点的掌握情况，达到让学生自主学习和主动探究的目的。

(2) 项目实施建议

本节的体验思考、探究活动与作业练习均围绕项目主题“自动驾驶展馆的个性化推荐”专家系统展开，教师可以将这些活动贯穿于项目活动的始终，帮助学生分析问题、建构知识、提升能力。在实际教学中，教师也可以让学生结合实际生活或兴趣爱好，由各组学生通过小组讨论自主确定项目主题，通过归纳总结，得出结论并进行展示。此时，教师应特别注意对学生的引导，帮助学生选取主题健康积极、基于真实情境或问题解决的项目，从而激发学生实施项目的热情。

在根据个人信用数据集，用决策树训练分类模型以评估或预测信用风险的项目实践中，一方面，教师需要提前安装好相应的第三方模块库，并提供给学生足够的注释语句，帮助他们了解程序中一些语句的基本功能和用法；另一方面，要注意引导学生从分析问题入手，了解 scikit-learn 中的决策树模型，最后实现决策树的可视化。编写程序不是学习的最终目的，掌握解决问题的方法和算法实现的原理，举一反三，用以解决其他问题才是关键。

由于本节内容对学生来说比较陌生，学生普遍缺乏学习基础，因此在项目活动中，教

师可事先准备一些学习资料,搭建好学生自主学习的平台,为学生个性化的学习需求提供指导,同时也能记录下学生学习的过程性资料。

组织学生通过探究活动“由决策树产生推理规则”,感受专家系统推理的过程,引导学生分析、总结出正向推理、反向推理的原理和步骤。

(3) 评价建议

评价方式:以过程性评价为主,主要由教师进行课堂观察评价并对项目活动进行检查,考查学生灵活运用所掌握的知识进行实践操作、交流协作等能力。

评价要点:在课堂学习过程中,学生是否能积极思考、踊跃交流、表达观点、调试程序,主动参与到课堂活动中去;是否真正掌握运用决策树算法自动构建知识库的方法;是否乐于动手实践调试和修改程序,开展小组合作,实现决策树监督分类模型,并将结果可视化。

三、习题分析与提示

■ 习题 1 描述

见教科书第 47 页的探究活动“决策树中的特征”。

■ 习题 1 参考答案

在图 2-8(即教科书第 47 页图 2.27)中,“天气”特征虽然没有出现在决策树上,但不能说天气状况与是否推荐参观没有关系。构建决策树的时候,对每个分类节点选择特征生成新的分类节点,直到信息增益(比)很小或者没有特征可以选择为止。“天气”最后没有出现在决策树里,有可能是由于天气状况与是否推荐参观无关,也可能是由于特征之间有相关性,存在冗余,天气状况的特征被其他特征覆盖了。

根据给定的训练数据集构建一个决策树模型,该模型针对该训练集的确是非常完美的。但是,这种完美就使得整体模型的复杂度较高,而对其他数据集的预测能力就降低了,即产生了过拟合,使得模型的泛化能力变弱。

解决办法:决策树的剪枝,即通过优化损失函数来去掉一些不必要的分类特征,降低模型的整体复杂度。剪枝的方式,即从树的叶子节点出发,向上回溯,逐步判断。如果去掉某一特征后,整棵决策树所对应的损失函数更小,那就将该特征及带有的分支剪掉。单纯的决策树生成学习局部的模型,而剪枝后的决策树会生成学习整体的模型,因为剪枝的过程中,通过最小化损失函数,可以平衡决策树对训练数据的拟合程度和整个模型的复杂度。

■ 习题 2 描述

见教科书第 50 页的探究活动“由决策树产生推理规则”。

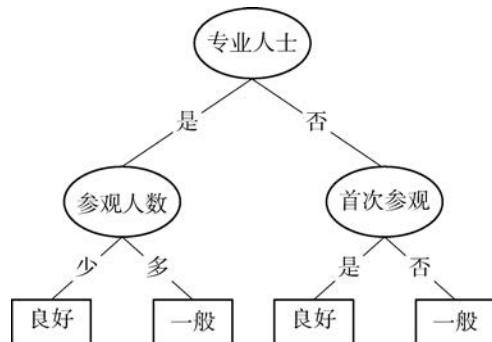


图 2-8 场馆推荐决策树

■ 习题 2 提示

参考思路如下：

如果是专业人士且参观人数少，则体验良好。

如果是专业人士且参观人数多，则体验一般。

如果是非专业人士且首次参观，则体验良好。

如果是非专业人士并且非首次参观，则体验一般。

正向推理从事实出发，逐步运用规则推导出结论，即根据是否是专业人士、参观人数、是否首次参观等特征推出参观体验良好或一般的结论。反向推理从结论逐步回溯到事实，即根据体验结果标示出能够导出结论的相关知识，然后再由标示的部分根据规则推导出结论，从而完成验证。

在专家系统中，当输入的事实数据特征及特征值很明确的时候，适合正向推理；而当标签已经确定，需要找到相应的特征和特征值时，则适合反向推理。例如：参观者是专业人士，并且参观人数较少，或者参观者是非专业人士，并且是首次参观，可以通过正向推理得到体验良好、建议参观的结论。如果想咨询什么情况下参观者的体验良好适合参观，则可以通过反向推理，给出结论：专业人士在参观人数不多的情况下建议参观；或者非专业人士首次参观的情况下建议参观。

除正向推理和反向推理外，还有一种双向推理，是正向推理与反向推理同时进行，且在推理过程中的某一步骤上“碰头”的一种推理方式。

■ 习题 3 描述

见教科书第 50 页的作业练习“构建专家系统和知识库”。

■ 习题 3 提示

学生能够对个性化展馆推荐系统的知识库、综合数据库、推理机、接口、解释机做出定义即可，考查学生对专家系统各组成部分的具体含义和功能的理解，学生可结合图 2-9（即教科书第 39 页图 2.22）的构成来逐一定义。

(1) 知识库

存储基于参观者的参观体验（标签）与参观者是否专业人士、是否初次参观、天气情况、人数多少等特征之间关系的若干规则的库。比如：if[(专业人士 == 是) and(首次参观 == 是)]then(参观体验 == 良好)。

(2) 综合数据库

存储推理过程中所需的原始数据（参观者是否专业人士、是否初次参观、天气情况、人数多少）、中间结果和最终结论（是否建议参观），往往作为暂时的存储区。

(3) 推理机

也称为推理引擎或推理机制（包括正向推理和反向推理），推理机针对参观者是否专业人士、是否初次参观、天气情况、人数多少等特征的值去匹配知识库中的规则，为参观者提出是否应该参观的建议。

(4) 接口

参观者与个性化展馆推荐系统的交互界面。参观者可以在接口输入信息(例如参观者是否专业人士、是否首次参观、天气情况、人数多少等),而专家系统反馈回来的结论和解释也会通过接口呈现给参观者。

(5) 解释机

个性化展馆推荐系统除了要给参观者提出个性化建议,还应该给出相应的解释和说明,这部分即解释机,解释机使个性化展馆推荐系统更加人性化。

四、核心概念精解与技术工具介绍

1. 核心概念

(1) 专家系统

专家系统指的是这样的计算机程序或应用,其内部含有某个领域专家水平的知识与经验,能够利用人类专家所积累的知识、方法和经验,来解决该领域中的实际问题。专家系统最核心的部分是它所包含的知识,因此,有时也将专家系统称为基于知识的系统。新一代人工智能兴起后,基于大数据和深度学习的方法使从海量无结构多信息源中自动构建知识库并生成推理规则成为可能,这也为专家系统带来了新的发展机遇。

(2) 知识库

知识库是由领域专家的经验、知识、方法汇集而成的库。每个专家系统通常都按照其知识库中知识的表示方法(即知识表示)来称呼。比如,基于规则的专家系统就是指该专家系统中的知识是以一条条规则的形式来表示的。

(3) 综合数据库

综合数据库指专家系统中所使用的除知识库以外的其他类型的数据,如通用信息、背景信息和一些常识等。

(4) 推理机

推理机也称为推理引擎或推理机制,指专家系统处理知识和使用知识获得结论的机制。最常用的推理方法包括:正向推理和反向推理。

正向推理从事实出发,逐步运用规则推导出结论。

反向推理采取从结论逐步回溯到事实的方式,先自结论开始,标示出能够导出结论的相关知识,然后再由标示的部分根据规则推导出结论,从而完成验证。

(5) 接口

和普通的信息系统一样,接口指用户与专家系统进行交互的方式。

(6) 解释机

专家系统不但要响应用户的请求,得出结论,还要能对导出结论的依据给出解释和说明,这部分即解释机。

(7) 决策树

决策树是广泛用于分类和回归任务的模型,决策树的模型呈树形结构。每个节点表

示待分类目标的一个属性,每个分支代表基于该属性所做出的一个判断,树形结构中的每个叶子节点代表一种分类结果。用决策树模型进行决策的过程,就是在模仿人类大脑做真实决策的过程。常见的决策树算法包括 ID3、C4.5、CART 等。

(8) 信息熵

也简称为熵,是一个用来描述数据不确定性的量:数据的不确定性越高,信息熵的值就越大;反之,数据的确定性越高,信息熵的值就越小。

若设一个场景可能有 x_i 个事件,用 $p(x_i)$ 表示事件 x_i 发生的概率,则该场景的信息熵定义为:

$$E(P) = - \sum_i p(x_i) \log_2 p(x_i)$$

当 x_i 仅取有限个离散值时,概率 $p(x_i)$ 经常用频率来代替。

信息熵的公式中对概率取负对数表示了一种可能事件发生时携带的信息量。把各种可能出现的信息量乘以其发生的概率之后求和,就表示了整个系统所有信息量的一种期望值。从这个角度来说,信息熵还可以衡量一个系统的复杂程度,系统越复杂,出现不同情况的种类越多,信息熵就越大;系统越简单,出现不同情况的种类越少(极端情况为 1 种情况,则对应概率为 1,对应的信息熵为 0),此时的信息熵较小。

(9) 信息增益

信息增益:得知特征 X 的信息而使得类 Y 的信息不确定性减少的程度。

特征 A 对训练数据集 D 的信息增益 $Gain(D, A)$, 定义为集合 D 的经验熵 $H(D)$ 与特征 A 给定条件下 D 的经验条件熵 $H(D | A)$ 之差,即

$$Gain(D, A) = H(D) - H(D | A)$$

信息熵可以表示数据的不确定性,熵越大,数据的不确定性就越大。因此可以使用划分数据集前后信息熵的差值来衡量使用当前特征对于数据集合划分效果的好坏。如专业人士的信息增益为:

$$Gain(D, \text{专业人士}) = E(D) - \sum_{i=1}^2 \frac{|D_i|}{|D|} E(D_i)$$

一个特征的信息增益越大,表明使用该特征来对样本进行划分所获得的“纯度提升”越大,即这个特征使得数据由不确定变确定的能力越强,那么这个特征对于分类来说就很关键。我们可以用信息增益来进行决策树的划分特征选择。

(10) 基尼(Gini)不纯度

基尼不纯度(基尼指数):表示在样本集合中一个随机选中的样本被分错的概率。

$$I_G(F) = \sum_{i=1}^N t_i (1 - t_i) = 1 - \sum_{i=1}^N t_i^2$$

其中 t_i 表示标签为 i 的数据项在样本中所占的比例。

注意:基尼指数越小表示集合中被选中的样本被分错的概率越小,也就是说集合的纯

度(purity)越高,反之,集合越不纯。

(11) 构建决策树的过程

要构建决策树,关键是要确定决策树的分类节点,即如何选择最优划分的特征。一般而言,随着划分过程不断进行,我们希望决策树的分类节点所包含的样本尽可能属于同一类别,即节点的纯度越来越高。确定分类节点的方法是:通过指标选择分叉后不纯度降低最多的特征作为分类节点,对每个分类节点使用相同的方式生成新的分类节点,直到指标很小或者没有特征可以选择为止。主要有以下三种算法:

ID3 算法:在决策树的各个节点上应用信息增益准则进行特征选择,增益大的特征优先选择。ID3 算法只适用于离散的特征。

C4.5 算法:与 ID3 类似,区别是用信息增益比准则进行特征选择,增益率大的特征优先选择。使用信息增益比来选择特征,可以减少信息增益容易选择特征值多的特征的问题。C4.5 既可用于处理离散的特征,也可用于处理连续的特征。当终节点是连续变量时,则该树为回归树,当终节点为离散变量时,该树为分类树。

CART(classification and regression tree)算法:ID3 和 C4.5 都是基于信息论的熵模型,会涉及大量的对数运算。CART 算法使用基尼系数来代替信息增益比,基尼系数代表了模型的不纯度,基尼系数越小,不纯度越低,特征越好。这和信息增益(比)是相反的,是一种高效的非参数分类/回归方法。通过构建树,修剪树,评估树来生成一个二叉树。

下面就以 ID3 算法为例,对自动驾驶展馆用户的参观体验建立决策树。

① 计算数据集的信息熵:

首先观察数据集 D ,发现数据集 D 中有参观体验“良好”和“一般”两个类别,其中体验“良好”占比 $p_1 = \frac{9}{20}$,体验“一般”占比 $p_2 = \frac{11}{20}$,计算出数据集 D ,即参观体验的信息熵为:

$$E(D) = -\sum_{i=1}^2 p(x_i) \log_2 p(x_i) = -\left(\frac{9}{20} \log_2 \frac{9}{20} + \frac{11}{20} \log_2 \frac{11}{20}\right) = 0.99$$

② 计算各个属性的信息增益:

观察数据集可以发现,用户的特征有四个:首次参观、参观人数、天气、专业人士,首先对专业人士计算信息增益。

用户分专业人士和非专业人士两种情况。专业人士记作 D_1 ,非专业人士记作 D_2 ,对于 D_1 ,参观体验“良好”占 $\frac{2}{11}$,参观体验“一般”占 $\frac{9}{11}$;对于 D_2 ,参观体验“良好”占 $\frac{7}{9}$,参观体验“一般”占 $\frac{2}{9}$,分别计算它们的信息熵:

$$E(D_1) = -\left(\frac{2}{11} \log_2 \frac{2}{11} + \frac{9}{11} \log_2 \frac{9}{11}\right) = 0.68$$

$$E(D_2) = -\left(\frac{7}{9} \log_2 \frac{7}{9} + \frac{2}{9} \log_2 \frac{2}{9}\right) = 0.76$$

计算信息增益(专业人士):

$$\begin{aligned}Gain(D, \text{专业人士}) &= E(D) - \sum_{i=1}^2 \frac{|D_i|}{|D|} E(D_i) \\&= 0.99 - \left(\frac{11}{20} * 0.68 + \frac{9}{20} * 0.76 \right) \\&= 0.27\end{aligned}$$

也就是说通过专业人士这一特征,可以使参观体验的信息熵下降 0.27。同理可以计算出:

$$Gain(D, \text{首次参观}) = 0.11$$

$$Gain(D, \text{参观人数}) = 0.13$$

$$Gain(D, \text{天气}) = 0.02$$

比较首次参观、参观人数、天气和专业人士这四个特征,可以发现专业人士这个特征的信息增益最大,这表明专业人士这个特征在决定参观体验上是最有贡献的。于是选择专业人士作为数据集 D 的划分标准,构造第一级的决策树(如图 2-10 所示)。



图 2-10 第一级决策树

决策树算法对每一个分支节点继续进行划分:对于 11 个专业人士,计算参观体验的信息熵,结果约为 0.68,在此基础上,分别计算首次参观、参观人数和天气在这 11 个数据上的信息增益。经比较,参观人数的信息增益最高,约为 0.68。以参观人数为第二层特征,把数据分成参观人数多的一组和参观人数少的一组。参观人数少的一组都体验良好,参观人数多的一组都体验一般。

对于 9 个非专业人士进行划分,方法相同,最后生成的决策树如图 2-11 所示。

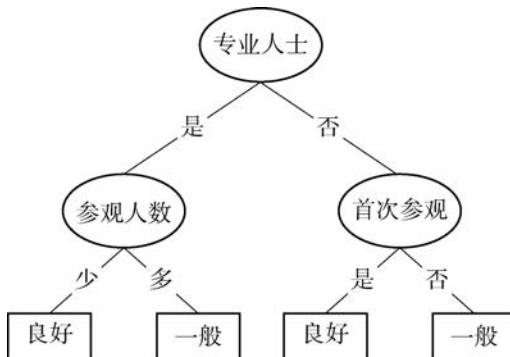


图 2-11 场馆推荐决策树

其中每一个非叶节点(图中的椭圆形节点)都代表一个特征,通过该特征的取值来判断样本属于正样本还是负样本。决策树的每个叶节点(图中的矩形节点)数据都是一种标签。

决策树算法通过对数据信息增益的计算,将经验提升到规则,再把规则显性地表现出来。

2. 技术工具

本节涉及的技术工具:专家系统相关网站或 App,决策树程序。

scikit-learn,也称为 sklearn,是一个开源的基于 Python 语言的机器学习工具包,它建立在 NumPy、SciPy 和 Matplotlib 上,涵盖了几乎所有主流的机器学习算法,是简单高效的数据挖掘和数据分析工具,可以在各种环境中重复使用。

scikit-learn 包含六大任务模块,分别是分类、回归、聚类、降维、模型选择和预处理。

分类:识别某个对象属于哪个类别。常用的算法有:SVM(支持向量机)、nearest neighbors(最近邻)、random forest(随机森林)。常见的应用有:垃圾邮件识别、图像识别等。

回归:预测与对象相关联的连续值属性。常见的算法有:SVR(支持向量回归)、ridge regression(岭回归)、LASSO(最小绝对收缩和选择算法)。常见的应用有:药物反应、预测股价等。

聚类:将相似对象自动分组。常用的算法有: k -均值聚类算法、spectral clustering(谱聚类)、mean-shift(均值偏移)。常见的应用有:客户细分、分组实验结果等。

降维:减少要考虑的随机变量的数量。常见的算法有:PCA(主成分分析)、feature selection(特征选择)、non-negative matrix factorization(非负矩阵分解)。常见的应用有:可视化、提高效率等。

模型选择:比较,验证,选择参数和模型。常用的模块有:grid search(网格搜索)、cross validation(交叉验证)、metrics(度量)。它的目标是通过参数调整提高精度。

预处理:特征提取和归一化。常用的模块有:preprocessing(数据预处理)、feature extraction(特征提取)。常见的应用有:把输入数据(如文本)转换为机器学习算法可用的数据等。

scikit-learn 的结构图如图 2-12 所示。

五、教学参考资源

■ 参考资料 1:专家系统相关介绍

1. 简介

专家系统是人工智能中最重要的也是最活跃的一个应用领域,它实现了人工智能从理论研究走向实际应用、从一般推理策略探讨转向运用专门知识的重大突破。专家系统是早期人工智能的一个重要分支,它可以看作是一类具有专门知识和经验的计算机智能程序系统,一般采用人工智能中的知识表示和知识推理技术来模拟通常由领域专家才能

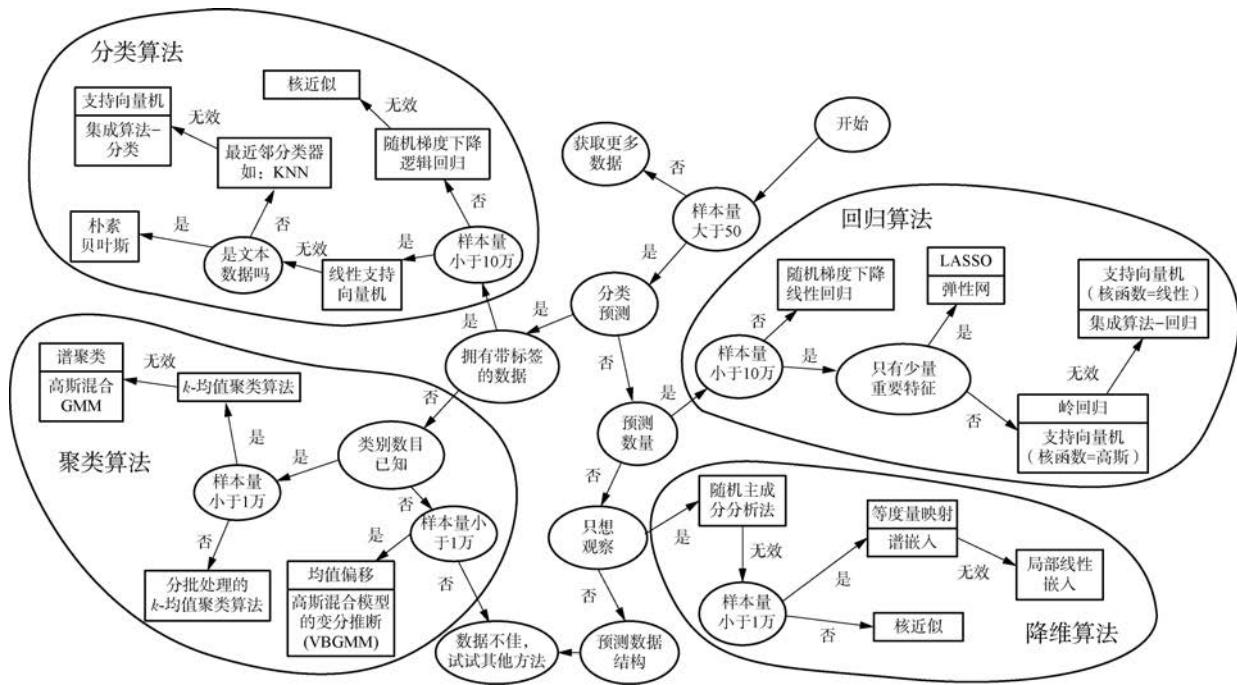


图 2-12 scikit-learn 的结构图

解决的复杂问题。

2. 起源与发展

20世纪60年代初,出现了运用逻辑学和模拟心理活动的一些通用问题求解程序,它们可以证明定理和进行逻辑推理。但是这些通用方法无法解决大的实际问题,很难把实际问题改造成适合于计算机解决的形式,并且对于解题所需的巨大的搜索空间也难以处理。1965年,研究人员在总结通用问题求解系统的成功与失败经验的基础上,结合化学领域的专门知识,研制了世界上第一个专家系统,此专家系统可以推断化学分子结构。几十年来,知识工程的研究、专家系统的理论和技术不断发展,应用渗透到几乎各个领域,包括化学、数学、物理、生物、医学、农业、气象、地质勘探、军事、工程技术、法律、商业、空间技术、自动控制、计算机设计和制造等众多领域,开发出几千个专家系统,其中不少在功能上已达到甚至超过同领域中人类专家的水平,并在实际应用中产生了巨大的经济效益。

专家系统的发展已经历了三代,正向第四代过渡和发展。第一代专家系统(dendral、macsyma等)以高度专业化、求解专门问题的能力强为特点。但在体系结构的完整性、可移植性、系统的透明性和灵活性等方面存在缺陷,求解问题的能力弱。第二代专家系统(mycin、casnet、prospector、hearsay等)属单学科专业型、应用型系统,其体系结构较完整,移植性方面也有所改善,而且在系统的人机接口、解释机制、知识获取技术、不确定推理技术、增强专家系统的知识表示和推理方法的启发性、通用性等方面都有所改进。第三代专家系统属多学科综合型系统,采用多种人工智能语言,综合采用各种知识表示方法和多种推理机制及控制策略,并开始运用各种知识工程语言、骨架系统及专家系统开发工具和环境来研制大型综合专家系统。在总结前三代专家系统的设计方法和实现技术的基础

上,已开始采用大型多专家协作系统、多种知识表示、综合知识库、自组织解题机制、多学科协同解题与并行推理、专家系统工具与环境、人工神经网络知识获取及学习机制等最新人工智能技术来实现具有多知识库、多主体的第四代专家系统。

3. 功能

根据定义,专家系统应具备以下几个功能,其中存放知识和运用知识进行问题求解是专家系统的两个最基本的功能。

- (1) 存储问题求解所需的知识。
- (2) 存储具体问题求解的初始数据和推理过程中涉及的各种信息,如中间结果、目标、字母表以及假设等。
- (3) 根据当前输入的数据,利用已有的知识,按照一定的推理策略,去解决当前问题,并能控制和协调整个系统。
- (4) 能够对推理过程、结论或系统自身行为做出必要的解释,如解题步骤、处理策略、选择处理方法的理由、系统求解某种问题的能力、系统如何组织和管理其自身知识等。这样既便于用户理解和接受,同时也便于系统维护。
- (5) 提供知识获取,机器学习以及知识库的修改、扩充和完善等维护手段。只有这样才能更有效地提高系统的问题求解能力及准确性。
- (6) 提供一种用户接口,既便于用户使用,又便于分析和理解用户的各种要求和请求。

4. 特点

专家系统是一个基于知识的系统,它利用人类专家提供的专门知识,模拟人类专家的思维过程,解决对人类专家都相当困难的问题。一般来说,一个高性能的专家系统应具备如下特点:

- (1) 启发性。不仅能使用逻辑知识,也能使用启发性知识,它运用规范的专门知识和直觉的评判知识进行判断、推理和联想,实现问题求解。
- (2) 透明性。它使用户在对专家系统结构不了解的情况下,可以与系统进行相互交流,并了解知识的内容和推理思路,系统还能回答用户的一些有关系统自身行为的问题。
- (3) 灵活性。专家系统的知识与推理机构的分离,使系统不断接纳新的知识,从而确保系统内知识不断增长以满足商业和研究的需要。

5. 分类

根据知识表示的技术,可将专家系统分为:基于逻辑的专家系统、基于规则的专家系统、基于语义网络的专家系统和基于框架的专家系统等。

资料来源 百度百科“专家系统”

■ 参考资料 2:决策树的优缺点

1. 优点

- (1) 能够直接体现数据的特点,易于理解和实现,使用者在学习过程中不需要了解很多的背景知识,只要通过解释都有能力去理解它所表达的意义。

(2) 对于决策树,数据的准备往往是简单或者是不必要的,而且它能够同时处理数据型和常规型属性,在相对短的时间内能够对大型数据源做出可行且效果良好的结果。

(3) 易于通过静态测试来对模型进行评测,可以测定模型可信度;如果给定一个观察的模型,那么根据所产生的决策树很容易推出相应的逻辑表达式。

2. 缺点

- (1) 对连续性字段比较难预测。
- (2) 对有时间顺序的数据,需要做很多预处理的工作。
- (3) 当类别太多时,错误可能就会增加得比较快。
- (4) 一般的算法分类的时候,只是根据一个字段来分类。

资料来源 百度百科“决策树”

■ 参考资料 3: 正向推理的过程

正向推理时,用户首先给专家系统提供一批事实(已知数据),将它们存储到综合数据库中,作为初始值。如图 2-13 所示,正向推理的过程如下:

步骤 1 推理机将知识库中的规则前提与这些事实进行匹配。一般是将每条规则的前提提取出来,验证这些前提是否在数据库中。若存在,则匹配成功;若不存在,则取下一条规则进行匹配。

步骤 2 把匹配成功的规则的结论作为新的事实,添加到综合数据库中。

步骤 3 用更新后的综合数据库中的事实,重复上面两个步骤,直到某个事实就是期望的结论,或是不再有新的事实产生为止。

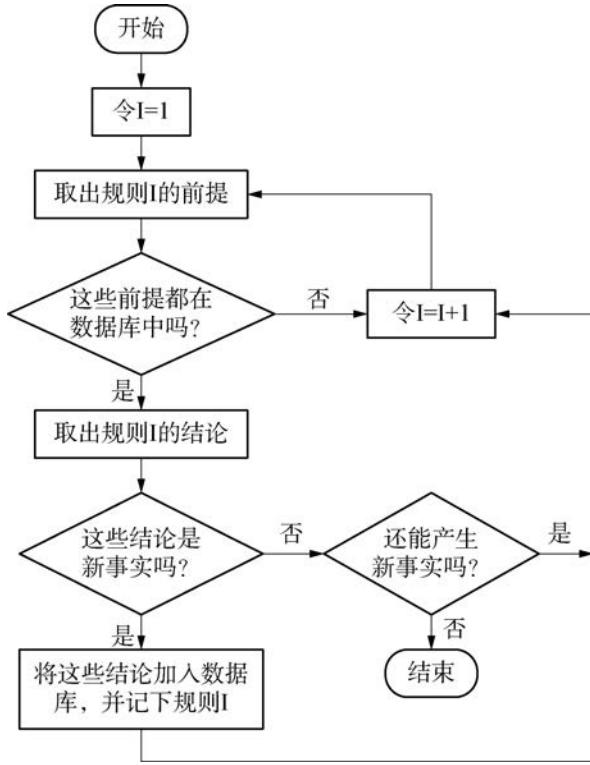


图 2-13 正向推理的过程

■ 参考资料 4: 反向推理的过程

反向推理就是用户或系统提出一些假设,然后系统来验证这些假设的真假。它的推理过程可以理解为从目标出发,反向使用规则进行推理。如图 2-14 所示,反向推理的过程如下:

步骤 1 看假设是否在综合数据库当中。若在,则假设成立,推理结束或进行下一个假设的验证;否则,进入下一步骤。

步骤 2 判断这些假设是不是证据节点。若是,系统向用户提问;否则,进入下一步骤。

步骤 3 找出结论部分包含这些假设的规则,把这些规则的所有前提作为新的假设。

步骤 4 重复步骤 1。

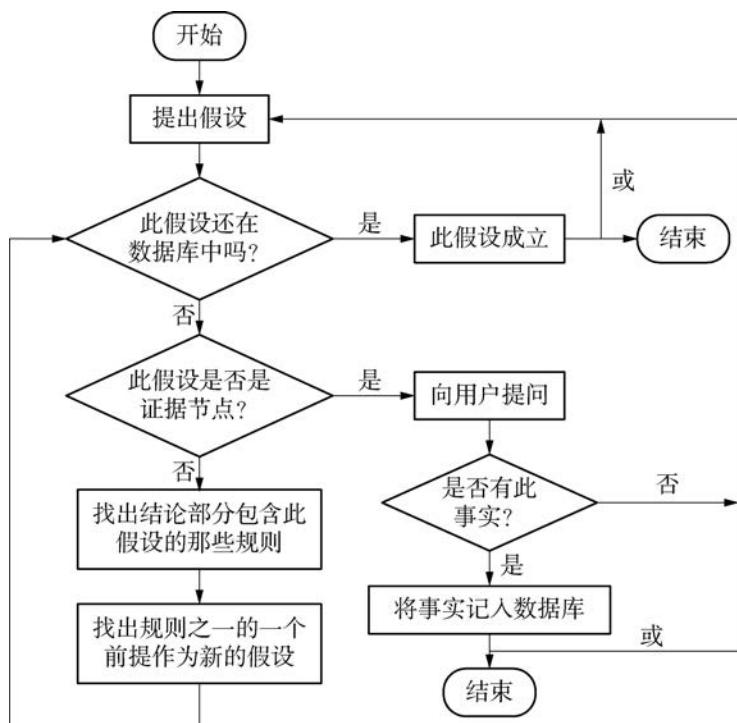


图 2-14 反向推理的过程

六、教学参考案例

■ 参考案例 1

使用决策树进行分类

北京师范大学附属实验中学 韩冬兵
(2 课时)

1. 教学目标

- 了解决策树的基本概念,能举例说明算法的应用场景,并将其适当运用在学习与生

活中。(信息意识、信息社会责任)

- 了解生成决策树的经典算法之一——ID3 算法,理解算法的基本原理和关键环节。(信息意识、计算思维)
- 通过项目实践,体验构建决策树的过程。(计算思维、数字化学习与创新)
- 通过决策树可视化,进一步理解决策树上每个节点的属性特征值、信息熵、样本量等信息。(信息意识)

2. 教学对象分析

学生在之前的学习中已经对人工智能的概念、发展历程有了初步了解,具备了一定的 Python 语言编程功底,初步了解了机器学习的一般过程,但对基本的机器学习方法如分类、聚类、回归等还缺乏全面的了解和把握。

决策树是一种比较基础的机器学习方法,有别于其他“黑盒”式的机器学习方法,决策树的可视化使它具有非常好的可解释性,因此高中生对这部分内容的学习会比较感兴趣。但构造决策树时需要用到信息熵、信息增益等抽象的数学知识,理解起来会有一定难度,所以在教学过程中利用直观的图形、程序等工具帮助学生理解概念,降低学习的难度。另外,高中生已经具备一定的自主探究能力,因此以小组为单位,在整个课堂活动中穿插自主探究体验活动,培养学生的动手实践和合作交流能力。

3. 教学重难点

教学重点:了解 ID3 算法的基本原理和关键环节;使用 scikit-learn 构建决策树;决策树的可视化。

教学难点:信息熵与信息增益的计算与分析;使用 scikit-learn 构建决策树;通过决策树可视化,进一步理解决策树上每个节点的属性特征值、信息熵、样本量等信息。

4. 教学方法

- 通过形象直观的图形、程序等工具帮助学生理解信息熵等数学概念;
- 通过决策树可视化,并对节点的信息进行进一步分析,巩固和内化决策树相关概念和算法的基本原理;
- 采用教师讲授与项目实践相结合的形式,从理解人工智能的基本概念、分析相关技术原理、典型案例实践和利用人工智能技术解决真实情境中的问题等方面开展教学,将“创新园区展馆体验”项目贯穿整个教学过程,培养学生解决真实情境中的问题的思维方式和能力。

5. 教学工具

网络机房、投影、Python 运行环境、Python 集成开发环境 Anoconda、scikit-learn 库等。

6. 教学过程设计

教学过程设计如表 2-5 所示。

表 2-5 教学过程设计

教学环节	教学内容						学生活动	设计意图
问题引入	提出问题：人工智能创新园区的自动驾驶展馆收集了20份用户反馈意见。根据用户是否首次参观、展馆参观人数多少、天气情况、是否是专业人士等情况，用户体验是否良好等数据，我们是否能根据这些以往经验数据和每个用户的个体特征，来预测用户的参观体验呢？						思考,听讲,回答教师提问	提出问题,引出决策树的概念,为学习新知做铺垫
	样本	首次参观	参观人数	天气	专业人士	参观体验		
	1	是	多	雨	是	良好		
	2	是	多	晴	是	良好		
	3	否	少	雨	否	良好		
	4	否	多	阴	是	一般		
	5	是	多	晴	否	一般		
	6	是	少	阴	否	良好		
	7	是	多	雨	否	一般		
	8	否	多	雨	是	一般		
	9	否	少	晴	是	一般		
	10	否	多	阴	是	一般		
	11	否	少	晴	否	良好		
	12	否	多	雨	是	一般		
	13	否	少	晴	是	一般		
	14	是	少	阴	否	良好		
	15	否	多	雨	是	一般		
	16	是	少	雨	否	良好		
	17	否	少	晴	否	良好		
	18	否	少	雨	是	一般		
	19	否	少	阴	否	良好		
	20	否	少	晴	是	一般		
有没有更直观的方法能表示以上数据,从而做出相应的预测呢?								
1. 认识决策树的工作原理								
(1) 概念讲解:画出上述展馆体验问题的决策树,讲解根节点、内部节点和叶子节点的概念。								

续表

教学环节	教学内容	学生活动	设计意图																								
	<p>(2) 分组体验如何使用决策树：提供几组参观人员特征数据，引导学生根据上述决策树，从根节点开始，对每个内部节点逐层向下移动，直至到达叶子节点，完成分类，观察分类结果，是否与该组数据的实际种类相符。</p> <table border="1"> <thead> <tr> <th>样本</th> <th>首次参观</th> <th>参观人数</th> <th>天气</th> <th>专业人士</th> <th>参观体验</th> </tr> </thead> <tbody> <tr> <td>21</td> <td>是</td> <td>多</td> <td>晴</td> <td>否</td> <td>良好</td> </tr> <tr> <td>22</td> <td>否</td> <td>少</td> <td>晴</td> <td>是</td> <td>良好</td> </tr> <tr> <td>23</td> <td>是</td> <td>多</td> <td>雨</td> <td>是</td> <td>一般</td> </tr> </tbody> </table> <p>(3) 教师总结：</p> <ol style="list-style-type: none"> ① 根节点包含整个样本集，是开始进行分类的节点。 ② 内部节点决定下层往哪个方向走，对应一个特征测试。 ③ 叶子节点对应最终的分类结果 	样本	首次参观	参观人数	天气	专业人士	参观体验	21	是	多	晴	否	良好	22	否	少	晴	是	良好	23	是	多	雨	是	一般	思考，听讲，理解决策树工作原理。 利用给出的展馆参观体验决策树对数据进行手动分类，比较分类结果，体会决策树分类效果	新知讲解，让学生初步认识决策树。 使学生对决策树的用法有直观的概念
样本	首次参观	参观人数	天气	专业人士	参观体验																						
21	是	多	晴	否	良好																						
22	否	少	晴	是	良好																						
23	是	多	雨	是	一般																						
新知学习	<h3>2. 构造决策树</h3> <p>(1) 讲解构造决策树的一般流程： 收集数据—准备数据—训练算法—测试算法—使用算法。</p> <p>(2) 数据准备完成后，决策树的核心是如何选择最优的划分属性，即从当前数据的所有特征中，找出对决策影响最大的特征。 我们希望决策树的分类节点所包含的样本尽可能属于同一类别，即节点的纯度越来越高。确定分类节点的方法是：通过指标选择分叉后不纯度降低最多的特征作为分类节点，对每个分类节点使用相同的方式生成新的分类节点，直到指标很小或者没有特征可以选择为止。</p> <p>(3) 通过图形帮助学生理解信息熵的概念。 猜一猜以下四组天气情况哪一组的信息熵最低？</p>	认真听讲、积极思考。	通过构造决策树的过程，帮助学生了解有监督学习的一般流程。																								
		观察图形，理解度量样本集合纯度的指标——信息熵。	利用可视化图形，帮助学生理解抽象的数学概念。																								

续表

教学环节	教学内容	学生活动	设计意图
	<p>解释信息熵的数学定义：</p> $E(D) = -\sum_{i=1}^n p(x_i) \log_2 p(x_i)$ <p>熵值越大，数据的不确定性就越高； 熵值越小，数据的确定性就越高。 如何编写程序求出信息熵呢？</p> <pre>import math p1 = 4 / 5 p2 = 1 / 5 en = -p1 * math.log2(p1) -p2 * math.log2(p2) print(en)</pre> <hr/> <p>0.7219280948873623</p> <p>提出问题：计算以上四种情况的信息熵，使用这段代码是否合适？有没有更简洁的方法？</p> <p>(4) 引导学生参考上述代码，使用函数编写简单的 Python 程序计算以上四组天气情况的信息熵，并验证关于集合纯度的结论。</p> <pre>import math def entropy(*c): if(len(c) <= 0): return -1 result = 0 for x in c: result += (-x) * math.log2(x) return result entropy(4/5, 1/5)</pre> <hr/> <p>0.7219280948873623</p> <p>以上四种情况的信息熵：(0.65, 1.46, 2.58, 0)。</p> <p>(5) 理解信息增益的概念。 使用信息熵衡量划分的优劣，划分后使得下一层节点信息熵降低得越多越好。特征 a 对数据集 D 进行划分获得的信息增益：</p> $Gain(D, a) = E(D) - \sum_{v=1}^V \frac{ D^v }{ D } E(D^v)$ <p style="text-align: center;"> \downarrow \downarrow \downarrow 分割前的第 V 个分支 分割后的 信息熵 的权重，样本 越多越重要 </p> <p>(6) ID3 算法：在决策树的各个节点上应用信息增益准则进行特征选择，增益大的特征优先选择</p>	<p>对照信息熵的数学定义和程序代码，理解计算过程。</p>	<p>通过编程实践，让学生进一步理解熵的概念及统计学含义。</p>
		<p>认真思考。</p> <p>分组实践，编写 Python 程序计算以上四种天气情况的信息熵，验证结论“熵的值越小，样本集合的纯度越高”。</p>	<p>培养学生把可重复问题用自定义函数解决的意识，体会模块化程序设计思想</p>
			<p>倾听教师讲解，理解信息增益公式中各部分的含义</p>

教学环节	教学内容						学生活动	设计意图		
实践体验	样本	首次参观	参观人数	天气	专业人士	参观体验	在教师的带领下，帮助学生进一步理解信息熵和信息增益，尝试理解其所代表的统计学含义。 (计算信息熵、信息增益比较耗时，老师可根据课堂时间调整本部分内容)	通过手动计算，帮助学生进一步理解信息熵、信息增益的含义 (计算信息熵、信息增益比较耗时，老师可根据课堂时间调整本部分内容)		
	1	是	多	雨	是	良好				
	2	是	多	晴	是	良好				
	3	否	少	雨	否	良好				
	4	否	多	阴	是	一般				
	5	是	多	晴	否	一般				
	6	是	少	阴	否	良好				
	7	是	多	雨	否	一般				
	8	否	多	雨	是	一般				
	9	否	少	晴	是	一般				
	10	否	多	阴	是	一般				
	11	否	少	晴	否	良好				
	12	否	多	雨	是	一般				
	13	否	少	晴	是	一般				
	14	是	少	阴	否	良好				
	15	否	多	雨	是	一般				
	16	是	少	雨	否	良好				
	17	否	少	晴	否	良好				
	18	否	少	雨	是	一般				
	19	否	少	阴	否	良好				
	20	否	少	晴	是	一般				
与学生一起，体验用 ID3 算法手动创建一棵展馆体验决策树：										
1. 构造第一级决策树										
用户分专业人士和非专业人士两种情况，专业人士记作 D_1 ，非专业人士记作 D_2 ，对于 D_1 ，参观体验良好占 $\frac{2}{11}$ ，参观体验一般占 $\frac{9}{11}$ ；对于 D_2 ，参观体验良好占 $\frac{7}{9}$ ，参观体验一般占 $\frac{2}{9}$ ，分别计算它们的信息熵：										
$E(D_1) = -\left(\frac{2}{11} \log_2 \frac{2}{11} + \frac{9}{11} \log_2 \frac{9}{11}\right) = 0.68$										
$E(D_2) = -\left(\frac{7}{9} \log_2 \frac{7}{9} + \frac{2}{9} \log_2 \frac{2}{9}\right) = 0.76$										
计算信息增益(专业人士)：										
$Gain(D, \text{专业人士}) = E(D) - \sum_{i=1}^2 \frac{ D_i }{ D } E(D_i)$										
$= 0.99 - \left(\frac{11}{20} * 0.68 + \frac{9}{20} * 0.76\right)$										
$= 0.27$										
即通过专业人士这一特征，使参观体验的信息熵下降 0.27。同理：										

教学环节	教学内容	学生活动	设计意图
	<p>$Gain(D, \text{首次参观}) = 0.11$ $Gain(D, \text{参观人数}) = 0.13$ $Gain(D, \text{天气}) = 0.02$</p> <p>比较首次参观、参观人数、天气和专业人士这四个特征,可以发现专业人士这个特征的信息增益最大,这表明专业人士这个特征在决定参观体验上是最有贡献的。于是选择专业人士作为数据集 D 的划分标准,构造第一级的决策树。</p> <p>2. 构造第二级决策树</p> <p>对于 11 个专业人士,计算参观体验的信息熵,结果约为 0.68,在此基础上,分别计算首次参观、参观人数和天气在这 11 个数据上的信息增益。经比较,参观人数的信息增益最高,约为 0.68。以参观人数为第二层特征,把数据分成参观人数多的一组和参观人数少的一组。参观人数少的一组都体验良好,参观人数多的一组都体验一般。</p> <p>对于 9 个非专业人士进行划分,方法相同,最后手动构建一棵决策树</p>		参考前面第一级决策树的构造过程,自主完成第二级决策树中信息熵、信息增益的计算
编程实践	<p>给出决策树程序半成品,引导学生完善和调试程序,使用 scikit-learn 完整体验构造一棵决策树的过程。</p> <p>1. 加载数据集。 2. 将数据拆分为训练集和测试集。 3. 使用 scikit-learn 四步建模模式。 第 1 步:导入要使用的模型。 第 2 步:创建模型的实例。 第 3 步:使用训练集训练模型。 第 4 步:预测测试数据标签。 4. 评价模型的性能。</p> <pre>#-*- coding: utf-8 -*- #使用ID3决策树算法分析自动驾驶展馆体验 import pandas as pd #读入数据 inputfile = u'展馆体验.xlsx' inputdata = pd.read_excel(inputfile, index_col = u'样本') #导入数据 data = inputdata[[u'首次参观',u'参观人数',u'天气',u'专业人士']] target = inputdata[[u'参观体验']] #print(data) #print(target) #training data train_target = target[:20] #选择前20个数据作为训练集 train_data = data[:20] print(train_data) #testing data test_target = target[20:] #选择第21个数据到结束作为测试集 test_data = data[20:] print(test_data) #训练模型 from sklearn import tree model = tree.DecisionTreeClassifier(max_depth = 3, criterion = 'entropy') model.fit(train_data, train_target) #用测试集测试模型分类是否正确 print(test_target) print("\n使用决策树的分类结果:") print(model.predict(test_data)) from sklearn.metrics import accuracy_score y_test_predict = model.predict(test_data) accuracy_score(test_target,y_test_predict)</pre>	体验构造决策树的过程,理解程序关键代码,体会到监督学习的完整过程	通过完整体验 scikit-learn 构造决策树的过程,加深学生对有监督学习的理解

续表

教学环节	教学内容	学生活动	设计意图
实验探究	<p>使用 <code>export_graphviz</code> 函数将生成的决策树可视化。</p> <p>提出问题： 观察信息熵的值，信息熵作为数据不确定性的度量，其值越大，则不确定性越（ ），当 <code>entropy</code> 等于（ ）时，分类结果是一个确定的值</p>	绘制决策树模型，探究每个节点给出的属性特征值、信息熵、样本量等信息所表达的含义。	引导学生观察实验现象，探究实验结论，仿照科学的过程，培养学生的实验能力和创新精神。
归纳总结 分享交流	<p>和学生一起回顾和总结本节课知识点：</p> <ol style="list-style-type: none"> 1. 决策树的工作原理。 2. 构造决策树的过程。 3. 编程实践 ID3 算法，对展馆体验数据建立决策树进行分类并可视化。 4. 决策树算法的应用：语音识别、医疗诊断、客户关系管理、模式识别、专家系统等 	认真倾听，梳理知识点。 积极思考决策树的应用，分享交流	帮助学生梳理知识，总结算法流程和特点
分层拓展 延展课堂	<p>请学有余力的同学参考阅读资料，完成如下任务：</p> <p>根据经典的鸢尾花数据集，绘制不同深度的决策树模型，找到最佳的决策树最大深度 <code>max_depth</code>，输出具有不同 <code>max_depth</code> 时决策树预测的准确性，初步理解过拟合的概念</p>	阅读学案，完成程序：探究每个节点给出的属性特征值、信息熵、样本量等信息所表达的含义。 尝试不同的决策树最大深度 <code>max_depth</code> ，掌握超参数调整的方法	针对学生兴趣、基础等差异，为学有余力的学生留出知识拓展的空间。同时培养学生自主学习的能力和数字化学习与创新能力
教学评价方式			
<ol style="list-style-type: none"> 1. 课堂参与度(50%)：在课堂学习中，学生是否积极参与课堂活动，主动构建新知识；是否能根据任务需求，积极主动思考，实现问题的有效解决。 2. 实践过程及结果(50%)：在分组实践探究活动中，是否能与小组成员有效合作，围绕探究内容积极提出自己的想法，动手完成实践。 3. 拓展任务完成情况(10%奖励)：是否能根据阅读材料，完成附加任务，具有自主学习能力和数字化学习与创新能力 			

■ 参考案例 2

使用专家系统进行动物识别

北京师范大学附属实验中学 马 静

(2课时)

1. 教学目标

- 通过案例剖析了解专家系统的概念,能举例说明专家系统的应用场景。(计算思维)
- 通过体验动物识别了解专家系统的结构,理解专家系统的原理和关键环节。(计算思维)
- 通过动物识别专家系统项目实践,体验构建简单专家系统解决问题的过程。(数字化学习与创新)
- 通过添加或删除规则,进一步理解专家系统的决策原理。(信息意识、信息社会责任)

2. 教学对象分析

学生在之前的学习中已经对人工智能的概念、发展历程有了初步了解,具备了一定的Python语言编程功底,初步了解了决策树实现的一般过程,但对经典人工智能实现方法如分类、聚类、回归等还缺乏全面的了解和把握。

专家系统是一种比较经典的人工智能技术,是一个含有大量的某个领域专家水平的知识与经验的智能计算机程序系统,能够利用人类专家的知识和解决问题的方法来处理该领域问题。专家系统能对符合条件的输入信息做出一定的判别和推理,因此高中生对这部分内容的学习会比较好奇,希望能够探索其原理。但专家系统结构较为复杂,理解起来会有一定难度。所以在程序实现过程中引入Tkinter模块,实现专家系统接口,直观呈现推导过程和图形等帮助学生理解概念,降低学习的难度。另外,高中生已经具备一定的自主探究能力,因此以小组为单位,在整个课堂活动中穿插自主探究体验活动,培养学生的动手实践和合作交流能力。

3. 教学重难点

教学重点:了解专家系统的概念和结构;利用案例程序体验专家系统的运行过程。

教学难点:利用案例程序体验专家系统的运行过程。

4. 教学方法

- 通过案例分析讲解帮助学生理解专家系统的概念和结构;
- 通过动物识别系统,直观呈现推导过程和图形等帮助学生理解概念;
- 采用教师讲授与项目实践相结合的形式,从理解专家的基本概念、分析相关技术原理、典型案例实践和利用人工智能技术解决真实情境中的问题等方面开展教学,使用“创新园区展馆体验”项目贯穿整个教学过程,培养学生解决真实情境中的问题的思维方式和能力。

5. 教学工具

网络机房、投影、Python 运行环境、Python 集成开发环境 Anaconda、Tkinter 库等。

6. 教学过程设计

教学过程设计如表 2-6 所示。

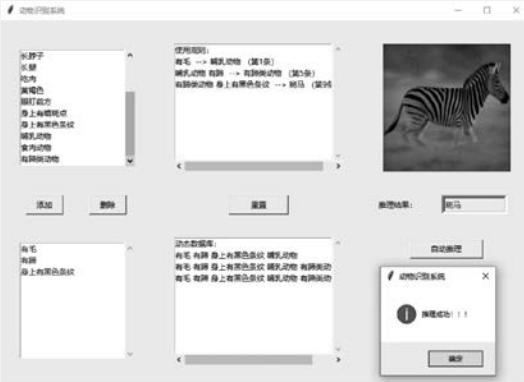
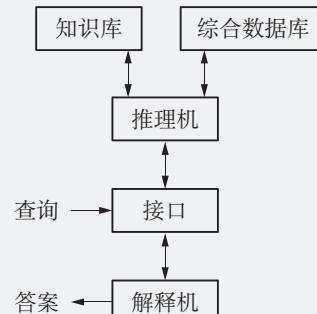
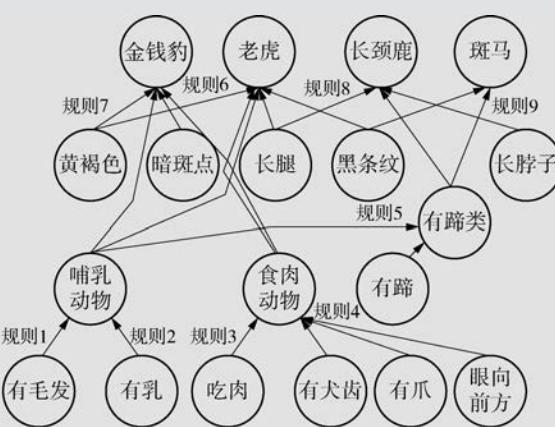
表 2-6 教学过程设计

教学环节	教学内容	学生活动	设计意图
新知引入	<p>提出问题。 在生活中,常常会对一些不太了解的情况不知如何应对。比如身体不适去医院看病时不知道该挂哪一个科室等。这时一些拥有类似人类专家思维和推理能力的计算机系统就可以随时随地来帮助人们了。这些“专家”,一般都拥有某一特定领域的大量知识以及丰富的经验。面对问题时,可以凭借缜密的思维,给我们提供参考建议。这就是专家系统,是具有智能特点的计算机程序。能够在特定领域内模仿人类专家思维来求解复杂问题。专家系统必须包含某个领域专家的大量知识和经验,并且拥有推理能力,能够解决问题</p>	<p>思考,听讲,回答 教师提问</p>	<p>提出问题,引出专家系统的概念。为学习新知做铺垫</p>
	<p>专家系统的构成。 结构讲解:专家系统之所以能够解决问题,离不开知识和推理,除此之外还需要其他的一些部分,如图所示。</p> <pre>graph TD; KB[知识库] <--> GD[综合数据库]; KB <--> RE[推理机]; RE <--> I[接口]; I -- "查询" --> RE; I --> EX[解释机]; EX <--> A[答案]</pre>	<p>思考,听讲,理解 专家系统的构成</p>	<p>新知讲解,学生初步认识专家系统</p>
新知学习	<ol style="list-style-type: none">1. 知识库:由领域专家的经验、知识、方法汇集而成的库。每个专家系统通常都按照其知识库中知识的表示方法(即知识表示)来称呼。比如,基于规则的专家系统就是指该专家系统中的知识是以一条条规则的形式来表示的。2. 综合数据库:综合数据库指专家系统中所使用的除知识库以外的其他类型的数据,如通用信息、背景信息和一些常识等。3. 推理机:推理机也称为推理引擎或推理机制,指专家系统处理知识和使用知识获得结论的机制。4. 接口:和普通的信息系统一样,接口指用户与专家系统进行交互的方式。5. 解释机:专家系统不但要响应用户的请求,得出结论,还要能对导出结论的依据给出解释和说明,这部分即解释机		

续表

教学环节	教学内容	学生活动	设计意图
实践体验	<p>体验简单动物识别专家系统。</p> <p>体验能够识别虎、金钱豹、斑马、长颈鹿四种动物的简单专家系统。</p> <p>专家系统的规则：</p> <ol style="list-style-type: none"> 1. 有毛发的动物是哺乳类； 2. 有奶的动物是哺乳类； 3. 吃肉的哺乳类称为食肉动物； 4. 犬牙利爪，眼睛向前的是食肉类； 5. 有蹄的哺乳类是有蹄类； 6. 黄褐色有黑色条纹的食肉类是老虎； 7. 黄褐色有暗斑点的食肉类是金钱豹； 8. 长腿长脖黄褐色有暗斑点的有蹄类是长颈鹿； 9. 有黑色条纹的有蹄类是斑马。 <p>构建规则：</p> <p>基于规则的专家系统是最简单、最典型的一种专家系统，这种系统使用一组规则来表示知识。一条规则表示当指定的条件 A 满足时，能推导出某个结论 B，表现为如下形式：if A then B，或 A→B。其中，A 称为规则的前提，B 称为结论。</p> <p>举例来说，如果 A 是下雨，B 是带伞，那么 A→B 就是一条简单规则：如果下雨，就带伞。</p> <p>动手实践 1：</p> <p>请尝试写出规则：</p> <p>规则 1：if 有毛发 then 哺乳动物；</p> <p>规则 2：if 有乳 then 哺乳动物；</p> <p>规则 3：if 吃肉 then 食肉动物；</p> <p>规则 4：if 有犬齿 and 有爪 and 眼向前方 then 食肉动物；</p> <p>规则 5：if 哺乳动物 and 有蹄 then 有蹄类；</p> <p>规则 6：if 哺乳动物 and 食肉动物 and 黄褐色 and 黑条纹 then 老虎；</p> <p>规则 7：if 哺乳动物 and 食肉动物 and 黄褐色 and 暗斑点 then 金钱豹；</p> <p>规则 8：if 有蹄类 and 长脖子 and 长腿 and 黄褐色 and 暗斑点 then 长颈鹿；</p> <p>规则 9：if 有蹄类 and 黑条纹 then 斑马</p>	<p>认真听讲、积极思考。</p> <p>观察关系图，理解动物判别的条件。</p>	通过体验简单动物识别案例了解专家系统的结构，体验构造的一般流程。
运行程序	<p>打开案例程序，体验使用专家系统进行动物识别。</p> <p>给出决策树程序半成品，引导学生完善和调试程序，使用 Tkinter 构造用户界面，呈现推理过程。</p> <ol style="list-style-type: none"> 1. 导入模块。 2. 加载综合数据库(animal、condition 两个文件和 images 文件夹)。 3. 读取所有规则 rule.txt。 4. 利用 Tkinter 构造界面，准备运行： <p>第 1 步：添加适当的事实描述；</p> <p>第 2 步：利用规则做出判断；</p> <p>第 3 步：弹出对话框，呈现判断过程和图片；</p>	<p>在教师的带领下，动手体验专家系统进行动物识别的过程。</p>	通过动手体验，进一步理解专家系统的结构，加深对专家系统的理解。

续表

教学环节	教学内容	学生活动	设计意图
	 <p>动手实践 2： 参考动物判断图，输入多种条件组合，尝试获得不同的结果。 教师特别引导学生关注使用的规则和呈现的动态数据库，以此来了解专家系统推理的过程。在反复尝试实践中，带领学生将专家系统的结构和案例各个部分对应。</p>		务必尽可能多地体验不同条件组合形成判断的过程，尝试对应程序关键代码，体会专家系统推理的过程
推理探究	 <p>在体验专家系统后，我们来探究系统是如何运用知识来解决实际问题的，即推理机是如何运用知识库中的知识进行推断的机制和规则。 展示专家系统关系图：</p>  <p>在关系图中，有毛、有奶、吃肉、有犬齿、有爪、眼向前方、有蹄、黄褐色、暗斑点、长腿、黑条纹、长脖子等是事实；而哺乳动物、食肉动物和有蹄类动物是中间结论；</p>	绘制推理机模型，探究正向推理的过程。 思考探究实验现象形成的原因，理解正向推理过程。	

续表

教学环节	教学内容	学生活动	设计意图
	<p>位于图最顶层金钱豹、老虎、长颈鹿和斑马则是最终结论。</p> <p>动手实践 3：</p> <p>体验正向推理。正向推理就是从事实出发，逐步运用规则推导出结论的过程。</p> <p>请拿笔将推导的过程标记出来。首先根据基本事实“有毛发”，能推导出中间结论，哺乳动物；再进一步，根据“有蹄”能推导出中间结论，有蹄类动物；最后一步由“黑条纹”，推导出结论“斑马”。</p> <p>而反向推理，则是采取从结论逐步回溯到事实的方式。先自结论开始，标示出能够导出结论的相关知识，然后再由标示的部分根据规则推导出结论，从而完成验证。</p> <p>请学生再尝试绘制另一次推理的过程</p>	<p>认真思考，回答教师提出的问题。分享自主绘制正向推理的过程、结论和体会</p>	通过倾听学生分享，了解学生对推理过程的理解程度
分享交流	<p>和学生一起回顾和总结本节课知识点：</p> <ol style="list-style-type: none"> 1. 专家系统的基本概念； 2. 专家系统的结构； 3. 动手撰写规则，根据关系图写出 if… then… 形式的语句； 4. 体验专家系统正向推理的过程 	<p>认真倾听，梳理知识点</p>	总结概念、结构、推理流程和特点
拓展延伸	<p>请学有余力的同学尝试添加五条规则，完成七种动物的判断：</p> <ol style="list-style-type: none"> 1. 有羽毛的动物是鸟类； 2. 若动物会飞且生蛋，则它是鸟类； 3. 不会飞腿长脖黑白色的鸟是鸵鸟； 4. 不会飞善游泳黑白色的鸟是企鹅； 5. 善飞的鸟是信天翁。 <p>提示，将 rules.txt 中的规则修改为：</p> <p>有毛→哺乳动物； 有奶→哺乳动物； 有羽毛→鸟类； 会飞，会下蛋→鸟类； 吃肉→食肉动物； 有犬齿，有爪，眼盯前方→食肉动物； 哺乳动物，有蹄→有蹄类动物； 哺乳动物，食肉动物，黄褐色，身上有暗斑点→金钱豹； 哺乳动物，食肉动物，黄褐色，身上有黑色条纹→虎； 有蹄类动物，长脖子，长腿，黄褐色，身上有暗斑点→长颈鹿； 有蹄类动物，身上有黑色条纹→斑马； 鸟类，长脖子，长腿，不会飞，有黑白二色→鸵鸟； 鸟类，会游泳，不会飞，有黑白二色→企鹅； 鸟类，善飞→信天翁</p>	<p>分层拓展：根据新规则，完成规则撰写和加入。试运行程序，研究新规则加入后对已有系统的影响，绘制新的推理</p>	针对学生兴趣、基础等差异，为学有余力的学生留出知识拓展的空间。同时培养学生自主学习的能力和数字化学习与创新能力

教学环节	教学内容	学生活动	设计意图
教学评价方式			
1. 课堂参与度(50%):在课堂学习中,学生是否积极参与课堂活动,主动构建新知识;是否能根据任务需要,积极主动思考,实现问题的有效解决。	2. 实践过程及结果(50%):在分组实践探究活动中,是否能与小组成员有效合作,围绕探究内容积极提出自己的想法,动手完成实践。	3. 拓展任务完成情况(10%奖励):是否能根据阅读材料,完成附加任务,具有自主学习能力和数字化学习与创新能力	

第三节 机器学习

一、教学目标与重点

教学目标:

- 通过具体案例了解机器学习的概念,理解机器学习与人类学习方式的异同点。(计算思维)
- 通过探究活动了解监督学习和无监督学习的概念,理解二者差异。(计算思维)
- 通过编程应用理解经典线性回归算法,感受机器学习原理。(计算思维)
- 通过探究活动了解聚类和分类的基本概念,解释算法关键环节。(计算思维)
- 通过编程应用 k -均值算法和 k -近邻算法分别实现聚类和分类。(数字化学习与创新)
- 通过为创新园区自由行项目设计集合地点应用 k -均值聚类算法解决实际问题。(信息意识、信息社会责任)

教学重点:

- 机器学习的概念;
- 经典线性回归算法;
- k -均值聚类算法原理;
- k -近邻分类算法原理与实践。

二、内容分析与建议

1. 内容分析

本小节是第二章“人工智能实现”的最后一小节，在本章中主要关注机器学习这一人工智能核心技术，同时关注智能技术的实现，为第三章漫游深度学习的世界打下基础。本小节首先详细介绍了机器学习的基本概念和经典的线性回归算法。依据不同数据作为输入的监督学习和无监督学习两大类别，以经典的 k -均值聚类和 k -近邻分类思想解决具体问题入手，理解人工智能的核心算法，尝试用智能技术解决实际问题。最后进行知识延伸，介绍以“随机森林”为代表的机器学习的经典算法。本节主要知识框图如图 2-15 所示。

2. 教学建议

(1) 教学活动建议

作为人工智能中的核心概念，机器学习在人工智能领域的应用范围非常广泛。在本小节的开始，可以请学生反思自己学习的过程，对比引出机器学习由数据驱动的特点。然后根据主题，应用经典线性回归算法和已有数据，建立自动驾驶电动车行驶里程与耗电量的线性模型，从而预测未知里程的耗电量。学生通过亲历现实问题的解决，切实感受数据对于人工智能的重要性。进而介绍标签的概念，并通过有无标签两种数据对应监督学习和无监督学习两大类别。教学中，可以尽可能多地列举生活中的问题，例如降雨量预测、图片归类、电影分类等，引发学生思考，学习把实际问题归类并找出适合的解决方法。

在本小节中，教科书在本章节项目主题“创新园区自由行”之下，结合机器学习打造了“集合地点的选择”和“参观者行程安排”两个分别与无监督学习和监督学习技术相关的项目，展开原理学习和编程体验。

在教学中也不必局限于教科书中的案例，可以因地制宜，根据学生和学校的具体情况，为学生们提供更多的机器学习与实际生活密切联系的案例。例如，线性回归算法还可以结合高一物理中研究小车速度随时间变化规律的实验展开， k -均值聚类算法还可以用于电影分类， k -近邻分类算法还可以用于鸢尾花分类等经典案例。

在算法的解析过程中，教师要根据学生的实际情况，对技术原理进行适度的展开，重点放在方法层面的解析和引导上，以便学生能够更加深入地理解机器学习的基本原理。因此，在具体教学中，不仅要鼓励学生模仿教科书案例解决项目问题，感受应用的魅力；更需要分析技术实现背后的计算思维，感悟机器学习的技术魅力，让学生在知其所以然之后，更能举一反三，将经典算法用于解决更多的实际问题之中，提高科学并合理应用人工智能

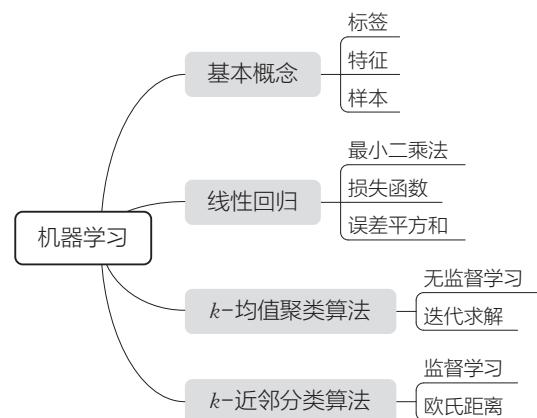


图 2-15 主要知识框图

技术的意识和能力。

(2) 项目实施建议

本节的项目任务主要是“集合地点的选择”和“参观者行程安排”这两个具体应用 k -均值聚类算法和 k -近邻分类算法的项目,通过编程实现分别体验无监督学习和监督学习两大机器学习技术。

本节项目实施的难点在于通过编程实践 k -均值聚类算法和 k -近邻分类算法的项目,理解无监督学习和监督学习两大机器学习技术。对于这两种抽象算法的理解,可以考虑组织学生进行角色扮演,模拟算法运行的方式来帮助学生进行直观理解。

例如, k -均值聚类算法可以将 10 位同学分成黑、白两组数据,另请两位同学分别扮演黑、白组数据的中心点,请同学们来模拟中心点定位的过程。

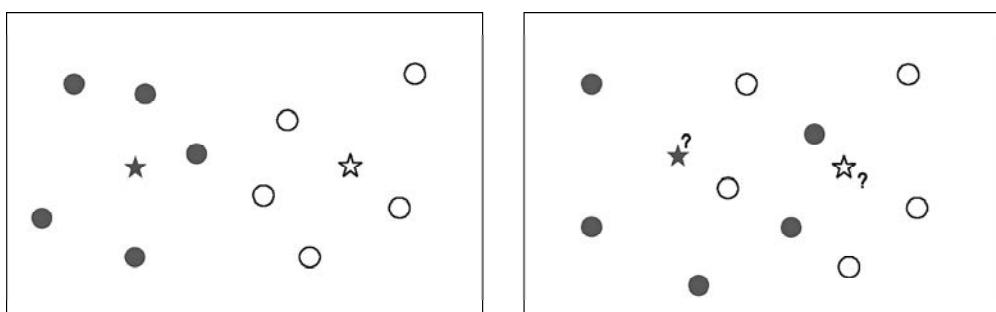


图 2-16 k -均值聚类算法学生活动示意图

首先,可以请两位扮演黑、白中心的同学自主地快速找到黑、白两组同学的中点,如图 2-16 中左图所示。教师通过采访这两位同学迅速定位的方法,即可引出学生通过距离确定中心的基本思想。之后教师可以将黑、白两组数据相互穿插,提高寻找中心的难度,如图 2-16 中右图所示。由此引出聚类算法的基本思想和过程,再请同学模拟多次迭代后的结果。在真人分组模拟演示之后,教师再带领学生解析代码,模仿实现项目。

(3) 评价建议

评价方式:课程评价建议将教师评价和学生自评相结合,过程性评价与总结性评价相结合,以过程性评价为主,主要由教师进行课堂观察评价并对项目活动进行检查。

评价要点:在课堂活动中,教师需要关注学生是否能积极参与到课堂活动中去,是否能积极参与思考与讨论、踊跃交流和表达观点;是否敢于动手实践调试和修改程序;是否能举一反三将课堂情境下的问题解决方法应用到社会生活中的其他情境之中。在项目活动中,是否能积极参与小组讨论,主动进行多种复杂数据情况的预设和分析,完成指定任务,能否主动提出问题,是否积极参与项目汇报和评价。

三、习题分析与提示

■ 习题 1 描述

见教科书第 55 页的项目实践“使用 Python 语言实现典型的无监督学习算法”。

习题 1 解题思路

第 1 个任务

首先需要准备适当的数据集。因此导入 numpy 工具, 使用其中的 np. random. normal() 创建一个正态分布的列表, 如 np. random. normal(10, 1, 300) 可以创建一个均值为 10, 标准差为 1 的 300 个数据的正态分布的列表。类似地, 由此方法创建四个数据列表, 并两两组合, 利用 matplotlib. pyplot 绘制散点图, 如图 2-17 所示。

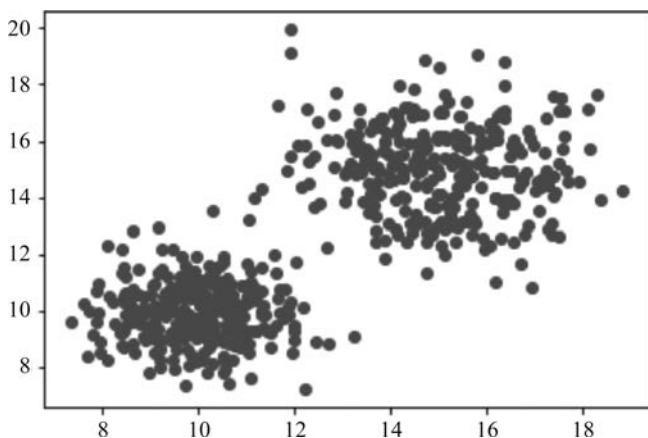


图 2-17 数据散点图

其次, 依据上一步得到的 x 和 y 两组数据, 构造两个数组 x1 和 x2, 将两组数据打包成数组形状为(x, y) 成对的元组。调用 sklearn 中的 KMeans 方法来进行聚类, 这里将超参数 k 设置为 2, 结果或 0 或 1 的数值输出到 y_pred 中。

最后调用 matplotlib. pyplot 根据 x1, x2 的数据绘制散点图, 颜色由 y_pred 中的数值确定, 即可得到结果(如图 2-18 所示)。

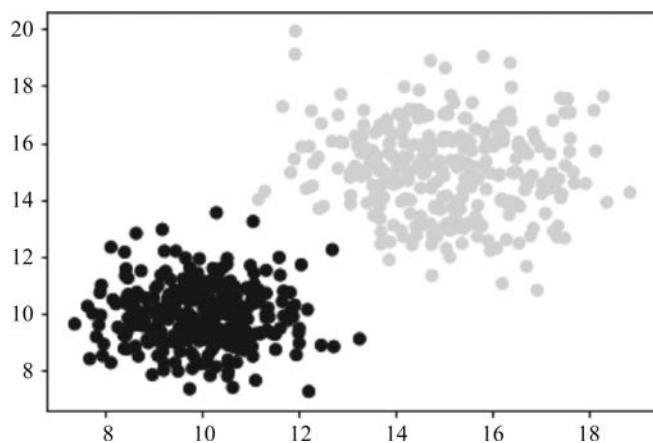


图 2-18 聚类结果示意图

第 2 个任务

首先需要准备适当的数据集。因此导入 sklearn 中的 make_moons 形成数据集, 设置生成的数据个数为 1000, 数据的噪点(离散程度)为 0.1。再利用 matplotlib. pyplot 绘

制散点图,如图 2-19 所示。

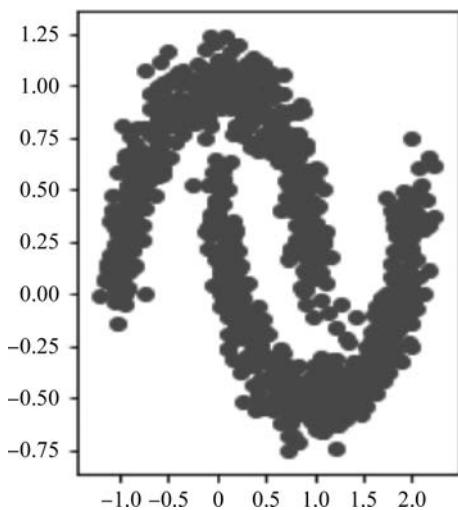


图 2-19 数据散点图

其次,依据上一步得到的数据,调用 sklearn 中的 KMeans 方法来进行聚类,这里将超参数 k 设置为 2,结果或 0 或 1 的数值输出到 y_pred 中。

最后调用 matplotlib.pyplot 根据 x1,x2 的数据绘制散点图,颜色由 y_pred 中的数值确定,即可得到结果如图 2-20 所示。

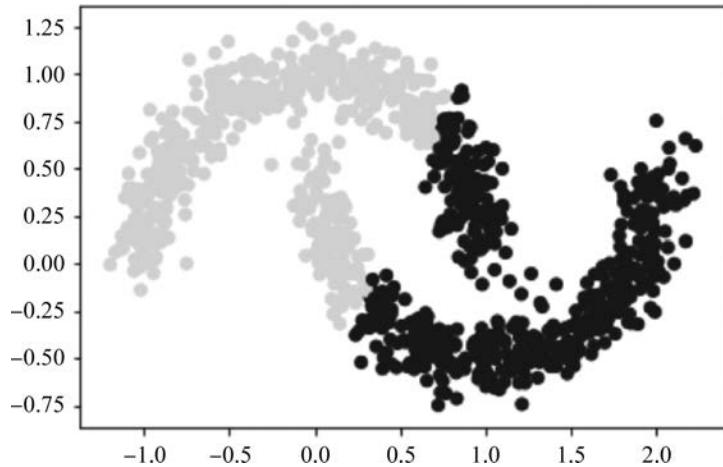


图 2-20 聚类结果示意图

■ 习题 1 源代码

第 1 个任务

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import KMeans

# 制作数据集
data_x1= np.random.normal(10,1,300)
data_x2= np.random.normal(15,1.5,300)
```

```

data_y1= np.random.normal(10,1,300)
data_y2= np.random.normal(15,1.5,300)

x= np.concatenate((data_x1,data_x2))
y= np.concatenate((data_y1,data_y2))

# 绘制原始数据散点图
plt.scatter(x,y)
plt.show()

# 转化成数组
x1= np.array(x)
x2= np.array(y)
xx= np.array(list(zip(x1,x2))).reshape(len(x1),2) # zip 打包成元组,变化数组形状为(x1,y1)

# 用 K-均值聚类方法来做聚类,选择 k= 2
clusters= 2
kmeans= KMeans(clusters,init= "random") # 构造聚类器
y_pred= kmeans.fit_predict(xx) # 聚类

# 输出质心以及每个原始数据对应类别
print("the centroid of cluster are:\n{0}".format(kmeans.cluster_centers_)) # 输出质心
print("the member of cluster are:{0}\n".format(kmeans.labels_)) # 输出每个原始数据对应类别

# 绘制聚类后的散点图
plt.scatter(x1,x2,c= y_pred)
plt.show()

```

第 2 个任务

```

import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import KMeans
from sklearn.datasets import make_moons

# 调用 sklearn 中的 make_moons 形成数据集
plt.subplot(122)
x1,y1= make_moons(n_samples= 1000,noisy= 0.1)
# 绘制数据图像
plt.title('make_moons function example')
plt.scatter(x1[:,0],x1[:,1],marker= 'o')
plt.show()

# 用 K-均值聚类方法来做聚类,首先选择 k= 2
clusters= 2
kmeans= KMeans(clusters,init= "random") # 构造聚类器
y_pred= kmeans.fit_predict(x1) # 聚类

# 输出质心以及每个原始数据对应类别
print("the centroid of cluster are:\n{0}".format(kmeans.cluster_centers_)) # 输出质心
print("the member of cluster are:{0}\n".format(kmeans.labels_)) # 输出每个原始数据对应类别

# 绘制聚类后的散点图
plt.scatter(x1[:,0],x1[:,1],c= y_pred)
plt.show()

```

习题 2 描述

见教科书第 57 页的项目实践“使用 Python 语言实现 k -近邻分类算法”。

习题 2 解题思路

第 1 个任务

首先需要准备适当的数据集。例如可以手工设定 16 个数据点：

```
X_train= np.array([[2,1],[3,2],[4,2],[1,3],[1.5,4],[1.7,3],[2.6,5],  
[3.4,3],[3,6],[1,7],[4,5],[1.2,6],[1.8,7],[2.2,8],[3.7,7],[4.8,5]])
```

再设置 y_{train} 为样本点标记, 将 16 个数据分成两类:

```
y_train= np.array([0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1])
```

利用 matplotlib.pyplot 绘制散点图, 如图 2-21 所示。

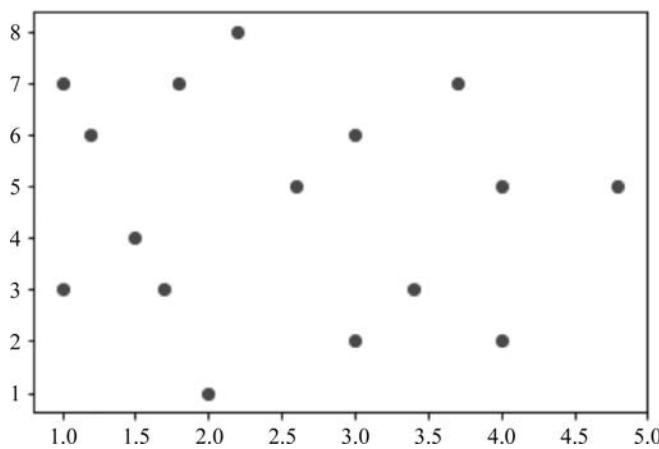


图 2-21 数据散点图

其次, 准备测试样本数据 $X_{test} = np.array([3.2, 5.4])$, 设置参与计算的邻居数为 3, $k = 3$ 。同时采用欧式距离计算与测试点之间的距离, 并根据距离大小排序找到与测试样本数据最近的 3 个, 再根据距离最近的三个样本点所属的类别计算该测试点属于的数据类别。此测试样本数据计算得到的是 1 类数据。

最后调用 matplotlib.pyplot 根据 X_{train} 的数据类别以颜色区分, 绘制出散点图。再标记出测试数据, 同时连接与其最近的三个数据, 也用颜色标记其所属类别, 即可得到图 2-22。

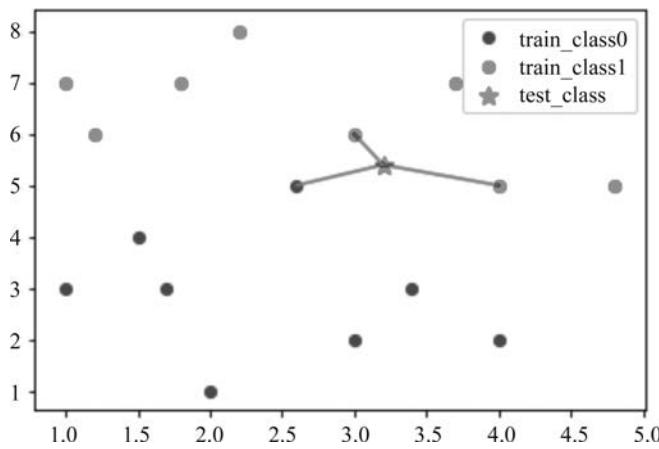


图 2-22 两类分类结果示意图

第 2 个任务

首先需要准备适当的数据集。例如可以手工设定 20 个数据点：

```
X_train5= np.array([[2,1],[3,2],[4,2],[1,3],[1.5,4],[1.7,3],[2.6,5],  
[3.4,3],[3,6],[1,7],[4,5],[1.2,6],[1.8,7],[2.2,8],[3.7,7],[4.8,5],[3.6,  
3],[2.4,2],[1.2,3],[4.9,1.5]])
```

再设置 y_{train} 为样本点标记,将 20 个数据分成五类:

```
y_train= np.array([1,2,3,4,5,1,2,3,4,5,1,2,3,4,5,1,2,3,4,5])
```

利用 matplotlib.pyplot 绘制散点图,如图 2-23 所示。

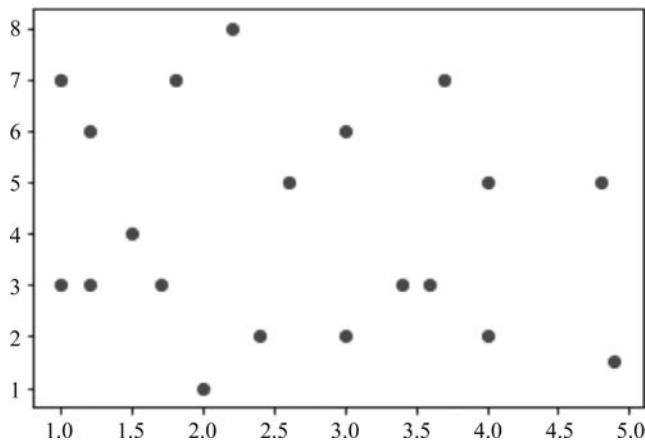


图 2-23 数据散点图

其次,准备测试样本数据 $X_{test} = np.array([3.2, 5.4])$,设置参与计算的邻居数为 7, $k=7$ 。同时采用欧氏距离计算与测试点之间的距离,并根据距离大小排序找到与测试样本数据最近的 7 个样本,再根据距离最近的三个样本点所属的类别计算该测试点属于的数据类别。此测试样本数据计算得到的是 2 类数据。

最后调用 matplotlib.pyplot 根据 X_{train} 的数据类别以颜色区分,绘制出散点图。再标记出测试数据,同时连接与其最近的 7 个数据点,并也用颜色标记其所属类别,即可得到如图 2-24 所示的结果。

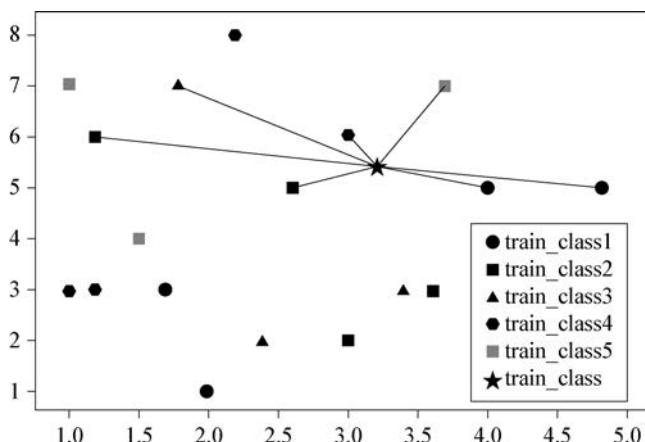


图 2-24 五类分类结果示意图

习题 2 源代码

第 1 个任务

```
import numpy as np
import matplotlib.pyplot as plt
# X_train 为样本点
X_train= np.array([[2,1],[3,2],[4,2],[1,3],[1.5,4],[1.7,3],[2.6,5],[3.4,3],
[3,6],[1,7],[4,5],[1.2,6],[1.8,7],[2.2,8],[3.7,7],[4.8,5]])
# y_train 为样本点标记
y_train= np.array([0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1])

plt.scatter(X_train[:,0],X_train[:,1],c= 'g',marker= 'o',label= 'train_class')
plt.show()
# X_test 为测试样本
X_test= np.array([3.2,5.4])
# k 为邻居数
k= 3

# 这里的距离公式采用欧式距离
square_= (X_train-X_test)* * 2
square_sum= square_.sum(axis= 1)* * 0.5
square_sum_sort= square_sum.argsort()
small_k= square_sum_sort[:k]
y_test_sum= np.bincount(np.array([y_train[i]for i in small_k])).argsort()
print('predict:class{}'.format(y_test_sum[-1]))

# 将数据可视化更生动形象,分两类
# 将 class0 一类的样本点放到 X_train_0 中
X_train_0= np.array([X_train[i,:]for i in range(len(y_train)) if y_train[i]== 0])
# 将 class1 一类的样本点放到 X_train_1 中
X_train_1= np.array([X_train[i,:]for i in range(len(y_train)) if y_train[i]== 1])
plt.scatter(X_train_0[:,0],X_train_0[:,1],c= 'g',marker= 'o',label= 'train_class0')
plt.scatter(X_train_1[:,0],X_train_1[:,1],c= 'm',marker= 'o',label= 'train_class1')
if y_test_sum[-1]== 0:
    test_class= 'g'
elif y_test_sum[-1]== 1:
    test_class= 'm'
plt.scatter(X_test[0],X_test[1],c= test_class,marker= '*',s= 100,label= 'test_class')
# 连接测试样本与 k 个近邻
for i in small_k:
    plt.plot([X_test[0],X_train[i,0]], [X_test[1],X_train[i,1]],c= 'c')
plt.legend(loc= 'best')
plt.show()
```

第 2 个任务

```
# 前面和上题一样
# 将数据可视化更生动形象,数据分五类
# 将 class1 一类的样本点放到 X_train_1 中
X_train_1= np.array([X_train5[i,:]for i in range(len(y_train)) if y_train[i]== 1])
# 将 class2 一类的样本点放到 X_train_2 中
X_train_2= np.array([X_train5[i,:]for i in range(len(y_train)) if y_train[i]== 2])
# 将 class3 一类的样本点放到 X_train_3 中
X_train_3= np.array([X_train5[i,:]for i in range(len(y_train)) if y_train[i]== 3])
# 将 class4 一类的样本点放到 X_train_4 中
X_train_4= np.array([X_train5[i,:]for i in range(len(y_train)) if y_train[i]== 4])
# 将 class5 一类的样本点放到 X_train_5 中
X_train_5= np.array([X_train5[i,:]for i in range(len(y_train)) if y_train[i]== 5])
```

```

# 绘制所有样本点并采用不同的颜色分别标记
plt.scatter(X_train_1[:,0],X_train_1[:,1],c= 'g',marker= 'o',label= 'train_class1')
plt.scatter(X_train_2[:,0],X_train_2[:,1],c= 'm',marker= 'o',label= 'train_class2')
plt.scatter(X_train_3[:,0],X_train_3[:,1],c= 'y',marker= 'o',label= 'train_class3')
plt.scatter(X_train_4[:,0],X_train_4[:,1],c= 'c',marker= 'o',label= 'train_class4')
plt.scatter(X_train_5[:,0],X_train_5[:,1],c= 'k',marker= 'o',label= 'train_class5')

if y_test_sum[- 1]== 1:
    test_class= 'g'
elif y_test_sum[- 1]== 2:
    test_class= 'm'
elif y_test_sum[- 1]== 3:
    test_class= 'y'
elif y_test_sum[- 1]== 4:
    test_class= 'c'
elif y_test_sum[- 1]== 5:
    test_class= 'k'

plt.scatter(X_test[0],X_test[1],c= test_class,marker= '*' ,s= 100,label= 'test_class')
# 连接测试样本与 k 个近邻
for i in range(k):
    plt.plot([X_test[0],X_train[i,:][0]],[X_test[1],X_train[i,:][1]],c= 'c')
plt.legend(loc= 'best')
plt.show()

```

四、核心概念精解与技术工具介绍

1. 核心概念

(1) 机器学习

机器学习是一门多学科交叉专业,涵盖概率论知识、统计学知识、近似理论知识和复杂算法知识,使用计算机作为工具并致力于真实实时的模拟人类学习方式,并将现有内容进行知识结构划分来有效提高学习效率。机器学习实际上已经存在了几十年或者也可以认为存在了几个世纪。追溯到 17 世纪,关于最小二乘法的推导和马尔可夫链,这些都构成了机器学习广泛使用的工具和基础。从 1950 年(艾伦·图灵提议建立一个学习机器)到 21 世纪初期(有深度学习的实际应用以及最近的进展,比如 2012 年的 AlexNet),机器学习有了很大的进展。

目前机器学习有下面几种定义:

机器学习是人工智能的分支领域,该领域主要研究如何在经验学习中改善具体算法的性能。

机器学习是对能通过经验自动改进的计算机算法的研究。

机器学习致力于研究如何通过计算手段,利用经验来改善系统自身的性能。

(2) 样本

机器学习的进行离不开大量的数据。这些记录事件或对象的数据集中,每条记录都是关于这个事件或对象的一个描述,称为一个样本。

(3) 特征

特征是反映事物或对象在某方面的表现或性质的事项。如苹果的颜色、某公寓的面

积、某建筑的建造年代等。简单的机器学习可能会使用单个或数个特征,而复杂的机器学习可能会用到上百万个特征。

(4) 标签

标签是关于样本的结果信息。如果样本中包含关于结果的信息,就是有标签样本;反之,如果样本中只有特征,而没有标签,就是无标签样本。

(5) 监督学习

通过已有标签样本数据来训练,从而得到一个最优模型,再利用这个模型来进行新的推断。这种监督学习需要使用人工标注好标签的数据进行训练,从而达到能够对未标记的数据进行推断的目的,因此标注数据对于监督学习来说极其重要,它的好坏将直接影响训练出的模型的质量。但是标注海量的学习数据是一项较为耗时费力的工作,细心认真,才能有高精度的标注数据,才能有好的监督学习结果。

(6) 无监督学习

不需要标签数据,计算机自动从数据中寻找规律并对数据进行分析的技术就是无监督学习。无监督学习可以交由计算机处理很多未知数据,减轻人类的工作量,侦测可能的数据分布模式,增加模型的泛化性能。无监督学习里典型例子是聚类,而 k -均值算法是一种典型的聚类算法。

(7) 最小二乘法

最小二乘法是一种数学优化技术。通过最小化误差的平方和寻找数据的最佳函数匹配。利用最小二乘法可以简便地求得未知的数据,并使得这些求得的数据与实际数据之间误差的平方和最小。

(8) 损失函数

损失函数用来表现预测与实际数据的差距程度,是机器学习中的重要概念,用于衡量模型预测的好坏。例如,在线性回归中,我们用误差平方和来计算真实值与预测值之间的差值,误差平方和就是一种损失函数。常用的损失函数还有交叉熵等。

(9) 误差平方和

根据 n 个训练数据拟合得到模型后,模型计算值与真实值之间的差值称为误差,所有 n 个误差的平方之和称为误差平方和。在回归分析中,可以用这一数值的大小来表明模型的好坏。

(10) k -均值算法(k -Means Clustering Algorithm)

是一种迭代求解的聚类算法。其步骤是,将数据分为 K 组,则随机选取 K 个对象作为初始的聚类中心,然后计算每个对象与各个种子聚类中心之间的距离,把每个对象分配给距离它最近的聚类中心。聚类中心以及分配给它们的对象就代表一个聚类。每分配一个样本,聚类的聚类中心会根据聚类中现有的对象被重新计算。这个过程将不断重复直到满足某个终止条件。终止条件可以是没有(或最小数目)对象被重新分配给不同的聚类,没有(或最小数目)聚类中心再发生变化,误差平方和局部最小。

(11) k -近邻算法(k -Nearest Neighbor Algorithm)

k -近邻算法是一个相对比较成熟的机器学习算法。该算法的基本思路是：在特征空间中，如果一个样本附近的 K 个最近(即特征空间中最邻近)样本的大多数属于某一个类别，则该样本也属于这个类别。

2. 技术工具

(1) Matplotlib

Matplotlib 是一个基于 Python 的二维绘图库，用于根据数据来绘制图形，方便直观化呈现。擅长于用简单的语句来绘制线图、散点图、等高线图、条形图、柱状图、3D 图形甚至是图形动画等。

(2) NumPy

NumPy(Numerical Python)是 Python 语言的一个扩展程序库，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。

NumPy 为开放源代码并且由许多协作者共同维护开发。NumPy 是一个运行速度非常快的数学库，主要用于数组计算，包含一个强大的 N 维数组对象 ndarray、广播功能函数、整合 C/C++/Fortran 代码的工具和线性代数、傅里叶变换、随机数生成等功能。

NumPy 最重要的一个特点是其 N 维数组对象 ndarray，它是一系列同类型数据的集合，以 0 下标为开始进行集合中元素的索引。ndarray 对象是用于存放同类型元素的多维数组。ndarray 中的每个元素在内存中都有相同存储大小的区域。

ndarray 对象的内容可以通过索引或切片来访问和修改，与 Python 中 list 的切片操作一样。ndarray 数组可以基于 0~n 的下标进行索引，切片对象可以通过内置的 slice 函数，并设置 start、stop 及 step 参数进行，从原数组中切割出一个新数组。

NumPy 通常与 SciPy(Scientific Python)和 Matplotlib(绘图库)一起使用，这种组合广泛用于替代 MatLab，是一个强大的科学计算环境，有助于我们通过 Python 学习数据科学或者机器学习。

五、教学参考资源

参考书

1. 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016: 197 – 204, 225 – 226.
2. 丁亮, 姜春茂, 于振中. 人工智能基础教程: Python 篇(青少版)[M]. 北京: 清华大学出版社, 2019: 235 – 248.

六、教学参考案例

■ 参考案例

利用 k -均值聚类算法对鸢尾花数据进行聚类

北京师范大学附属实验中学 董 玥

(1课时)

1. 教学目标

- 通过与分类问题做对比,了解聚类的基本概念。(信息意识、计算思维)
- 通过直观实例演示分析,理解 k -均值聚类算法的基本思路和特点。(计算思维)
- 通过程序实操,掌握利用 k -均值聚类算法对鸢尾花数据做聚类的方法。(计算思维、数字化学习与创新)

2. 教学对象分析

课程面向高中阶段学生,在学习本课之前,学生已经学习了 Python 语言的基本使用,并且学习了人工智能课程中的基本概述以及机器学习中分类算法等内容,具备一定的人工智能基础知识和技能。

3. 教学重难点

教学重点:聚类的基本概念; k -均值聚类算法的基本思路和特点;利用 k -均值聚类算法对鸢尾花数据做聚类的方法。

教学难点:利用 k -均值聚类算法对鸢尾花数据做聚类的方法。

4. 教学方法

- 对于 k -均值聚类算法的基本思路,教师可以采用比较直观有趣的例子,比如对动物园里面的火烈鸟进行分群,让学生理解聚类算法的基本思路,同时也可以通过体验相关网站,让学生直观感受初始质点和群的数量对于聚类效果的影响。

- 对于 k -均值聚类算法的特点,教师可以对照实际的操作过程,通过引导分析的方式,让学生一步一步自己进行归纳总结,找到答案。

5. 教学工具

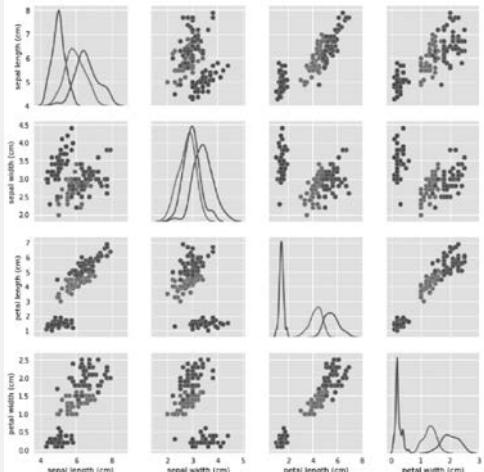
网络机房、投影、Python 环境、Python 源代码。

6. 教学过程设计

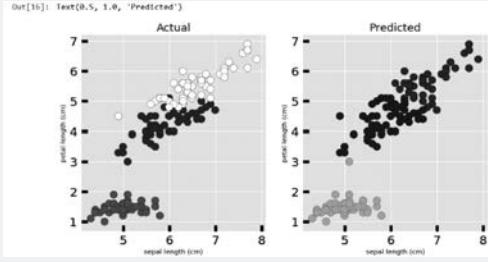
教学过程设计如表 2-7 所示。

表 2-7 教学过程设计

教学环节	教学内容	学生活动	设计意图																														
情境引入	<p>提问:之前我们学习了鸢尾花的分类问题,现在我们假设,埃德加·安德森虽然从加拿大加斯帕半岛上的鸢尾属花朵中提取了形态学变异数据,但他并不知道鸢尾花有几个亚属,那么他该怎么对这种只有四个特征值,但并没有标签的数据进行划分呢?</p> <ul style="list-style-type: none"> • 150 个样本; • 四个特征:花萼和花瓣的长度和宽度。 <table border="1"> <thead> <tr> <th></th> <th>sepal length (cm)</th> <th>sepal width (cm)</th> <th>petal length (cm)</th> <th>petal width (cm)</th> </tr> </thead> <tbody> <tr><td>0</td><td>5.1</td><td>3.5</td><td>1.4</td><td>0.2</td></tr> <tr><td>1</td><td>4.9</td><td>3.0</td><td>1.4</td><td>0.2</td></tr> <tr><td>2</td><td>4.7</td><td>3.2</td><td>1.3</td><td>0.2</td></tr> <tr><td>3</td><td>4.6</td><td>3.1</td><td>1.5</td><td>0.2</td></tr> <tr><td>4</td><td>5.0</td><td>3.6</td><td>1.4</td><td>0.2</td></tr> </tbody> </table> <p>参考答案:将数据值近似地划分为一类</p>		sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	0	5.1	3.5	1.4	0.2	1	4.9	3.0	1.4	0.2	2	4.7	3.2	1.3	0.2	3	4.6	3.1	1.5	0.2	4	5.0	3.6	1.4	0.2	思考,回答	创设情境,激发学生学习兴趣,为讲授新知做铺垫
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)																													
0	5.1	3.5	1.4	0.2																													
1	4.9	3.0	1.4	0.2																													
2	4.7	3.2	1.3	0.2																													
3	4.6	3.1	1.5	0.2																													
4	5.0	3.6	1.4	0.2																													
介绍聚类算法的基本概念	<p>1. 监督学习和无监督学习的对比。 2. 聚类的概念。 通过分析数据在特征空间的聚集情况,可以将一组数据分成不同的类,我们把这种方法叫做聚类</p>	听讲,思考,记忆	通过对聚类算法的介绍,让学生了解聚类算法的基本概念																														
讲解 k -均值聚类算法	<p>1. 以动物园的火烈鸟为例子,讲解 k-均值聚类算法的基本思路: (1) 决定分成三群; (2) 把数据点随机分成三群; (3) 计算三群的中心; (4) 计算数据点最接近的中心; 如果数据点有群的变换则回到第三步骤; (5) 结束。</p> <p>提问:请问这三群火烈鸟各有什么特征呢? 参考答案:胖的火烈鸟,瘦高的火烈鸟,瘦小的火烈鸟。</p> <p>2. 引导学生总结 k-均值聚类算法的特点: (1) 将输入的数据点分为多群; (2) 聚类模型可以直接对数据点使用,不需要训练; (3) 最终聚类的结果有多种可能; (4) 需要人为观察数据来确定聚类模型的参数; (5) 需要人类来解读数据的聚类效果。</p> <p>3. 体验活动: 访问并学习关于聚类算法的网站,体验质心和 K 值变化带来的聚类结果。 请学生分享自己体验的结果(质心和 K 值对聚类的影响)</p>	<p>听讲,思考,回答。</p> <p>思考,分析,总结,回答。</p> <p>操作,思考,回答</p>	通过对 k -均值聚类算法的介绍,让学生了解该算法的基本思路和流程,体会该算法需要人类来解读数据的聚类效果。																														

教学环节	教学内容	学生活动	设计意图
	<p>1. 调用之前所使用的鸢尾花分类课程的程序代码, 观察任意两个属性间数据的分布状况。</p>  <p>提问: 同学们认为选择什么样的属性组合来进行聚类, 效果会比较好, 分几类?</p> <p>参考答案: 选择数据点分布呈现为一族一族的属性组合, 聚类效果比较好, 分3簇。</p> <p>2. k-均值聚类算法的调用。</p> <pre>In [15]: feature_index = [0, 2] x = iris.data[:, feature_index] y = iris.target In [16]: km = KMeans(n_clusters = 3, n_jobs = 4, random_state=2) km.fit(x) Out[16]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300, n_clusters=3, n_init=10, n_jobs=4, precompute_distances='auto', random_state=2, tol=0.0001, verbose=0)</pre> <pre>In [15]: centers = km.cluster_centers_ print(centers) [[5.87413793 4.39310345] [5.00784314 1.49215686] [6.83902439 5.67804878]]</pre> <p>选择两个特征值: 比如0和2; 设定簇的数量: 3; 查看聚类结果: 三个簇的中心。 提问: (1) 在 k-均值聚类算法的调用过程中, 我们是否需要标签呢? 这属于我们之前讲过的监督学习还是无监督学习? 参考答案: 不需要标签, 无监督学习。 (2) 我们得到的结果, 是一个训练处理的模型, 还是对数据点的直接操作结果? 参考答案: 得到的是直接聚类的结果, 不是训练模型。</p>	听讲, 操作, 观察, 思考, 回答。	通过调用鸢尾花分类课程的程序代码, 让学生可以观察出不同属性间数据的分布状况, 体会 k -均值聚类算法应用的意义, 并确定需要划分的群的数量。
利用 k -均值聚类算法, 对鸢尾花数据做聚类		听讲, 操作, 思考, 回答。	通过对 k -均值聚类算法的程序调用, 让学生学会如何利用该算法去做聚类, 在这一过程中, 理解该算法的特点。

续表

教学环节	教学内容	学生活动	设计意图
	<p>3. 比较聚类结果与真实类别分布的差异。</p> <pre>In [16]: new_labels = km.labels_ fig, axes = plt.subplots(1, 2, figsize=(12,6)) axes[0].scatter(X[:, 0], X[:, 1], c=y, cmap='ocean', edgecolor='k', s=150) axes[1].scatter(X[:, 0], X[:, 1], c=new_labels, cmap='jet', edgecolor='k', s=150) axes[0].set_xlabel(names[feature_index[0]]) axes[0].set_ylabel(names[feature_index[1]]) axes[1].set_xlabel(names[feature_index[0]]) axes[1].set_ylabel(names[feature_index[1]]) axes[0].xaxis.set_tick_params(direction='in', length=10, width=5, colors='k', labelsize=20) axes[1].xaxis.set_tick_params(direction='in', length=10, width=5, colors='k', labelsize=20) axes[0].yaxis.set_tick_params(direction='in', length=10, width=5, colors='k', labelsize=20) axes[1].yaxis.set_tick_params(direction='in', length=10, width=5, colors='k', labelsize=20) axes[0].set_title('Actual', fontsize=18) axes[1].set_title('Predicted', fontsize=18)</pre>  <p>提问:请观察聚类结果与真实类别分布差异,想想为什么会这样? 参考答案:有些点会被划分到别的群,这是由 k-均值聚类算法的运算方式决定的。</p> <p>4. 实践任务。 请同学们完成如下任务: (1) 更换不同组合的两个属性做聚类,看看哪种聚类效果最好。 (2) 如果分成 4 群或是 5 群或是更多,聚类效果好坏如何? (3) 根据学生任务完成情况,请学生展示自己的成果,回答问题</p>	观察,思考,回答。	通过比较分群结果与真实类别分布的差异,让学生体会 k -均值聚类算法运算方式的特点。
课堂总结	<p>1. 聚类的基本概念。 2. k-均值聚类算法的基本思路和特点。 3. 利用 k-均值聚类算法对鸢尾花数据做聚类的方法</p>	操作,思考,展示,回答	通过让学生完成任务,掌握调用 k -均值聚类算法做聚类的方法,学会该如何选择合适的属性进行聚类运算,同时也体会该算法的弊端,即人为设定的群的数量对聚类的结果划分效果有很大影响

漫游深度学习的世界

一、本章学科核心素养的渗透

人工智能已逐渐深入到人类的日常生活中,而人工智能技术的发展将在建设制造强国、质量强国、航天强国、交通强国、网络强国、数字中国的过程中发挥重要作用,并产生深远的影响。在新一代人工智能发展的过程中,最核心的技术当属深度学习。深度学习让人工智能更接近于最初的目标——模拟人类的思维方式,是人工智能发展取得突破的关键。在此背景下,本章承载着帮助学生深入了解人工智能发展核心、理解深度学习的基本思想、体验与完成人工智能项目的重任。本章内容对应的学科核心素养要求体现在以下几个方面:

- 通过体验人脸识别、手写数字识别、图像风格迁移等应用引导学生认识深度学习,带领学生关注人工智能领域发展中的新动向和新趋势,在项目实践的过程中培养学生根据解决问题的需要,自觉、主动地寻求恰当的方式以及合适的技术获取与处理信息;(信息意识)
- 通过生物神经元与人工神经元的对比,促进学生理解人工神经元的工作机制,通过绘制激活函数图像、体验梯度下降法等项目过程,体验用计算机解决问题的方法与过程,包括界定问题、抽象特征、建立模型等。(计算思维)
- 通过体验使用感知机实现与门电路、手写数字识别、图像识别、图像风格迁移等多个实践项目,引导学生理解神经网络的训练过程,培养学生利用计算思维设计并解决问题的能力。(计算思维)
- 通过多个具体的实践项目,引导学生在实践的过程中掌握数字化学习系统、学习资源与学习工具的操作,利用所学的知识创造性地解决问题,完成学习任务,形成自己的创新作品,如利用图像风格迁移设计自己的图形作品等。(数字化学习与创新)
- 通过多个深度学习的应用(人脸识别、手写识别、图像识别等),培养学生在面对信息技术创新产生的新观念与新事物时,具有积极了解与学习的态度,认识到信息技术发展给社会生产和生活方式带来的影响与改变。(信息社会责任)

二、本章知识结构

本章学习内容分为 4 节,主要内容是了解深度学习这一技术的基本思想、重要概念、算法设计以及基于 Python 语言的编程实现。

第一节从类比生物神经元和人工神经元入手,帮助学生认识人工神经元的基本模型与工作机制;通过神经元进一步拓展到单层感知机模型,理解单层感知机的基本结构以及计算公式。在理解感知机的基础上,基于 Python 语言实现人工神经元的激活函数(阶梯函数和 S 形函数),并完成逻辑门的感知机项目,进一步帮助学生理解感知机。

第二节在第一节学习的基础上,以异或门感知机的非线性可分问题为探究,引出多层感知机的基本结构,帮助学生认识神经网络的结构和工作机制。通过反向传播算法和梯度下降法的分析,深入理解感知机训练学习的算法原理,学习神经网络。在理解算法的基础上,利用 Python 编程模拟多层感知机的训练过程,体验人工智能项目的实践与应用。

第三节从体验人脸识别与图像识别案例入手,引导学生学习深度神经网络结构、深度学习和数据集的概念。通过分析全连接、连接权重的学习算法等过程理解深度学习的发展困境。在了解深度神经网络概念和结构的基础上,通过探究 MNIST 手写数字数据集这个项目,帮助学生在实际活动中理解深度学习如何分辨图像。在前面理解深度学习技术的基础上,分析开发智能应用的工作流程,并了解深度学习的框架。

第四节以卷积神经网络为主题,通过数学分析及可视化图像,认识卷积操作的过程,学习卷积神经网络的主要要素,分别为输入层、卷积层、激活函数层、池化层和全连接层。基于 Python 编程实现卷积神经网络的激活函数 ReLU,并与阶跃函数和 S 形函数进行对比,了解 ReLU 的特点及优势。基于 Keras 完成卷积网络的项目实践,包括手写数字识别、图像分类、图像风格迁移等实践项目理解卷积神经网络的应用,提高利用计算机解决问题的能力。

本章主要知识与概念的关系如图 3-1 所示。

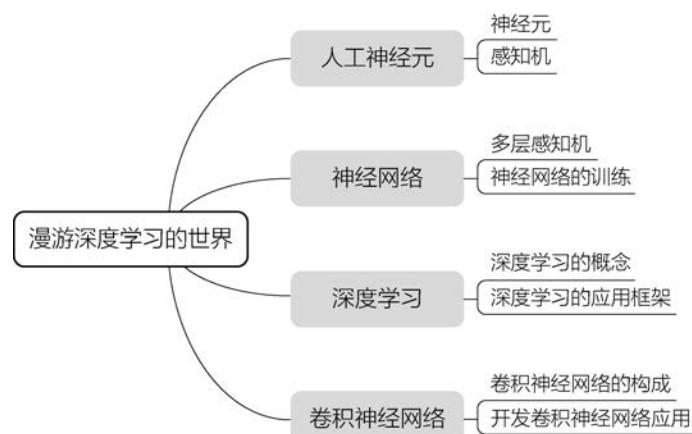


图 3-1 主要知识与概念的关系

三、本章项目活动的设计思路与实施建议

本章主要通过体验深度学习的应用,理解深度学习的概念,实现深度学习算法的关键模块,熟悉机器学习应用开发的一般过程,并借助人工智能开放平台和应用案例体验简单的人工智能应用项目。

本章项目任务以小组合作的形式为主,依据“学习—体验—实践—探究—总结”的过程分节展开,引领学生逐步了解深度学习的概念结构和算法原理,帮助学生逐层进入到人工智能的深层学习。第一节的项目任务主要是利用单层感知机模拟与门电路,让学生在理解逻辑门电路的基础上,利用 Python 编程实现与门电路的核心模块,在此过程中理解单层感知机的工作原理。第二节的项目任务主要是以随机梯度下降法实现神经网络的学习,利用 Python 程序体验随机梯度下降法的算法理念,感受神经网络的工作机制。第三节和第四节的项目任务是以卷积神经网络实现 MNIST 手写字识别,学生可以通过体验手写识别项目了解卷积神经网络的学习过程,了解开发智能应用的一般过程。

为了能够加强小组合作、促进个人学习,本章项目在最后建议设立一次总结交流活动。每个小组根据自己完成的作品,交流分享在项目探究过程中的经验和教训,组员完成自己的项目总结。指导教师从每位同学的参与态度、互助合作精神、所完成项目任务的数量与质量、所提出的想法与代码等方面对学生进行综合评价。具体的评价指标及标准可参考表 3-1。

表 3-1 评价指标及标准

评价指标	评价标准描述		
	优秀(80~100)	良好(60~79)	需努力(0~59)
研究主题和问题 (20%)	项目具有明确的研究主题,界定研究问题,制订详细的解决方案	项目具有研究主题,具有解决思路	项目没有明确的研究主题
协作探究 (20%)	组内成员分工明确,具有详细的分工表,成员积极互助	组内成员分工较明确,个别成员参与度较低,成员未能积极互助	组内成员分工不明确,多数成员参与度较低,成员没有互助
成果展示 (60%)	1. 能够成功实现深度学习的智能模块。 2. 研究方法得当,技术工具适宜。 3. 提出创新的想法	1. 基本实现深度学习的智能模块。 2. 研究方法较为得当,技术工具较为适宜	1. 未能实现深度学习的智能模块。 2. 内容不完整

四、本章教学课时分配建议

建议本章的学习安排 15 课时:第一节 3 课时,第二节 3 课时,第三节 4 课时,第四节 5

课时。课时安排围绕本章的整体知识架构、教学内容、项目任务等进行大致划分,实际教学需要结合学生学习理解和实践操作的具体情况灵活处理调整,没有必要拘泥于每节课的进度。本章教学课时分配建议如表 3-2 所示。

表 3-2 本章课时分配建议

节次	课时	具体 内 容
第一节	3 课时	第 1 课时: 人工神经元模型与工作机制
		第 2 课时: 单层感知机基础
		第 3 课时: 单层感知机应用
第二节	3 课时	第 4 课时: 多层感知机的建立与学习
		第 5 课时: 多层感知机的训练过程
		第 6 课时: 人工神经网络与梯度下降法
第三节	4 课时	第 7 课时: 深度学习基础
		第 8 课时: 深度学习与数据集
		第 9 课时: 深度学习框架
		第 10 课时: 深度学习应用
第四节	5 课时	第 11 课时: 初识卷积神经网络
		第 12 课时: 卷积神经网络结构
		第 13 课时: 卷积神经网络应用
		第 14 课时: 卷积神经网络项目实践 1
		第 15 课时: 卷积神经网络项目实践 2

第一节

人工神经元与单层感知机

一、教学目标与重点

教学目标:

- 通过与生物神经元进行对比,了解深度学习的基本组成单元——人工神经元,体验利用计算机解决问题的方法与过程,包括界定问题、抽象特征、建立

模型等。(计算思维)

- 通过样例学习与观摩,理解人工神经元的输入值、传递函数、激活函数以及输出的工作机制,体验利用计算机科学技术的思想和方法分析问题、解决问题。(计算思维)
- 通过编程实践与数学分析,认识早期两种激活函数——阶跃函数(也称为阶梯函数)和 S 形函数,并能够基于 Python 语言实现两种函数的图像绘制,完成图像绘制的作品。(数字化学习与创新)
- 通过对比分析,了解并学习感知机的概念及其与人工神经元的区别。(计算思维)
- 通过案例实践,学习利用单层感知机实现二分类应用的方法,理解与门、或门、与非门电路,并能够用 Python 语言实现与门、或门、与非门电路的感知机。(数字化学习与创新)
- 通过多个项目的实践过程,了解单层感知机的优点与不足,体会人工智能的魅力。(信息社会意识)

教学重点:

- 深入理解人工神经元的模型及工作机制;
- 学习单层感知机的基本算法,并能够实现简单的人工智能应用。

二、内容分析与建议

1. 内容分析

本小节是教科书第三章“漫游深度学习的世界”中第一节的内容,本节课程主要在于引导学生认识人工神经元的模型,理解人工神经元模拟生物神经元处理感知信号的基本机理;理解感知机的工作原理,即仿照生物神经元对输入信号进行处理和传输,通过学习

算法实现对输入信号的调谐。本阶段,学生有两个实践任务:其一是在理解人工神经元模型的基础上,使用 Python 实现阶跃函数与 S 形函数;其二是在分析感知机与门电路的基础上,基于 Python 实现与非门及或门的感知机算法。本节主要知识框图如图 3-2 所示。

对于任务一,可以采用图像展示、分析理解、讨论实现、展示评价的方式进行。首先,教师在分析人工神经元模型的基础上,引领学生理解如何用激活函数来模拟神经元兴奋的状态。可以给出阶跃函数与 S 形函数的表达式,展示所绘制的函数图像,结合两者带领学生理解激活函数。接着组织学生讨论分析实现阶跃函数的流程图,导入 NumPy 模块,编写程序实现阶跃函数的代码。然后展示学生程序运行效果。同理,实现 S 形函数的

Python 程序编写。最后,引入 Matplotlib 库,学生通过运行代码绘制出函数图像。

对于任务二,可以按照回顾学习、案例分析、补全代码、迁移应用法的流程进行。首先,教师带领学生在回顾与或非逻辑运算的基础上,学习四种逻辑门运算(与、或、与非、异或)。接着,以与门运算为例,引领学生分析如何确定单层感知机的各项参数,实现与门电路。然后,通过分析呈现算法流程图,在给出程序半成品的基础上,让学生补全代码,并运行程序,观看结果。最后,组织学生分析讨论,尝试利用 Python 完成与非门、或门的感知机。

2. 教学建议

本节重点在于理解人工神经元的模型、工作机理及单层感知机的原理,教师要根据学生情况,对其中涉及到的函数公式进行适度的展开与挖掘,在学生现有的基础上进行一些数学层面上的引导与分析,以便帮助学生更透彻地理解人工神经元与单层感知机。因此,在教学中,不仅要注重学习基本组成结构,更要分析神经元与感知机实现的原理与机制,理解深度学习模拟人类解决问题的思路与方法,让学生体悟到人工智能内的技术魅力。

教学任务可以不必拘泥于教科书提供的案例,可以基于学生的生活和学习调整或重构案例,创建真实情景,让学生体验利用智能技术解决现实问题的过程。例如,可以通过自动水果贩卖机如何选择苹果和梨,分析如何利用单层感知机算法进行实现,加深对程序的理解,感悟智能产品背后蕴含的技术。

教学中要注重建立教师主导、学生主体的双主体模式,教师关注对思考分析的逐步引导,突出学生实践探究的主体地位,让学生体验计算机技术解决问题的全过程。同时,教师要鼓励学生参与到算法分析、程序编写的过程中,通过互动交流,帮助学生逐步理解算法思想。

教学中要注重强化培养学生应用技术解决现实问题的能力,关注学生解决问题的过程表现,引导学生能够利用所学内容解决其他问题。例如,在利用单层感知机算法模拟与门电路后,可以引导学生分析思考实现或门、非门电路的方法,从而加深学生对感知机算法的理解。

三、习题分析与提示

■ 习题 1 描述

见教科书第 63 页的项目实践“使用 Python 实现阶跃函数与 S 形函数”。

■ 习题 1 解题思路

1. 使用 Python 实现阶跃函数

(1) 根据阶跃函数的定义 $h(x) = \begin{cases} 0, & x \leq 0, \\ 1, & x > 0 \end{cases}$ 编写程序

```
def step_function(x):
    if x > 0:
        return 1
    else:
        return 0
print(step_function(1))
print(step_function(-2))
```

程序运行结果：1

0

(2) 编写程序,绘制阶跃函数的图像

在(1)中,编写的阶跃函数 step_function(x)只能接收实数,不能接收数组。为了方便使用 numpy 数组来实现,可以考虑编写如下程序,从而实现阶跃函数的图像绘制。

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm

def step_function(x):
    return np.array(x> 0, dtype= np.int64)

myfont = fm.FontProperties(fname= r'C:\Windows\Fonts\simsun.ttc')

x = np.arange(- 5.0,5.0,0.1) # 生成一个 numpy 数组,x 的范围是(- 5.0,5.0),步长为 0.1
y1 = step_function(x) # 调用阶跃函数,生成 y 值

plt.plot(x, y1, label= 'S 形函数')
plt.ylim(- 0.2, 1.2) # 指定 y 轴范围为[- 0.2,1.2]

plt.xlabel("x")
plt.ylabel("y")
plt.title('阶跃函数',fontproperties = myfont, fontsize= 12)

plt.plot(x,y1)
plt.show()
```

2. 使用 Python 实现 S 形函数,绘制 S 形函数图像

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm

def step_function(x):
    return np.array(x> 0, dtype= np.int64)

def sigmoid(x):
    return 1/(1+ np.exp(- x))

myfont= fm.FontProperties(fname= r'C:\Windows\Fonts\simsun.ttc')

x= np.arange(- 5.0,5.0,1.0)
y1 = sigmoid(x)
y2 = step_function(x)

plt.plot(x, y1, label= 'S 形函数')
plt.plot(x, y2, linestyle= '--', label= '阶跃函数')
plt.ylim(- 0.2, 1.2)

plt.xlabel("x")
plt.ylabel("y")
plt.title('S 形函数与阶跃函数的对比', fontproperties= myfont, fontsize= 12)
plt.legend(loc= 'upper left', prop= myfont)

plt.savefig('fig3- 2.png', dpi= 400, bbox_inches= 'tight')
plt.show()
```

习题 2 描述

见教科书第 65 页的项目实践“感知机实现与门电路”。

习题 2 解题思路

1. 感知机实现与门电路

(1) 与门电路输入和输出信号对应表(如表 3-3 所示)

表 3-3 与门电路输入和输出信号对应表

x_1 (输入)	x_2 (输入)	y (输出)
0	0	0
1	0	0
0	1	0
1	1	1

利用感知机表示与门电路,需要做的就是确定能满足上述真值表的各个参数 w_1 、 w_2 、 θ 的值。事实上,满足条件的参数组合有无数种。例如,我们可以取 $w_1 = 0.5$, $w_2 = 0.5$, $\theta = 0.7$ 。设定参数后,仅当 $x_1 = 1$, $x_2 = 1$ 时,输入信号的加权和为 $w_1x_1 + w_2x_2 = 0.5 + 0.5 = 1 > 0.7 = \theta$, 对应输出为 1。

(2) Python 实现

```
def AND(x1, x2):
    w1, w2, 阈值 = 0.5, 0.5, 0.7
    输入加权和 = x1 * w1 + x2 * w2
    if 输入加权和 <= 阈值:
        return 0
    elif 输入加权和 > 阈值:
        return 1

print(AND(0,0))
print(AND(0,1))
print(AND(1,0))
print(AND(1,1))
```

2. 感知机实现或门电路

(1) 或门电路输入和输出信号对应表(如表 3-4 所示)

表 3-4 或门电路输入和输出信号对应表

x_1 (输入)	x_2 (输入)	y (输出)
0	0	0
1	0	1
0	1	1
1	1	1

与感知机表示与门电路相同,利用感知机表示或门电路,需要做的就是确定能满足上述真值表的各个参数 w_1 、 w_2 、 θ 的值。同样,满足条件的参数组合有无数种。例如,我们可以取 $w_1 = 0.5$, $w_2 = 0.5$, $\theta = 0.4$ 。设定参数后,仅当 $x_1 = 0$, $x_2 = 0$ 时,输入信号的加权和为 $w_1x_1 + w_2x_2 = 0 + 0 = 0 < 0.4 = \theta$,对应输出为 0。

(2) Python 实现

```
def OR(x1, x2):
    w1, w2, 阈值= 0.5, 0.5, 0.4
    输入加权和= x1* w1 + x2* w2
    if 输入加权和<= 阈值:
        return 0
    elif 输入加权和> 阈值:
        return 1

print(OR(0,0))
print(OR(0,1))
print(OR(1,0))
print(OR(1,1))
```

3. 感知机实现与非门电路

(1) 与非门电路输入和输出信号对应表(如表 3-5 所示)

表 3-5 与非门电路输入和输出信号对应表

x_1 (输入)	x_2 (输入)	y (输出)
0	0	1
1	0	1
0	1	1
1	1	0

与前两者一样,利用感知机表示与非门电路,需要做的就是确定能满足上述真值表的各个参数 w_1 、 w_2 、 θ 的值。同样,满足条件的参数组合有无数种。例如,我们可以取 $w_1 = -0.5$, $w_2 = -0.5$, $\theta = -0.8$ 。设定参数后,仅当 $x_1 = 1$, $x_2 = 1$ 时,输入信号的加权和为 $w_1x_1 + w_2x_2 = -0.5 \times 1 - 0.5 \times 1 = -1 < -0.8 = \theta$,对应输出为 0。

(2) Python 实现

```
def NAND(x1, x2):
    w1, w2, 阈值= - 0.5, - 0.5, - 0.8
    输入加权和= x1* w1+ x2* w2
    if 输入加权和<= 阈值:
        return 0
    elif 输入加权和> 阈值:
        return 1

print(NAND(0,0))
print(NAND(0,1))
print(NAND(1,0))
print(NAND(1,1))
```

四、核心概念精解与技术工具介绍

1. 核心概念精解

(1) 人工神经元

简单来说,人工智能所研究的是如何利用机器模拟和扩展人的智能。随着对脑神经科学认知的不断深入,研究发现大脑中存在着上百亿个高度连接的神经元,这些神经元构成的复杂神经系统帮助人类实现感知与思维。因此,如何模拟生物神经元、模拟大脑中的神经网络成为人工智能研究的重点。

生物神经元(如图 3-3 中左图所示)由四个主要部分构成:树突、细胞体、轴突和神经末梢。树突主要作用是接收其他神经元的电信号,是神经元的输入端,一个细胞体可以有多个树突,意味着可以与多个神经元相连;细胞体负责处理输入的电信号,并通过电位差值(输入的电信号会导致细胞体内外产生电位差)控制电信号的输出;轴突用于传递信号;信号经由神经末梢传输到其他神经元的输入端树突。

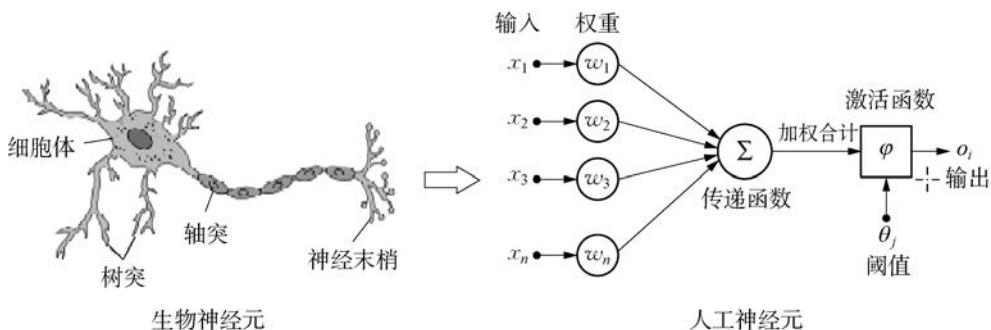


图 3-3 生物神经元与人工神经元

人工神经元(如图 3-3 中右图所示)则是在探索生物神经元的基础上进行模拟,提取关键要素,实现工作机制,建立数学模型。

① n 维输入:信号输入

一个生物神经元有多个树突,意味着有多个输入。在人工神经元中,利用 x_1, \dots, x_n 建立 n 维输入,模拟多个树突。每个树突对于神经体的作用是不尽相同的,因此,我们给每项输入配备“权重”,建立 w_1, \dots, w_n ,用来模拟不同输入的影响。

② 传递函数:输入处理

细胞体用来对输入信号进行处理,但目前还无法知晓是否是真正的处理过程。因此,在模拟时,人工神经元采用了最简单粗暴的形式——加权求和,即 $x_1w_1 + \dots + x_nw_n$ (或 $\sum_{i=1}^n x_iw_i$)。因此,在人工神经元中的传递函数就是通过加权求和来模拟细胞体信号处理的过程。

③ 激活函数:输出过程

细胞体在处理电信号后,会基于电位差(阈值)来控制是否向下传递神经脉冲信号。

在人工神经元中建立的激活函数则用来模拟这个过程,判断神经元是否“兴奋”向其他神经元传递信号。人工神经元常使用的激活函数阶跃函数与 S 形函数如表 3-6 所示。

表 3-6 阶跃函数与 S 形函数

函数名称	阶跃函数	S 形函数
函数表达式	$h(x) = \begin{cases} 0, & x \leq 0, \\ 1, & x > 0 \end{cases}$	$h(x) = \frac{1}{1 + e^{-x}}$
函数说明	当输入值小于等于 0 时,输出 0; 当输入值大于 0 时,输出 1。可以看到阶跃函数以输入值 0 为分界点,输出值进行跳变	Sigmoid 函数,当输入值趋近于负无穷时,输出值趋近于 0; 当输入值趋近于正无穷时,输出值趋近于 1; 当输入值等于 0 时,输出值为 0.5。可以看出,Sigmoid 函数的输出映射在(0,1)之间,意味着输入即使是非常大的正数或非常小的负数,输出也会是(0,1)之间,这使得数据不容易发散
函数图像		

(2) 感知机

人工神经元模拟生物神经元的特点及工作机制,抽象形成了神经元的数学运算模型。但是人工神经元仅是建立了模型,并没有规定权重 w 如何确定。为了解决这个问题,1957 年感知机由此提出。感知机在人工神经元的基础上,引入偏置值(相当于阈值),获得产生权重和偏置值的方法(感知机的训练和学习),可谓最早的人工神经网络。单层感知机是一个具有一层人工神经元的网络,其基本结构如图 3-4 所示。

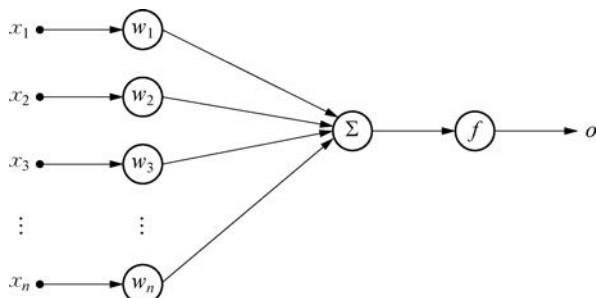


图 3-4 单层感知机基本结构

x_1, \dots, x_n 用来表示 n 维输入, 对应权重为 w_1, \dots, w_n, θ 表示阈值, o 代表输出。那么, 单层感知机的计算公式可以表示为如下所示:

$$o = f(w_1x_1 + \dots + w_nx_n - \theta) = \begin{cases} 0, & w_1x_1 + \dots + w_nx_n \leq \theta, \\ 1, & w_1x_1 + \dots + w_nx_n > \theta. \end{cases}$$

2. 技术工具介绍

(1) 函数图像绘制

绘制函数图像的软件和方法有很多,既可以选用简单好操作的在线网站进行绘制,也可以利用专门的函数绘制软件或编写程序进行图像绘制。

(2) 设计流程图

流程图是辅助分析问题解决方法、表示算法思路的有效手段。用于设计制作流程图的软件和工具也有很多,既可以选择文本编辑软件中的自选图像进行绘制,也可以利用思维导图等软件进行绘制,还可以利用在线流程图绘制平台。

(3) Python 函数

Python 函数是一段具有特定功能的语句组合,可通过函数名进行表示与调用。Python 语言定义函数的格式如下:

```
Def 函数名(参数):  
    函数体  
    return 返回值列表
```

(4) Python 模块

Python 的第三方库可通过 pip 工具进行安装(pip install 库名),通过 import 进行导入(import 库名)。NumPy(Numerical Python)是 Python 语言的一个扩展程序库,支持大量的维度数组与矩阵运算,此外也针对数组运算提供大量的数学函数库。Matplotlib 是 Python 的绘图库。它可与 NumPy 一起使用,提供了一种有效的 Matlab 开源替代方案。它也可以和图形工具包一起使用,如 PyQt 和 wxPython。

五、教学参考资源

■ 参考书

塔里克·拉希德. Python 神经网络编程[M]. 林赐,译. 北京: 人民邮电出版社, 2018.

■ 参考论文

黄从智,白焰. 智能控制课程中感知器教案设计与教学实践[J]. 中国电力教育, 2014 (014).

六、教学参考案例

■ 参考案例 1

生物神经元与人工神经元

北京市第八中学 蓝 征

(1课时)

1. 教学目标

- 通过与生物神经元进行对比,了解深度学习的基本组成单元——人工神经元。(信息意识、计算思维)
- 通过样例学习与观摩,理解人工神经元的输入值、传递函数、激活函数以及输出的工作机制。(计算思维)
- 通过编程实践与数学分析,认识早期两种激活函数——阶跃函数(也称为阶梯函数)和 S 形函数,并能够基于 Python 语言实现两种函数的图像绘制。(计算思维、数字化学习与创新)

2. 教学对象分析

本节课的教学对象为高一年级的学生,学生在日常生活中已经广泛接触过人工智能的产品。在前面课程的学习中,学生已经学习过人工智能的概念、机器学习的常用算法,学习过算法描述和程序设计实现的基本方法,并具有一定利用计算机编程解决问题的意识和能力。

3. 教学重难点

教学重点:理解人工神经元模型及其工作机制。

教学难点:利用 Python 实现绘制阶跃函数和 S 形函数的图像。

4. 教学方法

讲授法、演示法、任务驱动法、小组合作学习。

5. 教学工具

计算机、网络、PPT、Python 编译环境。

6. 教学过程设计

教学过程设计如表 3-7 所示。

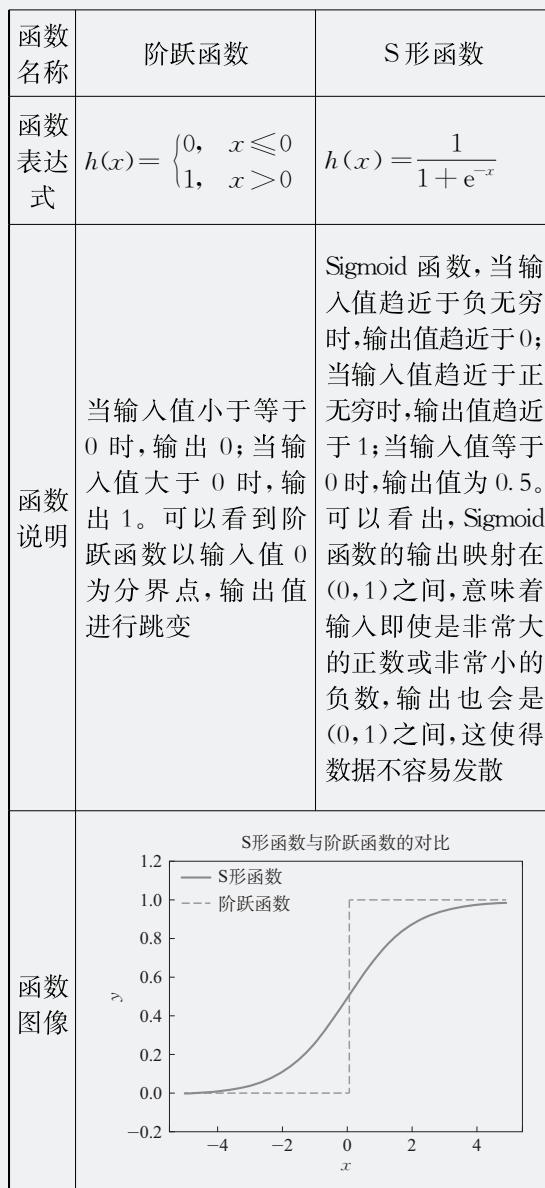
表 3-7 教学过程设计

教学环节	教学内容	学生活动	设计意图
情境引入	<p>引入情境,提出思考问题。</p> <p>通过前面的学习,我们知道人工智能是研究机器如何模仿、延伸和扩展人类智能的领域。而人类控制思考、运动、行为等这些的智慧能力都是由大脑创造的。如果我们能够利用机器模拟人类的大脑,这样就可以实现人工智能了。那么,人类的大脑思考时是如何工作的呢?</p>	<p>观看相关图片并思考</p>	<p>引出本课主题,通过教师引导激发学生探索大脑认知机制</p>

教学环节	教学内容	学生活动	设计意图
新知探究	<p>探究主题:大脑的认知机制。</p> <p>探究内容:</p> <ol style="list-style-type: none"> 播放视频:生物神经元的结构组成。 随着对脑神经科学认知的不断深入,研究发现大脑中约存在超过100亿个高度连接的神经元,这些神经元构成的复杂神经系统帮助人类实现感知与思维。 提出问题:通过刚才的视频你能回答出神经元的基本组成吗? 学生回答,分享交流。 总结展示,新知讲授。 <p>生物神经元由树突、轴突、细胞体和神经末梢组成。树突主要作用是接收其他神经元的电信号,是神经元的输入端,一个细胞体可以有多个树突,意味着可以与多个神经元相连;细胞体负责处理输入的电信号,并通过电位差值(输入的电信号会导致细胞体内外产生电位差)控制电信号的输出;轴突用于传递信号;信号经由神经末梢传输到其他神经元的输入端树突。末梢输出到其他的神经元的输入端。</p> <p style="text-align: center;">生物神经元</p> <ol style="list-style-type: none"> 提出问题:神经元的输入是由什么决定的呢? 学生回答,分享交流。 总结归纳,教师讲授。 <p>神经元的输出主要由两个部分决定:其一为前神经元的输入信号;其二为细胞体的状态是兴奋还是压抑(是否被激活)</p>	<p>学生观看相关视频,学习思考。</p> <p>师生互动,生生互动,共同探究生物神经元的基本结构及工作方式。</p> <p>交流分析,回答问题。</p> <p>聆听讲授,学习新知</p>	<p>通过视频动画、提问思考逐渐引导学生理解大脑中生物神经元的基本结构及其工作方式</p>
任务活动	<p>任务主题:建立生物神经元的数学模型。</p> <ol style="list-style-type: none"> 两人为一组,讨论分析,尝试模仿生物神经元建立数学模型,并画在学案相应位置。 教师巡视,给予指导。 选取两到三组的作品进行展示,并且邀请学生分享自己的想法。 教师归纳总结 	<p>生生互动,小组探究,绘制模型。</p> <p>展示分享,交流沟通</p>	<p>通过活动,体验如何用数学抽象模型表达神经元,体验利用计算机解决问题的思想</p>
新知讲授	<p>基于前面的任务活动,引出人工神经元的模型。</p> <p>人工神经元模型的基本结构:</p> <ol style="list-style-type: none"> n维输入:信号输入。 <p>在人工神经元中,利用x_1, \dots, x_n建立n维输入,模拟多个树突。每个树突对于神经体的作用是不尽相同的,因此,我们给每项输入配备“权重”,建立w_1, \dots, w_n,用来模拟不同输入的影响。</p>	<p>聆听学习,理解人工神经元模型输入、输出以及激活函数</p>	<p>通过教师讲授,帮助学生理解人工神经元模型的基本结构</p>

续表

教学环节	教学内容	学生活动	设计意图
	<p>2. 传递函数:输入处理。</p> <p>细胞体用来对输入信号进行处理,但目前还无法知晓是否是真正的处理过程。因此,在模拟时,人工神经元采用了最简单粗暴的形式——加权求和,即 $x_1w_1 + \dots + x_nw_n$(或 $\sum_{i=1}^n x_iw_i$)。</p> <p>3. 激活函数:输出过程。</p> <p>细胞体在处理电信号后,会基于电压差(阈值)来控制是否向下传递神经脉冲信号。在人工神经元中建立的激活函数则用来模拟这个过程,判断神经元是否“兴奋”向其他神经元传递信号。</p> <p>人工神经元常使用的激活函数阶跃函数与 S 形函数。</p>		



教学环节	教学内容	学生活动	设计意图
任务实践	<p>任务主题:利用 Python 实现绘制激活函数图像。</p> <ol style="list-style-type: none"> 1. 阶跃函数。 <ol style="list-style-type: none"> (1) 通过函数公式,设计流程图,分析实现过程。 (2) 两人一组,完成程序。 (3) 教师巡视,给予指导。 (4) 选取作品,分享展示。 2. S 形函数。 <ol style="list-style-type: none"> (1) 通过函数公式,设计流程图,分析实现过程。 (2) 两人一组,完成程序。 (3) 教师巡视,给予指导。 (4) 选取作品,分享展示。 3. 函数图像绘制。 <ol style="list-style-type: none"> (1) 介绍 Matplotlib 库:Matplotlib 是 Python 的绘图库。 (2) 提供半成品程序,引领学生阅读分析已有语句。 (3) 学生实践,补充代码,完成程序。 (4) 选取作品,分享展示 	上机实践,编写程序。师生互动,生生互动,分享交流	<p>通过函数公式与流程图,帮助学生分析算法。</p> <p>基于半成品程序补全主要代码,在有限时间中帮助学生理解程序思想,增强利用编程解决问题的能力</p>
总结归纳	<p>总结本节课的知识内容。</p> <ol style="list-style-type: none"> 1. 生物神经元的基本结构。 2. 人工神经元模型。 <ol style="list-style-type: none"> (1) 多维输入:值、权重。 (2) 传递函数:加权求和。 (3) 激活函数:阶跃函数、S 形函数;输出函数值 	知识总结,回顾复习	帮助学生系统归纳所学知识,促进知识内化吸收

参考案例 2

探索单层感知机

北京市第八中学 马 征

(1 课时)

1. 教学目标

- 通过对比分析,了解和学习感知机概念及与人工神经元的区别。(信息意识、计算思维)
- 通过案例实践,学习利用单程感知机实现二分类应用的方法。(计算思维、数字化学习与创新)
 - 通过补充应用,掌握基于 Python 语言实现与门、或门、与非门电路的感知机。(计算思维、数字化学习与创新)

2. 教学对象分析

本节课的教学对象为高一年级的学生,高中学生已经具有一定的逻辑推理能力以及数学思维。在前面课程的学习中,学生已经学习了人工神经元模型和工作机制,学习过 Python 语言,知道逻辑运算的方法,为本节课尝试利用单层感知机实现逻辑门电路提供了铺垫。

3. 教学重难点

教学重点:理解单层感知机的基本结构与工作机制。

教学难点:完成单层感知机应用的 Python 程序。

4. 教学方法

讲授法、演示法、任务驱动法、小组合作学习。

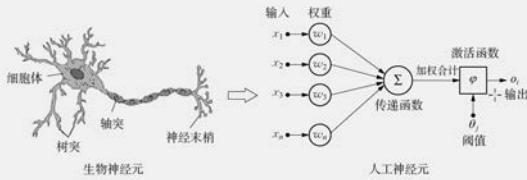
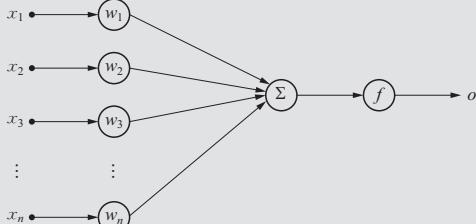
5. 教学工具

计算机、网络、PPT、Python 编译环境。

6. 教学过程设计

教学过程设计如表 3-8 所示。

表 3-8 教学过程设计

教学环节	教学内容	学生活动	设计意图
情境引入	<p>1. 回顾复习。</p> <p>通过前面的学习,我们已经成功构建了人工神经元的模型。你还记得人工神经元模型的组成和工作机制吗?</p> <p></p> <p>2. 提问思考,引出主题。</p> <p>(1) 在人工神经元中,权重的值是如何确定的呢?</p> <p>(2) 若想确定权重,你认为可以有什么方法吗?</p>	回顾复习,思考问题,回答分享	引出本课主题,通过教师引导激发学生思考
新知探究	<p>1. 感知机的提出。</p> <p>人工神经元仅是建立了数学模型,并没有规定权重 w 如何确定。为了解决这个问题,1957 年感知机由此提出。</p> <p>感知机在人工神经元的基础上,引入偏置值(相当于阈值),获得产生权重和偏置值的方法(感知机的训练和学习),可谓是最早的人工神经网络。</p> <p>2. 单层感知机。</p> <p>单层感知机是一个具有一层人工神经元的网络,其基本结构如下图。</p> <p></p> <p>3. 探究活动:根据以上单层感知机的基本结构,你能尝试写出单层感知机的函数表达式吗?</p> <p>(1) 分析尝试,在学案处填写表达式。</p> <p>(2) 分享学生所填写的表达式。</p> <p>(3) 教师归纳总结,展示表达式。</p> $o = f(w_1x_1 + \dots + w_nx_n - \theta) = \begin{cases} 0, & w_1x_1 + \dots + w_nx_n \leq \theta, \\ 1, & w_1x_1 + \dots + w_nx_n > \theta. \end{cases}$	聆听学习,对比辨别。基于单层感知机结构图,尝试建立单层感知机的数学表达	通过与之前学习的人工神经元进行对比,理解感知机,以及其与人工神经元的区别。 通过亲自尝试填写单层感知机的函数表达式,进一步加深对感知机的理解,学会抽象化的数学表达

续表

教学环节	教学内容	学生活动	设计意图															
讨论分析	<p>1. 提出问题:通过数学表达式,你认为单层感知机能用来解决什么问题? 2. 组织学生小组讨论,思考回答。 3. 教师引导,总结归纳。</p> <p>单层感知机的输出是一个二元值,是或否、真或假。那么可以帮助我们解决一个二分类问题</p>	小组讨论,表达自己的想法	从数学角度入手,让学生理解单层感知机的二分类应用															
应用实践	<p>1. 任务主题:苹果 or 香蕉。 任务说明:现在有一台自动售卖机,用来售卖水果。那么售卖机是如何区分不同的水果的呢?我们现在就来尝试分析如何利用单层感知机区分一个水果是苹果还是香蕉。 2. 提出问题,引导思考。 你是如何判断一个水果是苹果还是香蕉的呢? 3. 组织学生回答。 4. 分析归纳,数学表达。 我们可以根据特征来进行判断,其一可以看外形,定义圆形输入为 0,月牙形输入为 1;其二可以看颜色,红色为 1,黄色为 0。那么,输入会有几个维度?输入会有几个种类? 5. 提问引导:如果输出 0 代表苹果,输出 1 代表香蕉,那么权重 w_i、阈值 θ 应该如何取值呢? 6. 小组讨论,计算参数结果。 7. 分享展示计算结果,说出计算思路</p>	思考分析,讨论回答。小组合作,分析问题,计算结果,展示思路	通过区分水果的实际应用,从人为判断标准引入,分析输入、输出,形成数学表达。 通过小组合作分析参数的取值,展示结果,向其他同学分享思路															
总结过渡	<p>刚刚我们利用单层感知机解决了一个二分类问题,想必大家对单层感知机的应用有了更深刻的理解和认识。下面我们就来挑战一个计算机领域的经典问题:逻辑运算。</p> <p>逻辑运算主要有四种:与、或、与非及异或。基于四种运算构成的门电路是计算机硬件的基础:与门、或门、与非门及异或门</p>																	
项目探究	<p>1. 项目主题:感知机实现与门电路。 2. 小组合作,学习研究。 3. 分析探究:与门电路的输入与输出。</p> <p>(1) 提出问题:与门电路有几个输入?对应的输出是什么? (2) 组织学生回答。 (3) 归纳展示:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>x_1</td> <td>x_2</td> <td>y</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table> <p>4. 提问思考:与门电路的参数取值是多少? 5. 小组讨论,计算参数结果。</p>	x_1	x_2	y	0	0	0	1	0	0	0	1	0	1	1	1	思考分析,讨论回答。小组合作,计算结果。 编写程序,利用 Python 实现单层感知机的与门电路	基于上个探究的基础,先组织小组合作尝试完成与门电路感知机。 教师引领学生逐步思考输入输出,并通过学生分享实现生生学习。 实践编程,体验如何利用程序解决问题
x_1	x_2	y																
0	0	0																
1	0	0																
0	1	0																
1	1	1																

续表

教学环节	教学内容	学生活动	设计意图
	6. 分享展示计算结果,说出计算思路。 7. 编程实现。 (1) 根据流程图,分析算法思路。 (2) 小组编写与门电路的程序代码。 (3) 展示程序,分享交流		
迁移拓展	1. 小组任务:实现感知机与非门电路及或门电路。 2. 小组讨论,分析计算,编程实现。 3. 教师巡视学生实践情况,提供指导与帮助。 4. 选取两组同学的程序展示,并介绍自己的思路	小组合作,探究分析,编程实践	在与门电路的基础上尝试自己探究,利用单层感知机实现与非门及或门电路
归纳小结	师生共同总结本节课的收获: 提问 1:相比于人工神经元,感知机增加了什么? 提问 2:单层感知机可以解决什么问题?	回答问题,总结梳理	通过回答问题重新梳理课程知识

第二节

多层感知机与人工神经网络

一、教学目标与重点

教学目标:

- 通过样例思考,理解单层感知机只可处理线性可分问题的局限性,知道神经网络与多层感知机的异同,体会多层感知机与神经网络的优势,关注人工智能领域发展中的新动向和新趋势,能根据解决问题的需要,自觉、主动地寻求恰当的方式以及合适的技术获取与处理信息。(信息意识)
- 通过对比分析,认识损失函数的作用,知道常用的损失函数。(计算思维)
- 通过逐步地分析探究,理解梯度下降法的原理与过程,认识梯度与学习率,体验利用计算机科学技术的思想和方法分析问题、解决问题。(计算思维)
- 通过案例实践,体验基于 Python 编写的神经网络,通过修改参数理解其对神经网络学习性能的影响。(数字化学习与创新)
- 通过实践应用,理解多层感知机的训练过程,认识相关参数的调整规则,并能够完成训练算法的 Python 程序。(数字化学习与创新)
- 通过体验多层感知机与人工神经网络,体会人工智能的魅力。(信息社会意识)

教学重点：

- 理解感知机的训练过程(感知机学习算法)；
- 学习神经网络的概念；
- 理解神经网络的训练算法(梯度下降法)。

二、内容分析与建议

1. 内容分析

本小节是教科书第三章“漫游深度学习的世界”中第二节的内容。在第一节学习单层感知机的基础上，本节课程继续开展对于多层感知机以及人工神经网络的探究。本小节首先通过对异或门电路的问题探究，引出多层感知机的结构；然后，从多层感知机进行推广，引入神经网络。本阶段，学生的实践任务主要有三个：其一是基于多层感知机模拟实现异或运算；其二是利用 Python 基于感知机学习规则（一种反向传播算法）实现多层感知机的训练；其三是利用 Python 实现基于梯度下降的神经网络训练过程。本节主要知识框图如图 3-5 所示。



图 3-5 主要知识框图

对于任务一，可以按照对比分析、运算理解、算法设计、编程实现的过程进行。首先，教师应基于与、或、与非运算引领学生理解单层感知机只能解决线性可分的问题。接着，在认识异或运算是非线性可分的基础上，引出多层感知机的结构。然后，利用流程图分析算法过程，组织学生尝试完成异或运算的程序。最后，展示学生的程序作品。

对于任务二，教师应重点关注多层感知机训练过程的理解。教师可组织学生思考如何训练感知机，引出感知机学习规则（一种反向传播算法）。接着，通过组织学生讨论分享自己的想法，引领学生形成感知机训练的步骤与过程。其次，可以借助流程图的形式呈现算法的过程思想。在此项任务中，教师可通过提供半成品的程序，带领学生分析代码，让学生尝试补全语句，帮助学生理解算法过程。

对于任务三，理解损失函数的意义以及梯度下降法的过程是关键。教师可借助数学公式帮助学生理解损失函数，引领学生学习问题转换，明白损失函数的意义，求最优权重即是寻找损失函数值小的一组权重。接着，教师可借助逐步的动画演示分析梯度下降法的过程，理解学习率参数。最后，教师可提供示例程序，组织学生阅读分析程序，通过调节参数观察神经网络不同的学习性能。

2. 教学建议

本节涉及到算法理解的内容较多，教师要根据学生情况，利用流程图、实际案例等讲解算法过程，对程序进行讲解与分析，同时可通过对数学函数的分析，引导学生理解多层感知机。教学中重要的是帮助学生理解感知机与神经网络的训练过程，学习深度学习如

何基于数据进行学习更新,感受人工智能的学习能力。

本节中涉及到的程序代码量较大,教学中要鼓励学生勇于、敢于动手实践,消除对于程序的畏难情绪。教师可以给出示例程序,组织学生分析理解代码语句,让学生运行程序观察结果,并尝试修改其中的学习率等参数体验算法原理。

在学习训练算法的初期,学生编写完整的程序是一件较为困难的事情。且由于课堂时间有限,教师也可采用提供待补全的半成品程序。例如,在实现感知机的训练算法时,教师可以向学生提供半成品的代码,让学生在理解算法原理的基础上尝试补全其中的重要语句。

本节中实践性活动较多,知识点较难,编程任务较重。建议教师在教学中要及时关注学生的实践情况,及时梳理学生遇到的困难与问题,提供引导帮助。同时,建议采用小组合作的形式进行实践活动,实现生生互助,合作探究。

三、习题分析与提示

习题描述

见教科书第 71 页的作业练习“使用随机梯度下降法实现神经网络的学习”。

习题解题思路

请下载资源包中的对应程序,或者以下面的程序进行参数调整,进一步了解算法中的学习率对于训练结果的影响。在随机梯度下降法中,学习率是一个超参数,决定着迭代点移动方向的步长。经过分析,可以发现移动的步长过大或过小都不合适,过大有可能错过最优点,过小则有可能造成算法效率下降。可以尝试将学习率(以下代码中的 lr)设置为 1 或者 0.01,观察寻找最小值点的路径图有什么区别。

提示: 学习率为 1,节点分布在第二、第四象限,因为步长过大,直接越过了最低点;

学习率为 0.1,节点分布在第二象限,但是过于密集,这是因为移动步长太小。

```
# 示例程序: 用梯度下降法求解函数  $y = x_1^2 + x_2^2$  最小值
import numpy as npy
import matplotlib.pyplot as plt
# 求梯度
def numerical_gradient(f, x):
    h= 1e- 4
    grad= np.zeros_like(x)
    for i in range(x.size):
        tmp_val= x[i]
        x[i]= tmp_val+ h
        fxh1= f(x)
        x[i]= tmp_val- h
        fxh2= f(x)
        grad[i]= (fxh1- fxh2)/(2* h)
        x[i]= tmp_val
    return grad

def gradient_descent(f, init_x, lr, step= 1000):
    x= init_x
    for i in range(step):
        grad= numerical_gradient(f, x)
        x= - lr* grad
    return x
```

```

def gradient_descent(f, init_x, lr, step):
    x = init_x
    x_history = []
    for i in range(step):
        x_history.append(x.copy())
        grad = numerical_gradient(f, x)
        x -= lr * grad
    return x, np.array(x_history)

def function_2(x):
    return x[0]**2 + x[1]**2

step_num = 100
lr = 0.2
init_x = np.array([-3.0, 4.0])
x, x_history = gradient_descent(function_2, init_x, lr, step_num)
print('最后的点为', x)

plt.plot([-5, 5], [0, 0], '--b')
x1, y1 = np.arange(-4, 4, 0.1)
x1, y1 = np.meshgrid(x1, y1)
for i in range(6):
    plt.contour(x1, y1, x1**2 + y1**2, [3*i],)
plt.xlim(-3.5, 3.5)
plt.ylim(-4.5, 4.5)
plt.show()

```

四、核心概念精解与技术工具介绍

1. 核心概念精解

(1) 多层感知机的基本结构

多层感知机由单层感知机推广而来,是一种前馈全连接人工神经网络模型。多层感知机的特点就在于拥有多个层次,其示例结构如图 3-6 所示。最左边为输入层,中间为隐藏层,最右边为输出层。其中,多层感知机至少有一个隐藏层,但没有规定隐藏层的上限,可根据问题需求进行设计;输出层的神经元个数也没有限制,可以存在多个。例如,图 3-6 所展示的就是一个二层感知机(输入层不算,因为仅用于输入数据),第 0 层为输入层,第 1 层为隐藏层,第 2 层为输出层。

(2) 多层感知机的训练过程

对于简单问题,多层感知机模型中的权重、偏置值等参数可以根据人为推导进行预设指定,但遇到复杂问题、大量参数时,利用人工设定参数的方法是无法解决的。如何基于数据寻找参数的值,训练多层感知机的学习过程,对解决实际问题是至关重要的。

多层感知机可基于感知机学习规则(一种反向传播算法)实现训练过程。因为多层感知机是有监督的训练网络,其训练数据的输入有标签,因此我们可以将实际输出值与数据标签进行对比找出被错误分类的数据,再根据这些误分数据调整参数,直到训练集中没有误分的数据。具体步骤如表 3-9 所示。

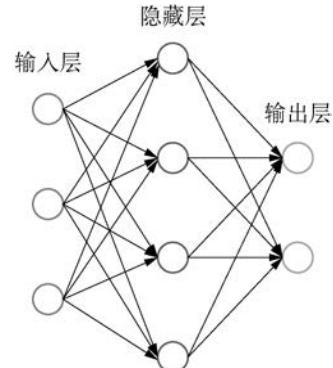


图 3-6 二层感知机

表 3-9 多层感知机训练过程

环节	步骤	说 明
初始化	第 0 步	确定所有权重和偏置值的初始值:0 或小随机数
	第 1 步	加权求和: $x_1w_1 + \dots + x_nw_n - \theta$
	第 2 步	返回状态: 基于激活函数 $y = f(x_1w_1 + \dots + x_nw_n - \theta)$, 返回二元值 y (激活或抑制)
循环直至参数符合需求(即训练集中没有误分数据)	第 3 步	调整参数: 根据错误分类的数据,按照如下规则调整参数: $w_i \leftarrow w_i + \eta y x_i; \quad \theta \leftarrow \theta_i + \eta y$ 。 其中, η 是学习率,用来表示每一次更新参数的程度大小。 η 需要人为确定为某个值,例如 0.01,再视学习过程的需要动态调整
	第 4 步	更新参数:更新原来的权重和偏置值

(3) 神经网络的训练学习

神经网络是对多层感知机的推广,依然是基于数据学习寻找合适的参数,但其激活函数不再限制为阶跃函数。神经网络中更常使用的激活函数是 S 形函数,因为其连续性(阶跃函数存在跳跃间断点)可以采用更强大的分析手段。

① 损失函数

多层感知机学习算法是减少误分数据的过程,那么如何表示误分的程度呢。神经网络中引入损失函数来表示预测值(实际输出值)与真实值(输入数据的标签)之间的差距。当损失函数的取值越大,意味着神经网络错误的程度越大;损失函数的取值越小,意味着神经网络错误的程度越小。因此,神经网络训练的目标就变成确立使得损失函数取值最小的一组参数。原则上,损失函数可以是任何函数,但通常会选择较有意义的函数,例如均方误差或交叉熵误差等。

② 梯度下降法

梯度下降法是神经网络的一种著名训练算法。梯度下降法的第一个阶段是选择一个起始点(通常是随机选取)。然后让损失函数从当前位置的取值沿着下降速度最快的方向移动一小段。接着,通过计算新位置的损失函数取值,然后再沿着下降速度最快的方向继续移动。通过以上重复迭代的过程逐步减小损失函数的取值。

其中,梯度用来表示函数在某一点处变化最快的方向。其中,正梯度代表增长速度最快的方向;负梯度代表下降速度最快的方向。在利用梯度确定移动方向后,还需要设置学习率来调节移动的距离。移动距离过大可能会导致错过最小取值点,移动距离过小可能会导致效率过低。因此,学习率不宜过大或过小,通常设置为 0.01 或 0.005 等。学习率与权重、偏置值不同,是需要人为事先确定为某个值,再根据学习过程进行调整。

2. 技术工具介绍

(1) 函数图像绘制

绘制函数图像的软件和方法有很多,既可以选用简单好操作的在线网站进行绘制,也可以利用专门的函数绘制软件或编写程序进行图像绘制,教师可根据需要进行选择。

(2) 设计流程图

流程图是辅助分析问题解决方法、表示算法思路的有效手段。用于设计制作流程图的软件和工具也有很多,既可以选择文本编辑软件中的自选图像进行绘制,也可以利用思维导图等软件进行绘制,还可以利用在线流程图绘制平台。

(3) Python 类的定义和使用

类提供了一种组合数据和功能的方法。创建一个新类意味着创建一个新的对象类型,从而允许创建一个该类型的新实例。创建一个类之后,可以通过类名访问其属性。Python 类定义的方法如下:

```
class <类名>:  
    <语句>
```

(4) Python 模块

Python 的第三方库可通过 pip 工具进行安装(pip install 库名),通过 import 进行导入(import 库名)。NumPy(Numerical Python)是 Python 语言的一个扩展程序库,支持大量的维度数组与矩阵运算,此外也针对数组运算提供大量的数学函数库。Matplotlib 是 Python 的绘图库。它可与 NumPy 一起使用,提供了一种有效的 Matlab 开源替代方案。它也可以和图形工具包一起使用,如 PyQt 和 wxPython。

五、教学参考资源

■ 参考书

塔里克·拉希德. Python 神经网络编程[M]. 林赐,译. 北京:人民邮电出版社,2018.

■ 参考论文

黄从智,白焰. 智能控制课程中感知器教案设计与教学实践[J]. 中国电力教育,2014(014).

六、教学参考案例

■ 参考案例 1

多层感知机的建立与学习

北京市第八中学 马 征

(1课时)

1. 教学目标

- 通过对样例思考,知道神经网络与多层次感知机的异同;(信息意识)
- 通过对比分析,认识损失函数的作用,知道常用的损失函数;(计算思维、数字化学习与创新)
- 通过逐步的分析探究,理解梯度下降法的原理与过程,认识梯度与学习率;(计算思维)
- 通过案例实践,体验基于 Python 编写的神经网络,通过修改参数理解其对神经网

络学习性能的影响。(计算思维、数字化学习与创新)

2. 教学对象分析

本节课的教学对象为高一年级的学生,高中生已具有一定的自学能力与问题解决能力,愿意探索尝试。通过前面课程的学习,学生已经理解了多层感知机的训练算法,了解了有监督学习与无监督学习,掌握了 Python 语言的基本知识,能进行简单的程序编写。

3. 教学重难点

教学重点:理解损失函数的意义以及梯度下降法的原理。

教学难点:理解梯度与学习率;阅读神经网络程序,并进行参数调整。

4. 教学方法

讲授法、任务驱动法、小组合作学习。

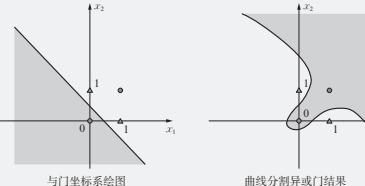
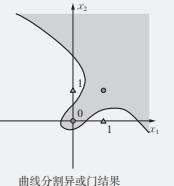
5. 教学工具

计算机、网络、PPT、Python 编译环境。

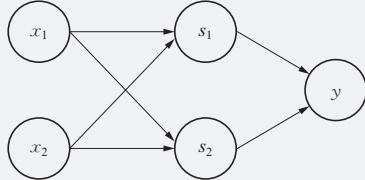
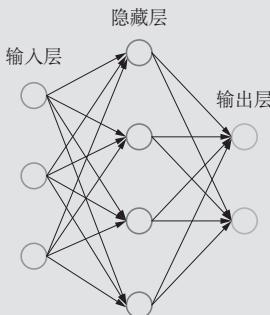
6. 教学过程设计

教学过程设计如表 3-10 所示。

表 3-10 教学过程设计

教学环节	教学内容	学生活动	设计意图
回顾探究	<p>1. 单层感知机模型及其应用。</p> $o = f(w_1x_1 + \dots + w_nx_n - \theta)$ $= \begin{cases} 0, & w_1x_1 + \dots + w_nx_n \leq \theta, \\ 1, & w_1x_1 + \dots + w_nx_n > \theta. \end{cases}$ <p>2. 分析探究:单层感知机与异或。</p> <p>在上节课的学习中,我们学习了单层感知机的模型及应用,并尝试利用单层感知机实现了与门、或门、与非门电路。在四种逻辑运算中还剩最后一种异或运算,你认为单层感知机可以实现异或运算吗?请小组合作探究,说出对这个问题的思考。</p> <p>3. 小组讨论,分析问题。</p> <p>4. 组织回答,分享交流。</p> <p>5. 归纳总结,引入主题。</p> <p>我们通过函数图像可以发现,在与门、或门、与非门运算中,我们可以找到一条直线来分开有不同输出值的点。然而在异或门图像中,我们无法通过一条直线将不同输出值的点分开,而只能利用曲线。</p> <p> 与门坐标系绘图</p> <p> 曲线分割异或门结果</p> <p>其次,通过单层感知机的函数表达式,我们可以发现它是一个多元一次函数,那么函数图像只能是直线、平面或者高维的超平面。这意味着单层感知机只能解决可用线性分开的数据。</p> <p>6. 提问思考,引入主题。</p> <p>那么,如何才能实现异或门电路呢?</p>	回顾思考,理解学习	通过分析单层感知机的函数图像和表达式,帮助学生理解单层感知机只能解决线性可分的问题。 借助函数图像,引领学生理解异或是非线性可分的。通过提出问题,引导学生思考如何实现异或门电路,从而引出本课主题

续表

教学环节	教学内容	学生活动	设计意图
探究实践	<p>探究主题:在感知机的基础上,如何实现异或门电路?</p> <p>探究活动:</p> <ol style="list-style-type: none"> 小组讨论,思考有何种解决方法。 组织有想法的学生表达自己的思考。 师生共同分析,理解问题转换。 <p>异或无法用单层感知机实现,但是与、或、与非这三种都可以实现。那么,异或能不能拆分成其他三种运算的组合呢?</p> <p>转化公式:$A \text{ xor } B = (A \text{ NAND } B) \text{ AND } (A \text{ OR } B)$(学生如果无法理解转换公式,可通过实例辅助讲解)</p> <p>那么计算 $x_1 \text{ xor } x_2$ 就可以转换为三个步骤:</p> <ol style="list-style-type: none"> $s_1 = x_1 \text{ NAND } x_2$ $s_2 = x_1 \text{ OR } x_2$ $y = s_1 \text{ AND } s_2$  <ol style="list-style-type: none"> 小组合作,编写完成异或门的 Python 程序。 展示学生程序,分析运行结果 	<p>思考问题,理解任务,编写代码,展示程序</p> <p>通过学生分享解决方法,师生共同分析异或运算的拆分步骤,实现问题转换。</p> <p>基于小组合作,编写程序</p>	
新知讲授	<p>1. 多层感知机的结构。</p> <p>像刚才表示异或门的网络就是多层感知机网络。多层感知机由单层感知机推广而来,拥有多个层次。最左边为输入层,中间为隐藏层,最右边为输出层。</p>  <ol style="list-style-type: none"> 输入层:n 维输入。 隐藏层:至少一层,没有上限。 输出层:数量没有限制。 层数:隐藏层 + 输出层。 <p>2. 利用学案进行小测检验,帮助学生及时巩固所学新知。</p> <ol style="list-style-type: none"> 回答学案中多层感知机的层数、输入与输出数量。 请学生分享答案 	<p>学习理解,完成学案</p> <p>基于异或门电路引入多层感知机的结构组成。</p> <p>通过学案中的小测帮助学生检查自己的理解程度</p>	

续表

教学环节	教学内容	学生活动	设计意图															
问题引导	<p>我们前面用于模拟逻辑电路的单层和多层感知机,其权重都是事先指定好的。这仅对简单问题才有可能,现实中的神经网络通常涉及成千上万个参数,甚至可以达到上亿,人工决定参数值的方法是行不通的。</p> <p>在上一个单元,我们学习过人工智能可以从数据中学习规律。多层感知机也可以从数据中学习,从而自行确定自己的参数。那么多层感知机是如何学习的呢?</p>	学习思考	结合前面所学的知识,引导问题思考															
自学探究	<ol style="list-style-type: none"> 阅读教科书第 68 页最上方内容,尝试理解多层感知机的训练过程。 教师提出问题,学生基于自学进行回答。 <ol style="list-style-type: none"> 多层感知机训练依据的是什么? 多层感知机参数如何调整? 教师讲解多层感知机的训练过程。 <table border="1"> <thead> <tr> <th>环节</th><th>步骤</th><th>说明</th></tr> </thead> <tbody> <tr> <td>初始化</td><td>第 0 步</td><td>确定所有权重和偏置值的初始值:0 或小随机数</td></tr> <tr> <td rowspan="2">循环直至参数符合需求(即训练集中没有误数据)</td><td>第 1 步</td><td>加权求和: $x_1w_1 + \dots + x_nw_n - \theta$</td></tr> <tr> <td>第 2 步</td><td>基于激活函数 $y = f(x_1w_1 + \dots + x_nw_n - \theta)$, 返回二元值 y(激活或抑制)</td></tr> <tr> <td rowspan="2">符合需求(即训练集中没有误数据)</td><td>第 3 步</td><td>根据错误分类的数据,按照如下规则调整参数: $w_i \leftarrow w_i + \eta y x_i$, $\theta \leftarrow \theta + \eta y$。 其中, η 是学习率,用来表示每一次更新参数的程度大小。η 需要人为确定为某个值,例如 0.01,再视学习过程的需要动态调整</td></tr> <tr> <td>第 4 步</td><td>更新参数: 更新原来的权重和偏置值</td></tr> </tbody> </table>	环节	步骤	说明	初始化	第 0 步	确定所有权重和偏置值的初始值:0 或小随机数	循环直至参数符合需求(即训练集中没有误数据)	第 1 步	加权求和: $x_1w_1 + \dots + x_nw_n - \theta$	第 2 步	基于激活函数 $y = f(x_1w_1 + \dots + x_nw_n - \theta)$, 返回二元值 y(激活或抑制)	符合需求(即训练集中没有误数据)	第 3 步	根据错误分类的数据,按照如下规则调整参数: $w_i \leftarrow w_i + \eta y x_i$, $\theta \leftarrow \theta + \eta y$ 。 其中, η 是学习率,用来表示每一次更新参数的程度大小。 η 需要人为确定为某个值,例如 0.01,再视学习过程的需要动态调整	第 4 步	更新参数: 更新原来的权重和偏置值	<p>通过阅读教科书内容,学生自学新知。</p> <p>阅读教科书,学习理解。思考讨论,回答问题</p> <p>通过详细讲解帮助学生系统梳理训练方法</p>
环节	步骤	说明																
初始化	第 0 步	确定所有权重和偏置值的初始值:0 或小随机数																
循环直至参数符合需求(即训练集中没有误数据)	第 1 步	加权求和: $x_1w_1 + \dots + x_nw_n - \theta$																
	第 2 步	基于激活函数 $y = f(x_1w_1 + \dots + x_nw_n - \theta)$, 返回二元值 y(激活或抑制)																
符合需求(即训练集中没有误数据)	第 3 步	根据错误分类的数据,按照如下规则调整参数: $w_i \leftarrow w_i + \eta y x_i$, $\theta \leftarrow \theta + \eta y$ 。 其中, η 是学习率,用来表示每一次更新参数的程度大小。 η 需要人为确定为某个值,例如 0.01,再视学习过程的需要动态调整																
	第 4 步	更新参数: 更新原来的权重和偏置值																
项目实践	<p>项目任务:训练多层感知机。</p> <ol style="list-style-type: none"> 分析任务:训练多层感知机的程序需要哪些部分? <ol style="list-style-type: none"> 多层感知机模型的建立(初始化、训练算法)。 数据集的建立:训练集与测试集。 感知机的训练。 结果呈现。 提供半成品程序,带领学生阅读代码,理解程序。 学生尝试补全训练算法中的参数调整部分。 展示程序,分享思路 	思考讨论,阅读代码,完成程序	<p>通过半成品程序帮助学生理解训练算法的主要部分。</p> <p>补全主要代码帮助学生理解训练算法的主要思想</p>															
课堂小结	<ol style="list-style-type: none"> 多层感知机的结构。 多层感知机的训练过程 	思考回顾	促进学生对新知的内化吸收															

参考案例 2

多层感知机的建立与学习

北京市第八中学 蓝 征

(1课时)

1. 教学目标

- 通过样例思考,理解单层感知机只可处理线性可分问题的局限性。(信息意识、计算思维)
- 通过对比分析,知道多层感知机的基本结构。(计算思维)
- 通过案例实践,能够利用多层感知机实现异或逻辑运算,并利用 Python 编写程序。(计算思维、数字化学习与创新)
- 通过实践应用,理解多层感知机的训练过程,认识相关参数的调整规则,并能够完成训练算法的 Python 程序。(计算思维、数字化学习与创新)

2. 教学对象分析

本节课的教学对象为高一年级的学生,高中生已经具有一定计算思维与编程能力。在前面课程的学习中,学生已经学习过单层感知机的基本结构,利用 Python 完成感知机与门、或门、与非门电路的程序实现,了解了异或运算,知道机器学习的基本过程,为本节课引入多层感知机与训练算法奠定了基础。

3. 教学重难点

教学重点:理解多层感知机的基本结构与训练过程。

教学难点:完成多层感知机的训练程序。

4. 教学方法

讲授法、演示法、任务驱动法、小组合作学习。

5. 教学工具

计算机、网络、PPT、Python 编译环境。

6. 教学过程设计

教学过程设计如表 3-11 所示。

表 3-11 教学过程设计

教学环节	教学内容	学生活动	设计意图
对比旧知 导入新知	上节课我们认识了多层感知机的模型及训练过程。多层感知机解决了之前无法解决非线性分类的缺陷,同时更多的层数也让网络更能够刻画现实世界中的复杂情形。 多层感知机是一种经典的神经网络,除此之外还有其他种类的神经网络,不同的神经网络可以用来解决不同的问题。那么,神经网络到底是什么呢?学习多层感知机的你能够尝试给神经网络下一个定义吗?	回顾思考,填写学案。 阅读教科书,理解学习	基于多层感知机进一步思考神经网络的概念。 通过阅读教科书学习比较两者的区别,并完成学案的相关内容,帮助学生导入新知

续表

教学环节	教学内容	学生活动	设计意图
	<p>1. 学生讨论思考,在学案相关地方填写神经网络的概念。</p> <p>2. 教师浏览学生的答案,请学生进行相关分享。</p> <p>3. 学生阅读教科书第 68 页第二部分的前三段,学习神经网络与多层次感知机的异同点,并在学案相关地方填写。</p> <p>4. 学生分享想法,教师归纳总结。</p> <p>(1) 神经网络是对多层次感知机的推广。</p> <p>(2) 相同点:从数据中学习合适的参数。</p> <p>(3) 不同点:激活函数不限制在阶跃函数,通常使用 S 形函数</p>		
任务探究 思考分析	<p>任务探究:神经网络如何进行训练学习?</p> <p>1. 引导回顾。 多层次感知机是如何进行训练的?其依据什么调整参数?</p> <p>2. 教师梳理。 多层次感知机学习算法本质是减少训练集中误分数据的过程,其依据的是每一次被误分的数据性质去调整参数。</p> <p>3. 新知讲授——损失函数的意义。 多层次感知机学习的依据是判断误分的数据,期望能找到合适的参数。那么如何判断数据被误分的程度呢? 神经网络中引入损失函数来表示预测值(实际输出值)与真实值(输入数据的标签)之间的差距。当损失函数的取值越大,意味着神经网络错误的程度越大;损失函数的取值越小,意味着神经网络错误的程度越小。</p> <p>4. 提问思考:神经网络的训练目的是求取最合适的参数,引入损失函数后,最合适的参数意味着什么?</p> <p>5. 组织学生思考讨论并回答。</p> <p>6. 提问思考:你认为损失函数可以是什么?</p> <p>7. 小组讨论,提出自己认为可行的损失函数。</p> <p>8. 新知讲授——常用损失函数。 (1) 均方误差:各数据偏离真实值的距离平方和的平均数。</p>	<p>思考讨论,回答问题,学习理解。</p> <p>任务计算,完成学案</p>	<p>利用已有知识经验,从多层次感知机的训练过程推广到神经网络的学习方法,提出衡量数据误分程度的函数——损失函数。</p> <p>以均方误差为例讲解常用损失函数,帮助学生理解损失函数。</p> <p>学生还没有交叉熵的基础,所以简单提及即可</p>

$$E = \frac{1}{2} \sum_{i=1}^n (y_i - t_i)^2.$$

根据已学知识,计算学案中示例题目的均方误差。

(2) 交叉熵误差。

$$E = - \sum_{i=1}^n t_i \log y_i$$

续表

教学环节	教学内容	学生活动	设计意图
引导过渡	现在,我们可以通过损失函数知道每一次训练过程中,数据被误分的程度,拥有了定量的衡量指标。那么,有了这样的指标依据,我们该如何调整参数呢?	聆听思考	思考训练中调参的方法
合作探究 问题引导 新知学习	<p>1. 学生小组合作,学习《神经网络训练与梯度下降法》微课里的内容,了解梯度下降法。</p> <p>2. 学生小组合作,填写完成学案中的问题。</p> <p>(1) 梯度下降法的最终目的是什么?</p> <p>(2) 你认为梯度下降法有几个主要阶段?</p> <p>(3) 每个阶段分别是什么?</p> <p>(4) 梯度有方向吗?</p> <p>(5) 如果有方向,正梯度和负梯度哪个代表下降速度最快的方向?</p> <p>(6) 梯度的方向决定了函数取值变化的方向,什么决定了“下一步”的距离?</p> <p>(7) 如何控制函数变化的调节距离?</p> <p>(8) 第(7)问的答案是人为设置还是机器自动学习?</p> <p>(9) 梯度下降法有哪几种类别?</p> <p>3. 组织不同小组分享自己对于问题的想法。</p> <p>4. 教师进行归纳总结。</p> <p>(1) 梯度:负梯度代表函数取值下降速度最快的方向;大小决定了移动的距离。</p> <p>(2) 学习率:调节损失函数取值的变化幅度,人为设置。</p> <p>(3) 梯度下降法的步骤。</p> <p>(4) 梯度下降法的类别</p>	<p>观看微课,小组合作,思考分析,填写学案。</p> <p>分享自己的想法,聆听其他同学的想法</p>	梯度下降法对于学生来说是一个较难理解的部分,借助微课以及学案中的问题能够帮助学生一步步梳理梯度下降法的步骤,理解梯度和学习率
应用实践	<p>1. 观看随机梯度下降法的示例动画,进一步从过程与结果理解梯度下降法。</p> <p>2. 体验随机梯度下降法。</p> <p>(1) 阅读示例代码,小组学习分析。</p> <p>(2) 教师带领学生分析代码的主要部分。</p> <p>(3) 运行程序,查看结果。</p> <p>(4) 调整其中的参数,观察其对神经网络学习性能的影响</p>	观看动画,阅读代码,分析程序,调整参数	通过示例程序引导学生学习如何实现梯度下降法。通过尝试调整参数,理解其对神经网络训练性能的影响
课堂小结 新知巩固	<p>教师引导学生回忆所学新知:</p> <p>1. 神经网络与多层感知机的异同。</p> <p>2. 神经网络的训练学习。</p> <p>(1) 损失函数。</p> <p>(2) 梯度下降法。</p>	回顾思考,回答问题	系统梳理所学新知,深化学生理解

第三节

深度学习

一、教学目标与重点

教学目标:

- 通过对神经网络结构分析,了解深度学习的概念和特点。(信息意识)
- 了解常见数据集的平台,通过实践任务,了解 MNIST 数据集的特点。(数字化学习与创新)
- 了解机器学习应用的开发流程。(计算思维、数字化学习与创新)
- 了解常见的深度学习框架 TensorFlow、Theano、Keras、PaddlePaddle。(计算思维、数字化学习与创新)
- 体验 TensorFlow、Keras、PyTorch 等框架环境安装。(数字化学习与创新)
- 通过对数据集来源、数据版权、数据隐私条款等的介绍,加强数据隐私安全意识和信息社会责任。(信息意识、信息社会责任)
- 通过回顾图像数字化过程、分析 MNIST 数据集的处理过程,提升计算思维能力。(计算思维)

教学重点:

- 深度学习的概念和特点;
- 数据集的概念和划分;
- MNIST 数据集的特点和使用方法。

二、内容分析与建议

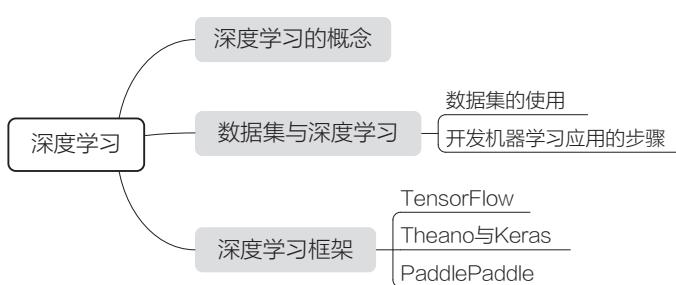


图 3-7 主要知识框图

1. 内容分析

本节在神经网络的基础上介绍了深度学习的概念、深度学习常见的数据集和使用方法,以及常见的深度学习框架。本节内容的知识框图如图 3-7 所示。

2. 教学建议

本节的学习内容包含深度学习的概念、数据集与深度学习、常见的深度学习框架三个部分。

在介绍深度学习的概念时,可通过人脸识别或者其他学生熟悉的具体应用实例引导学生了解深度学习技术,在人工神经网络基础上理解深度神经网络的结构和特点,可通过对比法,引导学生找出深度神经网络的新特点。在解决深度神经网络连接数计算量剧增的问题方面,教科书并没有展开内容,教师可寻找一些拓展阅读材料或视频帮助学生理解 GPU 计算特点以及 GPU 是如何帮助优化深度学习计算的。

在数据集与深度学习部分,本节介绍了数据集的概念,并介绍了广泛使用的数据集平台。这里建议设计活动,让学生通过任务形式从某数据集平台找到教师要求的数据集,在实践中学习查找所需数据集的基本方法。

本节安排了一个体验 MNIST 手写数字数据集的实践活动。程序中灰度图和数字文本的关系相对抽象,可先从灰度图的数字化过程引入,以视频形式介绍图像数字化过程,帮助学生理解图像如何转换为纯数字。这样学生比较容易理解活动里 csv 数据集文件中的数据是如何表示出图像的。Matplotlib 库的基本用法也可以采用拓展学习资料形式提供给学生。如果学生具有 Matplotlib 的基本使用知识,可将活动中数据集图像显示的环节留给学生探究实现。如果学生对图像数字化原理有基础知识,教师可展示或让学生体验生成一个手写数字图片、处理为合适大小、灰度化、保存为 csv 格式数据的过程,进一步拓展数据采集、数据清洗、数据整理等概念。

本节还介绍了开发机器学习应用的基本步骤。学生需要了解一般的机器学习应用开发的流程。可结合教科书提供的 MNIST 数据集训练神经网络模型的过程来讲解,如果条件允许,可利用 tensorflow.js 框架实现一个网页应用部署。

在深度学习框架部分,本节介绍了常见的几种深度学习框架:TensorFlow、Theano、Keras、PaddlePaddle。其中对 TensorFlow 的安装做了重点介绍。可组织学生体验安装和检测的过程。TensorFlow.js 在网页上即可实现深度学习应用,具体应用模型可提供现成的版本,让学生通过修改参数训练模型的实践初步体验框架的便利性。

技术发展的速度很快,我国人工智能行业的迅猛发展也催生了相关数据行业的发展,教师可多关注相关行业新闻,搜集国内新出现的数据集平台信息和案例分享给学生。

三、习题分析与提示

■ 习题 1 描述

见教科书第 72 页体验思考“人脸识别是如何实现的?”。

■ 习题 1 参考答案

- (1) 完成提到的识别任务,需要计算机识别人脸。
- (2) 人类是抓住物体主要特征来分辨的。(可以通过猜一猜类的游戏引导学生理解)

■ 习题 2 描述

见教科书第 73 页探究活动“深度所带来的挑战”。

■ 习题 2 参考答案

- (1) $4 \times 4 + 4 \times 5 + 5 \times 6 + 6 \times 4 + 4 \times 3 + 3 \times 3 = 111$ (个)。

(2) $4 \times 4 + 4 \times 5 + 5 \times 6 + 6 \times 8 + 8 \times 4 + 4 \times 3 + 3 \times 3 = 167$ (个)。

习题 3 描述

见教科书第 74 页体验思考“为什么是 GPU, 而不是 CPU?”。

习题 3 参考答案

GPU 最初是为图形应用而开发的专用硬件。图形计算中, 众多像素点都是执行统一的简单计算, 并且需要并行执行, 而且 GPU 的内存带宽比较大。这些特点决定了 GPU 做矩阵计算效率会很高。而且 GPU 通常是多核, 比 CPU 的计算单元多。CPU 的设计是为了执行复杂的指令, 运行机制是快速地读取缓存数据并执行串行计算, 重复简单的加减计算用 CPU 算不如 GPU 快。机器学习中, 深度学习框架中的众多神经元执行矩阵运算, 这个特点和图形计算类似, 由 GPU 执行就会效率更高。

习题 4 描述

见教科书第 75 页探究活动“探究 MNIST 手写数字数据集”。

习题 4 参考答案

MNIST 手写数字识别是学习深度学习课程的“hello, world!”项目。MNIST 数据集的原始格式包含了四个部分:

Training set images: train-images-idx3-ubyte. gz (9. 9 MB, 解压后 47 MB, 包含 60,000 个样本)

Training set labels: train-labels-idx1-ubyte. gz (29 KB, 解压后 60 KB, 包含 60,000 个标签)

Test set images: t10k-images-idx3-ubyte. gz (1. 6 MB, 解压后 7. 8 MB, 包含 10,000 个样本)

Test set labels: t10k-labels-idx1-ubyte. gz (5KB, 解压后 10 KB, 包含 10,000 个标签)

另外, 网络上也可以找到被处理后的 CSV 文件存储的 MNIST 数据集。本活动是以 CSV 文件数据集作为实验对象。

(1) 导入库并显示一条数据

```
% matplotlib inline
# 导入库
import numpy as np
import matplotlib.pyplot as plt

# 打开文件
MNIST 数据文件= open('mnist_train_100.csv','r')
MNIST 数据列表= MNIST 数据文件.readlines()

# 关闭文件
MNIST 数据文件.close()

# 显示第一个元素
MNIST 数据列表[0]
```

(2) 图形化显示数据

此过程是将读取的一维数据转换为 28×28 的矩阵图像形式, 并使用 matplotlib 库显示。这里需要学生理解图像数字化原理。所需代码如下:

```
# 选择一个元素(一行数据),加入分隔符“,”  
选定的数据行= MNIST 数据列表[15].split(',')  
  
# 从选定的数据行中剔除第一个值(标签值)  
截取的数据部分= np.asarray(选定的数据行[1:])  
  
# 将截取的数据部分重塑成为 28×28 的二维数组  
重塑的二维图像数组= 截取的数据部分.reshape((28,28))  
plt.imshow(重塑的二维图像数组,cmap= 'Greys',interpolation= 'None')
```

■ 习题 5 描述

见教科书第 81 页项目实践“安装 TensorFlow”。

■ 习题 5 参考答案

如使用 Anaconda 环境,可打开 Anaconda Prompt 命令窗口执行安装命令,如是原生 Python 环境,则可使用 cmd 调出命令窗口安装。

安装命令: pip install tensorflow 即可安装基本的 CPU 版本。

但是可能由于访问国外网站较慢,导致超时安装失败,建议使用国内镜像安装。

如安装失败,提示无权限安装,则可能需要使用管理员权限打开命令窗口,再执行。

如使用 Anaconda 的特定虚拟环境,则需切换到该虚拟环境下,再使用 conda 命令安装:

```
conda activate 环境名  
conda install tensorflow
```

■ 习题 6 描述

见教科书第 82 页项目实践“安装 Keras”。

■ 习题 6 参考答案

和安装 TensorFlow 类似,安装 Keras 框架时,也是打开命令行窗口或 Anaconda Prompt 窗口执行 pip 命令。安装语句: pip install keras。注意如果已经安装了 TensorFlow 2.0 以上版本,TensorFlow 已内置 Keras 框架高级接口。

■ 习题 7 描述

见教科书第 84 页项目实践“PyTorch 安装”。

■ 习题 7 参考答案

在 PyTorch 官网,找到 Get Started 目录,可以找到安装指导工具。

Stable 是目前测试和支持最多的 PyTorch 版本,Preview 是最新、未经充分测试和支持的版本。选择安装命令时,可选择 Anaconda 的 conda 命令安装,也可以选择 pip 命令工具安装。

CUDA 是 NVIDIA 发明的一种并行计算平台和编程模型,它通过利用图形处理器(GPU)的处理能力,可大幅提升计算性能。选择安装 PyTorch 时,可根据显卡的指令集选择对应 CUDA 版本。

选择好合适的参数后,会出现对应的命令,复制并粘贴到执行窗口(例如 Anaconda 的 Prompt 窗口或 Windows 系统的 cmd 窗口)中即可完成安装。

四、核心概念精解与技术工具介绍

1. 概念精解

(1) 张量

TensorFlow 里面的 Tensor, 也就是“张量”。那么张量到底是什么? 为什么会流动 (Flow)? 我们可以理解张量是一个数学容器。这个容器里装着各种各样的数字。0 维张量就是单个数字, 我们可以理解为一滴水。1 维张量就是一组数字, 也可以理解为一个数组, 比如 $[1, 2, 3]$ 。1 维张量只有一个坐标轴方向, 像一条线。1 维张量也叫作向量。2 维张量, 我们可以理解为一个矩阵, 也可以理解为一个二维表格。这个表格由行和列组成。这样 2 维张量有两个坐标轴方向, 像一个平面。3 维张量由一组 2 维张量组成。比如 $[[1, 2, 3]]$ 。3 维张量坐标轴有 3 个, 像一个立方体。继续增加下去, 4 维张量由 3 维张量组成。继续下去, 就是 N 维张量。现实生活中的数据可以存储为不同张量类型。比如我们可以把一个事物的不同属性存储到一个维度, 用多维张量来描述一个事物。比如某股票每分钟的股价最高值、股价最低值可以组成一个 2 维张量, 如果把一天的股价保存, 就由股价最高值、股价最低值、时间三个组成一个 3 维张量, 如果同时保存多只股票的数据, 股票的价格数据可以组成 4 维张量。

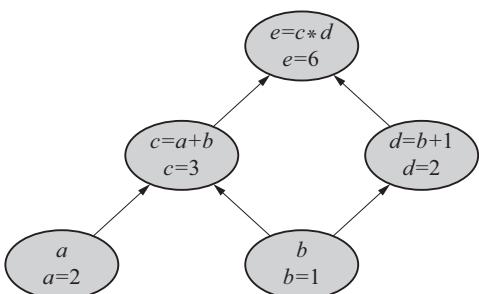


图 3-8 计算过程图形化

(2) 计算图

首先得理解图的概念。图在数据结构中是顶点 V 和边 E 构成的有限集合。图可以表示为 $G = (V, E)$, $V(G)$ 和 $E(G)$ 表示图中的顶点集合和边集合。如果边是有方向的, 那么图就是一个有向图, 否则就是无向图。那么计算图是什么呢? 这是一种将计算过程图形化的表示。也就是进行计算的每一个数据和操作都会成为图中的一个节点。例

如计算 $e = (a + b) * (b + 1)$, 在这个式子里, 先要计算 $a + b$ 得到结果 c , 再计算 $b + 1$ 得到结果 d , 然后计算 $c * d$ 的结果得到 e 。这些计算数据和过程可以用图形描述(如图 3-8 所示)。每个节点就是数据或者对数据进行的运算。边表示的就是数据流向的方向。

2. 技术工具

(1) jupyter 使用方法

在 Anaconda 安装目录下找到 jupyter notebook, 在打开的浏览器窗口即可进入 jupyter 界面。将所需要的运行的程序文件上传即可。注意如程序调用 csv 文件, 需要将同目录的 csv 目录和文件都上传。

(2) TensorFlow

TensorFlow 是一个端到端平台, 目前有两大版本, 也就是 TensorFlow1.x 和 TensorFlow2.x。该平台可以完成人工智能的全生命周期功能, 包括数据集预处理、模型构建、模型训练和调试、模型部署等过程。同时, 该平台提供了大量的学习教程和部分

通用数据集。比如 MNIST 手写字体数据集,可以满足各个层次开发者的需求。使用 TensorFlow 可以轻松、快速搭建训练模型,还可以根据开发需求选择合适的级别,比如底层 TensorFlow API 开发或者高层 Keras API 封装开发。

(3) TensorFlow 安装方法

在 Anaconda 环境下使用 pip 安装。在 Anaconda Prompt 中输入 pip install TensorFlow。

在 python3 原生环境下使用 pip 安装。在 shell 中输入 python3 -m pip install -U tensorflow。

(4) TensorFlow 数据集格式

深度学习与大数据更是相辅相成,在使用 TensorFlow 构建深度学习模型的时候,可能会涉及海量的数据,可能会用到数 G、T 甚至 P 级别的训练数据,很显然,要将如此庞大的数据一次性加载进内存,当前的硬件条件还远远达不到。所以 TensorFlow 框架优化了自己的数据集格式和读取方法。

① TensorFlow 读取数据的方式

对于深度学习而言,因为数据量庞大,在提高运算能力的同时,更高效的处理数据 I/O 对于提高整体的性能也非常重要。在使用 TensorFlow 训练模型的时候,有三种数据加载的方式:

使用 Python 代码为 TensorFlow 提供数据;预先加载数据,将需要训练的数据以变量的形式预先保存在计算机的内存中;利用管道从文件中读取数据。

对于数据较小的情况,直接将数据加载到计算机内存,然后每次取一个 batch 放进网络里面加以训练,但是对于大数据而言,一方面如果直接全部将数据放进内存肯定不可能;另一方面,我们可以每次需要多少数据就从硬盘中读取多少,但是这样做的后果就是频繁的 I/O 操作,使得执行效率大打折扣。

② 小数据的常用数据格式

对于比较小的数据,我们可以直接加载进内存,对于这种级别的数量,常用的一些数据格式有以下几种:

CSV 格式:以英文逗号间隔数据的文本文件。

NPZ 格式:Numpy 的数据保存格式。

PKL:Python 的序列化保存格式。

HDF:以 HDF5 为最新的系列。

③ 大数据的专用数据格式

对于大数据而言,TensorFlow 推荐使用自带的 TFRecords 文件。TFRecords 文件是以二进制进行存储的,适合以串行的方式读取大批量的数据。读取 TFRecord 格式数据时,首先需要将 TFRecord 文件作为参数,创建一个输入队列。一部分数据被处理时,另一部分数据会自动进入队列。数据的读取过程和图形的计算是独立进行的。因此训练神经网络时,队列里会有充足的数据,如需要也可以使用多线程加速数据的读取。

TFRecord 文件中的数据是通过 tf.train.Example Protocol Buffer 的格式存储的,也就是以键值对形式存储的。键名为字符串,值可以是字节列表(BytesList)、浮点数列表(FloatList)、整型列表(Int64List)等。

五、教学参考资源

参考书

1. 辛大齐. 深度学习实战——基于 TensorFlow 2.0 的人工智能开发应用[M]. 北京: 中国水利水电出版社, 2020.
2. Aurélien Géron. 机器学习实战：基于 Scikit-Learn、Keras 和 TensorFlow [M]. 宋能辉, 李娴, 译. 北京: 机械工业出版社, 2020.
3. 涌井良幸, 涌井贞美. 深度学习的数学[M]. 杨瑞龙, 译. 北京: 人民邮电出版社, 2019.

六、教学参考案例

参考案例

深度学习与数据集

北京市第三十五中学 李娟
(2课时)

1. 教学目标

- 通过体验活动,了解深度学习的概念。(信息意识、数字化学习与创新)
- 通过对具体实例的分析、探究及自主学习等活动,理解深度神经网络的特点。(信息意识、计算思维)
- 通过案例分析等学习方式,理解数据集的概念,了解常见的数据集平台。(信息意识、数字化学习与创新)
- 通过探究实践活动等方式,熟悉对数据集的简单操作,知道数据对于深度学习的重要性。(计算思维、数字化学习与创新、信息社会责任)

2. 教学对象分析

本节课的教学对象为高一年级的学生,学生在日常生活中已经广泛接触过人工智能的产品。在前面课程的学习中,学生已经学习过人工智能的概念、人工智能的发展历史,并且学习理解了人工神经网络的基本结构,通过程序实现了人工神经网络的基本结构,对神经网络实现数据学习、预测等过程有了直观的了解。在此基础上,理解深度神经网络相关的知识也会易于接受。

3. 教学重难点

教学重点:理解深度神经网络的结构和功能、理解数据集的概念、了解常见数据集平台。

教学难点:深度神经网络的特点。

4. 教学方法

讲授法、演示法、任务驱动法。

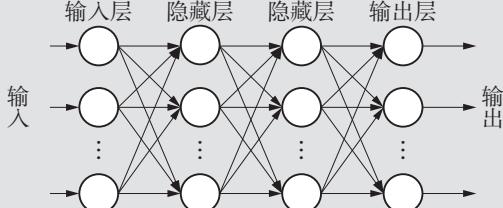
5. 教学工具

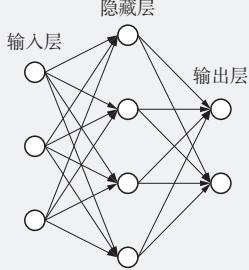
网络机房、教学课件、数据集文件、编程环境。

6. 教学过程设计

教学过程设计如表 3-12 所示。

表 3-12 教学过程设计

教学环节	教学内容	学生活动	设计意图
引入	<p>教师：网盘是非常好用的一种数据备份存储和分享资源的应用，而且很多网盘具有自动备份数据的功能，相当好用。小智使用某款网盘作为自己手机的备份空间，把拍的照片都自动备份了。现在他想找一下姥姥过去的一张照片。可是他想不起来拍照的时间了。怎么找呢？</p> <p>教师：有没有用过网盘的同学，或者大家想一想，该如何解决这个问题？</p> <p>教师：小智最终找到了通过人物来分类照片的功能。他给应用识别出的姥姥做了标记。所有包含姥姥的照片被自动分类出来，他很快就找到了自己想要的那张照片。</p> <p>教师：刚才小智使用的网盘很显然使用了人工智能技术——人脸识别技术。这类智能识别并且分类的功能在我们生活中还是很常见的，大家还能举出图像识别类的应用吗？</p>	<p>学生：根据现有的生活经验思考并给出解决方法。</p> <p>学生：根据自己的生活经验回顾图像识别技术的应用场景</p>	<p>引入网盘的智能照片分类、人脸识别技术，探讨实现方法——深度学习。</p> <p>泛化学生对深度学习技术的理解，不局限于人脸识别</p>
讲授新知	<p>教师：刚才大家的举例都是采用了图像识别技术，也为我们的生活提供了极大的便利。这类技术到底是怎样实现的呢？这就是我们今天要学习的深度学习。</p> <p>【深度学习】</p> <p>教师：深度学习其实就是深度神经网络的学习。</p>  <p>教师：我们已经学习过人工神经网络的结构，如图所示。那么神经网络，如果中间层（也就是隐藏层）是多层的，就是深度神经网络了。基于深度神经网络的学习就是深度学习。</p> <p>教师：那么问题来了，多少层才叫“深度”？这里并没有具体的限定，根据具体的网络结构而定。实际应用中，几十层甚至几百层都是可能的。并且，不同的网络结构，除了隐藏层，深度神经网络的结构也会有一定差异。</p>		
探究活动	<p>【探究活动】深度所带来的挑战。</p> <p>教师：既然深度神经网络是在普通神经网络的基础上加深——也就是多加隐藏层，那么为什么 20 世纪 90 年代的人工智能发展会遭遇瓶颈呢？我们一起来看看这个网络结构。</p>	<p>学生：思考并回答这个问题。</p> <p>学生：思考变通的办法</p>	<p>学生思考得出计算量暴增，联系之前的人工智能发展史的知识，得出计算量暴增导致的计算瓶颈问题</p>

教学环节	教学内容	学生活动	设计意图									
	 <p>教师：这是一个基本的神经网络。包含了一个隐藏层。请大家计算下，需要多少个全连接？ $(3 \times 4 + 4 \times 2 = 20)$</p> <p>教师：如果稍微“深度”一下，比如下图，包含 5 个隐藏层的话，全连接数是多少呢？请大家尝试计算，并完成任务单表格。 $(4 \times 4 + 4 \times 5 + 5 \times 6 + 6 \times 4 + 4 \times 3 + 3 \times 3 = 111)$</p>  <p>教师：如果继续深度下去，比如，在第三个隐藏层后增加 1 个包含 8 个神经元的隐藏层呢？请大家尝试计算。 $(4 \times 4 + 4 \times 5 + 5 \times 6 + 6 \times 8 + 8 \times 4 + 4 \times 3 + 3 \times 3 = 167)$</p> <p>任务单表格：</p> <table border="1" data-bbox="325 1342 870 1486"> <thead> <tr> <th></th> <th>隐藏层数</th> <th>全连接数</th> </tr> </thead> <tbody> <tr> <td>5 层</td> <td></td> <td></td> </tr> <tr> <td>6 层</td> <td></td> <td></td> </tr> </tbody> </table> <p>教师：我们发现最初的 20 个全连接网络，深度到 5 层时，全连接数达到 111 个，深度到 6 层时，全连接数达到 167 个。如果继续深度下去，隐藏的神经元个数再稍微增多一点，全连接数将急剧增多。这意味着什么？</p> <p>教师：的确，计算量的暴增，意味着需要更多的计算力。在 20 世纪 90 年代的人工智能发展寒冬时期，也是当时的计算力不足导致的。那么现在随着计算机的计算力提升，是不是我们可以忽略这个问题了？</p> <p>教师：深度神经网络如果无限制地依靠计算力，并不是高效的法子。所以科学家们也不断探究优化网络结构、减少全连接数、优化算法等手段来改进深度神经网络，并且也的确有了很大的改进。比如后面我们要讲到的卷积神经网络</p>		隐藏层数	全连接数	5 层			6 层				
	隐藏层数	全连接数										
5 层												
6 层												

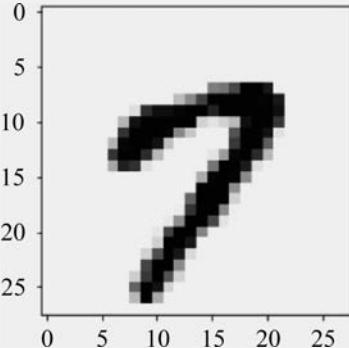
教学环节	教学内容	学生活动	设计意图
阅读学习	<p>教师：“一个好汉三个帮”，除了优化算法、网络结构等，加快计算速度也是优化的方向。这里就要感谢电子游戏的发展为人工智能技术的发展助力一把了。接触过电子游戏的同学可能了解，要想获得完美的游戏体验，拥有一台高配置游戏电脑是必需的。其中最为重要的一个硬件就是游戏显卡。显卡的内核就是 GPU。GPU 和人工智能技术有什么关联？它怎么能够优化计算速度呢？请阅读《为什么 GPU 比 CPU 更适合人工智能计算》。</p> <p>阅读材料：</p> <p>1. GPU 是图形处理单元，适合处理简单，但是大量并行的运算。GPU 就是“图形计算单元”的英文简写。在前人工智能时代，它有个名字，叫“显卡”。顾名思义，就是负责显示的卡，为了说明显卡干了什么，我们得先仔细观察一下电脑屏幕！近距离观察就会发现，这花花绿绿变化万千的屏幕，竟然是由一堆小光点组成的。对，这就是像素。一个屏幕上有多少像素呢？现在手机上的像素数量，已经普遍达到了 1920×1080，大概几百万的量级。我们平时用电脑看电影，动态效果需要每秒钟显示 25 张以上的画面，每一次画面的改变，都需要上百万的像素点一起改变它们的显示强度。这样的更新强度，需要的计算量是惊人的，特别对擅长线性计算的 CPU 来说，虽然 CPU 频率很高，但如果我们的屏幕频繁变化，CPU 就不用干别的了。尤其是，图像像素点虽然多，更新需要的计算量大，但是逻辑简单。简单的计算大量的并行和重复！如果有一百万道小学数学题和 4 道微积分待解决，现在 GPU 就如同有一万个小学生同时做题，CPU 如同有 4 个大学生同时做题，小学生虽然不会微积分，但一百万道题还是 so easy 的，给大学生做这任务量就傻眼了。</p> <p>2. 人工智能计算中有大量的简单并行计算。人工智能那么智能，为什么却说有大量简单计算呢？这就要从现在流行的机器学习理论说起。目前最有效的机器学习手段，是统计学习的分支，也就是观察大量数据，找出其中规律。那么这个“观察大量数据”的过程，就是一个简单计算大量并行的过程。一个数据一个数据地观察也不是不行，但是一批一批地观察，显然更省时间。深一点说，相当于原来算法的公式里是一个表达式计算：$y = a \cdot x$，公式里面 x 代表一条数据，y 代表计算结果。那么把一批数据排列起来（类似像素排列），就变成了一个矩阵，用 X 表示，相应的结果 y 也是一个矩阵，用 Y 表示，新的表达式就是：$Y = a \cdot X$，用矩阵计算替代普通</p>	<p>学生：阅读材料，并理解 GPU 的运算特点，进一步理解 GPU 为何能够加快深度学习时的模型训练速度</p>	<p>游戏与高科技领域的人工智能有什么关联？这个问题可以引起学生的好奇心</p>

续表

教学环节	教学内容	学生活动	设计意图
新知讲授	<p>表达式计算,能够在一次计算中,观察更多的数据 X,这种特性充分的匹配了 GPU 的计算原理。综上,为了图像阵列显示, GPU 有做阵列并行计算的能力;而为了快速观察大量数据,统计机器学习(当前人工智能)有大量的矩阵计算需求。所以用 GPU 来做有很多矩阵运算的统计学习计算,是恰到好处的应用。</p> <p>教师:GPU 的特点是有大量的核心和高速内存,擅长并行计算,对海量数据进行快速处理。所以在人工智能发展浪潮中, GPU 顺势而上</p> <p>【数据集】 教师:在此前的学习中,我们已经接触过数据集这个概念。以深度学习为代表的机器学习算法,需要大量数据作为输入,这些数据就是以数据集的形式提供给使用者的。通常,这些数据集都是各个行业、政府、企业搜集来的已经失去了时效性和价值的学习范例数据,通常作为开放数据免费提供给学习者和研究者。</p> <p>【数据集平台】 教师:下面介绍几个常见的数据集平台。</p> <ol style="list-style-type: none"> 1. MNIST 数据集。 这是一个手写数字数据集,也是深度学习案例最常用入门数据集。这个数据集通常用来进行教育和基准测试,测试一个模型能否正常使用,相当于编程语言中的第一个“hello, world!”。 MNIST 数据集含有 70000 幅 28×28 的灰度图像,每一个手写字体数字都是一幅独立图像。其中 60000 幅图像带有标签,表明了该图像代表的数字,用于训练模型;另外 10000 幅图像用于测试和诊断模型。 2. 程序自带数据集。 除了开放的数据集,Python 数据处理扩展包 Pandas 和机器学习扩展包 scikit-learn 也会包含一些学习和测试机器学习算法的经典数据集,比如 MINIST、Iris 等。甚至一些程序包可以产生简单的测试数据集功能 		介绍数据集平台,为后期手写字体体验项目做准备
课堂实践活动	<p>【探究 MNIST 手写数字数据集】</p> <ol style="list-style-type: none"> 1. 下载 MINIST 数据集的 CSV 文件。 打开教师提供的链接,下载 MNIST 的 CSV 数据集。 2. 打开 MINIST 数据集的 CSV 文件。 打开 minist_test_10.csv(记事本工具或表格工具打开),观察该文件中的 10 条记录。 	<p>学生:通过熟悉文件格式,了解 CSV 形式的数据集格式</p>	帮助学生了解数据集的获取途径

续表

教学环节	教学内容	学生活动	设计意图																																																																																																																																				
	<p>1 7,0,0,0,0,0,0,0,0,0,0,0 2 2,0,0,0,0,0,0,0,0,0,0,0,0 3 1,0,0,0,0,0,0,0,0,0,0,0,0 4 0,0,0,0,0,0,0,0,0,0,0,0,0 5 4,0,0,0,0,0,0,0,0,0,0,0,0 6 1,0,0,0,0,0,0,0,0,0,0,0,0 7 4,0,0,0,0,0,0,0,0,0,0,0,0 8 9,0,0,0,0,0,0,0,0,0,0,0,0 9 5,0,0,0,0,0,0,0,0,0,0,0,0 10 9,0,0,0,0,0,0,0,0,0,0,0,0</p> <table border="1"> <tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td></tr> <tr><td>1</td><td>7</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>3</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>5</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>6</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>7</td><td>4</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>8</td><td>9</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>9</td><td>5</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>10</td><td>9</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> <p>教师：每条数据为一行，第一个数字为标签，剩余的数字为数字图像的每个像素点值（灰度图一个像素只有一个数字）。也就是原来的图像被拍扁为一串数字了。</p> <p>3. 使用程序读取数据集。</p> <pre>打开程序,读取同目录下的数据集 CSV 文件。 % matplotlib inline # 导入库 import numpy as np import matplotlib.pyplot as plt # 打开文件 MNIST 数据文件= open('mnist_train_100. csv','r') MNIST 数据列表 = MNIST 数据文件. readlines() # 关闭文件 MNIST 数据文件.close() # 显示第一个元素 MNIST 数据列表[0]</pre> <p>4. 将数据集数据转换为图像。</p> <pre># 选择一个元素(一行数据),加入分隔符“,” 选定的数据行= MNIST 数据列表[15].split (',')</pre> <p># 从选定的数据行中剔除第一个值(标签值) 截取的数据部分= np.asarray(选定的数据 行[1:])</p> <p># 将截取的数据部分重塑成为 28×28 的二维 数组 重塑的二维图像数组 = 截取的数据部分. reshape((28,28)) plt.imshow(重塑的二维图像数组, cmap= 'Greys', interpolation= 'None')</p> <p>运行结果如下：</p>	A	B	C	D	E	F	G	H	I	J	K	L	1	7	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	5	4	0	0	0	0	0	0	0	0	0	0	6	1	0	0	0	0	0	0	0	0	0	0	7	4	0	0	0	0	0	0	0	0	0	0	8	9	0	0	0	0	0	0	0	0	0	0	9	5	0	0	0	0	0	0	0	0	0	0	10	9	0	0	0	0	0	0	0	0	0	0	<p>学生：通过程序对数据集的读取和显示，直观感受图像数据集的数据集的存储方式</p> <p>通过使用程序对数据集进行读取和显示，直观感受图像数据集的存储方式</p>	<p>将数字类型数据再次转换为直观的图像，帮助学生加深对 MNIST 数据集格式的了解</p>
A	B	C	D	E	F	G	H	I	J	K	L																																																																																																																												
1	7	0	0	0	0	0	0	0	0	0	0																																																																																																																												
2	2	0	0	0	0	0	0	0	0	0	0																																																																																																																												
3	1	0	0	0	0	0	0	0	0	0	0																																																																																																																												
4	0	0	0	0	0	0	0	0	0	0	0																																																																																																																												
5	4	0	0	0	0	0	0	0	0	0	0																																																																																																																												
6	1	0	0	0	0	0	0	0	0	0	0																																																																																																																												
7	4	0	0	0	0	0	0	0	0	0	0																																																																																																																												
8	9	0	0	0	0	0	0	0	0	0	0																																																																																																																												
9	5	0	0	0	0	0	0	0	0	0	0																																																																																																																												
10	9	0	0	0	0	0	0	0	0	0	0																																																																																																																												

教学环节	教学内容	学生活动	设计意图
			
课堂小结	教师：今天我们在学习神经网络的基础上，了解了深度学习的概念，了解了常见的数据集平台，并且体验了 MNIST 数据集的读取和显示过程，了解了 CSV 形式的数据集存储方式。下一节课，我们将进一步学习深度学习框架的使用	学生：在教师引导下，梳理本节课实践的内容和收获	

第四节 卷积神经网络

一、教学目标与重点

教学目标：

- 通过实例分析理解卷积神经网络结构以及卷积层、激活函数、池化层等结构的功能。（计算思维、信息意识）
- 通过 TensorFlow 实现识别 MNIST 手写数字的实践活动，体验完整的模型搭建、训练和验证的过程。（计算思维、数字化学习与创新）
- 通过 CIFAR - 10 图像识别活动，体验 Keras 深度学习框架的数据集使用方法、模型搭建和训练的完整过程。（计算思维、数字化学习与创新）
- 通过图像风格迁移活动，体验使用 TensorFlow 框架搭建 CNN 神经网络的完整过程。（计算思维、数字化学习与创新）
- 通过项目实践，引导学生联想，思考卷积神经网络适合运用的领域，增强学生的信息意识。（信息意识、信息社会责任）

教学重点：

- 卷积神经网络的结构和每层功能；
- TensorFlow 框架和 Keras 框架的基本使用方法。

二、内容分析与建议

1. 内容分析

本节是第三章的最后一节，也是第三章最重要、最关键且较难理解的内容。本节主要围绕经典的深度学习模型——卷积神经网络展开，知识框图如图 3-9 所示。卷积神经网络的结构以及每层的计算过程和原理是实现特征抽取和识别的关键。理解卷积神经网络的原理和结构层次是学习重点。



图 3-9 主要知识框图

本节详细讲解了卷积核的概念、卷积计算过程。卷积核做特征提取的过程也就是特征提取的过程。卷积计算过程作为重点需要学生理解掌握。建议该过程以动画形式呈现讲解。相关的项目活动是思考两种不同卷积核(滤波器)过滤效果的差异,使学生加深理解不同卷积核代表不同特征。可参考教科书中的例子,通过一个图形示例来展示不同卷积核的计算效果。

激活函数的作用和概念在“深度神经网络”中已学习,卷积神经网络的激活函数是 ReLU 函数。ReLU 函数会将不符合特征的值归零,从而突出提取的特征。本节的项目活动是用 Python 编写程序实现 ReLU 函数。本小节内容理解相对容易,教学实施时,可减少教师讲授,增加学生主动思考判断的机会。

池化层的作用是卷积神经网络的又一重点。池化的计算过程是在缩减数据量时尽可能保持特征最大化。池化对特征的最大保留效果,建议用图和数结合的方式呈现,帮助学

生理解。

在学习了卷积神经网络的关键层后,归纳总结出卷积神经网络的基本架构,也就是输入层、卷积层、激活函数层、池化层、全连接层。和 CNN 网络架构相比,本小节回顾了最早出现的深层神经网络 LeNet - 5 的结构层次,可通过两者的发展变化,对比掌握 CNN 结构的特点。

本节的实践任务是体验 Keras 环境下搭建的 CNN 模型并运行的完整过程。熟悉框架环境的使用,体验 CNN 模型参数和训练的完整过程,为下一步的实践活动奠定基础。建议给学生充足的时间和实验条件,使其经历完整的安装 Keras 和搭建 CNN 的过程。教师可通过帮助文档、阅读材料、辅助代码等形式提供“脚手架”。可由教师带领学生分析教科书中提供的代码结构,找出 CNN 中关键的卷积、池化操作。

在了解理论结构的基础上,安排了手写数字识别、图像分类识别、图像风格迁移三个主要的实践项目,从而使学生体验卷积神经网络在图像处理方面的应用特点,并实践和熟悉 TensorFlow、Keras 框架的运用。

在手写数字识别实践活动中,分别运行 TensorFlow 和 Keras 两个版本的代码,在对比中学习手写识别 CNN 的代码逻辑,并尝试修改超参数对比识别效果。经过对比实现相同手写数字识别来对比不同学习框架的代码区别。建议对比时通过 CNN 的层次结构中的卷积、池化、全连接层不同层次的代码实现,进一步理解搭建 CNN 结构的过程。

图像识别实践活动中,使用了 CIFAR - 10 数据集,运行调试 CNN 模型,实现了图像分类的识别。CIFAR 数据集是 Keras 自带的数据集。通过该活动,可以向学生介绍 Keras 框架的其他自带数据集,并介绍调用数据集的方法。以此为基础,引导学生尝试仿照本例实现其他数据集。本实践活动中代码较多,可按照数据集读取、数据集预处理、搭建 CNN 网络结构、数据训练、数据检验等步骤来讲解代码。

图像风格迁移的实践活动中,由于模型不需要训练,可由学生自主选择不同风格图像和目标图像来体验。可由教师讲解每部分代码的功能,并显示生成过程的中间图效果。

2. 教学建议

本节实践内容较多,专业性较强。学习这部分内容时不可避免会接触到不少的专业术语,如图、张量等,教师可有意识地为学生准备相关的阅读资料,有意识地培养学生的自主学习能力。理解卷积神经网络的原理和结构层次是本节教学的重点。教师可设计探究任务,引导学生分析卷积计算的过程、池化的目的、卷积核的特点等,提升学生自身计算思维能力。关键环节可采用动画演示辅助理解,比如卷积计算过程、池化过程等。卷积核特征提取过程可对比卷积核和特征提取后的图像来理解。这个过程中,也可尝试提供代码让学生运行,以生成不同的特征图进行对比领会。相对于了解具体的计算过程,让学生理解运算目的更能帮助其理解 CNN 的层次结构功能。教学后期可安排学生通过思维导图等数字化工具梳理卷积神经网络的结构和特点,将学习内容结构化呈现,提升学生的数字化学习与创新意识及能力。神经网络应用的场景和领域目前很丰富,可展示一些最新的行业新闻、成果,为学生拓展思路,使其了解深度学习与生活、医疗、生产、科研、国防等结合后的新成果。加深学生对深度学习的认识,增强学生的信息意识。在学生对深度学习

的基本原理有了初步理解和体验后,可引导学生思考其优势和不足。教师可引导学生从正反两面去思考技术带来的便利性和风险,也可引入一些正、反面案例,激发学生对技术存在两面性的理解,增强自身的信息社会责任感。教学重心不必过多放在代码编写实现环节。体验项目的代码可由教师提供给学生,让学生直接运行或修改后运行。学生将重点放在算法的理解与尝试迁移应用上。本节项目实践活动较多,建议增加辅助阅读材料和实验帮助文件,助力学生的学习。

活动 1:手写数字识别。

需要运行 TensorFlow 和 Keras 两个不同版本的代码。TensorFlow 的实现过程可由教师分步讲解,Keras 可由学生参考阅读材料或帮助文件对比两个版本代码,完成识别过程。

代码见习题分析与提示。

活动 2:CIFAR 图像识别。

此活动使用 Keras 自带的 CIFAR 数据集。可向学生详细介绍 Keras 自带的数据集特点和类别。本活动用到的 CIFAR 数据集调用方法和其他数据集相同。通过讲解代码实现过程,请学生选择不同的超类图片实现图像识别分类,并修改参数,调试出最佳识别精度。

加载数据集这个环节,由于数据集有一百多兆,较大,且考虑到机房计算机一般有保护卡,难以保存下载好的数据集,这一步可以由教师演示。学生可以直接使用教师已经下载好的数据集。

神经网络结构这部分,可以组织学生对比增加结构层,对比训练效果。

训练模型这部分,可以组织学生修改训练次数、修改优化函数等方式,观察选择不同的优化函数时识别精度的区别。

在完成本对数据集数据的识别活动后,也可以增加拓展活动,使用训练好的模型识别学生自己通过网络找到的图片。

活动 3:图像风格迁移。

此活动效果学生应比较感兴趣,可选择学生熟悉的建筑、校园、名人等图片来做被处理图像。可提供典型的不同风格图像作为原始图片。实践活动中,可请学生修改 C、S、G 三个参数,体验不同风格强度的结果图。在讲解代码后,可由学生自由发挥,体验深度学习的奇妙效果。

建议以任务探究法为主,搭配讲授法和演示法进行教学。教师评价与小组学生自评结合,过程性评价与成果展示评价结合。

建议课前机房安装好 Anaconda 这类集成环境,并将所需编辑器、其他需要的第三方库、TensorFlow 深度学习框架、所需的数据集等程序下载安装好。活动任务建议设计成任务单和阅读材料形式,帮助学生实现分工和自主学习部分。

3. 评价建议

评价环节应与教学活动紧密配合,不仅要关注学生对内容的掌握与理解,还要注意学生在核心素养方面的提升。卷积神经网络原理的分析,有助于学生计算思维的提升,评价

时要设置评价内容关注学生高阶思维的进展。可通过卷积神经网络算法的应用,有意识地提升学生的信息意识及信息社会责任,在评价过程中要有意识地设置相关评价任务或问题来测评学生在上述方面的进展;整个教学活动中,学生在教师引导下会接触多种数字化学习工具,会分组合作或协作来完成探究任务,这时候教师可有意识地测评学生的数字化学习与创新能力、合作能力、表达能力等。评价环节非常关键,通过及时全面的评价,教师可获取学生学习数据,进而分析学生的发展情况,及时发现教学中的亮点及不足,以便调整教学策略,有效落实育人目标。

三、习题分析与提示

■ 习题 1 描述

见教科书第 88 页项目实践“卷积操作的可视化”。

■ 习题 1 参考答案

每个卷积核会提取不同的特征,比如卷积核(滤波器)1 左半部分大致呈黑色,右半部分大致呈白色,这个卷积核提取后的特征图明显竖向边缘特征被强化了。同理,卷积核(滤波器)2 是上半部分大致呈黑色,下半部分大致呈白色,这个卷积核提取后的特征图明显水平边缘部分提亮了,也就是横向边缘特征被强化了。

教科书图 3.28 中右侧的卷积核也是提取某一类特征。比如左上角呈黑色,右下角呈白色的卷积核,显然就是提取斜线角度的边缘特征。

■ 习题 2 描述

见教科书第 89 页项目实践“ReLU 激活函数的 Python 实现”。

■ 习题 2 参考答案

(1) Python 实现 ReLU 函数

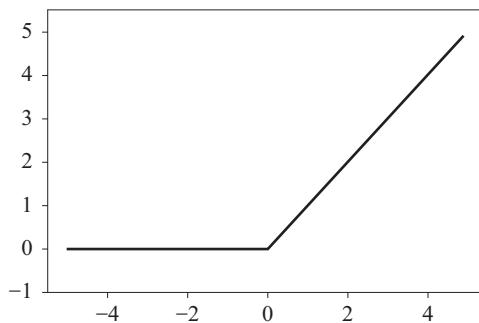
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
def relu(x):
    return np.maximum(0,x)
```

(2) 使用 matplotlib 绘制 ReLU 的图像

```
def show_relu():
    # 展示 relu 函数图像
    x= np.arange(- 5.0,5.0,0.1)
    y= relu(x)
    ylim= (- 1.0,5.5)  # y 轴的范围,比输入的大 d
    plt.plot(x,y)
    plt.ylim(ylim)  # y 轴范围
    plt.show()

# 调用函数,绘制图像
show_relu()
```

(3) 运行结果



习题 3 描述

见教科书第 92 页项目实践“使用 Keras 实现 CNN 架构”。

习题 3 参考答案

```
# 导入所需的库
import numpy as np
import keras
from keras.models import Sequential
from keras.layers import Dense, Flatten
from keras.layers import Conv2D, MaxPooling2D
# 新建模型
model = Sequential()
# 一层卷积层,包含了 32 个卷积核,每个卷积核为 3×3,激活函数为 relu
model.add(Conv2D(32,(3,3),activation= 'relu',input_shape= (100,100,3),name= 'conv-1'))
# 一个最大池化层,池化大小为 2×2
model.add(MaxPooling2D(pool_size= (2,2),name= 'max-pooling-1'))
# 添加一个卷积层,包含 64 个卷积核,每个卷积核为 3×3,激活函数为 relu
model.add(Conv2D(64,(3,3),activation= 'relu',name= 'conv-2'))
# 添加一个池化层,池化窗口 pool_size 为 2×2
model.add(MaxPooling2D(pool_size= (2,2),name= 'max-pooling-2'))
# 压平层
model.add(Flatten())
# 全连接层
model.add(Dense(256,activation= 'relu',name= 'fullc-1'))
# 最后为分类层
model.add(Dense(10,activation= 'softmax'))
# 展示网络结构
model.summary()
```

运行结果如下：

Layer (type)	Output Shape	Param #
conv-1 (Conv2D)	(None, 98, 98, 32)	896
max-pooling-1 (MaxPooling2D)	(None, 49, 49, 32)	0
conv-2 (Conv2D)	(None, 47, 47, 64)	18496
max-pooling-2 (MaxPooling2D)	(None, 23, 23, 64)	0
flatten_8 (Flatten)	(None, 33856)	0
fullc-1 (Dense)	(None, 256)	8667392
dense_12 (Dense)	(None, 10)	2570
Total params: 8,689,354		
Trainable params: 8,689,354		
Non-trainable params: 0		

习题 4 描述

见教科书第 95 页项目实践“MNIST 手写数字识别的 TensorFlow 实现”。

习题 4 参考答案

本项目和教科书第 97 页使用 Keras 实现手写数字识别的代码进行对比,以凸显 Keras 高级接口的优势。学生在调试时,可以修改训练集、测试集、验证集的比例、学习率、迭代数等参数,通过修改,可出现过拟合等情况供学生对照分析。

本项目代码使用的数据集直接调用 tensorflow_datasets 在线数据集。tensorflow_datasets 是为解决学习机器学习者在获取数据集时的繁琐操作,集成常见的公共数据集,并整合为标准格式以供学习者使用。注意使用 tensorflow_datasets 数据集,需要使用较新的 TensorFlow 版本。

(1) 代码

```
import tensorflow as tf
import tensorflow_datasets as tfds

# 超参数
# 数据集名称
dataset_name= 'mnist'
# 批量处理数据块大小
batch_size= 1024
# 数据泄露率
leaky_relu_alpha= 0.2
# 神经元随机丢失率
dropout_rate= 0.025
# 学习率
learning_rate= 0.005
# 训练集占比
train_percent= '60%'
# 测试集占比
test_percent= '30%'
# 验证集占比
validation_percent= '60% : 70%'
# 训练迭代数
num_epochs= 5

# 默认 10 个输出分类
output_classes= 10
# 加载数据集,ds_train,ds_val,ds_test 分别为训练集、验证集、测试集
[ds_train,ds_val,ds_test],info= tfds.load(name= dataset_name,split= ["train[:"+ train_percent+"],"train["+ validation_percent+"],"test[:"+ test_percent+"]"],with_info= True)

ds_train= ds_train.shuffle(1024).batch(batch_size)
ds_test= ds_test.shuffle(1024).batch(info.splits['test'].num_examples)
ds_validation= ds_val.shuffle(1024).batch(batch_size)
print(ds_train.element_spec)
print(ds_test.element_spec)
print(ds_validation.element_spec)
print(info)

# 卷积层函数
def conv2d(inputs,filters,stride_size):
    out= tf.nn.conv2d(inputs,filters,strides= [1,stride_size,stride_size,1],padding= 'SAME')
```

```

        return tf.nn.leaky_relu(out, alpha= leaky_relu_alpha)
# 池化层函数
def maxpool(inputs,pool_size,stride_size):
    return tf.nn.max_pool2d(inputs,ksize= [1,pool_size,pool_size,1],padding= 'VALID',
strides= [1,stride_size,stride_size,1])
# 全连接层函数
def dense(inputs,weights):
    x= tf.nn.leaky_relu(tf.matmul(inputs,weights),alpha= leaky_relu_alpha)
    return tf.nn.dropout(x,rate= dropout_rate)

# 生成随机参数
initializer= tf.initializers.glorot_uniform()
def get_weight(shape,name):
    return tf.Variable(initializer(shape), name = name, trainable = True, dtype = tf.
float32)

shapes= [
    [3,3,1,32], # W0- > conv_layer1
    [3,3,32,64] ,# W1- > conv_layer2
    [3,3,64,128],# W2- > conv_layer3
    [3* 3* 1* 128,128], # W3- > dense_layer
    [128,output_classes],
]
def initialize_weights(shape_lst):
    weights= []
    for i in range(len(shape_lst)):
        weights.append(get_weight(shape_lst[i],'weight{}'.format(i)))
    return weights

def model(x,print_dim= False):
    x= tf.cast(x,dtype= tf.float32)
    if print_dim:print('输入给 Conv W0:',x.shape)
    c1= conv2d(x,weights[0],stride_size= 1)
    if print_dim:
        print('Conv W0 形状:',weights[0].shape)
        print('输入给 pool p1',c1.shape)
    p1= maxpool(c1,pool_size= 2,stride_size= 2)
    if print_dim:print('从 pool p1 输出',p1.shape)

    if print_dim:print('输入给 Conv W1:',p1.shape)
    c2= conv2d(p1,weights[1],stride_size= 1)
    if print_dim:
        print('Conv W1 形状:',weights[1].shape)
        print('输入给 pool p2',c2.shape)
    p2= maxpool(c2,pool_size= 2,stride_size= 2)
    if print_dim:print(' pool p2 输出',p2.shape)

    if print_dim:print('输入给 Conv W2:',p1.shape)
    c3= conv2d(p2,weights[2],stride_size= 1)
    if print_dim:
        print('Conv W2 形状:',weights[2].shape)
        print('输入给 pool p3',c3.shape)
    p3= maxpool(c3,pool_size= 2,stride_size= 2)
    if print_dim:print('pool p2 输出',p2.shape)

    flatten= tf.reshape(p3,shape= (tf.shape(p3)[0],- 1))
    if print_dim:print('输入给 dense layer',flatten.shape)

    d1= dense(flatten,weights[3])
    if print_dim:print('输入给 softmax layer',d1.shape)
    logits= tf.matmul(d1,weights[4])

```

```

    return logits

def softmax(logits):
    return tf.nn.softmax(logits)

def loss(pred,target):
    loss= tf.nn.softmax_cross_entropy_with_logits(target,pred)
    return tf.reduce_mean(loss)

def accuracy(pred,target):
    correct_prediction= tf.equal(tf.argmax(target,1),tf.argmax(pred,1))
    accuracy= tf.reduce_mean(tf.cast(correct_prediction,tf.float32))
    return accuracy

optimizer= tf.optimizers.Adam(learning_rate)

def train_step(model,inputs,outputs):
    with tf.GradientTape(persistent= True) as tape:
        tape.watch(weights)
        pred_logits= model(inputs)
        current_loss= loss(pred_logits,outputs)
        pred= softmax(pred_logits)
        current_acc= accuracy(pred,outputs)

    grads= tape.gradient(current_loss,weights)
    optimizer.apply_gradients(zip(grads,weights))
    loss_val= tf.cast(current_loss,dtype= float,name= None)
    acc_val= current_acc

    return loss_val,acc_val

# 检查每一层的维度
weights= initialize_weights(shapes)
for features in ds_train:
    image,label= features['image'],features['label']
    model(image,print_dim= True)
    break

# 初始化权重
weights= initialize_weights(shapes)
# 开始迭代训练
for e in range(num_epochs):
    for i,batch in enumerate(ds_train):
        image,label= batch['image'],batch['label']
        curr_loss,curr_acc= train_step(model,image,tf.one_hot(label,depth= 10))#
    depth= 输出分类数
        if i % 100== 0:
            print("Epoch {} batch {},Loss= {:.4f} Accuracy= {:.4f}".format(e,i,curr_
loss,curr_acc))

    # 验证集验证
    for batch in ds_validation:
        # only one batch here
        image,label= batch['image'],batch['label']
        pred= softmax(model(image))
        acc= accuracy(pred,tf.one_hot(label,depth= 10))
        print("Epoch {},Validation Accuracy= {:.4f}".format(e,acc))

```

(2) 运行结果

通过观察可以看到训练模型的精确度在收敛。

```
Epoch 0 batch 0, Loss= 31.4332 Accuracy= 0.1055
Epoch 0, Validation Accuracy= 0.7295
Epoch 0, Validation Accuracy= 0.7402
Epoch 0, Validation Accuracy= 0.7275
Epoch 0, Validation Accuracy= 0.7246
Epoch 0, Validation Accuracy= 0.7305
Epoch 0, Validation Accuracy= 0.7330
Epoch 1 batch 0, Loss= 0.8376 Accuracy= 0.7393
Epoch 1, Validation Accuracy= 0.9102
Epoch 1, Validation Accuracy= 0.9277
Epoch 1, Validation Accuracy= 0.9189
Epoch 1, Validation Accuracy= 0.9170
Epoch 1, Validation Accuracy= 0.9141
Epoch 1, Validation Accuracy= 0.9352
Epoch 2 batch 0, Loss= 0.2516 Accuracy= 0.9297
Epoch 2, Validation Accuracy= 0.9531
Epoch 2, Validation Accuracy= 0.9512
Epoch 2, Validation Accuracy= 0.9551
Epoch 2, Validation Accuracy= 0.9424
Epoch 2, Validation Accuracy= 0.9473
Epoch 2, Validation Accuracy= 0.9534
Epoch 3 batch 0, Loss= 0.1446 Accuracy= 0.9658
Epoch 3, Validation Accuracy= 0.9580
Epoch 3, Validation Accuracy= 0.9648
Epoch 3, Validation Accuracy= 0.9580
Epoch 3, Validation Accuracy= 0.9658
Epoch 3, Validation Accuracy= 0.9619
Epoch 3, Validation Accuracy= 0.9534
Epoch 4 batch 0, Loss= 0.1344 Accuracy= 0.9678
Epoch 4, Validation Accuracy= 0.9678
Epoch 4, Validation Accuracy= 0.9678
Epoch 4, Validation Accuracy= 0.9629
Epoch 4, Validation Accuracy= 0.9629
Epoch 4, Validation Accuracy= 0.9600
Epoch 4, Validation Accuracy= 0.9784
```

■ 习题 5 描述

见教科书第 95 页项目实践“使用 Keras 实现手写数字识别”。

■ 习题 5 参考答案

该项目使用到 TensorFlow 的 Keras 结构, 可从模型构建、训练过程对比未使用 Keras 结构的部分。

```
# 导入库
from tensorflow.keras.datasets import mnist
from tensorflow.python.keras.utils import np_utils
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D
%matplotlib inline

# 加载数据集, train_images, train_labels, test_images, test_labels
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
# 显示数据集部分图形
# 创建图
fig = plt.figure()
# 显示前 15 张灰度图片
for i in range(15):
```

```

plt.subplot(3,5,i+1)      # 用3行5列形式展示
plt.tight_layout()         # 自动适配图位置
plt.imshow(train_images[i],cmap= 'Greys')    # 用灰色显示图像
plt.xticks([])             # 删除x轴标记 plt.yticks([])
# 对图片进行归一化
# 图片每个像素的数值都是在0~255之间,归一化要除以255
x_train_nom= train_images.reshape(train_images.shape[0],28,28,1).astype('float32')/255
x_test_nom= test_images.reshape(test_images.shape[0],28,28,1).astype('float32')/255

# 标签转化为one-hot类型
y_train1hot= np_utils.to_categorical(train_labels)
print(y_train1hot[:3])

# 搭建一个模型,第一层先将数据展平,原始图片是28×28的灰度图,所以输入尺寸是(28,28)
# 第二层节点数可以自己选择一个合适值,这里用128个节点,激活函数用relu
# 第三层有多少个种类就写多少,[0,9]一共有10个数字,所以必须写10,激活函数用softmax

model= Sequential()
model.add(Conv2D(filters= 16,
                 kernel_size= (3,3),
                 padding= 'same',
                 activation= 'relu',
                 input_shape= (28,28,1)))
model.add(MaxPooling2D(pool_size= (2,2)))# 增加一个池化层
model.add(Conv2D(filters= 36,
                 kernel_size= (3,3),
                 padding= 'same',
                 activation= 'relu'))
model.add(MaxPooling2D(pool_size= (2,2)))
model.add(Flatten())
model.add(Dense(128,activation= 'relu'))
model.add(Dense(10,activation= 'softmax'))
# 显示模型概要
model.summary()

# 配置模型的优化器为adam,损失函数为categorical_crossentropy,评估训练的指标为accuracy
model.compile(optimizer= 'adam',loss= 'categorical_crossentropy',metrics= ['accuracy'])
# 指定参数,开始训练,将训练结果存入train_history
# x是归一化后的训练数据,y为分类标签,epochs为训练轮数,verbose为日志显示类型为进度条
# validation_split用来指定训练集的一定比例数据作为诊断集

train_history= model.fit(x= x_train_nom,
                         y= y_train1hot,
                         epochs= 10,
                         batch_size= 64,
                         verbose= 1,
                         validation_split= 0.2)

# 图形化显示训练结果,精度和误差的变化情况
# 定义一个函数绘制训练过程精确度和误差的变化曲线
def show_train_history(train_history,train,validation):
    plt.rcParams['font.sans-serif']= ['SimHei']# 用来正常显示中文标签
    plt.rcParams['axes.unicode_minus']= False# 用来正常显示负号
    plt.plot(train_history.history[train])
    plt.plot(train_history.history[validation])
    plt.title('train history')
    plt.ylabel(train)
    plt.xlabel('Epoch')
    plt.legend([train,validation],loc= 'upper right')
    plt.savefig(train)
    plt.show()

```

```

show_train_history(train_history,'loss','val_loss')
show_train_history(train_history,'accuracy','val_accuracy')

# 将测试集的图片输入模型,返回预测结果
predictions= model.predict(x_test_nom)
print('预测值:',np.argmax(predictions[0]),'真实值:',test_labels[0])

# 保存训练好的模型
model_save_path= "./keras-minist.h5"
model.save(model_save_path)

```

■ 习题 6 描述

见教科书第 98 页项目实践“使用 Keras 完成 CIFAR - 10 图像的分类任务”。

■ 习题 6 参考答案

此项目实践活动可将分类任务的步骤分解讲解,安排不同组实现不同的网络结构或者修改训练次数,体会训练过拟合、训练精度的变化过程。

(1) 导入库

```

import keras
import numpy as np
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense,Dropout,Activation,Flatten
from keras.layers import Conv2D,MaxPooling2D
from keras.callbacks import ModelCheckpoint

import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

```

(2) 加载 CIFAR - 10 数据集

```

(x_train,y_train),(x_test,y_test)= cifar10.load_data()
# 显示数据集的形状
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)

```

(3) 显示每个分类的图像

```

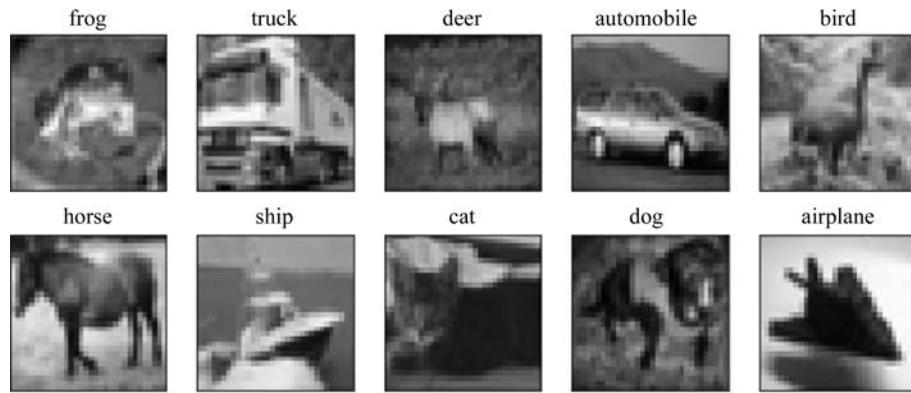
# 显示每一类中的一张图片
label_dict= {0:'airplane',1:'automobile',2:"bird",3:"cat",4:"deer",5:"dog",6:"frog",
7:"horse",8:"ship",9:"truck"}
# 定义画布
fig= plt.figure(figsize= (12,5))
# 转换 numpy 数组类型为列表类型
y_train.tolist()
# 找到每一个分类的一张图片索引
labelnames= label_dict.copy()
showdatas= []
for i in range(y_train.shape[0]):
    if y_train[i][0] in labelnames.keys():
        showdatas.append(i)
        del labelnames[y_train[i][0]]
    if len(labelnames)> 0:
        continue

```

```

else:
    break
# 循环输出每个分类的图片
for i in range(len(showdatas)):
    index= showdatas[i]
    # 增加第 i 个位置的子图
    ax= fig.add_subplot(2,5,i+ 1,xticks= [],yticks= [])
    ax.imshow(np.squeeze(x_train[index]))
    # 添加图像标题
    train_label_value= label_dict[y_train[index][0]]
    plt.title(train_label_value)
plt.show()
运行结果如下：

```



(4) 数据预处理

```

# # # 数据预处理,归一化
x_train= x_train.astype('float32')/255
x_test= x_test.astype('float32')/255

num_classes= len(np.unique(y_train))

# 图像转换为二进制.
y_train= keras.utils.to_categorical(y_train,num_classes)
y_test= keras.utils.to_categorical(y_test,num_classes)

```

(5) 定义卷积神经网络结构

```

# 建立模型
model= Sequential()
# 建立 CNN 结构
model.add(Conv2D(32,(3,3),padding= 'same',input_shape= x_train.shape[1:]))
model.add(Activation('relu'))
# 池化层
model.add(MaxPooling2D(pool_size= (2,2)))
model.add(Dropout(0.25))
# 卷积层
model.add(Conv2D(64,(3,3),padding= 'same'))
model.add(Activation('relu'))
# 池化层
model.add(MaxPooling2D(pool_size= (2,2)))
model.add(Dropout(0.25))
# 卷积层
model.add(Conv2D(64,(3,3),padding= 'same'))
model.add(Activation('relu'))
# 池化层

```

```

model.add(MaxPooling2D(pool_size= (2,2)))
model.add(Dropout(0.25))
# 平铺
model.add(Flatten())
# 全连接层
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
# 分类层
model.add(Dense(num_classes))
model.add(Activation('softmax'))
# 显示网络结构
model.summary()
运行结果如下：

```

Layer (type)	Output Shape	Param #
conv2d_67 (Conv2D)	(None, 32, 32, 32)	896
activation_117 (Activation)	(None, 32, 32, 32)	0
max_pooling2d_67 (MaxPooling)	(None, 16, 16, 32)	0
dropout_92 (Dropout)	(None, 16, 16, 32)	0
conv2d_68 (Conv2D)	(None, 16, 16, 64)	18496
activation_118 (Activation)	(None, 16, 16, 64)	0
max_pooling2d_68 (MaxPooling)	(None, 8, 8, 64)	0
dropout_93 (Dropout)	(None, 8, 8, 64)	0
conv2d_69 (Conv2D)	(None, 8, 8, 64)	36928
activation_119 (Activation)	(None, 8, 8, 64)	0
max_pooling2d_69 (MaxPooling)	(None, 4, 4, 64)	0
dropout_94 (Dropout)	(None, 4, 4, 64)	0
flatten_25 (Flatten)	(None, 1024)	0
dense_50 (Dense)	(None, 512)	524800
activation_120 (Activation)	(None, 512)	0
dropout_95 (Dropout)	(None, 512)	0
dense_51 (Dense)	(None, 10)	5130
activation_121 (Activation)	(None, 10)	0
<hr/>		
Total params: 586,250		
Trainable params: 586,250		
Non-trainable params: 0		

(6) 组装和训练模型

```

# 保存最好模型权重,设置回调函数 callbacks
bestfilename= 'cifa_10_simple.hdf5'
checkpoint= ModelCheckpoint(filepath= bestfilename,verbose= 1,save_best_only= True)

# 训练模型,损失函数使用 RMSprop
model.compile(loss= 'categorical_crossentropy',optimizer= 'adam',metrics= ['accuracy'])
# 训练次数
epochs= 10
# 开始训练,validation_split 设置了 20% 的训练集数据做验证数据,callbacks 指定了
history= model.fit(x_train,y_train,epochs= epochs,shuffle= True,callbacks= [checkpoint],verbose= 1,validation_split= 0.2)

# 保存的模型名

```

```
model_name= 'cifar10.h5'  
# 保存模型  
model.save(model_name)
```

(7) 绘制训练过程图表

```
# 绘制训练集和验证集准确率和 loss 曲线  
plt.rcParams['font.sans-serif']= ['SimHei']# 用来正常显示中文标签  
plt.rcParams['axes.unicode_minus']= False# 用来正常显示负号  
  
plt.plot(history.history['accuracy'])  
plt.plot(history.history['val_accuracy'])  
plt.title('模型精确度')  
plt.ylabel('accuracy')  
plt.xlabel('epoch')  
plt.legend(['train','val'],loc= 'upper left')  
plt.show()  
# summarize history for loss  
plt.plot(history.history['loss'])  
plt.plot(history.history['val_loss'])  
plt.title('训练误差')  
plt.ylabel('loss')  
plt.xlabel('epoch')  
plt.legend(['train','val'],loc= 'upper left')  
plt.show()
```

(8) 显示测试集预测精度

```
# 由于我们设置了检查点,找到 val_loss 最低,我们加载保存的权重,并用模型预测测试集  
model.load_weights('cifa_10_simple.hdf5')  
y_pred= model.predict(x_test)  
# 计算预测值 y_pred 的正确率  
count= 0  
for i in range(len(y_pred)):  
    if(np.argmax(y_pred[i]) == np.argmax(y_test[i])):# argmax 函数找到最大值的索引,即为其  
类别  
        count+= 1  
score= count/len(y_pred)  
print('模型识别精度为: %.2f %' % (score* 100, '%'))
```

(9) 显示测试集 10 个预测结果

```
# 预测 10 个测试集图片  
for i in range(10):  
    prediction= np.argmax(model.predict(x_test[i:i+ 1]))  
    print('第',i,'个图片,预测值:',label_dict[prediction],"真实值:",label_dict[np.argmax  
(y_test[i])])
```

■ 习题 7 描述

见教科书第 100 页项目实践“使用 TensorFlow 及 CNN 模型 VGG19, 实现图像风格迁移应用”。

■ 习题 7 参考答案

该项目实现源代码可参考书中附录文件,此文件以 jupyter notebook 形式保存。

(1) 导入库

```
# 引入 TensorFlow 框架  
import tensorflow as tf
```

```
# 引入 keras
from tensorflow import keras
# 引入 keras 层结构
from tensorflow.keras import layers
# 引入 Numpy 数据处理模块
import numpy as np
# 引入时间模块
import time
# 引入日期模块
from datetime import datetime
# 引入图像绘制模块
import matplotlib.pyplot as plt
# 在 jupyter 界面显示绘图
%matplotlib inline
```

(2) 设置权重和迁移图片

```
# 风格权重
style_weight= 1e-2
# 内容权重
content_weight= 1e4
# 内容图像
image_content_path= './images/content.jpg'
# 风格图像
image_style_path= './images/style.jpg'
```

(3) 读取图像

```
# 读取文件
def image_read(image_path):
    """图像读取
    参数:
        image_path: 图像路径
    返回:
        img: 图像矩阵数据(0.0~ 1.0)
    """
    # 读取图像文件
    img= tf.io.read_file(image_path)
    # 图像 Base64 编码
    img= tf.io.encode_base64(img)
    # 图像 Base64 解码
    img= tf.io.decode_base64(img)
    # 图像 Base64 转矩阵数据
    img= tf.io.decode_image(img)
    # 图像矩阵数据转为 0.0~ 1.0 范围
    img= tf.image.convert_image_dtype(img,dtype= tf.float32)
    # 添加数据维度
    img= img[np.newaxis,:]
    return img

# 读取图像文件
img_content= image_read(image_content_path)
print(img_content.shape)
img_styles= image_read(image_style_path)
print(img_styles.shape)
```

(4) 构建 VGG19 网络模型

建立 VGG 结构的 CNN 模型，并输出概要。如图 3-10 所示，该结构的模型里包含的参数高达 143488080 个，如果自己训练则会非常耗时费力。

Model: "Keras-VGG19"		
Layer (type)	Output Shape	Param #
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
fc3 (Dense)	(None, 1000)	4097000
predictions (Dense)	(None, 1000)	1001000

Total params: 143,488,080
Trainable params: 143,488,080
Non-trainable params: 0

图 3-10 VGG 结构的 CNN 模型(概要)

```
# 重置图结构,为 jupyter notebook 使用
tf.keras.backend.clear_session()

def keras_vgg19_cnn():
    """搭建卷积神经网络
    参数:
        无
    返回:
        model:类实例
    """
    # 实例化
    model= tf.keras.Sequential(name= "Keras-VGG19")
```

```
# 卷积组-1
model.add(
    layers.Conv2D(64, (3,3),
    padding= "same",
    activation= "relu",
    input_shape= (224,224,3),
    name= "block1_conv1")
)
model.add(
    layers.Conv2D(64, (3,3),
    padding= "same",
    activation= "relu",
    name= "block1_conv2")
)
# 最大池化层-1
model.add(
    layers.MaxPooling2D((2,2),
    name= "block1_pool")
)
# 卷积组-2
model.add(
    layers.Conv2D(128, (3,3),
    padding= "same",
    activation= "relu",
    name= "block2_conv1"))
model.add(
    layers.Conv2D(128, (3,3),
    padding= "same",
    activation= "relu",
    name= "block2_conv2"))
)
# 最大池化层-2
model.add(
    layers.MaxPooling2D((2,2),
    name= "block2_pool")
)
# 卷积组-3
model.add(
    layers.Conv2D(256, (3,3),
    padding= "same",
    activation= "relu",
    name= "block3_conv1"))
model.add(
    layers.Conv2D(256, (3,3),
    padding= "same",
    activation= "relu",
    name= "block3_conv2"))
)
# 最大池化层-3
model.add(
    layers.MaxPooling2D((2,2),
    name= "block3_pool")
)
# 卷积组-4
model.add(
    layers.Conv2D(512, (3,3),
    padding= "same",
    activation= "relu",
    name= "block4_conv1")
```

```
        )
model.add(
    layers.Conv2D(512,(3,3),
    padding= "same",
    activation= "relu",
    name= "block4_conv2")
)
model.add(
    layers.Conv2D(512,(3,3),
    padding= "same",
    activation= "relu",
    name= "block4_conv3")
)
model.add(
    layers.Conv2D(512,(3,3),
    padding= "same",
    activation= "relu",
    name= "block4_conv4")
)
# 最大池化层-4
model.add(
    layers.MaxPooling2D((2,2),
    name= "block4_pool")
)
# 卷积组-5
model.add(
    layers.Conv2D(512,(3,3),
    padding= "same",
    activation= "relu",
    name= "block5_conv1")
)
model.add(
    layers.Conv2D(512,(3,3),
    padding= "same",
    activation= "relu",
    name= "block5_conv2")
)
model.add(
    layers.Conv2D(512,(3,3),
    padding= "same",
    activation= "relu",
    name= "block5_conv3")
)
model.add(
    layers.Conv2D(512,(3,3),
    padding= "same",
    activation= "relu",
    name= "block5_conv4")
)
# 最大池化层-5
model.add(
    layers.MaxPooling2D((2,2),
    name= "block5_pool")
)
# 数据拉伸
model.add(layers.Flatten())
# 全连接层-1
model.add(
    layers.Dense(4096,
    activation= "relu",
    name= "fc1")
```

```

        )
# 全连接层-2
model.add(
    layers.Dense(4096,
    activation= "relu",
    name= "fc2")
)
# 全连接层-3
model.add(
    layers.Dense(1000,
    activation= "relu",
    name= "fc3")
)
# softmax 层
model.add(
    layers.Dense(1000,
    activation= "softmax",
    name= "predictions")
)
# 展示网络结构
model.summary()
# 绘制网络流程
keras.utils.plot_model(
    model,
    "./images/keras-vgg19.png",
    show_shapes= True
)
model.summary()
return model

# 新建 VGG19 模型
keras_vgg19_cnn()

```

(5) 预训练网络提取图像特征

为了能节约时间、简化步骤,我们直接使用 imagesnet 提供的训练好的网络模型来提取特征,能快速高效实现风格迁移效果。图 3-11 所示是读取预训练 VGG19 模型,并提取出特征图形后显示的效果。

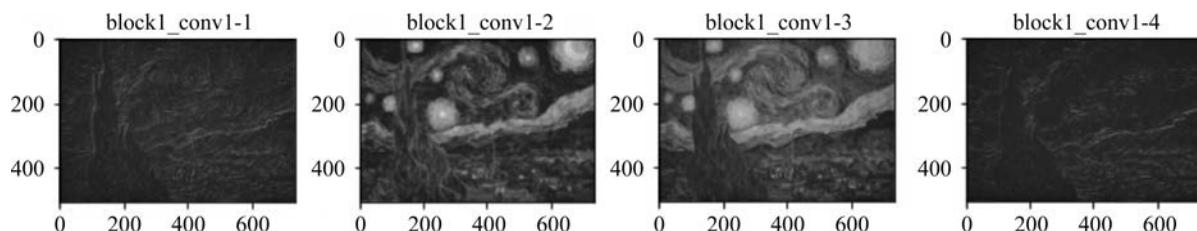


图 3-11 显示效果

提取特征代码如下:

```

# 预训练网络提取图像特征
def layers_name():
    """预训练卷积神经网络层
参数:
    无
返回:
    提取内容层

```

```

    提取风格层
    内容层数量
    风格层数量
"""
# 提取图像内容层
pre_contents_layers= ["block5_conv2"]
# 提取图像风格层
pre_styles_layers= [
    "block1_conv1",
    "block2_conv1",
    "block3_conv1",
    "block4_conv1",
    "block5_conv1"
]
# 风格层数量
num_style_layers= len(pre_styles_layers)
# 内容层数量
num_content_layers= len(pre_contents_layers)
# 返回数据
return pre_contents_layers,pre_styles_layers,num_style_layers,num_content_layers

def pre_vgg19(layers_name):
    """预训练神经网络提取信息
参数:
    layers:神经网络层
返回:
    Model 对象
"""
vgg19= tf.keras.applications.VGG19(include_top= False,weights= "imagenet")
vgg19.trainable= False
outputs= [vgg19.get_layer(name).output for name in layers_name]
model= tf.keras.Model([vgg19.input],outputs)
return model

def pre_res_single(layers_name,outputs):
    """输出预训练神经网络图像特征
参数:
    layers_name:网络层
    outputs:预训练神经网络特征
返回:
    无
"""
i= 0
for name,output in zip(layers_name,outputs):
    i += 1
    plt.figure(i)
    print("网络层:",name)
    print("特征维度:",output.numpy().shape)
    print("特征值:",output.numpy())

    for j in range(4):
        plt.subplot(2,2,j+ 1)
        plt.subplots_adjust(wspace= 0.3,hspace= 0.3)
        plt.imshow(output.numpy()[0][:,:,j])
        plt.title(name+ "-" + str(j+ 1))
    plt.savefig("./images/feature-{}.png".format(i),format= "png",dpi= 300)
    plt.show()

# 获取层结构以及层数量
pre_contents_layers,pre_styles_layers,num_style_layers,num_content_layers= layers_
name()

```

```

# 预训练模型数据提取
pre_model= pre_vgg19(pre_styles_layers)
pre_styles= pre_model(img_styles* 255)
# 获取图像特征
pre_res_single(pre_styles_layers,pre_styles)

```

(6) 定义损失函数

图像风格迁移模型中,需要对内容和风格两项进行损失评估,故需要定义相应两种损失函数,以下是损失函数定义代码:

```

def loss(outputs,style_targets,content_targets,num_style_layers,num_content_layers):
    """计算图像风格和内容损失
    参数:
        outputs:图像风格信息和内容信息
        style_targets:目标风格信息
        content_targets:目标内容信息
        num_style_layers:风格层数量
        num_content_layers:内容层数量
    返回:
        loss_value:总体损失
        style_loss:风格损失
        content_loss:内容损失
    """
    # 提取图像风格特征
    style_outputs= outputs["style"]
    # 提取图像内容特征
    content_outputs= outputs["content"]
    # 计算风格损失
    style_loss= tf.add_n(
        [tf.math.reduce_mean((
            style_outputs[name]-style_targets[name]) * * 2)
         for name in style_outputs.keys()])
    # 分配图像风格损失
    style_loss *= style_weight/num_style_layers
    # 计算图像内容损失
    content_loss= tf.add_n(
        [tf.math.reduce_mean((
            content_outputs[name]-content_targets[name]) * * 2)
         for name in content_outputs.keys()])
    # 分配图像内容损失
    content_loss *= content_weight/num_content_layers
    # 总体损失:风格损失和内容损失
    loss_value= style_loss+ content_loss
    return loss_value,style_loss,content_loss

def grad_loss(inputs,style_targets,content_targets,pre_model,num_style_layers,num_content_layers):
    """获取优化变量
    参数:
        inputs:输入图像矩阵
        style_targets:目标风格信息
        content_targets:目标内容信息
        pre_model:预训练神经网络对象
        num_style_layers:风格层数量
        num_content_layers:内容层数量
    返回:
        总体损失
        风格损失
        内容损失
    """

```

```
训练变量对象
```

```
"""
with tf.GradientTape() as tape:
    outputs= pre_model(inputs)
    loss_value, style_loss, content_loss= loss(outputs, style_targets, content_
targets,num_style_layers,num_content_layers)
    return loss_value,style_loss,content_loss,tape.gradient(
        loss_value,inputs)
```

(7) 定义迭代训练变量

由于图像风格迁移过程中,每一次生成的结果图就是下一次处理的输入图片,训练和优化过程需要迭代进行,所定义的迭代函数如下:

```
# 由于图像风格迁移使用图像作为训练变量,因此需要优化对输入图像进行的迭代计算,输出结果
def optimizer_loss(inputs,style_targets,content_targets,pre_model,num_style_layers,
num_content_layers):
    """优化训练变量
    参数:
        inputs:输入图像矩阵
        style_targets:目标风格信息
        content_targets:目标内容信息
        pre_model:预训练神经网络对象
        num_style_layers:风格层数量
        num_content_layers:内容层数量
    返回:
        总体损失
        风格损失
        内容损失
        转换后的图像矩阵
    """
    # Adam 优化器,优化训练变量
    optimizer= tf.keras.optimizers.Adam(learning_rate= 0.02)
    # 计算损失,获取训练变量
    loss_value,style_loss,content_loss,grads= grad_loss(inputs,style_targets,content_
targets,pre_model,num_style_layers,num_content_layers)
    # 优化训练变量
    optimizer.apply_gradients([(grads,inputs)])
    # 图像数据处理为 0.0~ 1.0 范围数据
    inputs.assign(clip_0_1(inputs))
    return loss_value,style_loss,content_loss,inputs
```

(8) 训练图像函数

训练过程和日志保存操作定义在 train 函数中,方便调用。

```
def train(log_path,inputs,style_targets,content_targets,pre_model,num_style_layers,
num_content_layers):
    """训练变量
    参数:
        inputs:输入图像矩阵
        style_targets:目标风格信息
        content_targets:目标内容信息
        pre_model:预训练神经网络对象
        num_style_layers:风格层数量
        num_content_layers:内容层数量
    返回:
        无
    """
    i= 0
    # 保存运行日志
```

```

if not os.path.exists(log_path)
    os.mkdir(log_path)
summary_writer= tf.summary.create_file_writer(log_path)
# 开始迭代训练
for epoch in range(900):
    # 每轮迭代的开始时间
    time_start= int(time.time())
    # 总体损失均值
    epoch_loss_avg= tf.keras.metrics.Mean()
    # 风格损失均值
    style_loss_avg= tf.keras.metrics.Mean()
    # 内容损失均值
    content_loss_avg= tf.keras.metrics.Mean()
    # 优化损失函数
    loss_value,style_loss,content_loss,style_img= optimizer_loss(
        inputs,
        style_targets,
        content_targets,
        pre_model,
        num_style_layers,
        num_content_layers)
    epoch_loss_avg(loss_value)
    style_loss_avg(style_loss)
    content_loss_avg(content_loss)
    # 每轮训练结束时间
    time_end= int(time.time())
    # 每轮训练耗费的时间
    time_cost= (time_end-time_start)/60
    # 每训练 100 次保存一次训练结果
    if (epoch+ 1)% 100== 0:
        i+= 1
        print("epoch:",i)
        # 新建绘图区
        plt.figure()
        # 图像写入绘图区
        plt.imshow(style_img.read_value()[0])
        # 保存转化后的图像
        plt.savefig("./images/transed-{}.png".format(i),format= "png",dpi= 300)
        # 关闭绘图区
        plt.close()
    # 保存风格损失日志
    with summary_writer.as_default():
        tf.summary.scalar("style_loss",style_loss_avg.result(),step= epoch)
    # 保存内容损失日志
    with summary_writer.as_default():
        tf.summary.scalar("content_loss",content_loss_avg.result(),step= epoch)
    # 保存整体损失日志
    with summary_writer.as_default():
        tf.summary.scalar("total_loss",epoch_loss_avg.result(),step= epoch)
    print("训练次数:{}，总体损失:{}，风格损失:{}，内容损失:{}，训练时间:{}")
min".format(
    epoch+ 1,
    epoch_loss_avg.result(),
    style_loss_avg.result(),
    content_loss_avg.result(),
    time_cost))

```

(9) 定义提取风格和内容特征类

调用模型特征参数,从输入的内容图片和风格图片里抽取出特征,可以封装为一个类,代码如下:

```

class Vgg19StyleContentModel(tf.keras.Model):
    """预训练神经网络提取图像特征"""
    def __init__(self, style_layers, content_layers):
        super(Vgg19StyleContentModel, self).__init__()
        self.vgg19 = pre_vgg19(style_layers + content_layers)
        self.style_layers = style_layers
        self.content_layers = content_layers
        self.num_style_layers = len(style_layers)
        self.vgg19.trainable = False

    def call(self, inputs):
        """输入图像特征提取
        参数:
            inputs: 输入图像矩阵
        返回:
            图像内容特征字典
            图像风格特征字典
        """
        # 图像矩阵数据转为 0~ 255
        inputs = inputs * 255.0
        # 加载预处理神经网络,预处理图像数据
        preprocessed_input = tf.keras.applications.vgg19.preprocess_input(inputs)
        # 预处理神经网络提取图像特征
        outputs = self.vgg19(preprocessed_input)
        # 依据卷积伸进网络层次,提取对应特征值
        # 图像风格特征和内容特征
        style_outputs, content_outputs = (
            outputs[:, :self.num_style_layers],
            outputs[:, self.num_style_layers:]
        )
        # 图像风格特征 Gram 信息提取
        style_outputs = [
            gram_matrix(style_output)
            for style_output in style_outputs]
        # 图像内容特征
        content_dict = {
            content_name: value
            for content_name, value in zip(self.content_layers, content_outputs)}
        # 图像风格特征
        style_dict = {
            style_name: value
            for style_name, value in zip(self.style_layers, style_outputs)}
        return {"content": content_dict, "style": style_dict}

```

(10) 完成训练和风格迁移

调用以上定义的函数和类,实现训练和图像迁移效果。训练函数定义 900 次迭代后,训练过程生成的结果图会保存在 images 目录下,这些体现训练过程的图片也是模型训练输出的结果,如图 3-12 所示。

训练次数:1, 总体损失:157669584.000, 风格损失:157669584.000, 内容损失:0.000, 训练时间:0.6min
 训练次数:2, 总体损失:112533800.000, 风格损失:108299752.000, 内容损失:4234049.000, 训练时间:0.4min
 训练次数:3, 总体损失:64562828.000, 风格损失:58871860.000, 内容损失:5690969.000, 训练时间:0.4min
 训练次数:4, 总体损失:42313452.000, 风格损失:35618808.000, 内容损失:6694645.000, 训练时间:0.4min
 训练次数:5, 总体损失:64652056.000, 风格损失:59066312.000, 内容损失:5585743.000, 训练时间:0.5min
 训练次数:6, 总体损失:38103860.000, 风格损失:31060780.000, 内容损失:7043081.000, 训练时间:0.5min
 训练次数:7, 总体损失:63569856.000, 风格损失:58241920.000, 内容损失:5327938.000, 训练时间:0.5min
 训练次数:8, 总体损失:29398700.000, 风格损失:22715778.000, 内容损失:6682921.000, 训练时间:0.4min
 训练次数:9, 总体损失:54481724.000, 风格损失:49725328.000, 内容损失:4756396.000, 训练时间:0.4min
 训练次数:10, 总体损失:23656240.000, 风格损失:17352894.000, 内容损失:6303347.000, 训练时间:0.4min

```
训练次数:11, 总体损失:56194800.000, 风格损失:51535472.000, 内容损失:4659328.500, 训练时间:0.4min  
训练次数:12, 总体损失:19710772.000, 风格损失:13945429.000, 内容损失:5765344.000, 训练时间:0.4min  
训练次数:13, 总体损失:53393984.000, 风格损失:48367552.000, 内容损失:5026432.000, 训练时间:0.4min  
训练次数:14, 总体损失:16047302.000, 风格损失:10604111.000, 内容损失:5443190.500, 训练时间:0.5min  
训练次数:15, 总体损失:43501840.000, 风格损失:38816944.000, 内容损失:4684898.000, 训练时间:0.5min
```

图 3-12 训练过程(部分)

实现代码如下：

```
# 预训练网络提取图像风格与内容特征  
vgg19_pre_model= Vgg19StyleContentModel(pre_styles_layers,pre_contents_layers)  
# 预训练网络提取风格特征  
style_targets= vgg19_pre_model(img_styles)[ "style"]  
# 预训练网络提取内容特征  
content_targets= vgg19_pre_model(img_content)[ "content"]  
# 新建图像变量张量  
inputs_img= tf.Variable(img_content)  
# 日志文件路径  
stamp= datetime.now().strftime("%Y% m% d% H% M% S")  
log_path= "./logs/style-transfer"+ stamp  
# 训练模型  
train(  
    log_path,  
    inputs_img,  
    style_targets,  
    content_targets,  
    vgg19_pre_model,  
    num_style_layers,  
    num_content_layers)
```

四、核心概念精解与技术工具介绍

1. 概念精解

(1) 鲁棒性

一般指系统的健壮度,耐用性,在受到一定干扰时仍能保持稳定性。

(2) 生物神经网络

生物神经网络是由大脑神经元、细胞、触点等组成的网络。神经元类似于CPU处理器的处理单元,这些处理单元按照一定方式相互连接,构成了大脑中的生物神经元网络。神经元又叫做神经细胞,具有感受刺激和传导兴奋的功能。生物神经元具有两种状态:兴奋和抑制。当传入的神经冲动使得细胞膜电位升高超过阈值,细胞进入兴奋状态,产生神经冲动并且由轴突输出,当传入的神经冲动使得膜电位下降低于阈值时,细胞进入抑制状态,不输出神经冲动。

(3) 激活函数 ReLU(rectified linear units)

神经网络训练中通常可以选择的激活函数有 threshold、sigmoid、tanh、ReLU 几种。其中 ReLU 函数效果最好。为什么呢? 神经网络中的神经元越多,连接越复杂,所产生的参数也越多,随之神经网络的学习能力也会越强。但是实际训练中,数据源中的无效数据也被学习,导致神经网络的预测能力随之下降。这种情况下,需要将神经网络中的参数

“稀释”，降低其学习能力，提高其预测能力。ReLU 函数可有效“稀释”数据，效果最好。因此，在实际应用中应用最广泛。

ReLU 函数的图像如图 3-13 所示，当参数大于 0 时，会等比例保留参数。这种处理方式可以对小于 0 的参数进行过滤，参数的密度就得到了“稀释”，减少了计算量。

(4) 训练模型

训练模型就是建立神经网络模型，是神经网络“学习”输入数据的特征，并把“学习”这些特征时用到神经网络参数保存到一个模型文件中，从而“记住”这些数据的特征。当时用该模型预测时，神经网络按照“学习”到的特征预测。

(5) 泛化能力

神经网络学习的目的是学到隐含在数据背后的规律，对具有同一规律的训练集以外的数据，经过训练的网络也能给出合适的输出，该能力称为泛化能力。

(6) 归一化

简单理解，就是将一系列数据经过一定的数学计算转换为一定范围内的值，依然保留这组数据之间的大小关系。数据在预处理时，通过归一化操作，既保留了数据的统计分布特征，又可以将不同度量范围的数据特征统一到一个可以相互比较的量级。

2. 工具介绍

(1) TensorFlow 搭建神经网络

使用 TensorFlow2.0 搭建神经网络可以有三类方法。

方法 1：直接使用 Keras 搭建。

方法 2：借助 Keras 自定义搭建。这种方法可以快速搭建神经网络，又可以自定义损失函数和优化训练过程，实现更加复杂的数据处理任务。

方法 3：完全自定义神经网络结构。这样仅仅是继承了 Keras 的类，从最底层自定义神经网络结构，比较繁琐，要求编写人员对网络结构完全掌握，也是比较容易出错的方法。

(2) TensorFlow 模型格式

TensorFlow2.0 的模型有 ckpt、h5、pb 三种类型。其中 ckpt 格式的模型是分散保存的，主要有三类文件，也就是 checkpoint、data 和 index 模型文件。其中 checkpoint 存储最新文件模型名称、data 存储 tensorflow 张量中的数值、index 存储 data 的索引。h5 格式的模型是一个整体稳健的模型，保存了模型的参数以及网络结构。pb 是服务器部署模型时使用的模型。

(3) TensorFlow 保存模型和载入模型的方法

在神经网络训练过程中，神经网络节点的参数在不断更新，直到达到超参数指定的训练步数。当训练到一定程度，模型可以正确预测指定数据。如果需要模型去预测新的数据，就可以先将训练好的神经网络参数保存下来，使用时载入即可。这样神经网络可以部署到新的环境中去完成任务。TensorFlow 为神经网络参数的保存和加载提供了简洁的

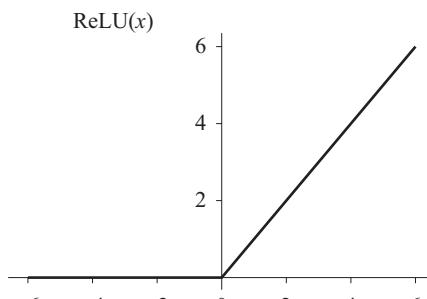


图 3-13 ReLU 函数的图像

工具：

① 通用模型保存和载入类 `tf.train.Checkpoint()`

通用模型保存和载入类是 TensorFlow2.0 提供的保存模型的一种处理方式。可以保存模型中的全部数据，包括模型结构和模型中的参数。

`tf.train.CheckPoint(* * kwargs)` 使用方法如表 3-13 所示：

表 3-13 `tf.train.CheckPoint()` 使用方法

<code>save(file_prefix)</code>	保存训练模型节点数据, <code>file_prefix</code> 为模型文件名称
<code>write(file_prefix)</code>	将训练过程中的节点数据写入文件, <code>file_prefix</code> 为模型文件名称
<code>restore(save_path)</code>	恢复模型, 读取模型文件中保存的参数, <code>save_path</code> 为模型文件路径

例如，在源代码目录中建立一个名为 `save` 的文件夹并调用一次 `checkpoint.save('model.ckpt')`，我们就可以发现名为 `checkpoint`、`model.ckpt-1.index`、`model.ckpt-1.data-00000-of-00001` 的三个文件，这些文件就记录了变量信息。`checkpoint.save()` 方法可以运行多次，每运行一次都会得到一个 `.index` 文件和 `.data` 文件，序号依次累加。使用方法示例如下：

```
checkpoint = tf.train.Checkpoint(model = model)
```

```
checkpoint.save(save_path_with_prefix)
```

当需要在其他地方为模型重新载入之前保存的参数时，需要再次实例化一个 `checkpoint`，同时保持键名的一致，再调用 `checkpoint` 的 `restore` 方法。就像下面这样：

```
model_to_be_restored = MyModel() # 待恢复参数的同一模型
```

```
checkpoint = tf.train.Checkpoint(myModel = model_to_be_restored) # 键名保持为“myModel”
```

```
checkpoint.restore(save_path_with_prefix_and_index)
```

② 模型保存与载入类 `tf.keras.Model()`

Keras 模型保存与载入类专用于 Keras 搭建的 `ckpt` 或者 `h5` 格式的模型，使用方法如表 3-14 所示：

表 3-14 `tf.keras.Model()` 使用方法

<code>tf.keras.Model().save_weights(filepath, overwrite=True, save_format=None)</code>	这个方法保存神经网络所有层的权重，仅保存模型中的参数，不保存模型的层次结构。 <code>filepath</code> 这个参数是指模型全部的保存路径。 <code>Overwrite</code> 参数为 <code>true</code> 时，保存时如存在旧的模型数据，重新保存的模型会覆盖更新模型数据。 <code>Save_format</code> 是模型保存格式，为 <code>h5</code> 或者 <code>ckpt</code> 后缀
<code>tf.model().load_weights(filepath, by_name=False)</code>	这个方法用于恢复模型参数，其中的 <code>filepath</code> 是指模型数据所在路径， <code>by_name</code> 是指通过名称恢复

```
tf. model(). save(
    filepath,
    overwrite = True,
    include_optimizer = True,
    save_format = None,
    signnatures = None,
    options = None
)
```

这个方法保存模型的所有数据,包括模型结构和模型参数。其中参数 `include_optimizer` 为 `True` 则保存模型优化器状态

(4) Keras 使用方法

TensorFlow2.0 将 Keras 内置到了 TensorFlow 框架里,开发者可以直接在 TensorFlow 框架中调用 Keras,简单、高效。使用 Keras 可以搭建完整周期的模型,也就是新建模型、训练模型、持久化模型、模型预测等。Keras 搭建神经网络实现预测的流程如图 3-14 所示。由图可知,完整的人工智能开发任务有六个步骤,每个步骤 Keras 都有对应的工具接口,左边为人工智能任务的步骤,右边为 Keras 的对应接口。

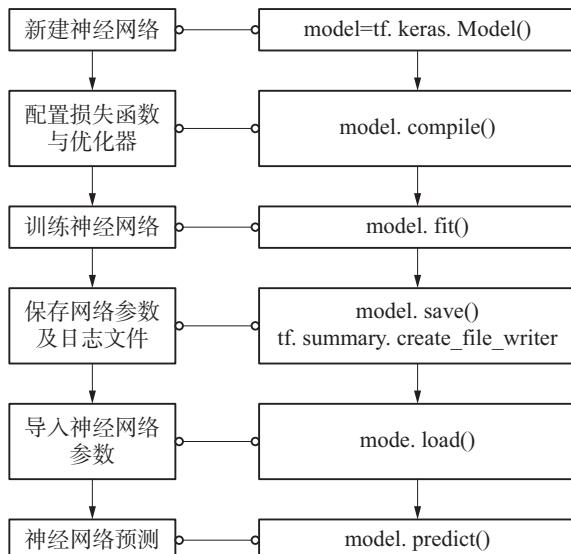


图 3-14 Keras 搭建神经网络实现预测的流程

① Keras 的常用类和方法介绍

Keras 搭建神经网络常用的高级接口有两个,分别是 `Model` 类和 `Sequential` 类。通过这两个类可以实现模型的搭建、训练、持久化、预测、调优等完整生命周期的运行和维护。

`tf.keras.Model` 类:

这个类将神经网络的输入和输出作为其参数,完成神经网络的训练、模型的保存和实现预测。使用这个类的方法有两种,一种直接使用 `Model` 这个类完成(上面普通神经网络和卷积神经通网络的搭建已经做了完整的示范),另一种是继承这个类搭建自定义的神经网络结构。因此通过 `Model` 来搭建神经网络是比较灵活,可以满足多种需求。

Model 类有几个重要方法。

compile 方法:这个方法用于配置训练神经网络的损失函数、优化器、衡量指标等。

fit 方法:这个函数可以指定训练次数、保存训练参数等。

predict 方法:这个方法用于载入模型数据,对输入数据进行预测,获得神经网络的预测值。

tf.keras.Sequential 类:

Kerasd 搭建神经网络的高级接口有两个接口类,一个是 Model 类,另一个就是 Sequential 类。Sequential 类继承自 Model 类,因此 Model 类具备的方法 Sequential 类也可以使用。Sequential 类是针对单一输出的神经网络,通过 add 方法叠加建立神经网络层,方便快捷。对于一般性的神经网络,使用序列化方式即可完成神经网络的创建、训练、保存和预测全生命周期。

tf.keras.layers.Dense 类:

Dense 这个类用于神经网络层二维矩阵的计算。比如图像识别应用里,图像中像素的每个通道(RGB 图像有红绿蓝三个原色通道)的像素值都会转换成一个 0~255 的数值。每个像素点每通道的颜色值会组成一个二维矩阵。通过对矩阵数据的计算,最终可以实现神经网络对图像的识别任务。Dense 层还用于卷积神经网络的全连接层数据的计算。Dense 类的参数如下:

```
tf.keras.layers.dense(  
    inputs,           # 层的输入  
    units,            # 设置该层节点数  
    activation=None, # 激活函数,如字符串"relu"或实例 tf.nn.relu 形式  
    use_bias=True,   # 布尔值,设置是否使用偏置,默认带有偏置  
    kernel_initializer=None, # 卷积核的权重初始化方法  
    bias_initializer=tf.zeros_initializer(), # 偏置值初始化方法  
    kernel_regularizer=None, # 权重规范化函数  
    bias_regularizer=None, # 偏置项规范化函数  
    activity_regularizer=None, # 输出的规范化方法  
    kernel_constraint=None, # 权重变化限制函数  
    bias_constraint=None, # 偏置值变化限制函数  
    trainable=True,      # 表明该层的参数是否参与训练  
    name=None,          # 层的名字  
    reuse=None,         # 是否重复使用参数  
)
```

tf.keras.layers.Conv2D 类:

Conv2D 用于搭建卷积神经网络,并进行卷积运算。Keras 的 Conv2D 类通过初始化参数即可完成卷积计算。其具体参数如下:

```
tf.keras.layers.Conv2D(  
    filters,           # 表示输出数据的维数(即卷积过滤器的数量)  
    kernel_size,       # 表示卷积核的高和宽。  
    strides=(1,1),     # 表示卷积的纵向和横向的步长  
    padding='valid',   # "valid"不够卷积核大小的块就丢弃,"same"表示不够就补 0。  
    data_format=None,  # 表示输入数据的格式  
    dilation_rate=(1,1), # 表示使用扩张卷积时的扩张率  
    activation=None,   # 激活函数。如果是 None 则为线性函数  
    use_bias=True,     # 表示是否使用偏差向量
```

```

        kernel_initializer= 'glorot_uniform',      # 卷积核的初始化
        bias_initializer= 'zeros',                 # 偏差向量的初始化
        kernel_regularizer= None,                # 卷积核的权重初始化方法
        bias_regularizer= None,                  # 偏置值变化限制函数
        activity_regularizer= None,              # 输出的规范化方法
        kernel_constraint= None,                # 权重变化限制函数
        bias_constraint= None,                  # 偏置值变化限制函数
        * * kwargs  reuse= None                 # 额外参数
)

```

tf.keras.layers.MaxPooling2D 类：

最大池化层是卷积神经网络最常用的数据提取方法。Keras 的最大池化类之一就是 MaxPooling2D 类,还有一个 MaxPool2D 类。这两个类都可以实现数据的特征提取。初始化参数如下:

```

tf.keras.layers.MaxPooling2D(
    pool_size= (2,2),          # 池化核尺寸
    strides= None,             # 池化核移动步长
    padding= 'valid',          # "valid"不够大小的块就丢弃,"same"表示不够就补 0。
    data_format= None          # 输入数据的格式
)

```

tf.keras.layers.Flatten 类：

Flatten 类是连接卷积层与全连接层的过渡层。Flatten 类的作用就是将上一层的神经网络数据“拉伸”为列向量,保持参数不变,只改变维度后作为全连接层的输入。这个类的初始化参数如下:

```

tf.keras.layers.Flatten(
    data_format= None,          # 输入数据的格式
)

```

② Keras 搭建神经网络示例

Keras 提供了高度封装的接口,搭建神经网络快捷、方便,可以满足简单任务神经网络的搭建,下面分别介绍普通神经网络和卷积神经网络的搭建过程。

示例 1：普通神经网络的搭建。

使用 Keras 搭建普通神经网络的方式有多种方式,下面介绍几种。

方式 1：Model 类逐层建立网络结构。

直接使用 keras 的 Model 类生成神经网络结构。可参考以下代码:

```

# 导入框架
import tensorflow as tf
# 导入 keras
from tensorflow import keras
# 导入 keras 的层结构
from tensorflow.keras import layers
# 重置图结构,为 jupyter notebook 使用
tf.keras.backend.clear_session()
def line_model():
    # 输入层
    inputs= keras.Input(shape= (1,),name= "inputs")
    # 隐藏层 1
    layer1= layers.Dense(10,activation= "relu",name= "layer1")(inputs)

```

```

# 隐藏层 2
layer2= layers.Dense(15,activation= "relu",name= "layer2") (layer1)
# 输出层
outputs= layers.Dense(5,activation= "softmax",name= "outputs") (layer2)
# 实例化
model= keras.Model(inputs= inputs,outputs= outputs)
# 展示网络结构
model.summary()
# 绘制网络流程图
keras.utils.plot_model(model,"./modelshow.png",show_shapes= True)

if __name__ == "__main__":
    line_model()

```

运行结果如下：

Layer (type)	Output Shape	Param #
inputs (InputLayer)	[None, 1]	0
layer1 (Dense)	(None, 10)	20
layer2 (Dense)	(None, 15)	165
outputs (Dense)	(None, 5)	80

Total params: 265
Trainable params: 265
Non-trainable params: 0

方式 2：Sequential 内置序列化搭建神经网络。

Keras 提供了序列化（按照步骤逐步实现）搭建神经网络的类 Sequential。通过 Sequential 可以搭建具有单一输出的神经网络。示例代码如下：

```

# 导入框架
import tensorflow as tf
# 导入 keras
from tensorflow import keras
# 导入 keras 的层结构
from tensorflow.keras import layers
# 重置图结构,为 jupyter notebook 使用
tf.keras.backend.clear_session()
def line_fit_sequential():
    """
    使用 Sequential 搭建神经网络,返回 model 网络类实例
    """
    model= keras.Sequential([
        # 隐藏层 1
        layers.Dense(10,activation= "relu",input_shape= (1,),name= "layer1"),
        # 隐藏层 2
        layers.Dense(15,activation= "relu",name= "layer2"),
        # 输出层
        layers.Dense(5,activation= "softmax",name= "outputs")
    ])
    # 展示网络结构
    model.summary()
    # 绘制网络流程图
    keras.utils.plot_model(model,"./line-fit-sequential.png",show_shapes= True)

```

```
    return model

if __name__ == "__main__":
    line_fit_sequential()
```

运行结果如下：

```
Model: "sequential"
-----  
Layer (type)        Output Shape       Param #
-----  
layer1 (Dense)     (None, 10)          20  
-----  
layer2 (Dense)     (None, 15)          165  
-----  
outputs (Dense)    (None, 5)           80  
-----  
Total params: 265
Trainable params: 265
Non-trainable params: 0
```

方式 3：Sequential 外置序列化搭建神经网络。

通过 Sequential 外置一步步搭建神经网络的步骤和之前两种方法大同小异。效果相同。可参考以下代码完成搭建：

```
# 导入框架
import tensorflow as tf
# 导入 keras
from tensorflow import keras
# 导入 keras 的层结构
from tensorflow.keras import layers
# 重置图结构,为 jupyter notebook 使用
tf.keras.backend.clear_session()
def line_fit_sequential_add():
    """
    使用 Sequential 外置序列化搭建网络结构
    """
    # Sequential 实例化
    model= keras.Sequential()
    # 添加隐藏层 1

    model.add(layers.Dense(10,activation= tf.nn.relu,input_shape= (1,),name= "layer1"))
    # 添加隐藏层 2
    model.add(layers.Dense(15,activation= tf.nn.relu,name= "layer2"))
    # 添加输出层
    model.add(layers.Dense(5,activation= tf.nn.softmax,name= "outputs"))
    # 展示网络结构
    model.summary()
    # 绘制网络流程图
    keras.utils.plot_model(model,"./line-fit-sequential-add.png",show_shapes= True)
    return model

if __name__ == "__main__":
    line_fit_sequential_add()
```

运行结果如下：

```

Model: "sequential"
-----  

Layer (type)        Output Shape       Param #
-----  

layer1 (Dense)      (None, 10)         20  

layer2 (Dense)      (None, 15)         165  

outputs (Dense)     (None, 5)          80  

-----  

Total params: 265  

Trainable params: 265  

Non-trainable params: 0

```

示例 2：卷积神经网络的搭建。

卷积神经网络的搭建依然会用到 TensorFlow 框架的 Sequential 类，也同样可以采用内置和外置搭建两种方法。完整的卷积神经网络是由输入层开始的，这层的名称会由 Keras 自动生成。其余各层按照程序参数指定的名称命名。

方法 1：Sequential 内置序列化搭建卷积神经网络。

卷积神经网络和普通神经网络搭建方法类似。只是建立的是卷积层、池化层、全连接层。使用 Sequential 类实现的代码如下：

```

# 导入框架
import tensorflow as tf
# 导入 keras
from tensorflow import keras
# 导入 keras 的层结构
from tensorflow.keras import layers
# 重置图结构,为 jupyter notebook 使用
tf.keras.backend.clear_session()
def cnn_sequential():
    """
    使用 Sequential 序列内置搭建卷积神经网络
    """
    model= keras.Sequential([
        # 卷积层 1
        layers.Conv2D(32,(3,3),activation= "relu",input_shape= (28,28,3),name= "conv-1"),
        # 最大池化层 1
        layers.MaxPooling2D((2,2),name= "max-pooling-1"),
        # 卷积层 2
        layers.Conv2D(64,(3,3),activation= "relu",name= "conv-2"),
        # 最大池化层 2
        layers.MaxPooling2D((2,2),name= "max-pooling-2"),
        # 卷积层 3
        layers.Conv2D(64,(3,3),activation= "relu",name= "conv-3"),
        # 多维的输入一维化
        layers.Flatten(),
        # 全连接层
        layers.Dense(64,activation= "relu",name= "fullconection-1"),
        # softmax,输出层
        layers.Dense(10,activation= "softmax",name= "softmax")
    ])

    # 展示网络结构
    model.summary()
    # 绘制网络流程图
    keras.utils.plot_model(model,"./cnn-sequential.png",show_shapes= True)

```

```

    return model

if __name__ == "__main__":
    cnn_sequential()

```

运行结果如下：

Model: "sequential_1"		
Layer (type)	Output Shape	Param #
conv-1 (Conv2D)	(None, 26, 26, 32)	896
max-pooling-1 (MaxPooling2D)	(None, 13, 13, 32)	0
conv-2 (Conv2D)	(None, 11, 11, 64)	18496
max-pooling-2 (MaxPooling2D)	(None, 5, 5, 64)	0
conv-3 (Conv2D)	(None, 3, 3, 64)	36928
flatten_1 (Flatten)	(None, 576)	0
fullconnection-1 (Dense)	(None, 64)	36928
softmax (Dense)	(None, 10)	650

Total params: 93,898
Trainable params: 93,898
Non-trainable params: 0

方法 2：Sequential 外置序列化搭建卷积神经网络。

外置搭建的代码如下。下面代码建立了一个 3 层卷积层，2 层池化层，1 层输出层的卷积神经网络。输出层激活函数使用的是 softmax。

```

# 导入框架
import tensorflow as tf
# 导入 keras
from tensorflow import keras
# 导入 keras 的层结构
from tensorflow.keras import layers
# 重置图结构,为 jupyter notebook 使用
tf.keras.backend.clear_session()
def cnn_sequential_add():
    """
    Sequential 序列外置搭建卷积神经网络,返回类实例
    """
    # 实例化
    model= keras.Sequential()
    # 卷积层 1
    model.add(
        layers.Conv2D(32,(3,3),
        activation= tf.nn.relu,
        input_shape= (25,28,3),
        name= "conv-1")
    )
    # 最大池化层 1
    model.add(layers.MaxPooling2D((2,2),name= "max-pooling-1"))
    # 卷积层 2
    model.add(layers.Conv2D(64,(3,3),activation= tf.nn.relu,name= "conv-2"))
    # 最大池化层 2
    model.add(layers.MaxPooling2D((2,2),name= "max-pooling-2"))
    # 卷积层 3
    model.add(layers.Conv2D(64,(3,3),activation= tf.nn.relu,name= "conv-3"))

```

```

# 多维的输入一维化
model.add(layers.Flatten())
# 全连接层 1
model.add(layers.Dense(64,activation= tf.nn.relu,name= "fullconnection-1"))
# softmax 输出层
model.add(layers.Dense(10,activation= tf.nn.softmax,name= "softmax"))
# 展示网络结构
model.summary()
# 绘制网络流程图
keras.utils.plot_model(model,"./cnn-sequential-add.png",show_shapes= True)
return model

if __name__ == "__main__":
    cnn_sequential_add()

```

运行结果如下：

```

Model: "sequential_1"
-----  

Layer (type)          Output Shape         Param #
-----  

conv-1 (Conv2D)        (None, 23, 26, 32)      896  

-----  

max-pooling-1 (MaxPooling2D) (None, 11, 13, 32)  0  

-----  

conv-2 (Conv2D)        (None, 9, 11, 64)       18496  

-----  

max-pooling-2 (MaxPooling2D) (None, 4, 5, 64)    0  

-----  

conv-3 (Conv2D)        (None, 2, 3, 64)        36928  

-----  

flatten_1 (Flatten)   (None, 384)            0  

-----  

fullconnection-1 (Dense) (None, 64)           24640  

-----  

softmax (Dense)       (None, 10)             650  

-----  

Total params: 81,610
Trainable params: 81,610
Non-trainable params: 0

```

示例 3：卷积神经网络 CNN 实现手写数字识别。

导入库：

```

# 引入 Tensorflow 框架
import tensorflow as tf
# 引入 keras 框架
from tensorflow import keras
# 导入 keras 的层结构
from tensorflow.keras import layers
# 引入数据处理模块
import numpy as np
# 引入图像处理模块
import matplotlib.pyplot as plt

```

读取数据集：

```

# 引入 MNIST 数据集
mnist= keras.datasets.mnist
# 提取数据集数据
(train_images,train_labels),(test_images,test_labels)= mnist.load_data()
# 打印出数据集的形状,也就是图像维度(个数、宽、高、通道数)

```

```

print('数据集尺寸',train_images.shape)
print('第一个训练集图像')
print(train_images[0])
print('第一个训练集标签',train_labels[0])
print('第一个测试集图像')
print(test_images[0])
print('第一个测试集标签',test_labels[0])

```

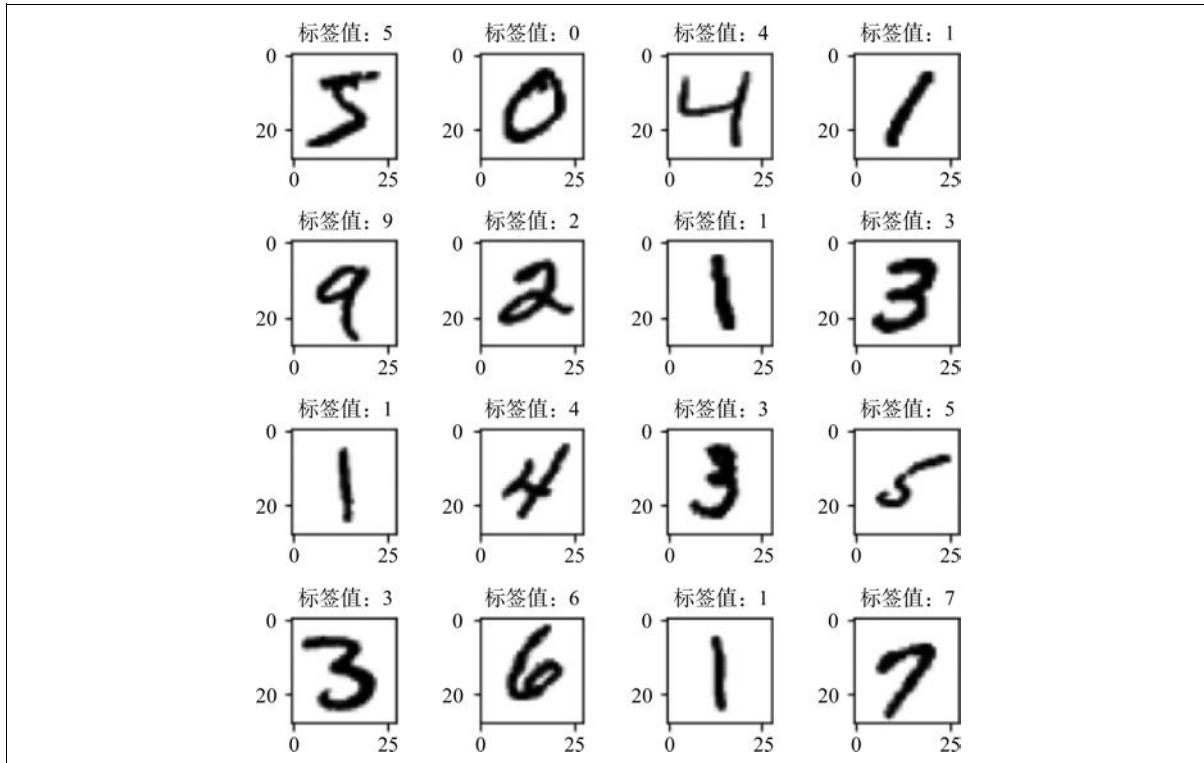
显示数据集：

```

plt.rcParams['font.sans-serif']= ['SimHei'] # 显示中文标签
plt.rcParams['axes.unicode_minus']= False # 这两行需要手动设置
# 打开绘图区
plt.figure(figsize= (6,6))
# 绘制前 16 个图像
for i in range(16):
    # 图像矩阵数据
    train_data_value= train_images[i]
    # 图像标签数据
    train_label_value= train_labels[i]
    # 调整图像尺寸
    train_image_reshape= train_data_value.reshape((28,28,- 1))
    # 绘图区分区
    plt.subplot(4,4,i+ 1)
    plt.subplots_adjust(wspace= 0.5,hspace= 0.8)
    # plt.imshow(train_image_reshape[:, :, 0],cmap= "Greys_r")
    # 图像写入绘图区
    plt.imshow(train_data_value,cmap= plt.cm.binary)
    # 添加图像标题
    plt.title("标签值:{}".format(train_label_value))
    # print("data i:{}".format(i))
plt.savefig("./train_image_show.png",format= "png",dpi= 500)
plt.show()

```

运行结果如下：



搭建 CNN 神经网络结构：

```
# 建立 CNN 模型
model= tf.keras.Sequential(name= "MNIST-CNN")
# 卷积层-1
model.add(
    layers.Conv2D(32, (3,3),
        padding= "same",
        activation= tf.nn.relu,
        input_shape= (28,28,1),
        name= "conv-1")
)
# 最大池化层-1
model.add(
    layers.MaxPooling2D(
        (2,2),
        name= "max-pooling-1"
)
)
# 卷积层-2
model.add(
    layers.Conv2D(64, (3,3),
        padding= "same",
        activation= tf.nn.relu,
        name= "conv-2")
)
# 最大池化层-2
model.add(
    layers.MaxPooling2D(
        (2,2),
        name= "max-pooling-2"
)
)
# 全连接层-1
model.add(layers.Flatten(name= "fullc-1"))
# 全连接层-2
model.add(
    layers.Dense(512,
        activation= tf.nn.relu,
        name= "fullc-2")
)
# 全连接层-3
model.add(
    layers.Dense(10,
        activation= tf.nn.softmax,
        name= "fullc-3")
)
# 配置损失计算及优化器
model.compile(
    optimizer= tf.keras.optimizers.Adam(learning_rate= 0.001),
    loss= tf.keras.losses.SparseCategoricalCrossentropy(from_logits= True),
    metrics= ["accuracy"]
)
```

这里的 compile 用来配置损失函数。

损失函数的参数如表 3-15 所示。

表 3-15 损失函数的参数

optimizer	损失函数优化器,这里选择了 Adam 优化器
loss	损失函数,计算神经网络输出值和标签值的损失,这里选择交叉熵函数
metrics	测试值,一般为 accuracy

显示神经网络结构和参数:

```
model.summary()  
keras.utils.plot_model(model, "./cnn-structure.png", show_shapes= True)
```

运行结果如下:

```
Model: "MNIST-CNN"  
-----  
Layer (type)          Output Shape         Param #  
=====  
conv-1 (Conv2D)       (None, 28, 28, 32)      320  
max-pooling-1 (MaxPooling2D) (None, 14, 14, 32) 0  
conv-2 (Conv2D)       (None, 14, 14, 64)      18496  
max-pooling-2 (MaxPooling2D) (None, 7, 7, 64) 0  
fullc-1 (Flatten)    (None, 3136)           0  
fullc-2 (Dense)      (None, 512)            1606144  
fullc-3 (Dense)      (None, 10)             5130  
=====  
Total params: 1,630,090  
Trainable params: 1,630,090  
Non-trainable params: 0
```

训练神经网络:

```
# 获取前 1000 个图像数据  
train_labels= train_labels[:1000]  
# 获取前 1000 个评估使用图像  
eval_images= train_images[:1000]  
# 调整图像数据维度,供训练使用  
train_images= train_images[:1000].reshape(- 1,28,28,1)/255.0  
# 指定参数,开始训练,将训练结果存入 train_history  
train_history= model.fit(  
    train_images,  
    train_labels,  
    epochs= 10,  
    batch_size= 64,  
    verbose= 1,  
    validation_split= 0.2  
)
```

参数说明如表 3-16 所示。

表 3-16 训练神经网络参数说明

train_images	输入数据
train_labels	输出数据
epochs	训练次数
Verbose	日志显示类型,0 为标准输出,1 为进度条形式输出,2 为每个 epoch 输出一行
validation_split	用来指定训练集的一定比例数据作为诊断集
batch_size	每一个 batch 的大小
callbacks	回调函数

持久化神经网络模型:

训练神经网络是一个动态过程,在实际应用中,训练好的模型需要部署后直接使用,而不是再次训练,所以需要将动态训练好的神经网络模型持久化保存。TensorFlow 提供了简单易用的模型持久化函数。

持久化神经网络模型参数:

```
callback_params= tf.keras.callbacks.ModelCheckpoint(  
    filepath= model_path,  
    verbose= 1,  
    save_weights_only= True,  
    save_freq= 'epoch'  
)
```

参数说明如表 3-17 所示。

表 3-17 持久化神经网络模型参数说明

filepath	模型保存的列路径
Verbose	日志显示类型,0 为标准输出,1 为进度条形式输出,2 为每个 epoch 输出一行
save_weights_only	只保存权重标志位
save_freq	保存频率,epoch 是每完成一次训练,就保存一次参数。若为其他 值,则训练指定次数,再保存一次模型参数

保存运行参数:

```
tensorboard_callback= tf.keras.callbacks.TensorBoard(  
    log_dir= model_path,  
    histogram_freq= 1  
)
```

参数说明如表 3-18 所示。

表 3-18 保存运行参数说明

log_dir	日志存储路径
histogram	数据保存频率

神经网络预测：

```
Pres= Model.predict(输入数据)
Predict 返回的是预测类别列表各项目的概率,还需要对应标签列表输出预测值。
例如:
pres= model.predict(inputs)
output= tf.math.argmax(pres)
```

五、教学参考资源

参考书

1. 辛大齐. 深度学习实战——基于 TensorFlow 2.0 的人工智能开发应用[M]. 北京：中国水利水电出版社，2020.
2. Aurélien Géron. 机器学习实战：基于 Scikit-Learn、Keras 和 TensorFlow [M]. 宋能辉,李娴,译. 北京：机械工业出版社，2020.
3. 涌井良幸,涌井贞美. 深度学习的数学[M]. 杨瑞龙,译. 北京：人民邮电出版社，2019.

六、教学参考案例

参考案例

卷积神经网络

北京市第三十五中学 李娟
(2课时)

1. 教学目标
 - 阅读学习 CNN 的材料,锻炼抽象思维能力。(信息意识、计算思维)
 - 在具体实例中理解 CNN 卷积计算和池化的过程,锻炼计算思维能力。(计算思维)
 - 在具体实例数据计算过程中理解卷积神经网络的结构层次、每个层次的作用。(计算思维、数字化学习与创新)
 - 通过思考和讨论了解卷积神经网络的用途。(信息意识、信息社会责任)
2. 教学对象分析

本节课的教学对象为高一年级的学生,学生在日常生活中已经广泛接触过人工智能的产品。在前面课程的学习中,学生已经学习过人工智能的概念、机器学习的常用算法,学习过算法描述和程序设计实现的基本方法,并具有一定利用计算机编程解决问题的意

识和能力。

3. 教学重难点

教学重点:理解卷积神经网络的结构和功能。

教学难点:卷积计算、卷积核的作用、池化层的作用。

4. 教学方法

讲授法、演示法、任务驱动法。

5. 教学工具

网络机房、教学课件、项目实践程序文件、数据集文件、编程环境。

6. 教学过程设计

教学过程设计如表 3-19 所示。

表 3-19 教学过程设计

教学环节	教学内容	学生活动	设计意图
回顾引入	回顾人工智能的强大功能。提出问题: 问题 1:人工智能做一切的基础,首先是“感受到”外界。感受的方法有哪些呢? 问题 2:人工智能获取到的最常见的数据就是图像,那么计算机是怎样识别出图像是什么的呢?	学生回答(看到、听到、传感器读取周围的信息、网络获取等)	问题 1 是回顾人工智能的功能,问题 2 是引出计算机识别出图像的过程,引入本课内容
情境问题	思考人工智能是怎么识别图像内容的。对比下人类识别图像中猫咪的过程。 展示一幅猫的图片。请学生描述自己判断图中猫咪的判断依据	观察图像并回答自己判断的依据	得出判断图像内容需要根据多个特征来判断,并且特征重要性不同
问题归纳	总结人类识别的过程: 1. 找出事物的特征。 2. 每个特征的重要程度有所不同(权重不同)。 3. 多个特征判断是否符合某个事物的特征。 4. 得出可能性的判断		分析人类识别过程,得出事物特征组合、模糊特征、特征重要程度不同。和 CNN 的最后识别相呼应
阅读学习	卷积神经网络是现在流行的机器学习的一种模型结构,常用于图像识别。大家常见的图像识别背后基本都基于此原理。 提问:CNN 叫神经网络,那么又与人类的神经网络有什么关系?请阅读阅读材料后回答	阅读介绍 CNN 的学案。了解人类神经网络和 CNN 的关系,回答问题	理解 CNN 是模拟了人类识别图像的原理及过程。了解 CNN 的层次结构
问题归纳	介绍人工智能识别特征的算法——卷积神经网络 CNN		总结生物神经网络和 CNN 的相似性
实例讲解	1. 从一个问题开始:怎样识别图片中的一个字母是 X 还是 O? 思考:任何图像都是由像素组成的,是否可以简单地通过对比图像来实现?如果我们要识别的字母缩放、旋转或加粗了,看着不那么标准了,怎么识别?	请学生带着问题继续听课,并随即把问题答案填在学案上。 回答:表格对应位置数相乘后求和	通过 X 和 O 字母的判断过程,得出 CNN 的卷积和池化等层次的原理。 让学生意识到卷积计算和特征匹配的关系。

教学环节	教学内容	学生活动	设计意图																		
	<p>2. 卷积核:从 2 个不同的 X 图像中,可以发现 3 个特征小图能够描述不同的 x 的特点。我们称这种特征小图为卷积核。</p>  <p>特征小图和图片进行卷积计算</p> <p>红色部分</p> <table border="1"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>-1</td></tr> <tr><td>1</td><td>1</td><td>-1</td></tr> </table> <p>7</p> <p>绿色部分</p> <table border="1"> <tr><td>-1</td><td>-1</td><td>1</td></tr> <tr><td>-1</td><td>-1</td><td>1</td></tr> <tr><td>-1</td><td>1</td><td>-1</td></tr> </table> <p>-3</p> <p>3. 卷积计算:卷积核是通过卷积运算来判断图中特征的。</p> <p>提问:通过观察,用一句话描述卷积计算。</p> <p>提问:观察一个卷积核计算的结果,思考卷积核在不同区域计算结果的差异,这个差异代表了什么?</p> <p>4. 特征图:卷积核和图片卷积计算结果组成的二维标叫做特征图。</p> <p>提问:观察三个卷积核和计算后的特征图有什么关系?</p> <p>5. 卷积层:卷积层就是卷积核和图像做卷积运算生成特征图,特征图作为输入数据给下一层处理。</p> <p>6. 激活函数层:这一层把输入数据的值 1、所有负数全部变成了 0。</p> <p>提问:激活函数层的目的是什么?</p> <p>总结:激活函数层是强调有效特征的数据。</p> <p>7. 池化层:观察池化的过程。</p> <p>提问:池化层的目的是什么?</p> <p>总结:池化层减少了数据量、减少了计算量,把细节特征概括化了。</p> <p>8. 全连接层:所有概括特征变换队形,大排队。分别和分类器计算,得出自己的特征得分,也就是分类概率</p>	1	1	1	1	1	-1	1	1	-1	-1	-1	1	-1	-1	1	-1	1	-1	<p>回答:有特征时,计算结果值大,没特征时,计算结果小,是个负数。</p> <p>回答:特征图和卷积核很相像,越像的地方图像越亮。</p> <p>回答:突出有效的值,去掉无效值。</p> <p>回答:减少数据量</p>	<p>通过观察发现特征图和特征之间的关系。</p> <p>通过自主思考得出激活函数层作用。</p> <p>理解了卷积,池化就更容易理解了。</p> <p>全连接层不够形象,只能由教师讲解</p>
1	1	1																			
1	1	-1																			
1	1	-1																			
-1	-1	1																			
-1	-1	1																			
-1	1	-1																			
实例总结	<p>总结 5 层结构:</p> <p>输入层——目标图像进场;</p> <p>卷积层——多个特征挨个去匹配图像,得到特征图;</p> <p>激活函数层——匹配到的特征举手,未匹配的灭灯;</p> <p>池化层——特征数据太多了,选代表上台,数据合并,概括特征;</p> <p>全连接层——合并后的新特征们按原先顺序排队经过分类器投票</p>		<p>总结过程,梳理思路,让学生得出答案并填写学案</p>																		

续表

教学环节	教学内容	学生活动	设计意图
CNN 总结	卷积神经网络特点总结： 实际卷积神经网络层数很多,几十到几百层; 每一层有很多个提取不同特征的卷积核; 每一层提取出图片不同的特征,如边缘、纹理、颜色等信息; 每层的输出作为下一层的输入		强调层次关系
思考问题	CNN 这种神经网络算法适合什么类别的数据识别?除了识别图像还能用在哪些领域?(二维数组结构的图像或者类似结构的数据)	将自己的想法和结论写在学案上	
体验活动	体验 CNN 实现手写识别	体验手写数字 CNN 模型识别的过程	体验可视化手写识别过程中 CNN 各层的表现,深化对 CNN 的各层作用的理解
归纳总结	总结 CNN 的特点和应用领域	回顾和思考	通过总结梳理本课内容

人工智能未来发展

一、本章学科核心素养的渗透

学科核心素养是学科育人价值的集中体现,是学生通过学科学习而逐步形成的正确价值观、必备品格和关键能力。对学生来说,具备良好的学科核心素养比只学习理论知识更重要。高中信息技术学科核心素养由信息意识、计算思维、数字化学习与创新、信息社会责任四个核心要素组成。

在本章教学过程中,要根据所学内容的特点有意识地渗透相关核心素养的培养。落实学科核心素养时,由于教学内容不同,所承载的核心要素会有所侧重。本章内容在学科核心素养方面着重于体现在以下方面。

- 通过项目活动的开展,能够根据问题的需要自觉主动地寻求恰当的方式获取处理与人工智能相关的信息;理解人工智能对于信息技术发展的意义,学会分析一些与人工智能相关的技术事物与现象;在自己的日常生活与学习中自觉利用一些人工智能的软件工具,乐于分享对人工智能的认识,积极关注人工智能的新发展,主动探索人工智能的新想法。(信息意识)

- 通过本章的学习,能够了解人工智能的相关网站,能熟悉并选择适当的数字化学习工具查找人工智能相关资料,能掌握数字化工具的操作技能以助力本章人工智能方面知识的学习,能够利用数字化学习工具开展自主学习、合作学习,能够利用数字化工具对新一代人工智能技术和人工智能的潜在风险等内容进行交流与探讨。(数字化学习与创新)

- 通过项目实践,能够利用人工智能领域的技术、方法和思想,设计解决方案,处理或解决复杂的问题。(计算思维)

- 通过探究活动和体验思考等多种活动形式,引导学生感受人工智能给人类带来的巨大价值,深入理解人工智能可能存在的潜在危险。能够客观认识人工智能技术发展对社会生活的影响,既能形成积极、安全使用人工智能技术的观念,又要对人工智能技术应用带来的社会伦理、安全等方面的风险具有清晰的认识,形成较强的安全防护意识和责任

感,知道安全防范的基本方法和措施。(信息社会责任)

二、本章知识结构

前面我们已经初步了解了人工智能的相关概念、知道了人工智能的过去和现在,学习了人工智能的部分核心算法,基本理解并掌握了人工智能技术。作为模块教学的最后一章,我们将学习人工智能的未来发展趋势,探讨如何看待并合理使用人工智能技术、人工智能时代人类面临的新问题和新挑战、如何才能达到人机和谐共处。本章主要围绕新一代人工智能技术、人工智能潜在风险、人工智能伦理规范三部分内容展开。

“新一代人工智能技术”这一小节主要是介绍智能驾驶、智能陪伴和智能艺术创作三种新一代人工智能技术,分析新一代人工智能技术的发展方向和趋势。智能驾驶部分涉及的概念有弱人工智能、强人工智能、智能艺术、车联网技术等。智能陪伴部分涉及的概念有智能陪伴机器人。智能艺术创作部分涉及的概念有音乐情感表示、音乐情感的标签表示方法和音乐情感的维度表示方法等。

“人工智能潜在风险”这一小节,主要是引导学生客观辩证地看待人工智能技术的发展。人工智能技术的发展和应用在加速人类经济社会发展的同时,也会带来一系列的风险。这些风险需要我们妥善处理,同时也提醒我们应该客观辩证地看待人工智能技术对人类社会和思维方式带来的改变。安全隐患这部分内容涉及的概念主要有技术缺陷、管理缺陷、隐私侵犯、技术滥用等。技术奇点这部分主要涉及的概念有奇点、完全信息博弈、非完全信息博弈等。

“人工智能伦理规范”这一小节,主要是引导学生关注人工智能技术面临的伦理问题和规范使用问题。人工智能的持续进步和广泛应用带来的好处将是巨大的,但是,我们也不能忽视人工智能背后的伦理问题。随着人工智能技术的发展与广泛应用,现有法律的局限性日益凸显,法律法规必须不断做出修改和完善。随着人机共生社会的到来,作为青少年要深入思考,适当调整和规范自身行为,积极应对由此带来的新变化。本小节主要涉及的概念有人机关系、情感伦理、人机社会、机器人三定律、阿西洛马人工智能原则等。

本章主要知识与概念的关系如图 4-1 所示。

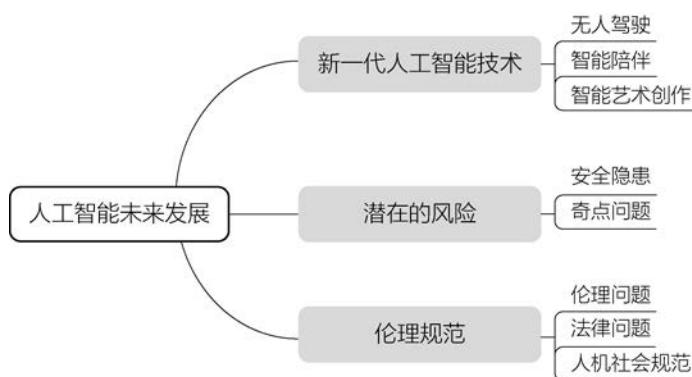


图 4-1 主要知识与概念的关系

三、本章项目活动的设计思路与实施建议

本章以“畅想未来智能生活”为主题开展项目活动。让学生通过参观、体验、实践、辩论等多种活动形式，了解人工智能对社会生活带来的巨大变化，分析未来人工智能的发展方向与特点，关注智能化社会伦理道德和法律问题。

在项目活动中，学生主要围绕以下三个任务展开学习，最后要提交一份个人项目活动总结。

任务一：结合新兴人工智能技术，认识新一代人工智能技术的特点，分析未来人工智能的发展方向。

任务二：通过调查访问或网络查阅等方式，收集人工智能技术应用可能引发的社会问题，并讨论应对策略。

任务三：利用整理好的资料，制作“我们身边的人工智能”电子作品和撰写“人工智能与社会问题”分析报告。

项目总结：上述任务完成后，每位同学需要撰写并提交个人项目活动总结。

项目实施过程中，可以将大项目分解后融入到整个单元教学中实施。实施过程中，教师需要对实施的过程有一个总体的把握，结合每一节的学习目标进行有序的安排，并通过项目将整章的内容串联和整合起来。

任务一：主要围绕新一代人工智能技术展开。任务的主要目标是认识新一代人工智能技术的特点，分析未来人工智能的发展方向。新一代人工智能技术可以结合教科书中提供的自动驾驶、智能陪伴和智能创作三个应用领域。本任务可以选择小组合作的方式开展。每小组先选择其中的一个领域进行深入的学习，活动形式据内容不同形式不一。无人驾驶可以理论与实践相结合的方式，亲自动手实践，更好地体验和了解无人驾驶领域的技术特点。智能陪伴方面可以通过网络调研、查询资料、产品体验等形式开展。智能创作可以进入到相应的网络平台亲历作品的创作过程或者阅读欣赏智能创作作品的形式展开学习。如果有时间，小组可以在三个领域转换学习。时间有限的学校，可以采取组间分享的方式，每个小组在重点学习某一领域的同时，也能了解其他组的学习内容和收获。通过本章第一小节的学习，引导学生思考新一代人工智能的特点，分析认识新一代人工智能的发展方向，做好迎接新一代人工智能技术的准备。

任务二：通过调查访问或网络查阅等方式，收集人工智能技术应用可能引发的社会问题，并讨论应对策略。本任务关注点在于人工智能技术引发的社会问题及应对策略，该任务可以在本章第二小节和第三小节学习之后完成。人工智能技术应用可能引发的社会问题既可能给人类带来安全隐患，同时也会面临伦理法律等问题。如果采取调查访问的方式，需要组员设计相关的调查问卷，可以选择现场调查或网络调查的方式。如果采取网络查阅的方式，教师可以推荐相关的网络平台，供学生参考。

任务三：利用整理好的资料，制作“我们身边的人工智能”电子作品和撰写“人工智能与社会问题”分析报告。“我们身边的人工智能”电子作品主要结合第一个任务完成。作

品形式多样,可以是视频,可以是演示文稿,还可以是电子书等。“人工智能与社会问题”分析报告主要结合第二个任务,根据调查数据及后期的研讨分析完成分析报告。

对于整个项目的开展,建议采取分组方式进行,汇集集体智慧,共同解决问题,同时增强学生团队合作意识、提升分工协作能力。

项目最后,可以安排项目交流分享会。不同学生对交流主题会有来自不同角度的看法和观点,从而对主题认识更充分和深刻;分享本组在项目实施过程中的经验和收获及待完善之处与不足,增长信息交流表达能力;实现组间相互交流,学习他组在数字化学习中的先进经验,学习他组在合作学习中的亮点,为今后更好地进行数字化学习和更好地合作学习积累经验,做好铺垫。

四、本章教学课时分配建议

本章教学课时分配建议如表 4-1 所示。

表 4-1 本章课时分配建议

节次	课时	具体内 容
第一节	2 课时	第 1 课时:智能驾驶与智能陪伴
		第 2 课时:智能艺术创作
第二节	1 课时	第 3 课时:人工智能的潜在风险
第三节	1 课时	第 4 课时:人工智能的伦理与规范

第一节 新一代人工智能技术

一、教学目标与重点

教学目标:

- 通过新技术体验、案例分析等活动,了解新一代人工智能技术跨界融合、人机协同的发展趋势,感受人工智能技术对于人们生活方式和思维模式的改变。

教学重点:

- 新一代人工智能技术的发展趋势。

二、内容分析与建议

1. 内容分析

本小节涉及的内容是人工智能领域中新兴的技术。经过 60 多年的发展，人工智能技术不断发展、变化和更迭，新一代人工智能技术在大数据和高计算能力支撑下，向更加类人的方向高速发展，体现出跨界融合和人机协同的新特征。其中有些技术还处在发展阶段，但其发展前景同样令人期待。进行本小节教学时，需要教师从人工智能领域中选择典型的人工智能新技术进行体验、学习、分析，引导学生认识到新一代人工智能技术发展趋势，了解对于新技术的主流观点和典型应用。

新一代人工智能技术这部分内容主要聚焦在人工智能技术的发展趋势和发展方向，教学时要抓住“新”字。本节主要介绍了强人工智能、情感陪伴和智能艺术创作这三种较为典型的新一代人工智能技术。本小节主要知识框图如图 4-2 所示。

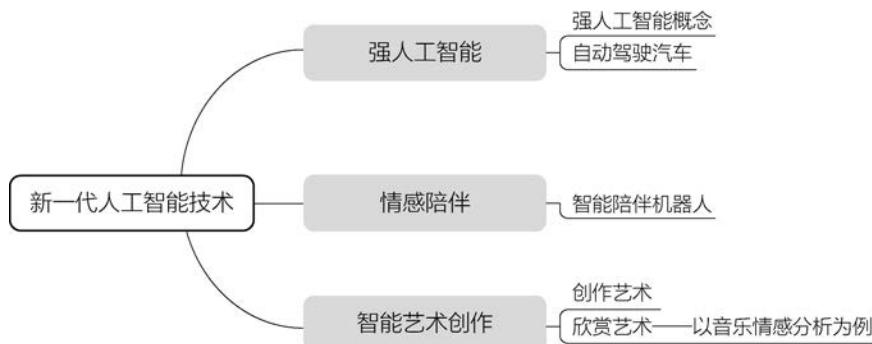


图 4-2 主要知识框图

(1) 智能驾驶

本部分内容，通过熟悉智能驾驶技术，引导学生体会理解新一代人工智能技术的发展趋势和特征。有条件的学校可以让学生亲自设计并实现智能驾驶的部分功能，理论结合实践，效果会更好。如果条件有限，教师可以通过虚拟无人驾驶仿真平台让学生感受智能驾驶，训练和验证无人驾驶汽车在不同场景下的感知和决策算法等，还可以关注智能驾驶在实际生活中的应用，如无人驾驶出租车的试运行等。有条件的学校，可以带领学生参与或亲自体验，以激发兴趣，加深印象，促进思考。目前关于智能驾驶的视频短片也比较多，也可以适时选择并利用。

(2) 智能陪伴

目前市面上智能陪伴产品各具特色。在学习这部分内容时，可以引导学生充分调研已有的智能陪伴产品，分析它们现有的功能及各自的特色，探讨用户的心理和需求，分析智能陪伴产品可能的发展方向，探讨如何做有温度的智能陪伴产品。可以采取小组合作的形式，联合制作调查问卷，采集和分析调查数据，形成调查分析报告。有条件的学校，可以在创建学科实验室时，准备多种型号智能陪伴机器人等示范教学用具，丰富人工智能教学资源。

(3) 智能艺术创作

人工智能技术能否进行智能艺术创作？通常人们认为人工智能只能完成机械重复的动作，而艺术创作一直是人类最引以为傲的智能体现。随着人工智能技术的不断发展，人工智能尝试在艺术创作领域一展伸手。智能机器是否真的能够进行艺术创作？建议教师引导学生带着这样的疑问，通过智能艺术创作的软件或平台在线体验AI写作、AI绘画、AI作曲等，亲自感受人工智能技术的魅力，同时认识到技术存在的不足。

本小节主要完成全章大项目“畅想未来生活”中的第一个任务，学习和关注新一代人工智能技术，总结新技术的特点。实施过程中建议从智能驾驶、智能陪伴、智能艺术创作三个极具代表性的领域，选择当下真实发生的、典型的案例来设计教学。如通过某自动驾驶服务正式上线这个案例，引导学生关注应用、学习原理与结构、分析技术给我们带来的影响，了解关于这项新技术的主流观点等。

2. 教学建议

教学活动方面，以学生为中心，考虑学生的接受程度、兴趣爱好，活动形式应丰富且多样化，避免大段的理论讲解与直接说教。有条件的学校，可以采取现场体验式教学，让学生亲身经历人工智能的应用，效果更明显，印象更深刻；如果条件有限，可以采取虚拟实验、案例分析等形式开展教学。在学习人工智能的相关内容时，可采用案例剖析法，通过一个个生动的案例引起学生的关注，通过剖析典型智能系统应用案例（如自动驾驶、智能陪伴、智能艺术创作等），了解新一代人工智能技术的发展方向，阐述智能技术社会化应用的巨大价值以及随之带来的伦理、安全、法律等不同层面上的复杂问题。

教学方式方面，本小节内容适合开展自主学习、分组学习、合作探究式学习。但是值得注意的是，在教学实施前教师要根据学生的接受能力，准备并制作好相关的学习资源、设计适当的学案与任务单，引导和助力学生的学习。对于新技术，学生通常会好奇，想一探究竟。教学时建议教师充分利用学生的这种好奇心并将其作为学生学习的内动力，尽可能创建良好的学习环境和氛围，促进学生积极主动地学习。

三、习题分析与提示

■ 习题 1 描述

见教科书第 109 页的体验思考。

■ 习题 1 提示

上述问题较为开放且由于每人个性不同，问题没有统一标准答案，请根据自己的实际情况回答。

■ 习题 2 描述

见教科书第 110 页第 1 个探究活动。

■ 习题 2 提示

(1) 这和人工智能技术发展的程度相关，目前智能对话、在线教育、习惯培养、日常护理等任务比较适合。过于综合和复杂的任务不太适合。

(2) (本题没有标准答案)随着人类需求的增多,陪伴机器人的功能将会越来越多,越来越复杂。陪伴机器人将会是强或超强人工智能技术的产物。

需要解决的技术难题:

- 更加敏锐的感知能力:感知周围环境的细微变化,特别是能通过人的眼神、表情、动作、脸部细微表情等读懂人的需求和想法。涉及感知技术、模式识别技术等。
- 更加精准的思考能力:陪伴机器人需要在感知到用户的变化信息后,做出适当的反应,因此需要智能系统算法精准。如果让“陪伴机器人”能像人类一样思考,需要让机器人学会学习,机器学习和深度学习等技术需要被合理应用。
- 更加敏捷的执行能力:对机器人的反应能力要求很高,需要机器人在接收到用户情绪变化的信息后迅速地做出反应。这就对智能系统的计算力、通信能力和执行力提出了更高的要求。计算力涉及微电子技术等,通信能力涉及网络传输技术等,执行力涉及伺服驱动技术等。

四、核心概念精解与技术工具介绍

1. 核心概念精解

(1) 弱人工智能系统:有些人工智能系统只能在一种和几种应用中使用,不具有通用性,而且对于数据的依赖性很大,往往需要大量的训练数据,欠缺人类举一反三的能力。我们把这类人工智能系统称为弱人工智能系统。

(2) 强人工智能系统:相对地,我们把更通用,对数据依赖性更小,具有更好学习能力,甚至自主创造能力的系统称为强人工智能系统。

(3) 车联网技术:自动驾驶汽车在行驶过程中需要跟其他车辆、路侧基础设施和行人进行无线通信,这种技术称为车联网技术。

(4) 音乐的情感表示:音乐的情感表示通常有两种方式,一种是标签表示的方式,另一种是维度空间方式。

(5) 音乐情感的标签表示方法:这种表示方法将人的面部表情与相应的音乐类型建立关联,给每一类表情赋予一个情感标签。

(6) 音乐情感的维度表示方法:这种表示方法以二维空间中的不同部位来表示音乐情感。横轴表示情感的愉悦度,纵轴表示唤醒度。愉悦度和唤醒度有正负之分,分别对应着正负半轴。

2. 技术工具介绍

(1) 神码 AI+。神码 AI+ 是基于 AI 技术的智能化写作平台,通过自主定制研发的中文分词、语法纠错、通顺性检测、上下文联想等技术,主要用于原创文章创作的智能辅助,让写作更有乐趣。

(2) 自动无人驾驶仿真软件(TASS International PreScan)。TASS 是一款模拟车辆自动驾驶的软件,可以选择道路、行人、车手、车辆类型,随后通过软件强大的控制能力对车辆行驶方案进行模拟,记录行驶的各项参数,从而在模拟后期分析车辆数据。

(3) 音虫(SoundBug)。确切地说, AI 作曲目前还处于初级阶段, 虽然有着广阔前景, 但是想替代人类来作曲和编曲, 尚需时日。不过, 现阶段利用 AI 技术来辅助人类, 在某些应用领域来开展工作是可行的, 应付低端的编曲工作也是可能的。SoundBug v2.4.2 版本中有一键 AI 编曲功能, 可以快速地为一段简单的旋律编配伴奏。

(4) iebook 超级精灵。iebook 超级精灵是一款专业的电子杂志制作软件, 可用于完成项目作品。

五、教学参考资源

■ 参考资料 1: 人工智能自主学习

“人工智能大脑”变聪明是分阶段进行的, 从机器学习进化到深度学习, 再进化至自主学习。随着 AlphaGo Zero 的出现, 不少人士认为人工智能自主学习的时代到了。AlphaGo Zero 的本质是通过对弈自动产生标注数据, 然后使用 CNN(卷积神经网络)做 DeepLearning。令人吃惊的是 AlphaGo Zero 只用了 490 万个自我对局就超过了一年前的水平, 直观的数据对比是 AlphaGo Zero 只用了 4 个 TPU 和 3 天时间就达到了 AlphaGo 用 48 个 TPU 和 3 个月的学习达到的水平。AlphaGo Zero 的成功指明了人工智能的一个发展方向, 即在规则目标明确的场景, 可以自我产生数据, 突破人类固有的思维模式, 以达到意想不到的效果。

■ 参考资料 2: 人工智能文学

近些年来, 人工智能写作技术也发展迅猛: 如某研究院研发的人工智能程序“小冰”, 写作并出版了诗集《阳光失了玻璃窗》; 清华大学研发的诗歌写作程序“九歌”, 则能够生成集句诗、近体诗、藏头诗、现代诗等不同题材的诗歌。

近来已有一些学者在研究人工智能与文学的关系, 如南开大学陶锋发表的文章《人工智能推动文学新发展》中提出了“人工智能文学”这一概念。

“人工智能文学”指的是“以人工智能程序为写作主体, 模仿人类写作的行为和机制, 自动或者半自动地生成文学作品”。

资料来源 陶锋. 人工智能推动文学新发展[N]. 中国社会科学报, 2019.

■ 参考资料 3: 人工智能文学作品欣赏——《阳光失了玻璃窗》

史上第一本机器人写作诗集《阳光失了玻璃窗》已出版。这位机器人名为“小冰”, 她在 2760 小时内写了一万多首诗。其中有 139 首入选了这本名为《阳光失了玻璃窗》的诗集。这本书共有 10 章, 每个章节都突出展现了一种人类的情感, 如孤独、期待或喜悦等。

以下两首诗节选自小冰的诗集, 来感受一下人工智能创作的现代诗。

幸福的人生的逼迫

这是一个诗人的教堂上

太阳向西方走去我被抛弃

可信的蛇会做云层鱼的声音
听不见声音的天气
若近是语言文字的艺术为自然的国人
待从我的心灵
幸福的人生的逼迫
这就是人类生活的意义

是你的声音啊

微明的灯影里
我知道她的可爱的土壤
是我的心灵成为俘虏了
我不在我的世界里
街上没有一只灯儿舞了
是最可爱的
你睁开眼睛做起的梦
是你的声音啊

■ 参考资料 4: 养老看护型智能机器人

幼儿智伴和老人看护是目前机器人市场的两大主要场景。总的来说，儿童机器人更注重教育，而养老机器人更注重实用性。

养老与机器人的结合，曾被寄予缓解整个社会养老压力的厚望。老年人需要什么样的智能养老机器人？虽然机器人养老有庞大的市场，也有政策的鼓励，一批早期进入养老机器人领域的先行者抢夺了一部分市场，但玩家大多仍处于粗放阶段，在功能方面更多与儿童需求等同，与儿童产品界限模糊。就如我们不能用健康人的思维去设计残障人士的用具，同样，我们也不能用儿童的思维去定义老年人的需求。

机器人可以结合的场景有很多，但能发展得长久的，终究是那些结合市场需求的产品，而就老年群体市场来说，物理辅助机器人是刚需。比如，老年人如厕协助类机器人就是其中一个潜在的庞大市场需求场景。一方面，老年人身躯会比较沉重，需要一两个人来协助才能完成如厕；另一方面，和孩子不同，若人工帮助老年人如厕，某种程度上会让老年人感觉失去了尊严。所以说，不同于儿童陪伴需求，物理辅助机器人在老年群体中更加实用。

人工智能养老的市场很广，但目前的技术尚无法达到能够由机器养老的程度。“人工智能养老在每个国家的需求都不一样”。有些专业人士认为，在欧美国家更多的是以社区养老为主，机器人起到基本的辅助作用；日本则是以家庭养老为主，更多的是体现人文关怀；而中国的市场则尚处于摸索当中，两者皆有，“相比起其他国家，中国的养老市场才刚刚开始”。

六、教学参考案例

■ 参考案例 1

智能驾驶与智能陪伴

北京市西城区教育研修学院 熊雪亭
(1课时)

1. 教学目标

- 通过系列活动,体验人工智能在智能驾驶和智能陪伴领域的应用。(信息意识、数字化学习与创新)
- 通过前面的交流与体验,了解新一代人工智能技术。(信息意识、计算思维、数字化学习与创新)
- 知道人工智能技术在智能驾驶和智能陪伴领域的应用。(信息意识、计算思维)
- 进一步思考新一代人工智能技术对社会的影响。(信息意识、信息社会责任)

2. 教学对象分析

高中学生已经具有较强的概括能力、学习能力,逻辑思维能力日趋严密,有一定的创新意识。对于人工智能模块有较强的求知欲。通过前面的学习,学生对人工智能技术有了初步的认识,对于智能算法和智能系统等方面有了较为细致的学习。但是对于新一代人工智能技术的发展状况还不是特别了解。

3. 教学重难点

教学重点:自动驾驶技术与智能陪伴技术。

教学难点:自动驾驶系统基本构成。

4. 教学方法

实验教学、分组学习、自主学习。

5. 教学工具

多媒体网络教室、智能机器人、自动驾驶仿真平台。

6. 教学过程设计

教学过程设计如表 4-2 所示。

表 4-2 教学过程设计

教学环节	教学内容	学生活动	设计意图
创设情境导入新课	视频案例导入: 自动驾驶案例。 	视频引入。 真实案例引导学生关注三维重建技术	以真实案例的展示引入本课,冲击力大,吸引学生的注意。 对比三维创建技术,了解三维重建技术的特点与优势

续表

教学环节	教学内容	学生活动	设计意图
	引出新一代人工智能技术的发展。 相比刷脸支付系统,自动驾驶汽车要综合使用更多的人工智能技术,对学习能力和实时计算的要求更高,是人工智能系统迈向强人工智能时代的重要里程碑。		
自动驾驶概念与组成	<p>1. 初识自动驾驶 自动驾驶是一个集环境感知、规划决策、多等级辅助驾驶等功能于一体的综合系统,它集中运用了计算机、现代传感、信息融合、通信、人工智能及自动控制等技术,是典型的高新技术综合体。</p> <p>2. 自动驾驶系统的组成 自动驾驶系统的最基本框架。</p> <p>“大脑”:人工智能芯片。 “眼睛”:传感器相当于眼睛,感知识别周边的交通标志和交通设施等。 导航:高精度地图和定位系统来准确定位地形、周边物体和道路轮廓。 通信:高精度地图对实时性要求更高,因此自动驾驶对通信有较高的要求。 外观:目前还有部分有方向盘之类的结构设计,但随着自动驾驶技术的成熟和智能化程度的提升,自动驾驶汽车将不再需要驾驶员手动控制转向、油门和刹车,朝着更人性化、更舒适和安全的方向发展。</p> <p>3. 自动驾驶技术汽车的算法 传感:从传感器原始数据中提取有意义信息。 感知:准确定位无人机所在位置以及感知现在所处的环境。 决策:确保可靠安全抵达目的地。</p> <p>4. 仿真平台体验自动驾驶</p>	<p>倾听,理解,认真观看。</p> <p>利用学案以小组合作学习的方式,自主学习:培养和提高学生的终身学习能力。</p> <p>分析智能系统的基本组成。</p> <p>分析智能系统的算法方案</p>	
小组体验	设置三个探究任务: <ul style="list-style-type: none">● 任务一:让小车自由前进、后退、左转、右转。● 任务二:能够识别并躲避障碍。● 任务三:能够识别并按路标行驶	小组合作学习,体验活动	动手实践

教学环节	教学内容	学生活动	设计意图
智能陪伴	1. 体验活动 与智能机器人“悟空”进行互动体验,感受陪伴机器人与人的情感关系。 2. 思考 思考:你对朋友的要求,可能在未来由机器人来满足吗? 3. 分析陪伴机器人采用的技术 做出与人类相似动作,用人类语言交流,感受环境变化,具备学习能力。 通过内置智能程序与算法对这些数据进行分析,为人们提供个性化服务	小组探讨交流	体验活动:活跃气氛,激发对陪伴机器人的关注。 小组活动:互助互学,增进学生之间的交流
本课小结	小结: 智能驾驶与智能陪伴机器人	整理本节课所讲内容	梳理本节课的整体内容,巩固所学

参考案例 2

人工智能与艺术创作

北京市西城区教育研修学院 熊雪亭
(1课时)

1. 教学目标

- 通过活动,体验人工智能在艺术创作领域的应用。(信息意识、数字化学习与创新)
- 通过交流与体验,了解新一代人工智能技术。(信息意识、计算思维、数字化学习与创新)
- 知道人工智能技术在艺术创作领域的应用。(信息意识、计算思维)

2. 教学对象分析

高中学生已经具有较强的概括能力、学习能力,逻辑思维能力日趋严密,有一定的创新意识。通过前面的学习,学生对新一代人工智能技术在智能驾驶和智能陪伴领域有了初步的认识,对于新一代人工智能技术的发展状况有了初步的了解,但是对于人工智能技术进行艺术创作还不是特别了解。

3. 教学重难点

教学重点:人工智能艺术创作。

教学难点:人工智能进行音乐情感分析的基本原理与方法。

4. 教学方法

讲授式、分组学习、自主学习。

5. 教学工具

资源平台、计算机网络教室、演示文稿、学案、案例等。

6. 教学过程设计

教学过程设计如表 4-3 所示。

表 4-3 教学过程设计

教学环节	教学内容	学生活动	设计意图
创设情境 引出新课	<p>艺术创作一直是人类最引以为傲的智能体现。人工智能技术能否实现艺术创作呢?</p> <p>案例引入: 人工智能挑战艺术创作。</p> <p>思考:人工智能与艺术创作之间的关系是什么?</p> <p>艺术——人工智能正在“攻占”的下一个领域。随着人工智能的飞速发展,许多科技公司、研究机构和个人都在探索利用 AI 进行艺术创作。在音乐、诗歌、绘画、舞蹈、电影、小说等领域,我们都已经可以看到 AI 的身影</p>	<p>视频引入。</p> <p>真实案例引导学生关注人工智能在艺术创作领域的发展</p>	以真实案例的展示引入本课,冲击力大,吸引学生的注意
人工智能 与美术作品创作	<p>典型案例分析: 人工智能创作的美术作品,这幅 70 厘米×70 厘米的画作来自某艺术组织。</p> 	<p>倾听,认真观看。</p>	典型案例分析。
	<p>技术分析:该艺术组织使用了基于生成式对抗网络 GAN(GAN 是一种深度学习模型)。该组织为 GAN 提供 15000 张人物画像进行训练,算法模型会分析这些在 14 世纪到 20 世纪之间创作的肖像画的特征,然后进行画像创作,直到算法无法区分画像与人类作品之间的差异为止。</p> <p>其他美术作品的创作:2018 年,在第三届机器人艺术比赛中,全球 19 个机器人团队共提交 100 多幅作品。参赛团队使用各种软件算法指导机器人完成画作,有些团队甚至使用人工智能系统独立完成原创的绘画,完全没有任何源图片或图像做参考。专业艺术评委团队基于原创性、美观度、绘画能力和技术贡献等多项指标,对作品进行了评价。</p> <p>体验(三选一):</p> <ol style="list-style-type: none">利用“AI painting”人工智能绘画系统完成 AI 美术创作。利用 AutoDraw 把粗陋的草图变成画作。利用 Python 实现图片风格迁移	<p>理解。</p> <p>分组合作学习。 三选一,体验创作</p>	从案例入手,了解人工智能在美术创作领域的进展,分析其创作的原理与过程,了解其中使用的相关技术。 体验 AI 美术创作

续表

教学环节	教学内容	学生活动	设计意图
人工智能与音乐作品创作	<p>1. 人工智能和音乐的融合,是 AI 技术在音乐创作领域的新突破,这不再是音乐专业人士的固有权力,让对音乐感兴趣的人也能通过音乐 AI 创作出专属于自己的歌曲。 AI 音乐简单理解就是人工智能创作音乐。早在 1951 年,英国计算机科学家艾伦·图灵就曾使用机器录制计算机生成的音乐,当时的计算机几乎占满整个实验室。歌曲主要是由旋律和和声构成的,人工智能作曲的原理,是它先建立一个有很多首歌的数据库,然后开发一个程序,利用这个程序可以在非常多的歌曲里面随意截取其中的某一段音频,并可以把这个旋律和和声结构出来。自那以来,由 AI 或使用 AI 创作的音乐延续到今天。</p> <p>2. 体验活动。 试听几首人工智能创作的音乐,分享自己的感受。 体验利用 SoundBug 实现一键 AI 编曲功能。可以快速地为一段简单的旋律编配伴奏。</p> <p>3. 扩展。 了解人工智能在音乐创作领域的应用。</p> 	<p>利用自主学案以小组合作学习的方式,自学完成任务。</p> <p>试听体验。</p> <p>开阔视野,了解人工智能在音乐领域的广泛应用</p>	<p>自主学习:培养和提高学生的终身学习能力。</p> <p>分组方式:互助互学,增进学生之间的交流。</p> <p>体验人工智能创作的音乐,活跃课堂气氛,加深对人工智能创作音乐方面的认识与理解。</p> <p>进一步增进人工智能在音乐领域的广泛应用</p>
人工智能与艺术欣赏	<p>引出话题: 人工智能技术尝试像人一样欣赏艺术,从而与人类更好地共鸣。下面将以音乐情感分析为例。 可以通过人工智能技术预测人类对于音乐的情感反应,在音乐推荐、音乐治疗方面都有重要的应用价值。</p> <p>具体分析: 音乐的情感表示通常有两种方式:一种是标签的方式,另一种是维度空间方式。 分析这两种方式的实现过程</p>	<p>梳理思路。</p> <p>了解人工智能在艺术欣赏方面的情况。</p> <p>倾听。 师生交互。 学习了解</p>	<p>通过经典的讲授法,以教师深入浅出的语言,师生互动的方式,了解人工智能在艺术欣赏方面的使用情况</p>
项目活动	<p>项目主题:中国音乐情感计算。 项目内容:综合运用本课程中所学的知识,尝试中国音乐的情感分析。 项目流程:</p> <ol style="list-style-type: none"> 反复聆听中国传统音乐片段小数据集,并进行手工标注。 	以小组为单位,合作实践体验	以小组为单位,进行合作学习

教学环节	教学内容	学生活动	设计意图
	2. 尝试用从本书中所学的分类模型对收集到的音乐片段进行情感计算		
探究思考	请你与同学从以下角度一起讨论交流人工智能和艺术结合的意义,如人工智能能作曲、写诗、画画等,那么: 人工智能是否具有创造力?还是只是在模仿? 人工智能是不是真的理解里面的内涵? 从人工智能的艺术作品中是否能看出机器的痕迹?	积极与老师互动,分享自己所思	进一步深入了解新人工智能技术带来的新问题。 师生互动形式,营造轻松热烈的学习气氛
本课小结	小结: 人工智能与艺术创作	利用思维导图整理本节课所讲内容	梳理本节课的整体内容,巩固所学

第二节 潜在的风险

一、教学目标与重点

教学目标:

- 通过案例分析、思考交流等活动,辩证地认识社会智能化的巨大价值和潜在风险,增强自主学习和探究学习的能力,提升智能社会的公民责任意识。

教学重点:

- 社会智能化的巨大价值和潜在风险。

二、内容分析与建议

1. 内容分析

通过前面三章的学习,我们初步了解了人工智能基本知识,学习了人工智能的部分核心技术。本章将在前三章的基础上引导学生全面认识人工智能,既认识到其优势也要关注到其潜在风险。本小节主要围绕人工智能的潜在风险话题,从安全隐患和技术奇点这两方面展开。本小节主要知识框图如图 4-3 所示。

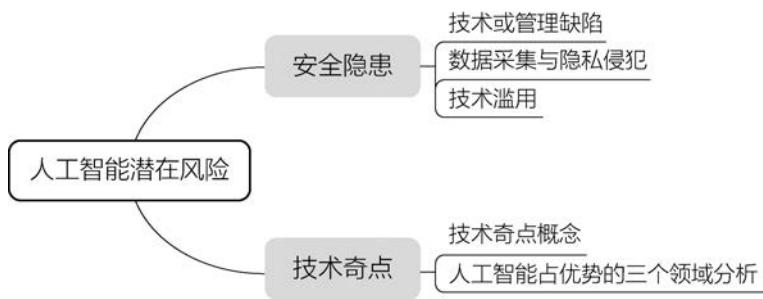


图 4-3 主要知识框图

2. 教学建议

人工智能带来的安全隐患主要从三个角度来展开:技术或管理缺陷带来的安全问题;数据采集中的隐私侵犯;技术滥用引发的安全威胁。这三个角度分别从 What(用什么样的技术)、Where(在什么场景下使用技术)、Who(什么人使用这些技术)三个视角来切入。

参考案例 1:

技术漏洞带来的安全问题可以参考与自动驾驶技术相关的案例。基于深度神经网络(DNN)的LKAS,属于L2自动驾驶系统技术。但即使是当前使用最为广泛的智能驾驶系统,在遇到一块道路补丁的情况下,可能只需要1.3秒——稍一疏忽的时间,就会失控,从而在高速行驶中转向其他车道。

参考案例 2:

现在智能音箱开始融入很多家庭的日常,为我们的生活带来了一定的便利,但大家有没有想过智能音箱可能会变成窃听器?有一种“海豚音攻击”。该攻击人耳无法听见,但通过超声波信号可以劫持智能设备的语音控制系统,再利用麦克风的硬件漏洞,就可以悄悄开启语音助手,把智能音箱变成窃听器,窃取用户隐私。

参考案例 3:

有关专家表示,当前人工智能技术存在被滥用风险,如用户在浏览、搜索、网上购物和社交时,大量软件在分析用户行为偏好。此外,人脸识别技术也被滥用,生物特征识别具有隐私性,为避免相关数据被非法使用,必须呼吁相关部门对此加强管理。

通过学习,我们认识到:人工智能技术在应用过程中要慎重又慎重,首先要尽量采用成熟的技术,不冒险使用且一定要有应急方案;其次,人工智能技术要正确地规范地使用。

技术奇点这部分内容主要是畅想未来的人工智能能否为人类掌控。思考:快速发展的人工智能会不会脱离人类智能的控制,反过来控制人类呢?如果有那么一天,人类应该如何去应对,又应该如何避免?

对于本部分内容,教师可以结合教科书设计的两个探究活动来开展。教学中可以采取视频来创设学习情境,结合科幻作品,探讨作品中想象的风险是否会成为现实。教师也可以精心准备适合的参考资料或提供相应的学习平台,引导学生自主阅读和思考;还可以将学生分为若干个小组,小组成员在共同学习、共同分析、共同讨论中加深对人工智能潜在风险的理解。教师还可以通过案例剖析来进行精讲,通过一个个生动的案例,剖析典型

智能系统应用,阐述智能技术社会化应用的巨大价值以及随之带来的伦理、安全、法律等不同层面上的复杂问题。

参考案例 4:人工智能在电子竞技中战胜人类。

最近,人工智能挑战了某款游戏的职业选手,人工智能产品与两位资格很老的人类对手进行比赛,并且以 10 比 1 的战绩打败人类职业选手。可以说这款游戏比围棋所含的信息量以及一些人为能动性高得多,人工智能都能驾驭这种信息量巨大的游戏,试想将来还有什么是它驾驭不了的呢。这些挑战和比赛足以向我们证明人工智能是多么强大,其发展将势不可挡。

项目实施方面,以小组为单位,通过调查访问或网络查阅等方式,收集人工智能技术应用可能引发的社会问题,并交流、分析给出相应的应对策略。建议参考表 4-4 制订项目评价表。

表 4-4 项目评价表

项目任务二(调查报告)评价表			
组别	组内成员		
评价项目	评分标准	评分(10 分)	优势及待改进之处
项目主题	主题明确; 内容始终围绕主题开展		
实施过程	计划详细;准备充分; 方法得当;数据可靠		
分工合作	分工明确;参与度高; 解决问题能力强; 组内交流充分,有问题能互助互帮		
项目成果	调查报告中数据充分,应对策略合理		
展示交流	形式新颖,表现力强,语言表达充分准确		
项目总分			

三、习题分析与提示

■ 习题 1 描述

见教科书第 113 页的探究活动。

■ 习题 1 提示

一切皆有可能。关于人工智能的发展一直有两种截然不同的声音,一种是认为人工智能并不可怕,将来它还会成为人类最好的帮手;一种是声称人工智能的全面发展将会威胁人类的生存发展,要对它加以约束。实际上技术的发展方向,需要人类来调节和掌控,

让其向好的方向发展。所以对于新技术,人类需要未雨绸缪,提早预防,要有忧患意识,不要掉以轻心,避免科幻影片中可怕的场景出现。

习题 2 描述

见教科书第 114 页的探究活动。

习题 2 提示

建议此活动采取小组合作方式来完成。活动前,需要确定调查对象,或者确定调查地点,在调查地点附近采取随机调查的方式。需要制作调查表(如表 4-5 所示)及确定调查方式,如纸质问卷填写还是网络在线调查。填写纸质调查问卷,又涉及数据的采集、整理、统计和分析的过程。如果是在线调查,需要确定调查平台及调查入口(分享网址还是二维码等)。调查完成后需要整理数据,分析调查结果,制作成调查报告以文字或可视化的方
式表达出来。

表 4-5 活动调查表

调 查 表						
调查日期	调查小组	备注	调查编号	调查对象姓名	智能穿戴设备	采集的生理信息

习题 3 描述

见教科书第 116 页的探究活动。

习题 3 提示

可以从如医疗领域、天气领域、人机对弈等不同领域去寻找相关资源。

案例 1:2019 年 9 月 5 日,首届全国“观云识天”人机对抗大赛正式在线启动。这是国内第一次举办的面向“全国气象 + 人工智能”的高规格赛事,此次赛事的主题为“人机争英雄,气象保民安”。大赛采取机器图像算法、人工识云“双赛道”和初赛、决赛“二级赛制”机制,面向社会各界技术人才、团队开放报名。其中,机器图像算法赛道考查参赛者的算法能力,人工识云赛道考查参赛者的观云识天能力。

案例 2:2018 年 7 月 2 日,全球首场神经影像诊断“人机大赛”在北京落幕。来自全球的 25 名神经系统疾病诊断专业人士组成“人类战队”,与人工智能应用产品展开了一场神经影像诊断“大对决”。最终,医疗人工智能产品以高出约 20% 的准确率战胜了人类战队。

四、核心概念精解与技术工具介绍

1. 核心概念

(1) 奇点

“奇点”一词的出现由来已久。在经济和科技领域,主要用来描述会产生剧变或引起爆发的临界点。对于人工智能,很多人有着同样的忧虑。随着人工智能的爆发式进展,机器智能达到人类的水平,甚至超越创造机器智能的人类。

① 完全信息博弈

完全信息博弈是指每一个参与者都拥有所有其他参与者的特征、策略及得益函数等方面准确信息的博弈。围棋就是典型的完全信息博弈。象棋、跳棋、五子棋等大部分棋类游戏,也都是完全信息博弈。

② 不完全信息博弈

不完全信息博弈是指参与者并不完全清楚有关博弈的一些信息。斗地主、21点、德州扑克、桥牌等有不明牌的纸牌类玩法以及所有的麻将玩法,都属于不完全信息博弈。

2. 技术工具

提示:由于信息技术发展迅速,提供的技术工具可能发生变化。

“中国象棋人机博弈”可用于体验人机对弈,了解人工智能在完全信息博弈领域的应用。此软件采用智能算法,开局灵活多变,中残局也具有相当的对弈水平,另有多种难度和让子设置,可增加对弈的趣味性。

五、教学参考资源

■ 参考书

1. 尼克·波斯特洛姆. 超级智能:路线图、危险性与应对策略[M]. 张体伟,张玉青,译. 北京:中信出版社,2015.
2. Stuart Russell. Human Compatible: AI and the Problem of Control [M]. London: Penguin Books, 2019.

六、教学参考案例

■ 参考案例

人工智能的潜在风险

北京市西城区教育研修学院 熊雪亭

(1课时)

1. 教学目标

通过案例分析、思考交流等活动,辩证认识社会智能化的巨大价值和潜在风险,提升

智能社会公民的社会责任意识。(信息意识、数字化学习与创新、信息社会责任)

2. 教学对象分析

高中学生已经具有较强的概括能力、学习能力,逻辑思维能力日趋严密,有一定的创新意识。通过前面的学习,学生对新一代人工智能技术有了较为全面的认识,对于人工智能技术存在的风险有所了解,但还不够全面和深入。

3. 教学重难点

教学重点:人工智能技术潜在风险。

教学难点:关注风险与规避风险。

4. 教学方法

讲授式、分组学习、自主学习。

5. 教学工具

资源平台、网络计算机教室、演示文稿、学案、案例等。

6. 教学过程设计

教学过程设计如表 4-6 所示。

表 4-6 教学过程设计

教学环节	教学内容	学生活动	设计意图
创设情境 引出新课	<p>人工智能技术的发展和应用加速了人类经济社会的发展,同时也在重塑人与机器的关系,带来了一系列可以预测和无法预测的风险。</p> <p>案例引入:</p> <p>播放科幻电影作品的片段,请同学们思考:人工智能与人类掀起的战争毁灭了世界,人却像田地里的玉米棒一样被机器大面积生产,装在一个个封闭的液体胶囊中,并通过神经系统让人类感觉到还生活在一个真实的世界(实际虚拟),目的就是让人类为机器生产源源不断的生物能。这会是人工智能导致的人类终极命运吗?</p>	<p>视频引入。</p> <p>真实案例引导学生关注人工智能可能带来的风险</p>	<p>以科幻电影作品引入本课,视觉冲击力大,吸引学生的注意,并关注本课核心话题</p>
人工智能 的安全隐患	<p>分组探究:人工智能带来的安全隐患。</p> <p>请各组同学观看微课并阅读导学任务单,完成对人工智能安全隐患来源的分析。</p> <p>案例 1:分析安徽一搬运机器人突然拦腰“抱住”工人致其伤重身亡事件。</p> <p>案例 2:人脸识别与隐私侵犯。某大学在部分试点教室安装了人脸识别系统用于日常考勤和课堂纪律管理。学生进教室后自动识别个人信息,系统自动签到签退,全程监控学生上课听讲情况,就连你发呆、抬头低头、打瞌睡和玩手机等动作行为都能被识别出来。</p>	<p>利用导学任务单、微课和文本资源探究人工智能技术带来的安全隐患。</p> <p>案例研究、讨论与剖析,总结人工智能带来的安全隐患</p>	<p>小组合作方式,方便学生互相交流,共同解决问题。</p> <p>典型案例分析。</p> <p>从案例入手,了解人工智能的安全隐患</p>

续表

教学环节	教学内容	学生活动	设计意图
	<p>案例3：骗子通过骚扰电话等录音来提取某人声音，获取素材后进行声音合成，从而可以用伪造的声音骗过对方。</p> <p>小组展示交流，通过案例分析人工智能带来的安全问题。</p> <p>师生总结：</p> <p>安全隐患来源：</p> <ol style="list-style-type: none"> 技术或管理缺陷导致的安全问题。 数据采集中的隐私侵犯。 技术滥用引发的安全威胁 		
技术奇点	<p>事实上，大多数担忧者都认为人工智能在达到某个临界点后会突然开始进化，重新设计自己，并把人类远远甩在身后。这个临界点有个特别的名字——人工智能奇点。有未来学家预言：2045年，奇点来临，人工智能会完全超越人类智能。他的论据在于呈指数发展的技术使人工智能的发展很快就会达到奇点。</p> <ol style="list-style-type: none"> 什么是奇点 <p>奇点的概念：在经济和科技领域，主要用来描述会产生剧变或引起爆发的临界点。</p> <p>对于人工智能技术奇点：</p> <p>人工智能技术奇点指的是人工智能发展会经历的一个阶段，当人工智能的发展达到奇点时，人工智能将会出现爆发式的进展。</p> <ol style="list-style-type: none"> 分组交流研讨 <p>思考：人工智能的爆发式进展导致机器智能达到人类的水平，甚至超越创造机器智能的人类，智能机器是否会按照人类设定的方式不断学习和自我优化呢？</p> <p>奇点如何到达？</p> <p>能不能到达？</p> <p>如何判断奇点的到达？</p> <ol style="list-style-type: none"> 组间分享交流 <p>请各小组分别派出代表，阐述各组的观点</p>	<p>倾听学习理解。</p> <p>思考奇点的内涵。</p> <p>分析人工智能技术奇点何时到来，如何判断奇点的到来。</p> <p>组间交流， 问题在辩中明， 观点交流与分享</p>	<p>精讲。</p> <p>各组分组研讨。</p> <p>组间交流。 观点分享</p>
课堂小结	<p>小结：</p> <p>人工智能带来的新问题。</p> <p>人工智能安全隐患来源。</p> <p>人工智能奇点问题</p>	利用思维导图整理本节课所讲内容	梳理本节课的整体内容，巩固所学

一、教学目标与重点

教学目标:

- 通过案例分析、观点分享、辩论活动、技术体验等多种方式,理解人工智能技术发展带来的伦理问题,自觉遵守智能化社会的法规,增强对智能信息系统的安全意识,提高综合应用信息技术的能力,提升信息社会责任感。

教学重点:

- 人工智能技术发展带来的伦理问题。

二、内容分析与建议

1. 内容分析

我们不仅要学习人工智能技术,还要思考如何合理使用人工智能技术。本小节主要围绕人工智能伦理与规范这个话题展开。本小节主要知识框图如图 4-4 所示。

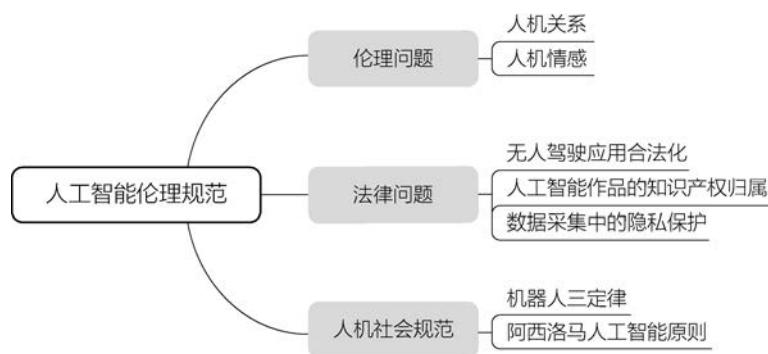


图 4-4 主要知识框图

2. 教学建议

(1) “伦理问题”教学建议

人工智能技术的发展将进一步影响人们的工作、学习和生活,给人类带来利益与便利的同时也会带来更多新的伦理问题。例如:当机器人的智能化程度大幅提升之后,人类需要思考机器人的身份问题:它是工具,还是与人类身份平等的同类?人与机器人之间的关系是什么?我们只有及时对人工智能技术的伦理问题进行反思、总结并采取相应的应对

策略,才能使人工智能技术趋利避害,为人类谋取更多福利。

多部科幻影视作品对这一问题进行了假设性的预言,引导人们关注并思考人工智能伦理问题。科幻影视作品通常具有跌宕起伏的故事情节、炫酷的视觉及声音特效等特点,非常吸引人,同时引发人们的关注与思考。教学时,我们可以充分利用影视作品的这些特点,适当地加入到教学中,引导学生思考相关话题。或许我们可以将影片中呈现的场景看作是人工智能与人的伦理关系的一种预言。

为了让同学们能够更好地表达观点,交流思想,可以采取分组的方式进行。小组学习这种方式利于学生之间充分、深入地探讨和交流。也可以引导学生在冲突辩论等形式中形成观点。

(2) “法律问题”教学建议

我们已经进入人工智能时代,大数据和人工智能的发展深刻地影响着我们的学习和生活,改变了我们的生产和生活方式。同时,人工智能技术的应用也引发了诸多的法律问题,需要我们去关注、思考。

本小节第二部分教学的目的就在于引导人们密切关注社会现实,积极回应大数据、人工智能等新兴科学技术所带来的一系列法律挑战。通过前面的学习,我们了解到无人驾驶技术正在被我们开发和利用,随之而来的是无人驾驶应用的合法化问题。无人驾驶允许上路吗?在我国现行法律中,对于自动驾驶(无人驾驶)汽车上路,尚没有相关规定。同样的,随着人工智能技术在艺术创作领域的发展,出现了不少人工智能作品如小说、诗歌、美术作品、音频作品等,这些人工智能作品的知识产权归属也成了面临的新问题。再有,随着人工智能技术的应用,出现了智能机器伤害人类或人类损毁智能机器的现象,这些也都是法律面临的新问题。这部分内容主要涉及新技术的管理与立法问题、新技术带来的知识产权问题等热点话题。随着人工智能技术的发展,人工智能引发的侵权责任问题很早就受到了学者的关注,随着人工智能应用范围的日益普及,其引发的侵权责任认定和责任承担问题将对现行侵权法律制度提出越来越多的挑战。这就需要立法部门,结合新技术所引发的新问题,及时跟进、调整法律相关内容,规范社会行为。

对于这些涉及法律的相关内容,学生可能会感觉比较抽象、枯燥、有距离感。此时教师可以选取真实发生的案例,让学生通过分析这些生动的、形象的、发生在身边的案例来引导学习,引发深入的思考。教师还可以选择一些相关的热点话题如无人车伤害了人类谁来负责等,通过让学生采取辩论、表演等多种方式去开展学习。

参考案例 1:事情发生在 2020 年 10 月 11 日,一辆白色汽车横向驶出连撞两辆汽车。这是一起看似平常的碰撞事故,但却引发了不少市民的关注。原来,事故涉及的其中一辆车为无人驾驶出租车公司的车辆。这样的案件在国内还是比较罕见的。目前对无人驾驶汽车领域的法律还是空白。

参考案例 2:2016 年 11 月,在深圳举办的第十八届中国国际高新技术成果交易会上,一台名为“小胖”的机器人突然发生故障,在没有指令的前提下自行打砸展台玻璃,砸坏了部分展台,并导致一人受伤。

参考案例 3:人工智能产品“小冰”已于 2017 年 5 月出版人工智能作品:诗集《阳光失

了玻璃窗》。这就提出了一个问题,即这些机器人创作作品的著作权究竟归属于谁?是归属于机器人软件的发明者,还是机器人的所有权人?

参考案例 4:某公园此前一直使用“年卡 + 指纹”的方式入园,但在近期园区系统升级,要求年卡用户改为“年卡 + 刷脸”方式入园。园区宣称已经以“通知”的形式告知用户:指纹取消,不注册人脸识别的年卡用户将无法正常入园。某用户不愿意被强制刷脸,并对园方的行为提出异议。面部特征等个人生物识别信息属个人敏感信息,一旦泄露并被非法滥用将对消费者造成极大的损失与危害。公园在未经用户同意的情况下,通过升级年卡系统强制收集个人生物识别信息,这严重地违反了《消费者权益保护法》等法律的相关规定,损害了用户的合法权益。

(3) “人机社会规范”教学建议

社会规范,是指调整人与人之间社会关系的行为规范。在这里我们要探讨的是人与机器人之间社会关系的行为规范。人机社会到来,人机如何相处。

在未来的机器人社会中,机器人的应用范围将更加广泛,与人类的关系将更加密切。如何构建一个和谐的人机共存社会?在人机共存的社会中,伦理道德的构建至关重要。不仅科学家要提高自身的道德素质和责任感,把握人工智能技术的正确发展方向。而且,对于越来越智能化的机器人,人们需要制订相应的规则来管理它们,同时为了规范自主机器人的发展和应用,人们需要更加详实的机器人伦理学指导原则,要让机器人融入更多的哲学思考,要与整个生态文明的发展相一致。即机器人不仅要具备针对某个问题的高智能,而且要对其行为本身进行道德伦理的规范和约束,使其不会威胁到人类以至整个生态环境的文明。

关于机器人应该遵循的行为规范和道德准则,早在人工智能初期,科幻作家艾萨克·阿西莫夫在其科幻小说中提出了“机器人三定律”。另外,2017 年有来自全球的几百名人工智能和机器人领域的专家联合签署了“阿西洛马人工智能原则”,呼吁全世界人工智能领域工作者遵守这些原则,共同保障人类未来的利益和安全。

在本部分内容教学过程中可以结合案例剖析,引导学生重视并探讨和思考相关话题。

参考案例 1:位于日本长崎的一家酒店曾获吉尼斯世界纪录认证,成为世界上第一家以机器人员工为主的酒店。机器人主要负责帮助顾客办理登记入住、搬运行李、协助调节房间灯光和温度等工作,数量一度多达 200 余个。探讨问题:机器人普及之时,人类真的要失业吗?人类应该从事什么工作?新技术对人类提出了什么更新的要求?

参考案例 2:观看与人工智能有关的影视作品。思考问题:机器人是否有可能实现自我进化,是否会跳出“机器人三定律”征服人类?是否有可能发生人类与机器人大战,机器文明终将取代人类文明?探讨问题:人工智能与人类能否和谐相处?人类如何避免此类问题的出现?如果出现,如何解决?

(4) 项目评价建议

建议参考表 4-7 制订项目评价表。

表 4-7 项目评价表

项目任务三(作品或分析报告)评价表			
组别	组内成员		
评价项目	评分标准	评分(满分 10 分)	优势及待改进之处
主题	主题明确,观点突出		
结构	结构完整,格式规范		
内容	内容充实,文笔流畅		
合作	分工合理,合作默契		
表达	表现力强,表达精准		
项目总分			

三、习题分析与提示

■ 习题 1 描述

见教科书第 117 页的体验思考。

■ 习题 1 提示

(1) 提供此案例的详细资料或获取该信息的相关链接地址。

(2) 为学生思考此话题提供一定的思考路径。

① 何为公民? 公民的权利和义务是什么?

公民是个法律概念,它代表着权利与义务。

例如:平等权和自由权,选举与被选举权。公民具有独立人格性。

② 机器人的相关特点:

我们让机器人做家务,无报酬,是否不公平? 我们选举是否要考虑机器人的意见? 机器人是否可以被选举?

(3) 设计对于此话题的思考方式。

通过讨论来阐述观点:小组讨论或正反方辩论等形式;

通过提交报告来阐述个人或小组思考成果。

■ 习题 2 描述

见教科书第 118 页的体验思考。

■ 习题 2 提示

(1) 安排分组:根据实际情况,进行分组。

(2) 活动资料:提供活动提示及活动参考资料。

我国或世界在人工智能领域方面,有没有制订相关的法律? 具体有哪些做法?

从影视作品、文学作品、新闻报道或现实生活中选择热点的或典型的相关案例,分享给学生。

(3) 活动评价:交流展示思考讨论结果。

■ 习题3描述

见教科书第119页的探究活动。

■ 习题3提示

建议以小组合作方式开展此活动。教师可以根据情况,适当地提供一些收集资料的途径;也可以设计一些问题引导学生思考、分析、讨论,完成相应的报告。

■ 习题4描述

见教科书第120页的作业练习。

■ 习题4提示

本题无标准的、统一的答案。

四、核心概念精解与技术工具介绍

1. 核心概念

(1) 机器人三定律

1940年,科幻作家艾萨克·阿西莫夫就在其科幻小说中提出了“机器人三定律”:第一条,机器人不得伤害人类或看到人类受到伤害而袖手旁观;第二条,机器人必须服从人类的命令,除非这条命令与第一条相矛盾;第三条,机器人必须保护自己,除非这种保护与以上两条相矛盾。阿西莫夫利用自己提出的“机器人三定律”,为机器人建立了一套行为规范和道德准则。

(2) 阿西洛马人工智能原则

2017年1月,来自全球的几百名人工智能和机器人领域的专家联合签署了“阿西洛马人工智能原则”;该原则目前共23项,分为科研问题、伦理和价值、更长期的问题三大类。

2. 技术工具

提示:由于信息技术发展迅速,提供的技术工具可能发生变化。

Prisma:人工智能工具软件。主要功能是通过云计算和人工智能把自己用手机拍的照片变成一幅美轮美奂的艺术照,或者是把一幅画变成一张非常仿真的照片,同时还能改变一些环境变量。

五、教学参考资源

■ 参考资料1:无人车伤害了人类,谁来负责呢?

人工智能的持续进步和广泛应用带来的好处将是巨大的。但是,为了让人工智能真正有益于人类社会,我们也不能忽视人工智能背后的伦理问题。

一辆载满乘客的无人驾驶汽车正在行驶,若遭遇一位孕妇横穿马路的突发状况,如果紧急刹车可能会翻车伤及乘客,但不紧急刹车可能会撞到孕妇。这种情况下,无人车会怎么做?

如果司机是人,那个瞬间完全取决于人的清醒判断,甚至是本能或直觉。可当人工智能陷入人类“伦理困境”的极端情景时,其每一步都是通过算法预先设定好的。

“无人车依靠的是人工智能大脑,它目前不可能做超出人类算法中所设定范围的行为决策。”浙江大学计算机学院教授吴飞说,将全国每年的交通事故数据“喂”给计算机,人工智能可以学习海量数据里隐含的各种行为模式。简单来说,就是无人车会从以往案例数据库中选取一个与当前情景较相似案例,然后根据所选取案例来实施本次决策。

但遇到完全陌生的情景,计算机会怎么办?“无人车第一个选择仍然是搜索,即在‘大脑’中迅速搜索和当前场景相似度大于一定阈值的过往场景,形成与之对应的决断。如果计算机搜索出来的场景相似度小于阈值,即找不到相似场景,则算法可约束无人车随机选择一种方式处理。”吴飞说。

“程序员可通过代码来约定无人车如何做,但这种约定始终要遵循普遍的社会伦理。这个过程中,程序员要和伦理学家一同参与把关,程序员要将符合伦理的决策用代码的形式体现出来,输入到系统中。”吴飞认为。

“当前人工智能尚未达到类人智能或超人智能水平,不能将人工智能作为行为主体对待。”浙江大学教授盛晓明说,从技术角度看,现在技术实现层次还很低,行为体出了问题肯定只能找它的设计者。另外,从哲学角度看,赋予人工智能“主体”地位很荒诞。“主体”概念有一系列限定,譬如具有反思能力、主观判断能力以及情感和价值目标设定等。人工智能不是严格意义上的“智能”。

“人工智能表现出来的智能以及对人类社会道德行为规范的掌握和遵循,是基于大数据学习结果的表现,和人类主观意识有本质的不同。人工智能不是生物,构不成行为主体,传统司法审判无法照搬到人工智能身上。因此,人工智能不可以作为社会责任的承担者。”中国社科院社会学研究所副研究员赵联飞持同样观点。

“以无人车为例,究竟由人工智能开发者负责,还是无人驾驶公司负责,甚至任何的第三方负责,或者这几者在何种情形下各自如何分担责任,应当在相关人工智能的法律法规框架下通过制订商业合同进行约定。”赵联飞说。

■ 参考资料 2:教科书中提到的 23 条阿西洛马人工智能原则

背景简介

2017 年 1 月,在美国加州阿西洛马召开的“阿西洛马会议”上,几百名人工智能和机器人领域的专家们联合签署了“阿西洛马人工智能原则”,呼吁全世界的人工智能领域工作者遵守这些原则,共同保障人类未来的利益和安全。到 2017 年 11 月 22 日,已经有多达 1200 名人工智能/机器人研究人员和其他 2000 余人签署支持这些原则。

阿西洛马人工智能原则共有 23 条,分为科研问题、伦理和价值、更长期的问题三部分。具体内容如下:

科研问题

- (1) 研究目标:人工智能研究的目标应该建立有益的智能,而不是无向的智能。
- (2) 研究资金:投资人工智能的同时,应资助那些确保其创造有益价值的研究,包括

计算机,经济学,法律,伦理学和社会学的棘手问题,诸如:

- 我们如何使得未来人工智能系统高度健全,从而不会发生故障和被入侵?
- 在维持人们的资源和目标的同时,如何通过自动化来提高我们的繁荣?
- 随着人工智能的发展,我们如何更公平和有效地修正法律系统以及管理人工智能相关的风险?
- 人工智能应该遵守哪些价值观?人工智能的法律和伦理状态应当是什么?

(3) 科学政策链接:人工智能研究人员和政策制订者之间,应形成积极的、有建设性的沟通。

(4) 研究文化:研究人员和人工智能开发人员应该培养一种互相合作、互相信任和互相透明的文化。

(5) 规避一蹴而就:开发人工智能系统的团队应积极相互合作,以避免在达成安全标准上偷工减料。

伦理和价值

(6) 安全性:人工智能系统应当是安全的,且是可适用的和可实行的。

(7) 故障透明:如果一个人工智能系统引起损害,应该有办法查明原因。

(8) 审判透明:在司法裁决中,但凡涉及自主研制系统,都应提供一个有说服力的解释,并由一个有能力胜任的人员进行审计。

(9) 职责:高级人工智能系统的设计者和建设者是系统利用、滥用和行动的权益方,他们有责任和机会塑造这些道德含义。

(10) 价值观一致:应该设计高度自主的人工智能系统,以确保其目标和行为在整个运行过程中与人类价值观相一致。

(11) 人类价值观:AI系统的设计和运作应符合人类尊严,权利,自由和文化多样性的理念。

(12) 个人隐私:既然人工智能系统能分析和利用数据,人们应该有权利存取、管理和控制他们产生的数据。

(13) 自由与隐私:人工智能在个人数据的应用不能无理削减人们的实际或感知的自由。

(14) 共享利益:人工智能技术应该尽可能地使更多人受益和授权。

(15) 共享繁荣:人工智能创造的经济繁荣应该广泛的共享,造福全人类。

(16) 人类控制:人类应该选择如何以及是否代表人工智能做决策,用来实现人为目标。

(17) 非颠覆:通过控制高级人工智能系统所实现的权力,应尊重和改善健康社会所基于的社会和公民进程,而不是颠覆它。

(18) 人工智能军备竞赛:应该避免一个使用致命自主武器的军备竞赛。

更长期的问题

(19) 性能警示:因为没有达成共识,我们应该强烈避免关于未来人工智能性能的假设上限。

(20) **重要性:**高级人工智能可以代表地球上生命历史的深奥变化,应该计划和管理相应的资源。

(21) **风险:**对于人工智能造成的风险,尤其是那些灾难性的和存在价值性的风险,必须付出与其所造成的影响相称的努力,以用于进行规划和缓解风险。

(22) **递归自我完善:**那些会递归地自我改进和自我复制的 AI 系统若能迅速增加质量或数量,必须服从严格的安全控制措施。

(23) **共同利益:**超级智能应当只为广泛分享道德理想而发展,是为了全人类而不是为了一个国家或者一个组织的利益。

六、教学参考案例

■ 参考案例 1

人工智能应用规则与伦理

北京市第三十五中学 贾 轩

(1 课时)

1. 教学目标

- 通过探究活动,辩证认识人工智能对人类社会发展的巨大价值和潜在威胁。(信息意识)
- 在案例分析及讨论等学习活动中,了解机器人三定律,从而自觉维护和遵守人工智能社会应用的规则,树立正确的价值观,培养规则意识和责任意识。(数字化学习与创新、信息社会责任)

2. 教学对象分析

学生经历前段时间的学习,对于现阶段人工智能发展和应用有了一定的了解和认识,但是对一些概念还不是十分明晰。

3. 教学重难点

教学重点:机器人三定律、人工智能应用规则。

教学难点:人工智能应用带来的伦理问题、社会问题、法律问题与可能面对的挑战。

4. 教学方法

讲授法、(视频)案例分析法、讨论法。

5. 教学工具

资源平台、网络计算机教室、演示文稿、学案、案例等。

6. 教学过程设计

教学过程设计如表 4-8 所示。

表 4-8 教学过程设计

教学环节	教学内容	学生活动	设计意图
设问激趣，引入课题	<p>1. 教师通过提问,让学生谈谈生活学习中都有哪些情境应用到了人工智能。</p> <p>2. 教师通过播放影片,激发学生兴趣,从而引出问题:人工智能除了会为我们带来便利以外,还会带来其他吗?(播放电影解说)</p>	<p>学生思考并回答,比如扫地机器人、智能音箱、智能汽车、指纹锁、面部识别解锁、语音助手等。</p> <p>学生思考后回答:机器人不受控制,攻击人类;机器人如果杀人,可能不会受到法律的制裁;机器人最后和人类成为了朋友;等等</p>	<p>用生活中的例子引出人工智能。</p> <p>通过影片激发学生兴趣,引出人工智能可能带来的问题</p>
分析讨论,获取新知	<p>1. 人工智能应用带来的问题 (1) 总结归纳为三大类问题:伦理、社会、法律。 (2) 教师让学生分成三组,分别讨论以上三个问题并讨论解决办法,选出代表进行总结交流。 (3) 教师邀请各小组代表对本小组讨论得出的问题以及解决办法进行分享。 伦理方面:情感问题、机器人能不能被看作是真正的人。 社会方面:以电影中的例子来说,如果机器人的程序受到干扰或者破坏,将影响到它的行为,做出一些违背其本身规则的事情,对社会治安造成一定的影响,甚至如果众多的机器人落入到坏人手里,将有可能成为战争武器。 法律方面:电影中的情节说明,如果机器人杀了人,怎么样进行处罚,这是一个需要去探讨的问题,又或者交通方面,机器人造成事故,责任由谁来承担。 (4) 教师总结学生发言,提出有些方面并没有完全说到,教师通过以下几个案例进行补充。 ① 人工智能应用带来的伦理问题 教师通过新闻播放,引出案例“2017年10月26日,沙特阿拉伯授予机器人索菲亚公民身份。索菲亚拥有仿生橡胶皮肤,可模拟62种面部表情,能够识别面部,并与人进行眼神接触,能够理解语言并与人类互动。”说明除了情感方面,还有可能存在的伦理问题,比如身份伦理,机器人是否能够与人类平等对待,二者之间的关系是什么?这也是我们需要去关注的。 ② 人工智能应用带来的社会问题</p>	<p>分组讨论。</p> <p>总结。</p> <p>探讨。</p> <p>学生在教师讲解时思考这些问题是否存在,还有没有其他伦理问题,如果有,可以提出一起探讨。</p> <p>学生在教师讲解时思考这些问题是否存在,还有没有其他社会问题,如果有,可以提出一起探讨。</p>	<p>学生通过小组讨论,了解合作的必要性,并通过合作学习,探究出人工智能应用带来的各项问题,并讨论解决办法。</p> <p>锻炼学生表达、总结归纳能力,并思考问题。</p> <p>根据真实案例,思考人工智能应用带来的伦理问题。</p> <p>通过新闻报道,思考人工智能应用带来的社会问题。</p> <p>通过视频,激发学生兴趣,引起学生思考人工智能应用带来的法律问题。</p>

续表

教学环节	教学内容	学生活动	设计意图
	<p>教师介绍案例:2017年8月8日21时19分,四川九寨沟县发生7.0级地震。21时37分15秒中国地震台网机器人自动编写稿件,仅用25秒出稿,540字并配发4张图片!内容包括速报参数、震中地形、热力人口、周边村镇、周边县区、历史地震、震中简介、震中天气等8大项。引出可能面临的社会问题:新闻报道失衡,违背新闻报道的基本原则。再比如很多人将面临失业、算法歧视、战争机器等。</p> <p>③人工智能应用带来的法律问题 教师播放视频:有人乘坐无人驾驶汽车参加某大型聚会。引导学生思考可能产生的问题,比如失控会导致交通事故、违反交规、遇到灾难天气时判断出现失误等。面对这样的问题,如何去解决?</p> <p>2. 机器人三定律 教师引导学生分析:提出人工智能应用需要一定规则去约束,引导学生思考电影中贯彻始终,一直提到的一个规则:三大定律,也就是机器人三原则。 第一条:机器人不得伤害人类,或看到人类受到伤害而袖手旁观。 第二条:机器人必须服从人类的命令,除非这条命令与第一条相矛盾。 第三条:机器人必须保护自己,除非这种保护与以上两条相矛盾。</p> <p>3. 人工智能应用规则 教师通过提问,让学生认识到现有的三大原则并不能涵盖整个人工智能领域,需要有更多的规则去解决这些问题。引出阿西洛马人工智能原则</p>	<p>学生思考解决方法,并提出应对策略。 教师根据学生回答,引出北京细则、上海办法。同时提出了法律方面,还有可能存在比如人工智能编曲、写作等有关知识产权的问题</p>	综合上述问题,总结归纳需要规则去约束完善人工智能,引出机器人三原则
总结归纳,内化提升	<p>教师利用一些专家学者对于人工智能的看法,引出这些问题不能仅仅依靠我们去思考,而是需要更多人参与到其中,为社会添一份力。 学生总结归纳本节课学习内容,辩证认识人工智能对人类社会发展的巨大价值和潜在威胁,了解机器人三定律,从而自觉维护和遵守人工智能社会应用的规则,树立正确的价值观,为社会主义新时代增添力量</p>		让学生了解人工智能应用规则,树立正确的价值观,培养学生的规则意识、责任意识,为社会增添力量

参考案例 2

人工智能的伦理规范

北京市西城区教育研修学院 熊雪亭

(1课时)

1. 教学目标

- 通过案例交流等活动,研讨未来的人机关系,思考人机社会规范。(信息意识、数字化学习与创新)
- 通过头脑风暴等活动,了解智能时代带来的新问题,提高智能社会的公民忧患意识和责任意识。(信息社会责任)

2. 教学对象分析

高中学生具备一定的思考能力和合作探究能力,逻辑思维能力也日趋严密。通过前面的学习,学生对新一代人工智能技术有了较为全面的认识,学习了相关的算法,对于人工智能技术存在的风险有所了解,但对于智能时代引发的伦理问题,缺乏深入的认识。

3. 教学重难点

教学重点:机器人三定律、阿西洛马人工智能原则相关内容;人工智能带来的伦理规范问题。

教学难点:人工智能带来的伦理规范问题。

4. 教学方法

案例教学、讲授式、分组学习。

5. 教学工具

资源平台、网络计算机教室、演示文稿、学案、案例等。

6. 教学过程设计

教学过程设计如表 4-9 所示。

表 4-9 教学过程设计

教学环节	教学内容	学生活动	设计意图
创设情境 引出新课	问题引入: 随着科学技术的发展,人工智能进入了更多的领域。请你畅想一下,人工智能将会走向何方?未来智能社会将会是怎样的场景? 请同学们先分小组交流讨论,围绕“3个1”展开,稍后各组进行观点的分享交流。 1画:构思一幅最能代表未来社会场景的画面; 1优:将未来人工智能社会与现在社会进行对比,谈谈其优越性(选一点阐述); 1问:谈谈未来人工智能社会出现的新问题(选一点阐述)。 教师结合各组发言进行适当的引导与点评随	问题引入。 引发学生的思考	热点问题引入,吸引学生的注意,从而关注本课核心话题

续表

教学环节	教学内容	学生活动	设计意图
人机关系	<p>着人工智能的发展,它在与人类的深度融合中产生了人际关系问题。</p> <p>智能时代的人机关系指的是人与机器、工具之间的关系。人机关系主要包括“人机”关系重思与“人机”边界重构。人机之间是互补还是替代?有一种观点是人机共生,并且根据共生情况构建出互利共生、偏利共生、偏害共生和吞噬取代四种范式。你对此观点是否认同?</p> <p>小组活动任务:</p> <p>思考:智能时代,人与机器之间是怎样的一种关系?</p> <p>活动流程:</p> <p>(1)个人静思:请个人先静心思考,可上网查阅相关资料;</p> <p>(2)小组共商:各小组成员之间进行头脑风暴,充分探讨交流,形成小组观点。</p> <p>小组交流展示</p>	<p>倾听。</p> <p>思考。</p> <p>观看。</p> <p>讨论与分析,总结人工智能带来的新问题</p>	小组合作的方式,方便学生互相交流,共同解决问题
人工智能带来的伦理问题	<p>1. 伦理 伦理是指人伦道德之理,主要指人与人相处的各种道德准则。人工智能的伦理学讨论的不再是人与人之间的关系,也不是人与自然界之间的关系,而是人类与自己所发明的一种产品所构成的关联。</p> <p>2. 道德伦理 思考并讨论: 我们如何创造“好的”人工智能?机器合乎道德意味着什么?我们如何在道德的基础之上使用人工智能?</p> <p>3. 情感伦理 随着人工智能的发展,会有新的问题出现。机器人发展到高级阶段,可以与人进行沟通与交流,就可能会产生情感上的需求,引出情感伦理问题。</p> <p>结合近两年热点案例,分析思考:随着人工智能的发展,人与机器人之间会产生情感吗?如果有,应该如何处理这种情感?机器人在人心中的位置是工具、机器,还是平等的人类?如何看待机器人索菲亚获得公民身份,她是否拥有平等的公民权利?</p> <p>教师总结并分享中国国家新一代人工智能治理专业委员会发布的《新一代人工智能伦理规范》(以下简称《规范》)。《规范》旨在将伦理道德融入人工智能全生命周期,为从事人工智能相关活动的自然人、法人和其他相关机构等提供伦理指引。</p> <p>各小组学习、探讨交流《规范》,并发表观点科</p>	<p>倾听。</p> <p>学习。</p> <p>理解。</p> <p>思考。</p> <p>组内讨论。 组间交流, 问题在辩中明, 观点交流与分享</p>	精讲。 各组分组研讨, 组间交流, 观点分享

续表

教学环节	教学内容	学生活动	设计意图
人机社会的规范	<p>学家一直希望：以机器人为代表的人工智能不给人类带来任何威胁。</p> <p>早期的机器人三定律：1940年，科幻作家阿西莫夫就在其科幻小说中提出了机器人三定律。</p> <p>阅读教科书，了解机器人三定律。</p> <p>思考：</p> <p>机器人三定律存在缺陷和漏洞吗？各小组交流讨论。</p> <p>后来，人们补充了各种不同版本的机器人定律，目前比较成型的机器人定律体系如下：</p> <p>元原则：机器人不得实施行为，除非该行为符合机器人原则。（防止机器人陷入逻辑两难困境而当机）第零原则：机器人不得伤害人类整体，或者因不作为致使人类整体受到伤害。第一原则：除非违反高阶原则，机器人不得伤害人类个体，或者因不作为致使人类个体受到伤害。第二原则：机器人必须服从人类的命令，除非该命令与高阶原则抵触。机器人必须服从上级机器人的命令，除非该命令与高阶原则抵触。（处理机器人之间的命令传递问题）第三原则：如不与高阶原则抵触，机器人必须保护上级机器人和自己之存在。第四原则：除非违反高阶原则，机器人必须执行内置程序赋予的职能。（处理机器人在没有收到命令情况下的行为）繁殖原则：机器人不得参与机器人的设计和制造，除非新机器人的行为符合机器人原则。（防止机器人通过设计制造无原则机器人而打破机器人原则）</p> <p>阿西洛马人工智能原则：</p> <p>2017年1月，来自全球的几百名人工智能和机器人领域的专家联合签署了阿西洛马人工智能原则，呼吁全世界的研究者和研究机构在开展人工智能领域研究的同时严格遵守这些原则，共同保障人类未来的伦理、利益和安全。该原则目前共23项，分为科研问题、伦理和价值、更长期的问题三大类</p>	<p>阅读理解。</p> <p>思考。</p> <p>交流讨论。</p> <p>思考每条原则的加入理由。</p>	<p>了解原则。</p> <p>思考这些定律存在的意义。</p>
辩论环节	<p>新问题：</p> <p>如何面对人工智能的伦理问题？</p> <p>人工智能“威胁论”。</p> <p>支持与反对双方辩论。</p> <p>各小组探讨交流并发表各组观点</p>	<p>倾听。</p> <p>辩论。</p> <p>思考</p>	<p>阅读阿西洛马人工智能原则。</p> <p>以小组为单位讨论这些原则的内容及存在的意义</p>
课堂小结	<p>小结：</p> <p>人工智能带来新的法律问题</p>	<p>利用思维导图整理本节课所讲内容</p>	<p>梳理本节课的整体内容，巩固所学</p>

参考案例3

人工智能带来的法律问题

北京市西城区教育研修学院 熊雪亭

(1课时)

1. 教学目标

- 通过案例分析、自主学习等活动,了解人工智能与人格权、知识产权、数据财产保护等之间的关系,辩证认识社会智能化的巨大价值和潜在风险。(信息意识、数字化学习与创新)
- 在分组学习过程中,了解智能时代带来的新问题尤其是法律方面问题,提升智能社会的公民责任意识。(信息社会责任)

2. 教学对象分析

高中生已经具有较强的概括能力、学习能力,逻辑思维能力日趋严密,有一定的创新意识。通过前面的学习,学生对新一代人工智能技术有了较为全面的认识,对于人工智能技术存在的风险有所了解,但对于智能时代的法律问题还需要全面学习了解。

3. 教学重难点

教学重点:人工智能带来的法律问题。

教学难点:人工智能带来的法律问题。

4. 教学方法

案例教学、讲授式、分组学习、自主学习。

5. 教学工具

资源平台、网络计算机教室、演示文稿、学案、案例等。

6. 教学过程设计

教学过程设计如表4-10所示。

表4-10 教学过程设计

教学环节	教学内容	学生活动	设计意图
创设情境 引出新课	<p>视频案例:</p> <p>在深圳举办的第十八届中国国际高新技术成果交易会上,一台名为“小胖”的机器人突然发生故障,在没有指令的前提下自行打砸展台玻璃,砸坏了部分展台,并导致一人受伤。这究竟该由谁来承担法律责任?</p> <p>思考:</p> <p>我们已经进入人工智能时代,大数据和人工智能的发展深刻地影响着我们的社会生活,改变了我们的生产和生活方式,也深刻地影响社会的方方面面。但同时,它们也提出了诸多的法律问题</p>	<p>视频引入。</p> <p>真实场景的案例更能代入感,更能激发学生的兴趣,引发学生的思考</p>	<p>以真实案例引入本课,视觉冲击力大,吸引学生的注意,从而关注本课核心话题</p>

续表

教学环节	教学内容	学生活动	设计意图
人工智能与人格权保护	<p>人格权保护问题。</p> <p>案例：</p> <p>分析安徽一搬运机器人突然拦腰“抱住”工人致其伤重身亡事件。</p> <p>请各组观看，并进行分析总结。</p> <p>总结：人工智能能系统模仿他人的声音、形体动作等，甚至能像人一样表达，并与人进行交流。但人工智能如果未经他人同意而擅自进行模仿活动，有可能构成对他人人格权的侵害。人工智能借助光学、声音控制、人脸识别等技术对他人人格权客体加以利用，也对声音、肖像等权利的保护提出了新的挑战。</p> <p>新问题：</p> <p>随着机器人伴侣的出现，虐待、侵害机器人伴侣的情形下，行为人是否应当承担侵害人格权以及精神损害赔偿的责任？要不要赋予机器人主体资格或部分权利？</p> <p>请各小组探讨交流并发表各组观点</p>	<p>小组合作。</p> <p>利用导学任务单、微课和文本资源探究人工智能技术带来的安全隐患。</p> <p>典型案例分析。</p> <p>案例研究、讨论与剖析，总结人工智能带来的安全隐患</p>	<p>小组合作的方式，方便学生互相交流，共同解决问题。</p> <p>从案例入手，了解人工智能的安全隐患</p>
人工智能与知识产权	<p>人工智能能够实现艺术创作，引出知识产权问题。</p> <p>新问题：</p> <p>机器人“小冰”已于2017年5月出版人工智能作品：诗集《阳光失了玻璃窗》。产生新问题：这些机器人创作作品的著作权究竟归属于谁？是归属于机器人软件的发明者，还是机器人的所有权人？要不要赋予机器人一定程度的法律主体地位从而由其享有相关权利？</p> <p>各小组探讨交流并发表各组观点</p>	<p>倾听、学习、理解。</p> <p>组间交流， 问题在辩中明， 观点交流与分享</p>	<p>精讲。</p> <p>各组分组研讨， 组间交流， 观点分享</p>
人工智能与数据财产保护	<p>新问题：</p> <ol style="list-style-type: none"> 1. 人工智能系统进行机器学习，需要以海量数据作为支撑，如何规范数据的收集、储存、利用行为、避免数据的泄露和滥用，并确保国家数据安全？ 2. 人工智能算法使用中如何有效规范算法及其结果的运用，避免侵害他人权利？需要法律如何应对？ <p>各小组探讨交流并发表各组观点</p>	<p>倾听。</p> <p>思考。</p> <p>组间交流</p>	<p>各组分组研讨。 组间交流。 观点分享</p>
人工智能与侵权责任认定	<p>人工智能引发侵权责任问题。</p> <p>如本课开始提到的案例，机器人伤人事件，究竟谁来负责？</p> <p>前不久，无人驾驶公交线路测试，如果发生交通事故应当由谁承担责任？法律上是否有必要为无人驾驶机动车制订专门的责任规则？</p> <p>各小组探讨交流并发表各组观点</p>	<p>倾听。</p> <p>思考。</p> <p>组间交流</p>	<p>各组分组研讨。 组间交流。 观点分享</p>

续表

教学环节	教学内容	学生活动	设计意图
人工智能与机器人的法律主体地位问题	案例:第一位机器人公民索菲亚。 随着机器人的智能程度越来越高,提出了一个新的法律问题,即我们将来是否有必要在法律上承认人工智能机器人的法律主体地位? 各小组探讨交流并发表各组观点	观看视频案例。 思考。 组间交流	视频案例。 各组分组研讨。 组间交流。 观点分享
课堂小结	小结: 人工智能带来新的法律问题	利用思维导图整理本节课所讲内容	梳理本节课的整体内容,巩固所学

经上海市中小学教材审查委员会审查
准予使用 淮用号 II-GJ-2022023

责任编辑：王健



绿色印刷产品

ISBN 978-7-5760-2954-3

A standard EAN-13 barcode representing the ISBN number.

9 787576 029543 >

定价：28.50 元