

# 揭阳职业技术学院

Jieyang Vocational & Technical College

## 教案

系（部）： 化学工程系

讲授课程： 人工智能技术

任课教师： 吴晓畅

专业班级： 应用化工技术251/252、新能源材料应用技术251

授课学期： 2025-2026 学年第一学期

揭阳职业技术学院化工系

2025 年 9 月

课程名称	人工智能技术	教案编号	01
教材名称	人工智能导论	专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第一课：人工智能的前世今生		
授课题目	人工智能的前世今生		
授课学时	2节（ ）；3节（ <input checked="" type="checkbox"/> ）；其它（ ）		
课 型	理论（ <input checked="" type="checkbox"/> ）；实验（ ）；见习（ ）；实训（ ）；其它（ ）		
教学目的	<ol style="list-style-type: none"> <li>1. 了解人工智能的萌芽与早期发展。</li> <li>2. 掌握人工智能元年及其重要事件。</li> <li>3. 理解人工智能的迭起兴衰过程。</li> <li>4. 熟悉中国人工智能的发展历程及现状</li> </ol>		
教学重点	<ol style="list-style-type: none"> <li>1. 人工智能概念的理解</li> <li>2. 人工智能早期发展的掌握</li> </ol>		
教学难点	<ol style="list-style-type: none"> <li>1. 人工智能“冬眠”期的分析。</li> <li>2. 中国人工智能发展的评价</li> </ol>		
课程思政	<ol style="list-style-type: none"> <li>1. 科技强国教育：以中国人工智能领域取得的突破性成果为例，如“萤火一号”训练平台和DeepSeek-V3模型，展现中国在人工智能领域的实力，激发学生的民族自豪感和投身科技事业的热情。</li> <li>2. 民族自信：介绍中国智能科学研究的领军人吴文俊的贡献，展现中国智慧的博大精深，增强学生的民族自信心。</li> <li>3. 创新精神：鼓励学生关注人工智能领域的最新研究进展，勇于探索未知领域，提出新的问题和解决方案，培养他们的创新意识和实践能力。</li> </ol>		
教学方法	讲授（ <input checked="" type="checkbox"/> ）；讨论（ <input checked="" type="checkbox"/> ）；指导（ ）；示教（ ）；其它（ ）		
电子教案	有（ <input checked="" type="checkbox"/> ）	Microsoft PowerPoint（ <input checked="" type="checkbox"/> ）；Author ware（ ）；其它（ ）	
	无（ ）		
教学资源	多媒体（ <input checked="" type="checkbox"/> ）；模型（ ）；标本（ ）；实物（ ）；音像（ <input checked="" type="checkbox"/> ）；其它（ ）		
教学过程时间安排	3学时		
思考题	<ol style="list-style-type: none"> <li>1. 人工智能“元年”中，“元年”指的是什么？</li> <li>2. 下载几款国产的AI智能大语言模型，体验其功能，思考总结他们的区别。</li> </ol>		

# 第一课：人工智能的前世今生（3 学时）

## 授课老师：吴晓畅

### 一、课程导入

1. 引入人工智能的概念，并简要介绍课程内容。
2. 提出问题：人工智能是如何改变我们的生活？
3. 通过问题导入，激发学生的学习兴趣，并引出本节课的主题。

### 二、人工智能的应用案例

以普通上班族的一天为例，展示人工智能在不同场景中的应用，如智能家居、智能交通、智能办公等。

详细讲解每个案例中人工智能技术的具体应用，例如：

- (1) 智能家居：语音助手、智能家电、智能安防等。
- (2) 智能交通：自动驾驶、智能导航、智能交通信号灯等。
- (3) 智能办公：智能会议系统、智能文档处理、智能客服等。

通过具体案例，让学生直观地感受人工智能带来的便利和效率提升，并思考人工智能对未来的影响。

### 三、相关人物介绍

介绍中国人工智能领域的杰出人物，如吴文俊院士、梁文锋等，展现中国在人工智能领域的成就和贡献。

详细介绍每位人物的主要成就和贡献，例如：

吴文俊院士：开创了数学机械化研究领域，提出的用计算机证明几何定理的方法在国际上被誉为“吴方法”。

梁文锋：DeepSeek创始人，带领团队自主研发了“萤火一号”训练平台和DeepSeek-V3模型，性能超过了ChatGPT 4.0。

通过人物故事，激发学生的民族自豪感和投身科技事业的热情，并树立榜样作用。

## 四、项目知识介绍

### 1. 人工智能的萌芽和早期发展

(1) 思想、理论与物质基础：从古代哲学家亚里士多德的三段论到现代科学归纳法、符号逻辑、神经网络模型、图灵机等，这些思想和理论为人工智能的发展奠定了基础。

(2) 早期文学作品中的“机器人”：如莱曼·弗兰克·鲍姆的《机器人小滴答》，展现了早期人类对人工智能的想象和探索。

(3) 图灵机与图灵测试：图灵机作为理想计算机模型，图灵测试作为衡量人工智能智能程度的标准，对人工智能的发展具有重要意义。

### 2. 人工智能元年

(1) 思考题：1. 人工智能“元年”中，“元年”指的是什么？

(2) 达特茅斯会议：1956年，达特茅斯会议的召开标志着人工智能学科的诞生，并确立了人工智能的研究方向和目标。

(3) 会议主题：会议围绕自动计算机、自然语言处理、神经网络、计算理论、自我提升、抽象概念、随机性和创造力等七个议题展开讨论，为人工智能的发展奠定了基础。

### 3. 人工智能的迭起兴衰

(1) 发展阶段：人工智能的发展历程可以分为六个阶段：第一次繁荣（1956-1974）、第一次“寒冬”（1974-1980）、重振旗鼓（1980-1987）、第二次寒冬（1987-1993）、稳健时代（1993-2011）、走向新的繁荣（2012至今）。

(2) 关键事件：每个阶段都有标志性的事件，如机器人、专家系统、深蓝战胜国际象棋冠军、AlphaGo战胜围棋冠军等，反映了人工智能技术的发展和应用。

### 4. 中国人工智能的发展

(1) 发展历程：中国人工智能的发展经历了起步、发展和快速提升三个阶段，从20世纪80年代初的探索，到21世纪的快速发展，中国人工智能取得了长足的进步。

(2) 成就与现状：中国在人工智能领域取得了许多重大成就，如华为、大疆等企业在人工智能硬件领域的突破，阿里巴巴、百度、腾讯等互联网巨头在人工智能领域的布局。

(3) 未来展望：中国政府高度重视人工智能发展，出台了一系列政策支持人工智能产业发展，未来中国人工智能发展前景广阔。

## 五、项目小结与展望

本项目我们学习了人工智能的发展史，从公元前伟大的哲学家亚里士多德的三段论的形式逻辑，到科学归纳法、数量逻辑、思维规律机械化、神经网络模型、计算机数学模型、世界上第一台电子计算机等成就奠定了人工智能发展思想、理论和物质基础。

人工智能历史上的第一座里程碑是图灵机与图灵测试；以时间轴看，从人工智能元年（1956年）到今天的2021年，人工智能经历了第一次繁荣（1956-1974）、第一次“寒冬”（1974-1980）、重振旗鼓（1980-1987）、第二次寒冬（1987~1993）、稳健时代（1993-2011）和走向新的繁荣（2012-今）的曲折的发展阶段后才有了今天的辉煌。

以史为鉴，面向未来，毫无疑问，人工智能的崭新时代已经来临，我们未来的生活和工作都离不开它。作为大学生，我们需要了解、认识人工智能，下一单元，我们会带大家从零开始认识人工智能。

## 六、课程作业

布置课后任务：下载几款国产的AI智能大语言模型，体验其功能，并注意提问的方式和方法。分析他们的区别。

课程名称	人工智能技术		教案编号	02
教材名称	人工智能导论		专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第二课：从零开始认识人工智能			
授课题目	AI的概念与基础技术			
授课学时	2节（ ）；3节（ <input checked="" type="checkbox"/> ）；其它（ ）			
课 型	理论（ <input checked="" type="checkbox"/> ）；实验（ ）；见习（ ）；实训（ ）；其它（ ）			
教学目的	<ol style="list-style-type: none"> <li>1. 理解AI的基本概念。</li> <li>2. 掌握AI的主要表现形式。</li> <li>3. 了解回归、分类和聚类在AI中的应用。</li> </ol>			
教学重点	<ol style="list-style-type: none"> <li>1. 学生应能理解什么是人工智能技术，并了解其在日常生活中的应用。</li> <li>2. 学生能理解并体验到人工智能技术如何改变我们的生活。</li> <li>3. 学生应能掌握一些基本的人工智能技术的应用和操作</li> </ol>			
教学难点	<ol style="list-style-type: none"> <li>1. 激发探索的内驱力。如何培养学生对人工智能技术的兴趣和积极探索。</li> <li>2. 独立思考学习能力。如何使学生能够独立操作并应用人工智能技术。</li> </ol>			
课程思政	<ol style="list-style-type: none"> <li>1. 科技强国教育：说明中国在各个领域推动人工智能技术的发展，体现中国在人工智能基础设施建设方面的领先地位。</li> <li>2. 民族自信：介绍中国在人工智能基础技术方面的研究进展，例如机器学习、神经网络、深度学习等，展现中国科学家在人工智能技术领域的深厚积累。</li> <li>3. 创新精神：引导学生学习人工智能技术研发过程中的工匠精神，例如对算法模型的不断优化、对数据的精细处理等，培养学生的精益求精精神。</li> </ol>			
教学方法	讲授（ <input checked="" type="checkbox"/> ）；讨论（ <input checked="" type="checkbox"/> ）；指导（ ）；示教（ ）；其它（ ）			
电子教案	有（ <input checked="" type="checkbox"/> ）	Microsoft PowerPoint（ <input checked="" type="checkbox"/> ）；Author ware（ ）；其它（ ）		
	无（ ）			
教学资源	多媒体（ <input checked="" type="checkbox"/> ）；模型（ ）；标本（ ）；实物（ ）；音像（ ）；其它（ ）			
教学过程 时间安排	3学时			
思考题	1. 人工智能需要具备哪些要素？			

## 第二课：从零开始认识人工智能（3 学时）

授课老师：吴晓畅

### 一、AI概念

#### 1. AI主流定义

目前，认可度较高的一种观点认为，人工智能是计算机科学的一个分支，是研究和开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门技术科学。

人工智能涉及到计算机科学、数学、认知科学、哲学、心理学、信息论、控制论、社会结构学等的众多学科内容。

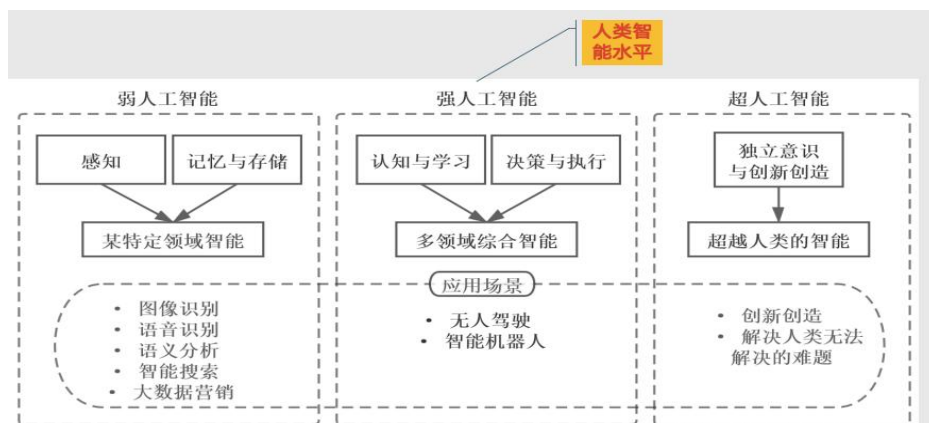
人工智能学科的基本思想和基本内容：人工智能研究的是人类智能活动的规律，研究如何让计算机去完成以往需要人的智能才能胜任的工作，从而构造具有一定智能的人工系统，也就是研究如何应用计算机的软硬件来模拟人类某些智能行为的基本理论、方法和技术。

#### 2. 什么是智能

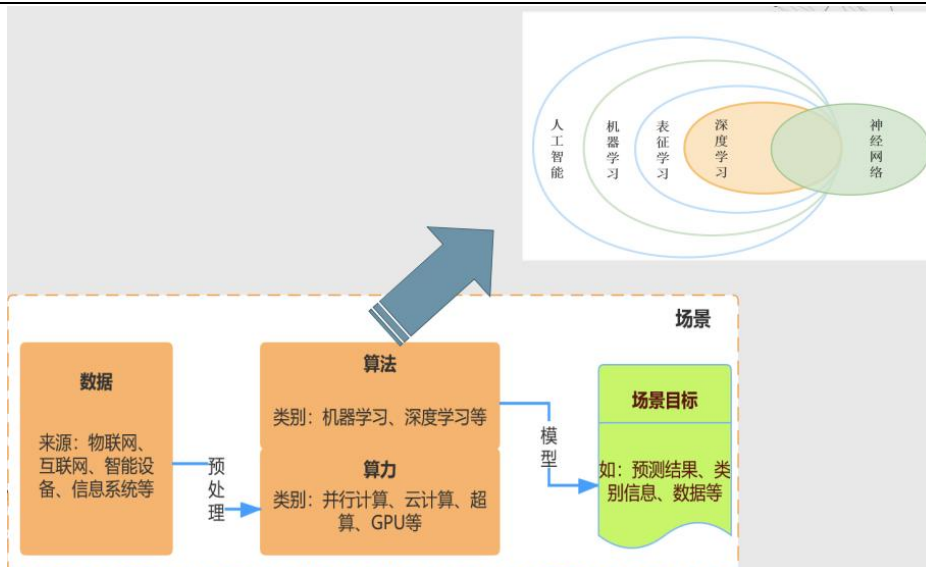
智能是指个体对客观事物进行感知、分析、判断以及有目的地行动和有效地处理周围环境事宜的综合能力。

作为人类各项能力的模拟和扩展，人工智能在“体力”、“感知能力”和“智力”三个领域发展迅速，在部分领域甚至已经远超人类，这种趋势在将来将会愈演愈烈。

#### 3. AI的三种形态：智能分强弱



#### 4. AI技术四要素



数据： 数据是人工智能的基础，机器学习算法需要大量的数据进行训练。

算力： 强大的算力是人工智能发展的保障，深度学习等算法需要大量的计算资源。

算法： 算法是人工智能的核心，机器学习算法的种类繁多，例如神经网络、决策树、支持向量机等。

场景： 是人工智能技术的具体使用。

### 5. AI技术体系



### 6. 人工智能主要研究内容

(1) 知识表示：研究如何将知识表示成计算机可以理解的形式，例如符号表示法和连接机制表示法。

(2) 机器感知：研究如何使计算机具有类似于人的感知能力，例如视觉、听觉、语音识别等。

(3) 机器思维：研究如何使计算机能进行推理和搜索，如逻辑推理等。

(4) 机器学习：研究如何使计算机能自主学习，例如深度学习、强化学习等。

(5) 机器行为：研究如何使计算机能够进行表达和行动，例如自然语言生成、机器人控制等。

## 二、AI的基础技术

### 1. 回归

回归是一种针对数值型连续数据进行预测和建模的有监督学习算法。

回归任务的特点是需要标注的数据集，其任务输入数据是连续性数据，并且任务目标也是连续型数值。也就是说，每一个观察样本都有一个数值型的标注真值。

典型的回归算法有线性回归、多项式回归、逻辑回归、Ridge回归等。

### 2. 分类

分类就是给数据贴标签的过程，要求尽可能为不同类别数据贴上正确的类别标签。人工智能的分类任务目标就是让机器通过经验学习从而具有正确贴标签的能力。

从数学角度来看，分类是一种对离散型随机变量建模或进行预测的一种有监督学习算法。与回归算法不同的是，分类算法通常适用于预测一个类别（或类别的概率）而不是连续的数值。

### 3. 聚类

聚类就是给相似的或相近的数据划分到同一类别，把不相似或者不相近的数据划分到不同类别，聚类一般不需要预先给数据打标签。

从技术上来讲，聚类是一种无监督学习任务，该算法基于数据的内部结构寻找观察样本的自然族群（即集群）。聚类常被用作分类算法的数据预处理步骤，用于处理缺少标签的数据。通过聚类分析，可以获得数据分布的基本概况。

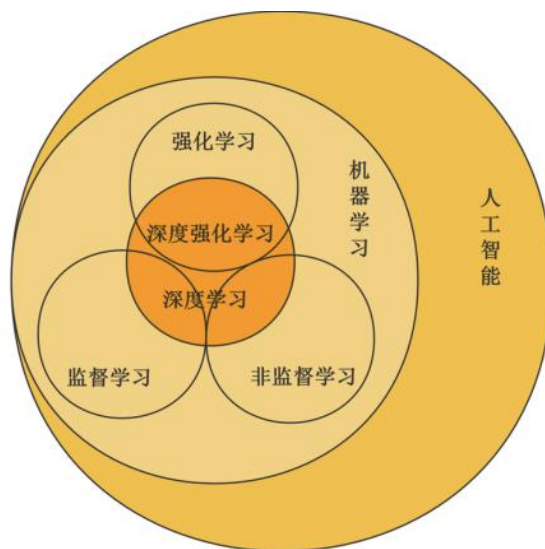
课程名称	人工智能技术		教案编号	03
教材名称	人工智能导论		专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第三课：从零开始认识人工智能2			
授课题目	AI的应用与伦理			
授课学时	2节（ ）；3节（ <input checked="" type="checkbox"/> ）；其它（ ）			
课 型	理论（ <input checked="" type="checkbox"/> ）；实验（ ）；见习（ ）；实训（ ）；其它（ ）			
教学目的	<ol style="list-style-type: none"> <li>1. 掌握实现AI的主要技术。</li> <li>2. 了解AI的应用领域。</li> <li>3. 理解人工智能伦理的重要性。</li> </ol>			
教学重点	<ol style="list-style-type: none"> <li>1. 人工智能技术流程</li> <li>2. 人工智能技术具体场景应用</li> </ol>			
教学难点	人工智能技术结合具体场景的应用实施			
课程思政	<ol style="list-style-type: none"> <li>1. 科技强国：强调中国在机器视觉、智能语音、自然语言处理等AI领域取得的成就，例如科大讯飞、百度语音识别等，展现中国在AI技术方面的实力和创新能力，激发学生的民族自豪感和对科技强国的认同。</li> <li>2. 安全意识：强调数据安全和隐私保护的重要性，引导学生了解相关法律法规，例如《网络安全法》，并培养学生在使用AI技术时的数据安全和隐私保护意识</li> <li>3. 智能化思维：引导学生思考AI对未来社会的影响，例如就业结构、教育模式等，并培养学生的未来意识和适应能力，使他们能够更好地适应智能化时代。</li> </ol>			
教学方法	讲授（ <input checked="" type="checkbox"/> ）；讨论（ <input checked="" type="checkbox"/> ）；指导（ ）；示教（ ）；其它（ ）			
电子教案	有（ <input checked="" type="checkbox"/> ）	Microsoft PowerPoint（ <input checked="" type="checkbox"/> ）；Author ware（ ）；其它（ ）		
	无（ ）			
教学资源	多媒体（ <input checked="" type="checkbox"/> ）；模型（ ）；标本（ ）；实物（ ）；音像（ ）；其它（ ）			
教学过程时间安排	3学时			
思考题	<ol style="list-style-type: none"> <li>1. 人工智能如何改变我们的生活？请举例说明人工智能在日常生活应用的例子，并分析其带来的利弊。</li> <li>2. 人工智能伦理问题有哪些？请举例说明人工智能可能带来的伦理风险。</li> </ol>			

## 第三课：从零开始认识人工智能2（3 学时）

### 授课老师：吴晓畅

#### 一、AI的技术基础

##### 1. AI基础技术体系



##### 1. 机器学习概述

机器学习是人工智能的核心技术和实现手段。简单来说，机器学习就是让计算机具有学习的能力，从而使得计算机能够模拟人的行为。



##### 2. 机器学习基本原理

机器学习一般包括三个步骤：一是收集历史数据，二是通过算法学习获得分布模式，三是应用模型处理新数据从而预测未来。

其中，步骤二是机器学习研究的重点，学习的过程就是根据数据确定模型参数的过程。

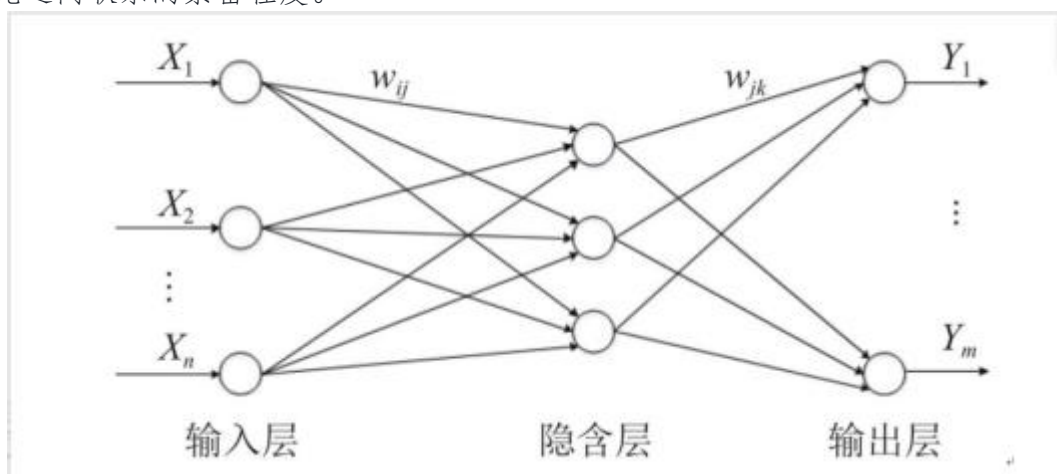
因此，机器“学习”的过程可以简化为寻找一个函数的过程，学习的结果也就是一个确定了参数的数学函数。

### 3. 人工神经网络

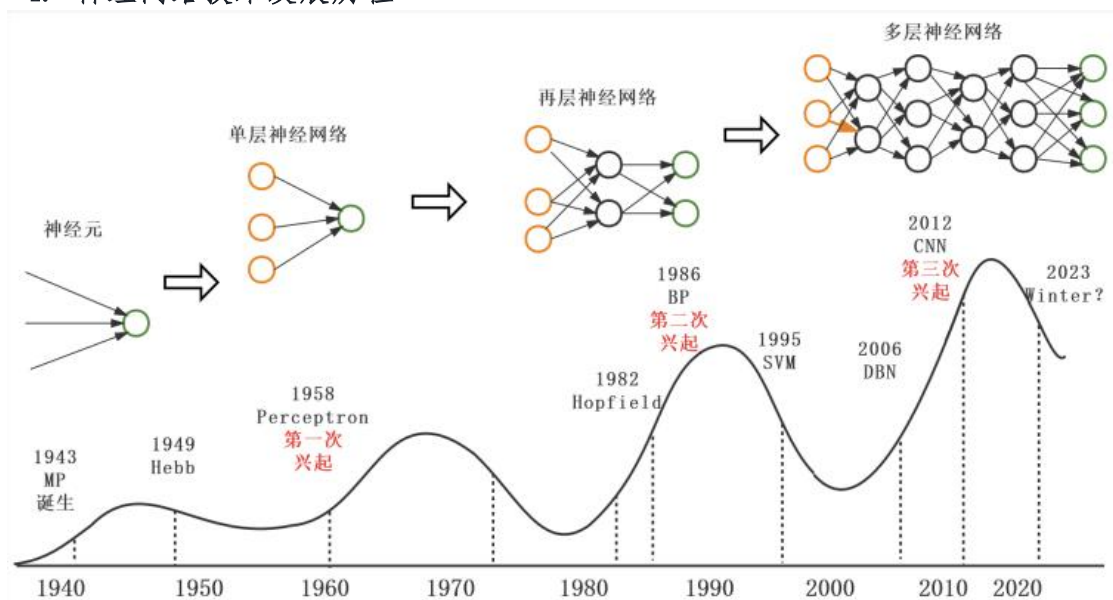
人工神经网络是一种模拟生物神经系统结构和功能的计算网络，因而生物神经网络是人工神经网络的技术原型。人类大脑皮层由大约140亿个生物神经元（简称神经元）组成，每个神经元又与大约10<sup>3</sup>个其它神经元相连接，形成一个高度复杂又高度灵活的不断变化的动态网络。

1943年，神经科学家麦卡洛克和数学家皮兹联手，根据生物神经元细胞的结构和工作原理，构造并提出了神经网络的数学模型MCP（McCulloch-Pitt: MCP），从而形成了“模拟大脑”、开启了人工神经网络（一般简称神经网络）的大门。

典型的神经网络由一个输入层、至少一个隐含层和一个输出层组成，每层网络由多个神经元构成，层与层之间一般采用全连接，神经元之间的连接强度 $\omega$ 表示神经元之间联系的紧密程度。



### 4. 神经网络技术发展历程



## 5. 深度学习

深度学习只是机器学习中的一个研究分支，是神经元层数较多的神经网络，因此深度学习网络也被称为深度神经网络。

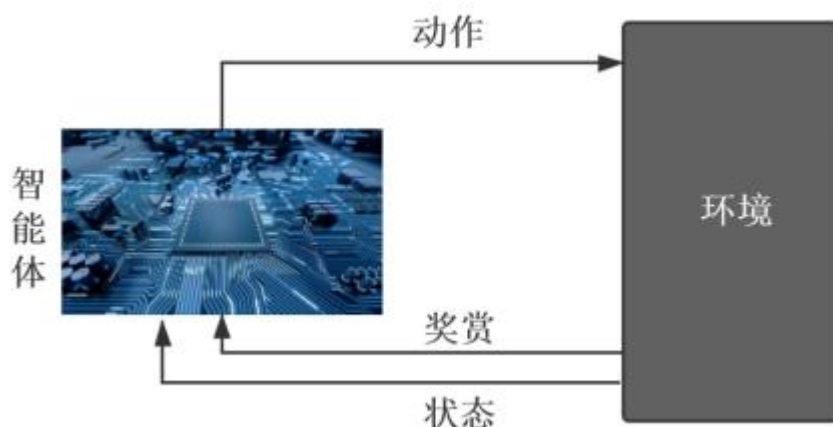
大家耳熟能详的深度学习网络有很多，比如卷积神经网络（CNN），循环神经网络（RNN），生成对抗网络（GAN），深度强化学习（DRL）等。

一般来说，其隐藏层的层数依具体问题可以是几层、几十层、几百层甚至数千层。

## 6. 强化学习

强化学习是机器学习领域的一个分支，强调如何基于环境而行动，以取得最大化的预期利益。

强化学习的学习方式是在获得样例过程中进行探索性学习（图2-21），在获得样例之后根据环境反馈的奖赏和状态更新自己的模型，利用更新后的模型来指导下一步的行动，下一步的行动获得奖赏反馈之后再更新模型，不断迭代重复直到模型收敛。



## 二、AI的应用领域

1. 计算机视觉（Computer Vision, CV）：CV让机器“看得见、看得懂”，是人工智能主要应用领域之一。我国在机器视觉方面处于世界领先地位。

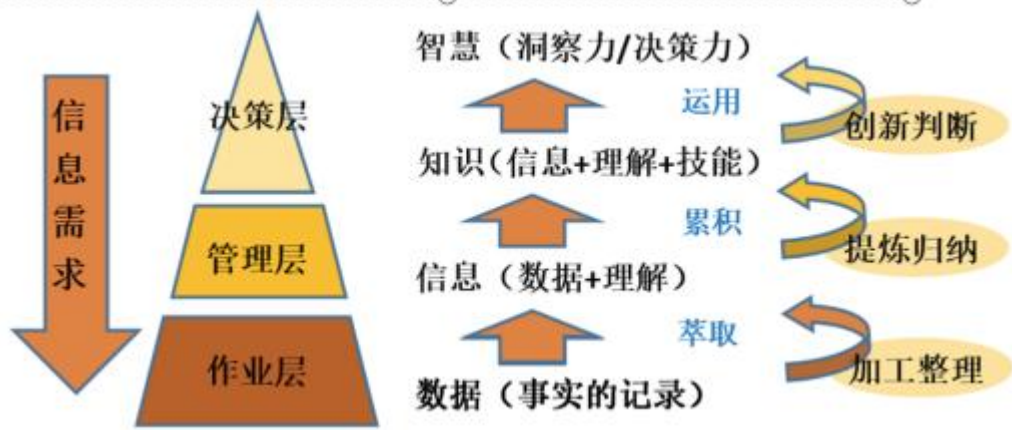
2. 智能语音：主要为机器人加上耳朵和嘴巴，让机器人能够“听得懂”、并且“说得好听”。

3. 自然语言处理（Natural Language Processing, NLP）：NLP被用于分析、理解和生成自然语言，以方便人和计算机设备进行交流、以及人与人之间的交流。

4. 无人驾驶：通过自动驾驶系统，部分或完全的代替人类驾驶员，安全地驾驶车辆、船舶、飞机等各种运输设备。



5. 数据挖掘 (Data Mining)，通过对海量数据的整理分析和归纳整合，分析并找出数据之间的潜在联系，为做出理想决策或预测发展趋势提供支撑性材料和建议，实现从海量数据中提取用于辅助决策的潜在的信息、知识、规律和模式。



### 三、AI伦理

#### 1. 概述

“科学技术都是一把双刃剑”。伴随着人工智能技术的成熟和广泛应用，人与机器之间的矛盾凸显，人工智能的伦理问题引起了社会和各行各业的日益关注。

#### 2. AI在军事上的应用

军用智能机器的性能和杀伤力远超人类士兵，无疑引起了人类对未来的担忧、以及对自主武器的争论，目前争论的核心是“自主武器该不该被使用”。



### 3. AI伦理问题的解决方法

人类对待AI出现了不同的观点，在科幻电影甚至传统媒体中均有体现。

阿西莫夫三定律：

- (1) 机器人不得伤害人类，或因不作为使人类受到伤害。
- (2) 除非违背第一定律，机器人必须服从人类的命令。
- (3) 除非违背第一及第二定律，机器人必须保护自己。

尽管阿西莫夫三定律被广为人知，但由于其可行性受到质疑，在现实中无论是AI安全研究者还是机器伦理学家，都没有真的使用它作为指导方案。

人工智能的伦理问题仍需人类的统一意见和行动，需要政府统一制定政策并推进执行。2017年7月，中国国务院发布《新一代人工智能发展规划》，关于人工智能伦理和法律制定了三步走的战略目标：到2020年，部分领域的人工智能伦理规范和政策法规初步建立；到2025年，初步建立人工智能法律法规、伦理规范和政策体系；到2030年，建成更加完善的人工智能法律法规、伦理规范和政策体系。

## 四、项目小结与展望

人工智能定位于人类能力的模拟和延伸，其定义和内涵随着时代的前进而不断变化，其智能程度也在不断提高，包括了早期的弱人工智能，现在正在发展的强人工智能，以及将来的超人工智能。人工智能需要解决的任务主要有三类：回归、分类和聚类，其实现的技术手段以机器学习和神经网络为基本技术、以深度学习、强化学习等为主流方法。伴随着人工智能技术在社会各行业、各领域的深入应用，人工智能的伦理问题也日益突出，成为政府和社会关注的焦点，也成为了影响人工智能走向的关键因素。

我国是AI技术研发和应用的大国，多项技术均处于国际领先地位。在我国“十四五”规划中，AI被明确列为优先发展的前沿科技领域之一，其基础研究工作更是被放到了战略地位，期待着更多的优秀青年加入到AI基础和核心研究中来。

随着人脸识别技术的逐步成熟，靠“脸”生存将不再是幻想。伴随着人脸识别等人工智能技术的发展，今后，从个人生活、教育、商业服务到城市管理都将全面迈入智能化时代。特别是高校在课堂上引入人脸识别技术、对学生面部表情进行识别，记录学生的学习状态，能更好的了解学生的学习情况，从而及时调整教学节奏。

课程名称	人工智能技术	教案编号	04
教材名称	人工智能导论	专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第四课 人工智能编程语言- Python		
授课题目	Python编程语言基础：认知、特性与环境搭建		
授课学时	2节（ ）；3节（ <input checked="" type="checkbox"/> ）；其它（ ）		
课 型	理论（ <input checked="" type="checkbox"/> ）；实验（ ）；见习（ ）；实训（ ）；其它（ ）		
教学目的	<ol style="list-style-type: none"> <li>1. 了解Python语言的特点及优势。</li> <li>2. 掌握Python电脑环境及软件安装方法。</li> </ol>		
教学重点	<ol style="list-style-type: none"> <li>1. Python历史、应用领域和语法特点等语言基础认知。</li> <li>2. Python优势介绍。</li> <li>3. Python电脑环境搭建及软件安装步骤。</li> </ol>		
教学难点	<ol style="list-style-type: none"> <li>1. Python语言特性的抽象理解。</li> <li>2. 环境配置中的常见问题解决。</li> </ol>		
课程思政	<ol style="list-style-type: none"> <li>1. 技术赋能与产业升级：强调Python在石油化工智能化转型中的应用（如智能工厂、预测性维护）。</li> <li>2. 科学精神与创新意识：引导学生思考：如何利用Python解决传统化工领域的效率问题（如工艺优化、能耗分析）。</li> </ol>		
教学方法	讲授（ <input checked="" type="checkbox"/> ）；讨论（ <input checked="" type="checkbox"/> ）；指导（ ）；示教（ ）；其它（ ）		
电子教案	有（ <input checked="" type="checkbox"/> ）	Microsoft PowerPoint（ <input checked="" type="checkbox"/> ）；Author ware（ ）；其它（ ）	
	无（ ）		
教学资源	多媒体（ <input checked="" type="checkbox"/> ）；模型（ ）；标本（ ）；实物（ ）；音像（ ）；其它（ ）		
教学过程 时间安排	3学时		
思考题	简述Python的三大核心优势，并举例说明其在石油化工领域的应用场景？		

## 第四课：人工智能编程语言- Python（3 学时）

授课老师：吴晓畅

### 一、Python简介

Python 已经成为最受欢迎的程序设计语言之一,目前在人工智能科学领域被广泛应用。Python 是人工智能（机器学习）的首选编程语言，它拥有众多模块，能完成人工智能开发的所有环节。使用Python进行AI的一个主要优点是它内置了许多与机器学习相关的库。Python提供了高效的高级数据结构，还能简单有效地面向对象编程。Python语法和动态类型，以及解释型语言的本质，使它成为多数平台上写脚本和快速开发应用的编程语言。主流的人工智能开发软件如谷歌的TensorFlow、Facebook的PyTorch和百度的PaddlePaddle等都使用python作为开发语言。可以说，学习人工智能，很难不学Python。

### 二、Python语言特点

Python作为一种高级编程语言，凭借其简洁易读的语法（代码接近自然语言，强制缩进提升可读性）、动态类型系统（变量无需声明类型，运行时自动推断）、跨平台兼容性（支持Windows、Linux、MacOS，一次编写多平台运行）、丰富的标准库（内置模块覆盖文件操作、数学计算、时间处理等常见需求）、强大的第三方生态（拥有NumPy、Pandas、Matplotlib等数据分析与可视化库，以及TensorFlow、PyTorch等AI框架）、面向对象与函数式编程结合（兼容多种编程范式）以及开源社区支持（全球开发者贡献代码，问题解决效率高）等核心特点，已成为数据科学、人工智能、Web开发及工程领域（如化工、能源）的主流工具，尤其适合初学者快速入门并高效解决复杂问题。

### 三、为什么要学习Python?

#### 1. 行业需求驱动

石油化工领域应用：

数据采集与分析：实时监控设备运行状态，预测故障（如使用scikit-learn进行机器学习建模）。

工艺优化：通过模拟优化反应条件（如用SimPy进行离散事件仿真）。

自动化控制：编写脚本控制阀门、泵设备（如用PySerial实现串口通信）。

## 2. 技术趋势推动

AI与大数据时代：Python是AI开发首选语言（如TensorFlow、PyTorch）。

工业4.0转型：智能制造、数字孪生等场景需要Python进行数据处理与算法开发。

低代码/无代码平台：Python可作为底层引擎，支持快速开发（如Streamlit构建数据可视化应用）。

## 3. 个人发展价值

就业竞争力：Python工程师需求量持续增长，薪资水平高于传统岗位。

跨领域迁移：掌握Python后，可轻松转向数据分析、AI开发、Web开发等方向。

科研工具：Python是科研领域主流编程语言（如用Biopython进行生物信息学分析）。

# 四、项目小结与展望

在本课程中，我们系统学习了Python这一目前最受欢迎的程序设计语言之一，深入探讨了其在人工智能、数据科学、Web开发及工程领域的广泛应用与核心优势。Python凭借其简洁易读的语法、动态类型系统、跨平台兼容性、丰富的标准库与强大的第三方生态，不仅成为AI开发的首选语言（如TensorFlow、PyTorch等主流框架均基于Python），还在石油化工等工业领域展现出巨大潜力，如数据采集与分析、工艺优化及自动化控制等场景均能通过Python实现高效解决方案。学习Python不仅是顺应AI与大数据时代的技术趋势，更是提升个人就业竞争力、实现跨领域迁移及科研创新的重要途径。

展望未来，随着工业4.0转型的加速，Python在智能制造、数字孪生等领域的应用将更加深入，其低代码/无代码平台的特性也将进一步降低技术门槛，促进更广泛的行业应用。对于学习者而言，持续深化Python技能，不仅有助于在现有领域内深耕细作，更能为探索新兴技术、把握未来机遇奠定坚实基础。我们期待每一位学员都能将所学知识转化为实践动力，用Python语言解锁更多可能，共创智能科技新篇章。

## 五、课堂作业

简述Python的三大核心优势，并举例说明其在石油化工领域的应用场景？

课程名称	人工智能技术	教案编号	05
教材名称	人工智能导论	专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第五课 人工智能编程语言- Python		
授课题目	Python基础入门：变量定义与注释规范		
授课学时	2节（ ）；3节（ <input checked="" type="checkbox"/> ）；其它（ ）		
课 型	理论（ <input checked="" type="checkbox"/> ）；实验（ ）；见习（ ）；实训（ ）；其它（ ）		
教学目的	<ol style="list-style-type: none"> <li>1. 理解Python变量的基本概念。</li> <li>2. 掌握变量名的命名规则。</li> <li>3. 理解变量值的概念。</li> <li>4. 掌握变量赋值的语法与方法。</li> <li>5. 理解代码注释的重要性及规范化书写方法。</li> </ol>		
教学重点	<ol style="list-style-type: none"> <li>1. Python变量的基本概念</li> <li>2. 变量命名规则</li> <li>3. 变量赋值操作</li> <li>4. Python注释的使用</li> </ol>		
教学难点	<ol style="list-style-type: none"> <li>1. 变量动态类型的理解，初学者易混淆“变量类型”与“数据类型”。</li> <li>2. 变量命名规范的实际应用。</li> <li>3. 注释的规范书写。</li> </ol>		
课程思政	<ol style="list-style-type: none"> <li>1. 代码规范与职业素养：强调变量命名、注释等规范的重要性，培养严谨的编程习惯，契合工程伦理中的“责任意识”。</li> <li>2. 动态思维与创新精神：通过Python动态类型特性，引导学生理解“灵活性”与“约束性”的平衡，培养适应快速变化的技术环境的能力。</li> </ol>		
教学方法	讲授（ <input checked="" type="checkbox"/> ）；讨论（ ）；指导（ ）；示教（ <input checked="" type="checkbox"/> ）；其它（ ）		
电子教案	有（ <input checked="" type="checkbox"/> ）	Microsoft PowerPoint（ <input checked="" type="checkbox"/> ）；Author ware（ ）；其它（ ）	
	无（ ）		
教学资源	多媒体（ <input checked="" type="checkbox"/> ）；模型（ ）；标本（ ）；实物（ ）；音像（ ）；其它（ <input checked="" type="checkbox"/> ）		
教学过程 时间安排	3学时		
思考题	<ol style="list-style-type: none"> <li>1. 判断题：Python变量名可以包含空格</li> <li>2. 选择题：以下哪个是合法的变量名？ A. 2name B. _age C. class D. user-name</li> </ol>		

## 第五课：人工智能编程语言- Python (3 学时)

授课老师：吴晓畅

### 一、Python基础语法

#### 1. 标识符

简单地理解，标识符就是一个名字，就好像我们每个人都有属于自己的名字，它的主要作用就是作为变量、函数、类、模块以及其他对象的名称。

Python 中标识符的命名不是随意的，而是要遵守一定的命名规则。标识符是由字符 (A~Z 和 a~z)、下划线和数字组成，但第一个字符不能是数字。标识符不能和 Python 中的保留字相同。Python 中的标识符中，不能包含空格、@、% 以及 \$ 等特殊字符。标识符区分大小写。

下面所列举的标识符是合法的：

```
UserID, name, model2, user_age
```

以下命名的标识符不合法：

```
4word #不能以数字开头
```

```
try #try是保留字，不能作为标识符
```

```
$money #不能包含特殊字符
```

#### 2. 缩进

Python 中的缩进 (Indentation) 决定了代码的作用域范围。这一点和传统的 C/C++ 有很大的不同 (传统的 C/C++ 使用花括号 {} 符决定作用域的范围；python 使用缩进空格来表示作用域的范围，相同缩进行的代码是处于同一范围)。每行代码中开头的空格数用于计算该行代码的缩进级别。

#### 3. 注释

注释用来向用户提示或解释某些代码的作用和功能，它可以出现在代码中的任何位置。Python 解释器在执行代码时会忽略注释，不做任何处理，就好像它不存在一样。Python 支持两种类型的注释，分别是单行注释和多行注释。单行注释以 # 开头，多行注释使用三个连续的单引号 ‘ ’ ‘ 或者三个连续的双引号 “ ” “。例如：

```
# 这是单行注释
```

‘ ‘ ‘这是多行注释’ ’ ’

#### 4. 关键字

Python 有一组关键字，这些关键字是保留字，不能用作变量名、函数名或任何其他标识符。如and、as、break、continue和class等，我们在后续的学习中会逐步认识这些关键字。

#### 5. 变量

我们定义一个计算机变量就好像向快递柜申请一个格子用来存放东西，这个格子的大小呢是根据我们变量的类型（就是想要存放的物品的可能尺寸）来分配的。而在申请一个格子的时候，里面可能会存在一些东西，当我们放新的东西的进去的时候就会把原来的东西给挤走（更准确说是覆盖，旧的东西找不回来了）。如果我们想让这个格子里的内容参与运算的时候，我们只需要告诉运算的指令这个格子的名称（即变量的名称）即可。

Python变量、类型和运算符所有的编程语言都支持变量，Python 也不例外。变量是编程的起点，程序需要将数据存储到变量中。

变量类型：变量可以指定不同的数据类型，这些变量可以存储整数，小数或字符。Python 定义了一些标准类型，用于存储各种类型的数据。这些类型包括数字、字符串、列表、元组和字典。数字数据类型用于存储数值。他们是不可改变的数据类型，这意味着改变数字数据类型会分配一个新的对象。当你指定一个值时，数字对象就会被创建。

## 二、代码演示及运行结果

### 1. 代码

# 1. 定义变量并赋值

```
name = "小明"      # 变量名: name, 变量值: "小明"
```

```
age = 18           # 变量名: age, 变量值: 18
```

```
height = 1.75     # 变量名: height, 变量值: 1.75
```

# 2. 输出变量的值

```
print("姓名:", name) # 输出: 姓名: 小明
```

```
print("年龄:", age)  # 输出: 年龄: 18
```

```
print("身高:", height, "米") # 输出: 身高: 1.75 米
```

# 3. 动态类型特性: 变量值可以重新赋值

```
age = 19 # 变量 age 的值从 18 改为 19
```

```
print("更新后的年龄:", age) # 输出: 更新后的年龄: 19
```

# 4. 多变量赋值

```
x, y, z = 10, 20, 30 # 同时给 x, y, z 赋值
```

```
print("x:", x, "y:", y, "z:", z) # 输出: x: 10 y: 20 z: 30
```

# 5. 使用注释解释代码逻辑

# 下面计算年龄加 10 年后的值

```
future_age = age + 10
```

```
print("10 年后的年龄:", future_age) # 输出: 10 年后的年龄: 29
```

## 2. 代码运行结果

```
1  姓名: 小明
2  年龄: 18
3  身高: 1.75 米
4  更新后的年龄: 19
5  x: 10 y: 20 z: 30
6  10 年后的年龄: 29
```

## 三、课堂作业

1. 判断题: Python变量名可以包含空格
2. 选择题: 以下哪个是合法的变量名?  
A. 2name B. \_age C. class D. user-name

课程名称	人工智能技术	教案编号	06
教材名称	人工智能导论	专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第六课 人工智能编程语言- Python		
授课题目	Python数据类型解析		
授课学时	2节 ( ) ; 3节 ( <input checked="" type="checkbox"/> ) ; 其它 ( )		
课 型	理论 ( <input checked="" type="checkbox"/> ) ; 实验 ( ) ; 见习 ( ) ; 实训 ( ) ; 其它 ( )		
教学目的	<ol style="list-style-type: none"> <li>1. 全面认识Python中的基本数据类型。</li> <li>2. 深入理解整数、浮点数、负数、布尔型、字符与字符串的特性及应用。</li> <li>3. 能够正确区分并使用这些数据类型进行编程。</li> </ol>		
教学重点	<ol style="list-style-type: none"> <li>1. 数据类型分类与特性。</li> <li>2. 整数、浮点数、布尔型、字符与字符串的表示方法及区别。</li> <li>3. 数据类型应用。</li> </ol>		
教学难点	<ol style="list-style-type: none"> <li>1. 浮点数精度与运算陷阱。</li> <li>2. 字符串的不可变性。</li> <li>3. 布尔型与条件判断的隐式转换。</li> </ol>		
课程思政	科学思维培养：浮点数精度问题→强调“误差容忍度”，有利于学生培养科学的思维意识。		
教学方法	讲授 ( <input checked="" type="checkbox"/> ) ; 讨论 ( ) ; 指导 ( ) ; 示教 ( <input checked="" type="checkbox"/> ) ; 其它 ( )		
电子教案	有 ( <input checked="" type="checkbox"/> )	Microsoft PowerPoint ( <input checked="" type="checkbox"/> ) ; Author ware ( ) ; 其它 ( )	
	无 ( )		
教学资源	多媒体 ( <input checked="" type="checkbox"/> ) ; 模型 ( ) ; 标本 ( ) ; 实物 ( ) ; 音像 ( ) ; 其它 ( <input checked="" type="checkbox"/> )		
教学过程 时间安排	3学时		
思考题	浮点数精度问题在哪些实际场景中可能引发严重后果？如何避免？		

## 第六课：人工智能编程语言- Python (3 学时)

授课老师：吴晓畅

### 一、数据类型介绍

#### 1. 数字

Python 中有三种数字类型：整数类型、浮点类型和复数类型。为变量赋值时，将创建数值类型的变量。

Python中布尔类型为bool，bool是int的子类。bool类型只有两个值：True和False。任何类型数据都可以通过bool()函数转换为布尔值，“没有的”或“空的”值会转换为False，反之转换为True。如：None（空对象）、False、0、0.0、0j（复数）、''（空字符串）、[]（空列表）、()（空元组）和 {}（空字典）这些值会转换为False，否则是True。

#### 2. 字符串

字符串是 Python 中最常用的数据类型，字符串通常是由数字，字母、符号组成的一串字符。我们可以使用引号（' 或 "）来创建字符串。创建字符串的过程十分简单，我们只要为变量分配被引号包围住的一个值即可。例如：

```
String1 = '张三',
```

```
String2 = 'abc'
```

### 二、代码演示

```
# 1. 认识数据类型
```

```
# 2. 整数
```

```
a = 10
```

```
b = 20
```

```
# 整数运算
```

```
sum_ab = a + b # 加法
```

```
diff_ab = a - b # 减法
```

```
prod_ab = a * b # 乘法
```

```
div_ab = a // b # 整除
```

```
mod_ab = a % b # 取余
```

```
exp_ab = a ** b # 幂运算
print("整数a:", a)
print("整数b:", b)
print("a + b =", sum_ab)
print("a - b =", diff_ab)
print("a * b =", prod_ab)
print("a // b =", div_ab)
print("a % b =", mod_ab)
print("a ** b =", exp_ab)

# 3. 浮点数
x = 10.5
y = 20.3

# 浮点数运算
sum_xy = x + y # 加法
diff_xy = x - y # 减法
prod_xy = x * y # 乘法
div_xy = x / y # 除法
print("浮点数x:", x)
print("浮点数y:", y)
print("x + y =", sum_xy)
print("x - y =", diff_xy)
print("x * y =", prod_xy)
print("x / y =", div_xy)

# 4. 负数类型
neg_a = -a
neg_b = -b
print("负数-a:", neg_a)
print("负数-b:", neg_b)

# 5. 布尔型
```

```
true_val = True

false_val = False

# 布尔型运算

and_op = true_val and false_val # 与运算

or_op = true_val or false_val # 或运算

not_op = not true_val # 非运算

print("True and False =", and_op)

print("True or False =", or_op)

print("not True =", not_op)

# 6. 字符与字符串

char = 'A'

string = "Hello, World!"

# 字符串创建和常见操作

# 字符串拼接

concat_string = string + " Python is fun!"

# 字符串长度

len_string = len(string)

# 字符串索引

first_char = string[0]

last_char = string[-1]

# 字符串切片

slice_string = string[0:5]

# 字符串查找（使用find方法，但这里为了展示传统格式化，我们稍后在
print中不使用它）

# find_index = string.find("World")

# 为了展示find的结果，我们可以先将其存储在一个变量中，但用%s格式
化时不会直接用它

# 而是用一个示例值来展示如何格式化字符串
```

```
example_find_index = 7 # 假设的索引值, 实际使用时应该替换为
find_index的结果

# 字符串替换 (但这里为了展示传统格式化, 我们直接在print中展示原字
字符串)

# replace_string = string.replace("World", "Python")

# 字符串转换为大写和小写

upper_string = string.upper()

lower_string = string.lower()

print("字符:", char)

print("字符串:", string)

print("字符串拼接:", concat_string)

print("字符串长度:", len_string)

print("字符串第一个字符:", first_char)

print("字符串最后一个字符:", last_char)

print("字符串切片 (前5个字符):", slice_string)

# 注意: 这里我们不用find的结果, 而是用%d来格式化一个示例索引值

print("假设的字符串中某部分的索引 (示例):%d" %
example_find_index)

# 如果要用find的结果, 应该这样 (但这里为了保持示例的一致性, 我们
注释掉它)

# print("字符串中'World'的索引 (实际):%d" % find_index)

print("字符串转换为大写:", upper_string)

print("字符串转换为小写:", lower_string)
```

### 三、课堂思考题

浮点数精度问题在哪些实际场景中可能引发严重后果? 如何避免?

课程名称	人工智能技术		教案编号	07
教材名称	人工智能导论		专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第七课 人工智能编程语言- Python			
授课题目	Python的条件和循环			
授课学时	2节 ( ) ; 3节 ( √ ) ; 其它 ( )			
课 型	理论 ( √ ) ; 实验 ( ) ; 见习 ( ) ; 实训 ( ) ; 其它 ( )			
教学目的	<ol style="list-style-type: none"> <li>1. 理解并掌握Python中的条件语句和循环语句。</li> <li>2. 能够根据实际需求设计合理的控制流结构。</li> <li>3. 培养学生运用控制流语句实现复杂逻辑的能力。</li> </ol>			
教学重点	<ol style="list-style-type: none"> <li>1. Python 条件语句 (if - elif - else) 的语法与应用。</li> <li>2. Python 循环语句 (while 和 for) 的语法与用法。</li> <li>3. break、continue 和 else 子句在控制流语句中的作用。</li> </ol>			
教学难点	<ol style="list-style-type: none"> <li>1. 学生根据实际需求设计合理的控制流结构。</li> <li>2. 复杂逻辑中控制流语句的嵌套与组合使用。</li> </ol>			
课程思政	<ol style="list-style-type: none"> <li>1. 培养严谨的逻辑思维：在讲解 Python 控制流语句时，强调逻辑的严密性和准确性。引导学生认识到在编程中，每一个条件判断和循环结构都需要经过仔细思考和精心设计，就像在解决生活中的问题时，也需要严谨的逻辑思维一样。通过编程训练，培养学生严谨的治学态度和思维方式，让他们明白只有严谨的逻辑才能构建出稳定、可靠的程序。</li> <li>2. 强调团队协作与分享：在课程中，可以组织小组项目，让学生分组完成一些具有挑战性的编程任务。在小组中，学生需要相互协作、相互学习，共同解决问题。通过团队协作，培养学生的沟通能力和团队合作精神。同时，鼓励学生分享自己的编程经验和技巧，营造良好的学习氛围，让学生明白知识的分享和交流能够促进共同进步。</li> </ol>			
教学方法	讲授 ( √ ) ; 讨论 ( ) ; 指导 ( ) ; 示教 ( √ ) ; 其它 ( )			
电子教案	有 (√)	Microsoft PowerPoint (√) ; Author ware ( ) ; 其它 ( )		
	无 ( )			
教学资源	多媒体 (√) ; 模型 ( ) ; 标本 ( ) ; 实物 ( ) ; 音像 ( ) ; 其它 (√)			
教学过程 时间安排	3学时			
思考题				

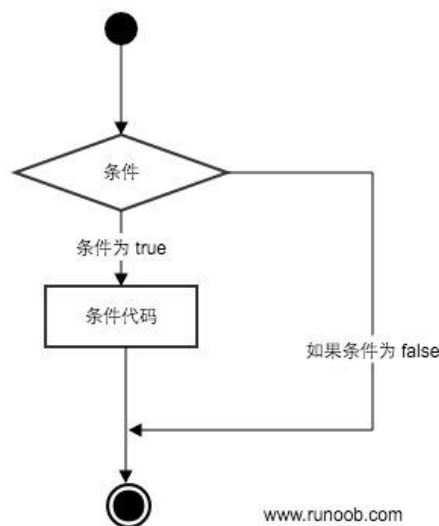
## 第七课：人工智能编程语言- Python（3 学时）

授课老师：吴晓畅

### 一、条件控制语句

程序在一般情况下是按顺序执行的，就像流水账一样，一条一条从上往下按照顺序执行。有时候我们需要根据条件来有选择地执行某些语句，这就要使用到Python的条件判断语句——if。Python 条件语句是通过一条或多条语句的执行结果（True 或者 False）来决定执行的代码块。

可以通过下图来简单了解条件语句的执行过程：



条件判断实例

```
# 1. 定义年龄变量
age = 18

# 2. 判断是否满 18 岁
# if 语句以及缩进部分的代码是一个完整的代码块
if age >= 18:
    print("你是一个成年人了……")
```

### 二、循环

#### 1. 循环的概念

循环是大多数编程语言都有的基本功能，它可以帮助我们执行一些重复的代码工作，但不同的语言循环结构会有一些的差异。Python中有for和while两种循环结构。

## 2. for循环和while循环

for 循环是在希望创建循环时经常使用的工具。Python for循环可以遍历任何可迭代对象，包括列表、元组、集合、字典甚至字符串。for循环的一般代码格式如下：

for 迭代变量 in 字符串|列表|元组|字典|集合：

代码块

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
```

while 循环的一般形式如下：

while (判断条件)：

代码块

## 三、代码演示和运行结果

### 1. 代码

```
# 1. While循环 + 判断语句
```

```
# 在while循环中，判断i是否为偶数，并打印相应的信息。
```

```
print("1. While循环 + 判断语句示例:")
```

```
i = 1
```

```
while i <= 5:
```

```
    if i % 2 == 0:
```

```
        print(f"While循环: i={i} 是偶数")
```

```
    else:
```

```
        print(f"While循环: i={i} 是奇数")
```

```
    i += 1
```

```
print() # 空行分隔输出
```

```
# 2. For-in循环 + 判断语句
```

```
# 在for-in循环中，判断x是否大于3，并打印相应的信息。
```

```
print("2. For-in循环 + 判断语句示例:")
```

```
for x in range(1, 6):
    if x > 3:
        print(f"For-in循环: x={x} 大于3")
    else:
        print(f"For-in循环: x={x} 小于或等于3")
print() # 空行分隔输出
# 3. While和for-in混合 + 判断语句
# 在while和for-in混合循环中, 判断m是否等于2, 并打印相应的信息。
print("3. While和for-in混合 + 判断语句示例:")
k = 1
while k <= 2:
    print(f"外层循环k={k}")
    for m in range(1, 4):
        if m == 2:
            print(f"混合嵌套: k={k}, m={m} 等于2")
        else:
            print(f"混合嵌套: k={k}, m={m} 不等于2")
    k += 1
print() # 空行分隔输出
# 4. 循环的退出 + 判断语句
# 在while循环中, 使用if判断p是否为偶数, 并打印相应的信息。
# 当p等于3时, 使用break退出循环。
print("4. 循环的退出示例 + 判断语句:")
p = 1
while True:
    if p % 2 == 0:
        print(f"检查: p={p} 是偶数")
    else:
        print(f"检查: p={p} 是奇数")
```

```
if p == 3:
    print("条件满足，退出循环。")
    break
p += 1

print("循环已退出。\\n") # 空行分隔输出

# 额外示例：在for-in循环中使用break + 判断语句
# 在for-in循环中，判断num是否大于3，并在满足条件时退出循环。
print("额外示例：在for-in循环中使用break + 判断语句:")

for num in range(1, 6):
    print(f"For-in循环检查：num={num}")
    if num > 3:
        print("在for-in循环中，num大于3时退出。")
        break

print("For-in循环已退出。\\n") # 空行分隔输出
```

## 2. 运行结果

### 1. While循环 + 判断语句示例：

```
While循环：i=1 是奇数
While循环：i=2 是偶数
While循环：i=3 是奇数
While循环：i=4 是偶数
While循环：i=5 是奇数
```

### 2. For-in循环 + 判断语句示例：

```
For-in循环：x=1 小于或等于3
For-in循环：x=2 小于或等于3
For-in循环：x=3 小于或等于3
For-in循环：x=4 大于3
For-in循环：x=5 大于3
```

### 3. While和for-in混合 + 判断语句示例:

外层循环k=1

混合嵌套: k=1, m=1 不等于2

混合嵌套: k=1, m=2 等于2

混合嵌套: k=1, m=3 不等于2

外层循环k=2

混合嵌套: k=2, m=1 不等于2

混合嵌套: k=2, m=2 等于2

混合嵌套: k=2, m=3 不等于2

额外示例: 在for-in循环中使用break + 判断语句:

For-in循环检查: num=1

For-in循环检查: num=2

For-in循环检查: num=3

For-in循环检查: num=4

在for-in循环中, num大于3时退出。

For-in循环已退出。

## 四、课堂小结

本堂课围绕 Python 编程中的控制流语句展开, 系统讲解了 if - elif - else 条件语句的语法规则与实际应用场景, 使同学们能依据不同条件执行相应代码块; 深入剖析了 while 和 for 循环语句的结构与用法, 帮助大家掌握重复执行代码的方法; 同时, 详细介绍了 break、continue 和 else 子句在控制流语句中的独特作用, 为优化程序逻辑提供了有力工具。通过丰富的实例练习, 同学们不仅熟悉了控制流语句的使用技巧, 更锻炼了根据实际需求设计合理控制流结构的能力, 为运用控制流语句实现复杂逻辑奠定了坚实基础。

## 五、课堂作业

根据课堂所讲代码, 自己在python编译器上打一遍代码, 将代码及运行结果截图上传到学习通。

课程名称	人工智能技术		教案编号	08
教材名称	人工智能导论		专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第八课 人工智能编程语言- Python			
授课题目	Python的元组和列表			
授课学时	2节 ( ) ; 3节 ( √ ) ; 其它 ( )			
课 型	理论 ( √ ) ; 实验 ( ) ; 见习 ( ) ; 实训 ( ) ; 其它 ( )			
教学目的	<ol style="list-style-type: none"> <li>1. 掌握元组的基本语法和创建方法，能够识别代码中的元组对象。</li> <li>2. 理解负索引和步长在访问元组元素时的应用。</li> <li>3. 了解如何将两个或多个元组合并成一个新的元组。</li> <li>4. 掌握如何从内存中“删除”整个元组对象</li> <li>5. 掌握列表的创建、遍历及常用操作方法（增删改查）</li> </ol>			
教学重点	<ol style="list-style-type: none"> <li>1. 元组相关：元组的基本语法、创建方法，负索引和步长在访问元组元素中的应用，元组合并操作。</li> <li>2. 列表相关：列表的创建、遍历方法，以及增删改查等常用操作。</li> </ol>			
教学难点	<ol style="list-style-type: none"> <li>1. 元组不可变性理解：深入理解元组不可变这一特性，以及在实际编程中如何合理利用这一特性，避免因误解而导致的错误操作。</li> <li>2. 列表高级操作应用：掌握列表切片、排序、反转等高级操作的具体实现方式，并能根据实际需求灵活运用这些操作解决问题。</li> </ol>			
课程思政	<ol style="list-style-type: none"> <li>1. 严谨态度培养：强调在编程过程中，元组和列表操作的严谨性，任何细微的错误都可能导致程序运行失败，引导学生养成严谨细致的学习和工作态度。</li> <li>2. 团队协作意识：通过介绍元组和列表在团队协作编程项目中的应用，让学生明白在团队中要相互配合，合理利用不同数据结构来实现项目目标，培养学生的团队协作意识。</li> <li>3. 创新精神激发：鼓励学生探索元组和列表在不同场景下的创新应用，激发学生的人工智能编程技术的创新热情，培养创新思维。</li> </ol>			
教学方法	讲授 ( √ ) ; 讨论 ( ) ; 指导 ( ) ; 示教 ( √ ) ; 其它 ( )			
电子教案	有 ( √ )	Microsoft PowerPoint ( √ ) ; Author ware ( ) ; 其它 ( )		
	无 ( )			
教学资源	多媒体 ( √ ) ; 模型 ( ) ; 标本 ( ) ; 实物 ( ) ; 音像 ( ) ; 其它 ( √ )			
教学过程 时间安排	3学时			
思考题	举例说明元组不可变性在实际编程中的优势和可能带来的不便。			

## 第八课：人工智能编程语言- Python (3 学时)

授课老师：吴晓畅

### 一、元组基础知识

#### 1. 元组的定义、特点和用途

讲解元组的定义：元组是一种不可变的序列类型，用圆括号 () 来定义。

例如 `my_tuple = (1, 2, 3)`。

强调元组的特点：不可变性，即元组一旦创建，其内容就不能被修改。同时说明元组的用途，如存储一组固定的数据，作为字典的键等。

#### 2. 元组的创建方法

演示直接用圆括号创建元组，如 `tuple1 = (10, 20, 30)`。

讲解不使用圆括号创建元组（在表达式中需要圆括号区分优先级时除外），如 `tuple2 = 40, 50, 60`。

介绍使用 `tuple()` 函数将其他可迭代对象转换为元组，例如 `list1 = [7, 8, 9]`，`tuple3 = tuple(list1)`。

#### 3. 使用索引访问元组中的单个元素

讲解正索引和负索引的概念，正索引从 0 开始，负索引从 -1 开始。演示如何使用正索引访问元组元素，如 `my_tuple = ('a', 'b', 'c')`，`print(my_tuple[0])` 输出 'a'。

演示如何使用负索引访问元组元素，如 `print(my_tuple[-1])` 输出 'c'。

介绍步长的概念，通过切片语法 `tuple[start:stop:step]` 来访问多个元素，例如 `print(my_tuple[0:3:2])` 输出 ('a', 'c')。

#### 4. 元组合并操作

演示使用加号 + 操作符连接两个或多个元组，如 `tuple4 = (1, 2)`，`tuple5 = (3, 4)`，`new_tuple = tuple4 + tuple5`，`print(new_tuple)` 输出 (1, 2, 3, 4)。

### 二、元组相关代码

```
# 目标：1. 认识元组 2. 访问元组 3. 连接元组 4. 删除元组
# 1. 认识元组
# 元组是一种不可变序列，使用圆括号 () 来定义
my_tuple = (1, 2, 3, 'apple', 'banana')
print("这是一个元组:", my_tuple)
```

```
# 元组可以包含不同类型的数据
mixed_tuple = (True, 3.14, 'Python', [1, 2, 3])
print("混合类型的元组:", mixed_tuple)

# 2. 访问元组
# 使用索引访问元组中的元素, 索引从0开始
print("元组的第一个元素:", my_tuple[0]) # 输出 1
print("元组的第四个元素:", my_tuple[3]) # 输出 'apple'

# 使用负索引从后往前访问
print("元组的最后一个元素:", my_tuple[-1]) # 输出 'banana'

# 3. 连接元组
# 可以使用 + 运算符连接两个元组
tuple1 = (10, 20, 30)
tuple2 = (40, 50, 60)
combined_tuple = tuple1 + tuple2
print("连接后的元组:", combined_tuple)

# 4. 删除元组
# 元组是不可变的, 不能直接修改或删除其中的元素
# 但是可以删除整个元组
del my_tuple # 删除 my_tuple 元组
# 尝试访问已删除的元组会引发 NameError
try:
    print(my_tuple)
except NameError:
    print("my_tuple 元组已被删除, 无法访问")

# 注意: 虽然不能删除元组中的单个元素, 但可以通过创建新元组来"排除"不需要的元素
# 例如, 从 combined_tuple 中排除第一个元素
new_tuple = combined_tuple[1:]
print("排除第一个元素后的新元组:", new_tuple)
```

### 三、列表基础知识

#### 1. 列表的定义与索引规则

讲解列表的定义: 列表是一种可变的序列类型, 用方括号 [] 来定义。例如 `my_list = [1, 2, 3]`。

强调列表的索引规则与元组类似, 也有正索引和负索引。

#### 2. 列表的遍历方法

演示使用 `for` 循环遍历列表, 如 `for item in my_list: print(item)`。

介绍使用 `range()` 函数结合索引遍历列表，如 `for i in range(len(my_list)): print(my_list[i])`。

### 3. 列表的常用操作方法（增删改查）

增加元素：演示使用 `append()` 方法在列表末尾添加元素，如 `my_list.append(4)`；使用 `insert()` 方法在指定位置插入元素，如 `my_list.insert(1, 5)`。

删除元素：演示使用 `remove()` 方法删除指定值的元素，如 `my_list.remove(2)`；使用 `pop()` 方法删除并返回指定位置的元素，如 `popped_item = my_list.pop(0)`；使用 `del` 语句删除指定位置的元素，如 `del my_list[1]`。

修改元素：演示直接通过索引修改列表元素，如 `my_list[0] = 10`。

查询元素：演示使用 `in` 关键字判断元素是否在列表中，如 `if 3 in my_list: print('Found')`；使用 `index()` 方法获取元素的索引，如 `index = my_list.index(4)`。

## 四、列表相关代码

```
# 创建列表
my_list = [10, 20, 30, 40, 50]

# 遍历列表
print("for 循环遍历:")
for item in my_list:
    print(item)

print("\n通过索引遍历:")
for i in range(len(my_list)):
    print(my_list[i])

# 增加元素
```

```
my_list.append(60)

my_list.insert(2, 25)

print("增加元素后的列表:", my_list)

# 删除元素

my_list.remove(30)

popped = my_list.pop(0)

del my_list[1]

print("删除元素后的列表:", my_list)

# 修改元素

my_list[0] = 15

print("修改元素后的列表:", my_list)

# 查询元素

if 40 in my_list:

    print("40 在列表中")

index = my_list.index(25)

print("25 的索引是:", index)
```

课程名称	人工智能技术		教案编号	09
教材名称	人工智能导论		专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第九课 人工智能编程语言- Python			
授课题目	Python的函数、全局和局部变量			
授课学时	2节 ( ) ; 3节 ( √ ) ; 其它 ( )			
课 型	理论 ( √ ) ; 实验 ( ) ; 见习 ( ) ; 实训 ( ) ; 其它 ( )			
教学目的	<ol style="list-style-type: none"> <li>1. 理解函数的定义与调用。</li> <li>2. 掌握函数的参数和返回值。</li> <li>3. 初步了解模块和函数存储。</li> <li>4. 区分全局变量和局部变量。</li> </ol>			
教学重点	<ol style="list-style-type: none"> <li>1. 函数定义与调用：清晰掌握函数的定义语法，明确如何正确调用函数，理解函数在代码结构化和复用中的关键作用。</li> <li>2. 函数参数和返回值：学会定义带有参数的函数，理解参数传递机制；掌握函数返回值的使用，区分无返回值和有返回值的情况。</li> <li>3. 全局变量与局部变量：准确区分全局变量和局部变量，了解它们在作用域和生命周期上的差异，能合理运用局部变量优化代码。</li> </ol>			
教学难点	<ol style="list-style-type: none"> <li>1. 函数参数传递机制：深入理解 Python 中函数参数传递是值传递还是引用传递，在不同数据类型（如可变对象和不可变对象）下的表现。</li> <li>2. 模块作用域和命名空间：理解模块的作用域和命名空间如何影响函数的可见性和使用，尤其是在大型项目中模块间函数的调用和交互。</li> <li>3. 局部变量合理使用：在复杂代码场景中，能够准确判断何时使用局部变量，避免因局部变量和全局变量混淆导致的代码错误，提高代码的可读性和可维护性。</li> </ol>			
课程思政	<ol style="list-style-type: none"> <li>1. 代码规范与职业素养：强调函数定义、变量使用等代码编写的规范，培养学生严谨、细致的职业态度，就像石油化工生产中严格遵循操作规程一样，确保代码质量和程序稳定性。</li> <li>2. 团队协作与模块化思维：通过模块和函数的学习，引导学生理解在软件开发中团队协作的重要性，每个模块和函数就像团队中的成员，分工明确、相互协作，共同完成项目任务，培养学生的团队协作精神。</li> <li>3. 创新思维与问题解决：鼓励学生在函数设计和使用中发挥创新思维，尝试不同的参数设置和返回值处理方式来解决实际问题，培养学生在面对石油化工领域复杂问题时勇于探索、积极创新的品质。</li> </ol>			
教学方法	讲授 ( √ ) ; 讨论 ( ) ; 指导 ( ) ; 示教 ( √ ) ; 其它 ( )			
电子教案	有 ( √ )	Microsoft PowerPoint ( √ ) ; Author ware ( ) ; 其它 ( )		
	无 ( )			
教学资源	多媒体 ( √ ) ; 模型 ( ) ; 标本 ( ) ; 实物 ( ) ; 音像 ( ) ; 其它 ( √ )			
教学过程 时间安排	3学时			
思考题	写一个函数，在函数内部定义局部变量和全局变量，对局部变量进行操作后返回，观察全局变量的值是否改变，思考如何在函数中合理使用局部变量来避免对全局变量的意外修改。			

## 第九课：人工智能编程语言- Python（3 学时）

授课老师：吴晓畅

### 一、函数基础知识

#### 1. 概念和作用

讲解函数是一段可重复使用的代码块，用于执行特定任务，可以提高代码的复用性、可读性和可维护性。

#### 2. 定义语法

演示函数定义的基本语法：`def 函数名(参数列表):`，然后是函数体，最后可以用 `return` 语句返回结果。例如：

```
def greet():  
    print("Hello, welcome to the petrochemical data processing!")
```

#### 3. 调用方法

演示如何调用函数，直接使用函数名加括号即可，如 `greet()`。

#### 4. 函数返回值的用法

无返回值情况

讲解有些函数只是执行一些操作，不需要返回结果，此时可以不使用 `return` 语句或者使用 `return` 但不返回任何值（相当于返回 `None`）。例如：

```
def print_info():  
    print("This is a petrochemical data processing function.")
```

```
# 无返回值
```

```
print_info()
```

有返回值情况

演示如何定义有返回值的函数，使用 `return` 语句返回计算结果等。例如：

```
python
```

```
def calculate_average(a, b, c):
```

```
    average = (a + b + c) / 3
```

```
    return average
```

```
result = calculate_average(80, 90, 70)
```

```
print("The average is:", result)
```

#### 5. 模块的作用域和命名空间

## 作用域

简单介绍模块的作用域，即模块内定义的函数和变量在模块内部是可见的，但在模块外部需要通过导入模块才能访问。

## 命名空间

讲解命名空间是用于存储变量和函数名的容器，不同模块有各自的命名空间，避免命名冲突。例如，在两个不同的模块中定义同名的函数，通过模块名可以区分它们。

## 二、函数相关代码

### # 1. 函数的定义与调用

```
print("===== 1. 函数的定义与调用 =====")
```

#### # 定义一个简单的函数

```
def greet():  
    """这是一个问候函数，打印问候语"""  
    print("Hello, 同学们！欢迎来到Python课堂！")
```

#### # 调用函数

```
greet() # 输出：Hello, 同学们！欢迎来到Python课堂！
```

#### # 带参数的函数

```
def greet_by_name(name):  
    """根据名字问候"""  
    print(f"Hello, {name}！今天学习Python开心吗？")
```

#### # 调用带参数的函数

```
greet_by_name("小明") # 输出：Hello, 小明！今天学习Python开心吗？
```

### # 2. 函数的参数和返回值

```
print("\n===== 2. 函数的参数和返回值 =====")
```

#### # 带默认参数的函数

```
def calculate_area(width, height=1):  
    """计算矩形面积，height默认为1"""  
    return width * height
```

#### # 调用函数

```
area1 = calculate_area(5) # 只传一个参数，使用默认高度  
area2 = calculate_area(3, 4) # 传两个参数  
print(f"面积1: {area1}, 面积2: {area2}") # 输出：面积1: 5, 面积2: 12
```

#### # 返回多个值（实际上是返回一个元组）

```
def circle_calculations(radius):  
    """计算圆的周长和面积"""  
    circumference = 2 * 3.14159 * radius  
    area = 3.14159 * radius ** 2
```

```
return circumference, area # 返回元组

# 调用并解包返回值
circ, arr = circle_calculations(5)
print(f"半径为5的圆：周长={circ:.2f}，面积={arr:.2f}")

# 3. 将函数存储在模块中
print("\n===== 3. 将函数存储在模块中 =====")

# 假设我们有一个模块文件 math_utils.py，内容如下：
"""
# math_utils.py
def add(a, b):
    return a + b

def subtract(a, b):
    return a - b
"""

# 在当前文件中导入并使用
# import math_utils # 导入整个模块
# print(math_utils.add(5, 3)) # 输出：8

# 也可以只导入特定函数
# from math_utils import add
# print(add(10, 20)) # 输出：30
```

### 三、全局变量和局部变量基础知识

#### 区别

强调全局变量是在模块顶层定义的变量，在整个模块内都可以访问；局部变量是在函数内部定义的变量，只在函数内部有效。

#### 联系

说明局部变量可以基于全局变量进行操作，但直接修改全局变量需要使用 `global` 关键字（简单提及，不深入展开复杂用法）。

### 四、全局变量和局部变量相关代码

```
global_var = 10 # 全局变量

def example_function():

    local_var = 20 # 局部变量
```

```
print("Inside function - global_var:", global_var)

print("Inside function - local_var:", local_var)

example_function()

print("Outside function - global_var:", global_var)

# print("Outside function - local_var:", local_var) # 这行代码会报错,
# 因为 local_var 是局部变量
```

## 五、课程小结

同学们，今天我们学习了 Python 中函数的关键知识。函数就像是一个小工具，能帮我们重复执行特定任务，提高代码的复用性。我们掌握了函数的定义和调用方法，知道怎么给函数传参数以及获取返回值。同时，我们也了解了模块的作用域和命名空间，这能让我们更好地组织代码，避免变量冲突。另外，全局变量和局部变量的区别也很重要，合理使用局部变量能让我们的代码更清晰、更易维护。希望大家课后多练习，把这些知识运用到石油化工相关的编程实践中。

## 六、课程作业

### 函数定义与调用练习

让学生定义一个函数，实现两个数的减法运算，并调用该函数计算 10 和 3 的差。

### 返回值练习

编写一个函数，判断一个数是否为偶数，有返回值（True 或 False），然后在主程序中调用该函数并输出结果。

课程名称	人工智能技术		教案编号	10
教材名称	人工智能导论		专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第十课 项目实践			
授课题目	AI驱动：生成石油化工技术专业的综合认识报告			
授课学时	2节（ ）；3节（√）；其它（ ）			
课 型	理论（ ）；实验（ ）；见习（ ）；实训（ ）；其它（√）			
教学目的	<ol style="list-style-type: none"> <li>1. 理解AI生成式工具在专业报告撰写中的应用原理。</li> <li>2. 能够使用AI工具完成综合报告框架搭建、内容填充与格式优化。</li> <li>3. 培养对AI工具的批判性使用思维，避免过度依赖技术而忽视专业知识积累。</li> </ol>			
教学重点	<ol style="list-style-type: none"> <li>1. AI生成式工具在专业报告撰写中的应用原理：让学生理解AI工具如何基于算法和模型，对石油化工相关数据和知识进行分析处理，从而生成报告框架和内容。</li> <li>2. 使用AI工具完成综合报告框架搭建、内容填充与格式优化：掌握利用AI工具快速搭建报告结构、填充专业内容以及优化报告格式的方法和步骤。</li> <li>3. 批判性使用思维培养：引导学生认识到AI工具的局限性，学会结合专业知识对AI生成的内容进行评估和修改，避免盲目依赖。</li> </ol>			
教学难点	<ol style="list-style-type: none"> <li>1. 理解AI生成式工具的内在逻辑：石油化工专业学生可能对AI算法和模型背后的原理较陌生，理解AI如何从数据中提取信息并生成报告有一定难度。</li> <li>2. AI工具与专业知识的有效融合：学生难以把握如何将AI生成的内容与自身专业知识相结合，使报告既符合AI生成的高效性，又具备专业深度和准确性。</li> <li>3. 格式优化的具体操作：不同石油化工行业报告有特定格式要求，学生可能不清楚如何利用AI工具实现精确的格式优化，如字体、段落、图表排版等。</li> </ol>			
课程思政	<ol style="list-style-type: none"> <li>1. 科技与专业融合：强调AI技术作为工具，要服务于石油化工专业知识的学习和应用，培养学生将现代科技与传统专业知识相结合的意识，推动行业创新发展。</li> <li>2. 知识产权与学术诚信：提醒学生AI生成的内容可能涉及知识产权问题，要正确引用和标注，遵守学术诚信原则，树立正确的价值观。</li> </ol>			
教学方法	讲授（√）；讨论（ ）；指导（ ）；示教（√）；其它（ ）			
电子教案	有（√）	Microsoft PowerPoint（√）；Author ware（ ）；其它（ ）		
	无（ ）			
教学资源	多媒体（√）；模型（ ）；标本（ ）；实物（ ）；音像（ ）；其它（√）			
教学过程 时间安排	3学时			
思考题	在利用AI工具优化石油化工专业报告格式时，如何确保格式既符合行业规范，又能体现报告的专业性和可读性？			

# 第十课：石油化工技术专业的综合认识报告（3 学时）

## 授课老师：吴晓畅

### 一、课程导入

#### 1. 回顾旧知

提问学生：“在之前的课程中，我们学习了哪些人工智能技术在石油化工领域的应用？这些技术如何帮助我们提高工作效率？”引导学生回顾相关知识点，为本次项目实践做铺垫。

#### 2. 引入新课

展示一份传统的石油化工行业分析报告，指出撰写报告过程中可能遇到的困难，如数据收集繁琐、内容撰写耗时等。然后介绍 AI 生成式工具在报告撰写方面的优势，引出本次项目实践的主题——基于石油化工数据集，编写程序生成行业分析报告。

### 二、知识讲解

#### 1. AI 生成式工具在专业报告撰写中的应用原理

文本生成：介绍 AI 工具基于自然语言处理技术，根据输入的数据和指令生成报告文本的原理。包括如何选择合适的语言模型、如何设置生成参数等。

格式优化：说明 AI 工具如何根据预设的格式要求，对生成的报告进行排版和美化。例如，设置字体、字号、段落间距、图表位置等。

#### 2. AI 工具操作要点

选择合适的 AI 工具：介绍几种常用的 AI 生成式工具，如 ChatGPT、文心一言等，并分析它们在报告撰写方面的特点和优势。

输入指令技巧：讲解如何编写清晰、准确的指令，让 AI 工具更好地理解需求。例如，明确报告的主题、受众、内容要点等。

评估与修改：强调对 AI 生成内容进行评估的重要性，介绍评估的标准和方法，如内容的准确性、逻辑性、专业性等。同时，讲解如何对不满意的生成内容进行修改和优化。

### 三、演示操作

#### 1. 报告框架搭建

打开 AI 工具，输入指令：“请根据提供的石油化工数据集，生成一份行业分析报告的框架，包括引言、市场分析、技术发展、竞争态势、结论与建议等部分。”

展示 AI 工具生成的报告框架，并解释每个部分的作用和主要内容。

#### 2. 内容填充

以“市场分析”部分为例，输入指令：“请根据数据集中的石油化工产品产量和价格数据，分析当前市场的供需情况和价格走势。”

展示 AI 工具生成的内容，并对内容进行分析 and 点评，指出优点和不足之处。

#### 3. 格式优化

输入指令：“请对生成的报告进行格式优化，设置标题为宋体二号加粗，正文为宋体小四，段落首行缩进 2 字符，添加页眉页脚等。”

展示格式优化后的报告，让学生直观感受格式优化的效果。

### 四、小组任务布置与实施

将学生分成若干小组，每组 4 - 5 人。为每个小组提供一份石油化工数据集，要求小组使用 AI 工具完成一份完整的行业分析报告，包括报告框架搭建、内容填充与格式优化。

强调小组分工合作的重要性，要求每个小组明确成员的职责，如数据收集员、指令编写员、内容审核员、格式优化员等。

课程名称	人工智能技术		教案编号	11
教材名称	人工智能导论		专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第十一课 图像识别			
授课题目	图像识别原理及其应用			
授课学时	2节 ( ) ; 3节 ( √ ) ; 其它 ( )			
课 型	理论 ( √ ) ; 实验 ( ) ; 见习 ( ) ; 实训 ( ) ; 其它 ( )			
教学目的	<ol style="list-style-type: none"> <li>1. 了解图像识别技术的发展阶段与核心任务。</li> <li>2. 掌握图像预处理、特征提取与分类算法的基本原理。</li> <li>3. 能够使用Python库（如OpenCV）完成简单图像处理。</li> </ol>			
教学重点	<ol style="list-style-type: none"> <li>1. 技术流程掌握：理解图像识别从采集、预处理、特征提取到分类的完整流程，以及各环节的相互关系和作用。</li> <li>2. 算法原理理解：掌握图像预处理、特征提取方法及常见分类算法（KNN、SVM）的基本原理，明白其实现逻辑。</li> <li>3. Python库应用：学会使用Python库（如OpenCV）进行简单图像处理操作，为后续实践项目奠定基础。</li> </ol>			
教学难点	<ol style="list-style-type: none"> <li>1. 算法原理深度：KNN和SVM算法涉及较多数学知识和复杂逻辑，学生较难深入理解其原理和参数设置对结果的影响。</li> <li>2. 特征提取技巧：如何从图像中提取有效、有区分度的特征是关键，学生难以把握合适的特征提取方法及特征与分类结果的联系。</li> <li>3. 实践代码调试：在Python实现手写数字识别时，会遇到代码报错、算法效果不理想等问题，学生缺乏调试和优化代码的经验。</li> </ol>			
课程思政	<p>社会责任意识：让学生明白图像识别技术若应用不当可能带来的风险，如错误识别导致化工事故等，引导学生树立正确的价值观和社会责任感，合理运用技术。</p>			
教学方法	讲授 ( √ ) ; 讨论 ( ) ; 指导 ( ) ; 示教 ( √ ) ; 其它 ( )			
电子教案	有 (√)	Microsoft PowerPoint (√) ; Author ware ( ) ; 其它 ( )		
	无 ( )			
教学资源	多媒体 (√) ; 模型 ( ) ; 标本 ( ) ; 实物 ( ) ; 音像 ( ) ; 其它 (√)			
教学过程 时间安排	3学时			
思考题	技术演进思考：回顾图像识别技术的发展阶段，结合石油化工行业需求，推测未来该技术可能会朝着哪些方向发展？			

# 第十一课：图像识别（3 学时）

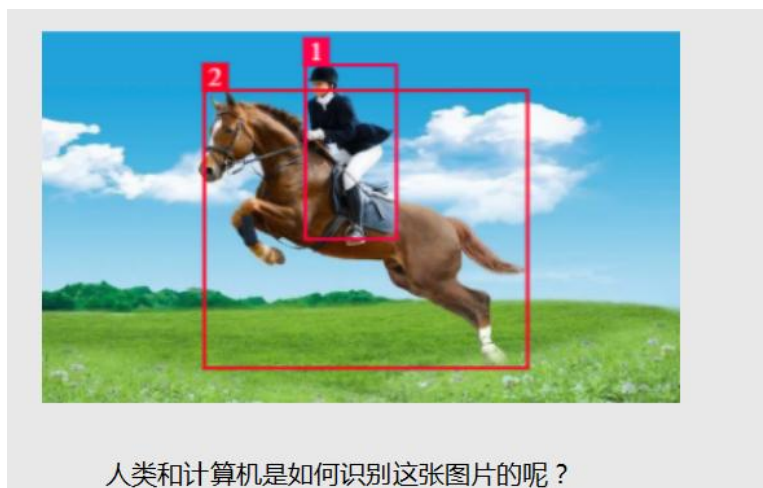
授课老师：吴晓畅

## 一、物体识别背景介绍

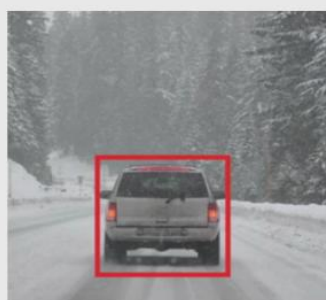
伴随着图片成为互联网中的主要信息载体，难题随之出现。当信息由文字记载时，我们可以通过关键词搜索轻易找到所需内容并进行任意编辑，而当信息由图片记载时，我们却无法对图片中的内容进行检索，从而影响了我们从图片中找到关键内容的效率。图片给我们带来了快捷的信息记录和分享方式，却降低了我们的信息检索效率。在这个环境下，借助计算机实现图像主体的识别技术就显得尤为重要。

物体识别，指利用计算机对图像进行处理、分析和理解，以识别图像中各种不同模式的目标和对象的技术。通俗地讲，物体识别是让计算机像人一样读懂图片的内容，识别出图像中有什么物体，并报告出这个物体在图像中的位置和方向。

## 二、物体识别的任务划分



a) 分类



b) 分类+定位



c) 多目标检测

物体识别任务划分

### 三、图像分类的相关知识

#### 1. 图像的预处理

平移：把我们感兴趣的、图中的对象移到图像的中间位置，便于尽可能准确地、定量地分析图像的特征。

缩放：一个数组越大，计算机处理该数组时消耗的内存越大、花费的时间越长，因而可以根据需要对图像大小进行调整。

填充：为了使待处理图片符合分类模型对图像像素尺寸的要求，用特定的像素值对图片的长或宽的像素数量进行补充。

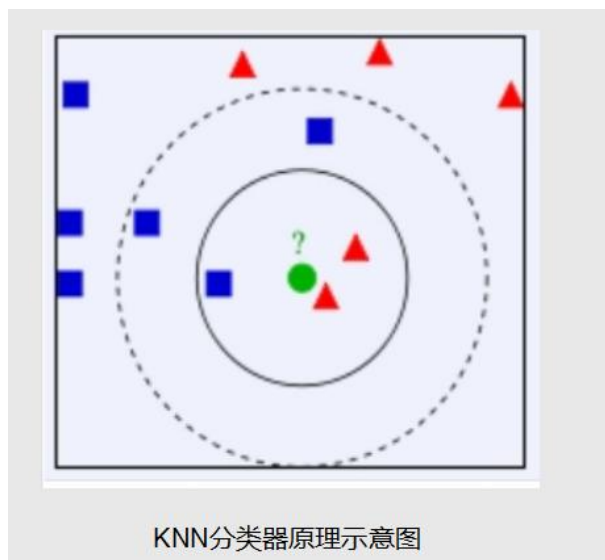
#### 2. 传统分类算法

特征提取：

(1) 颜色直方图、局部二值模式、方向梯度直方图特征、尺度不变特征变换、Gabor特征、区域协方差描述符。

(2) 主成分分析 (Principal Component Analysis, PCA)、线性判别分析 (linear discriminant analysis, LDA)、流形学习 (manifold learning) 和稀疏编码 (Sparse Coding, SC)。

(3) 分类器：K近邻(K-Nearest Neighbor, KNN)、支持向量机 (Support Vector Machine, SVM)、决策树、朴素贝叶斯、逻辑回归等。



### 3. 深度学习算法

深度学习算法的特点

优势：在训练集数量充足的情况下准确率高

局限性：（1）大数据集花费昂贵且耗时，并不容易获得

（2）训练阶段对硬件要求高

## 四、目标检测的相关知识

非极大值抑制（Non-Maximum Suppression, NMS）：当边框预测环节生成了大量建议边框后，接着要为每个矩形框做类别分类概率，最后需要判别哪些矩形框是没用的。所谓的非极大值抑制就是根据分类器类别分类概率做排序，从小到大排序，先拿最大概率候选框与其他框计算重叠度IOU，丢弃高于阈值的建议框。然后从没有被丢弃的建议框中再找最大概率建议框，重复上述操作，直到找到所有被保留下来的建议框。



滑窗法改进

1. 两步检测算法：R-CNN
2. 一步检测算法：YOLO



## 五、物体识别的特点和应用领域

### 1. 局限性

多视角和尺度：同一个物体的图像，它的角度和尺度是多变的，获取到的这个物体的图片也是多样的；

遮挡：目标物体可能被挡住，图片中只有目标物体的一小部分；

3照条件：在像素层面上，光照对图片的影响非常大；

类内差异：同一类别的个体之间有许多不同的对象，每个个体都有自己的外形。

### 2. 优势

在大数据时代，人类获取的图片资料越来越多，人们需要借助计算机去承担这些复杂沉重的工作，否则这些工作将由人类负担。

在一些极端环境下，比如水下、严寒或酷暑的环境下的物体识别，如果工作由人工智能来完成，人类面临的危险、伤害和压力将有所减轻。

人类判断物体类别的准确率总体上会比机器的识别率高，但在一些要求精密度的识别，比如缺陷识别、医学图片等识别，人类的优势可能会相应降低。在这类场合，通过机器替代人来实现物体识别，能够减少错误和缺陷。

## 六、课程小结

人类感觉信息中的80%都是视觉信息，图像是人类获取和交换信息的主要来源。机器视觉之于人工智能的意义就是视觉之于人类的意义，而决定着机器视觉技术之一就是物体识别技术。随着计算机技术的不断发展，图像识别技术也在不断地优化，其算法也在不断地改进。在某些应用场景，机器视觉比人类的生理视觉更具优势，它更加准确、客观和稳定。利用计算机代替人类实现物体识别，能够尽量缩短时间并减少资源。

## 七、课堂练习



## 数字识别--选择数字查看

手写数字识别是常见的图像识别任务。不同人的手写风格迥异，大小不一，造成了计算机对手写数字识别任务的一些困难。手写数字识别由于其...[展开全文](#)



课程名称	人工智能技术		教案编号	12
教材名称	人工智能导论		专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第十二课 无人驾驶			
授课题目	无人驾驶原理及其应用			
授课学时	2节（ ）；3节（√ ）；其它（ ）			
课 型	理论（√）；实验（ ）；见习（ ）；实训（ ）；其它（ ）			
教学目的	<ol style="list-style-type: none"> <li>1. 理解无人驾驶系统基本组成。</li> <li>2. 深入了解环境感知系统。</li> <li>3. 了解定位和导航技术</li> <li>4. 理解路径规划算法。</li> <li>5. 掌握决策、规划和控制策略</li> </ol>			
教学重点	<ol style="list-style-type: none"> <li>1. 无人驾驶系统基本组成：掌握无人驾驶系统的核心模块及其相互关系，理解各模块在整体系统中的作用。</li> <li>2. 环境感知系统：深入了解各类传感器（如激光雷达、摄像头、毫米波雷达等）的工作原理及其在环境感知中的应用。</li> <li>3. 定位和导航技术：理解GPS、惯性导航系统、地图匹配等定位技术的原理，以及它们在无人驾驶车辆定位中的应用。</li> </ol>			
教学难点	<ol style="list-style-type: none"> <li>1. 多传感器融合技术：如何将来自不同传感器的数据进行有效融合，以提高环境感知的准确性和鲁棒性。</li> <li>2. 复杂交通环境下的路径规划：在动态变化的交通环境中，如何设计高效的路径规划算法，确保无人驾驶车辆的安全行驶。</li> <li>3. 决策、规划和控制策略：理解无人驾驶车辆在面对复杂交通环境时的决策逻辑，以及如何将这些决策转化为具体的控制指令。</li> </ol>			
课程思政	<ol style="list-style-type: none"> <li>1. 伦理与安全责任：强调无人驾驶技术发展中的伦理问题，如数据隐私保护、安全责任界定等，培养学生的科技伦理意识。</li> <li>2. 创新驱动与产业发展：介绍无人驾驶技术在石油化工行业（如危险品运输、厂区巡检等）的应用前景，鼓励学生关注技术创新对产业发展的推动作用。</li> </ol>			
教学方法	讲授（√）；讨论（ ）；指导（ ）；示教（√）；其它（ ）			
电子教案	有（√）	Microsoft PowerPoint（√）；Author ware（ ）；其它（ ）		
	无（ ）			
教学资源	多媒体（√）；模型（ ）；标本（ ）；实物（ ）；音像（ ）；其它（√）			
教学过程 时间安排	3学时			
思考题	路径规划挑战：在面对复杂交通环境时，无人驾驶车辆的路径规划面临哪些挑战？如何应对这些挑战？			

## 第十二课：无人驾驶（3 学时）

授课老师：吴晓畅

### 一、课程导入

#### 1. 引入话题：

展示一段无人驾驶车辆在真实道路或特定场景（如石油化工厂区）行驶的视频，激发学生兴趣。

提问：“无人驾驶车辆是如何在没有人类驾驶员的情况下，实现安全、高效行驶的？”

#### 2. 课程目标：

明确本节课的学习目标，包括理解无人驾驶系统的基本组成、环境感知系统、定位和导航技术、路径规划算法，以及决策、规划和控制策略。

### 二、无人驾驶系统基本组成介绍

#### 1. 系统概述

车辆实现无人驾驶，必须经由三大技术环节才能实现，其中每一个环节都离不开人工智能技术。

##### (1) 感知系统。

感知也就是让车辆获取信息，各种系统所用的车辆传感器也各有不同的类型，最常用的包括红外雷达、超声雷达、激光雷达、毫米波雷达、图像传感器和轮速感测器等，以检测车辆的工作状态，收集车辆的实时信息，读取不断发生变化的状况参数。

##### (2) 决策系统。

在对周围物体作出识别进行了解并进行定位后，决策系统就可以开始做出相关的决策：利用算法决定汽车的下一步的行为。从最简单的“红灯停、绿灯行”到变道换路等，决策系统不但要利用实时的定位，还要结合当前道路拥堵

情况、路面质量、信号灯等待时间等一系列因素来进行决策以获取最佳的行驶路线。。

(3) 执行系统。

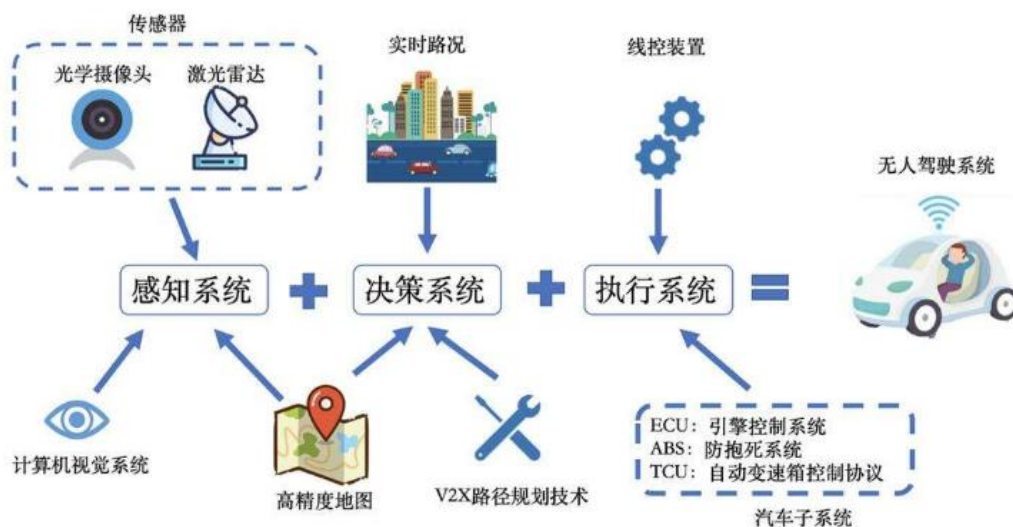
执行系统是在决策系统进行操控之后，对汽车的油门、刹车、方向盘、车灯等进行物理层面的控制，严格的执行决策系统的决策。

2. 关键组件

感知层：传感器（激光雷达、摄像头、毫米波雷达等）、V2X通信等。

决策层：路径规划、决策算法、控制策略等。

执行层：车辆控制系统（转向、加速、制动等）。



3. 互动讨论

引导学生思考各组件在无人驾驶系统中的作用，以及它们之间的相互关系。

三、无人驾驶感知系统



## 1. 传感器原理与工作方式

激光雷达：发射激光束并测量反射时间，获取三维环境信息。

摄像头：捕捉图像信息，用于目标识别、车道线检测等。

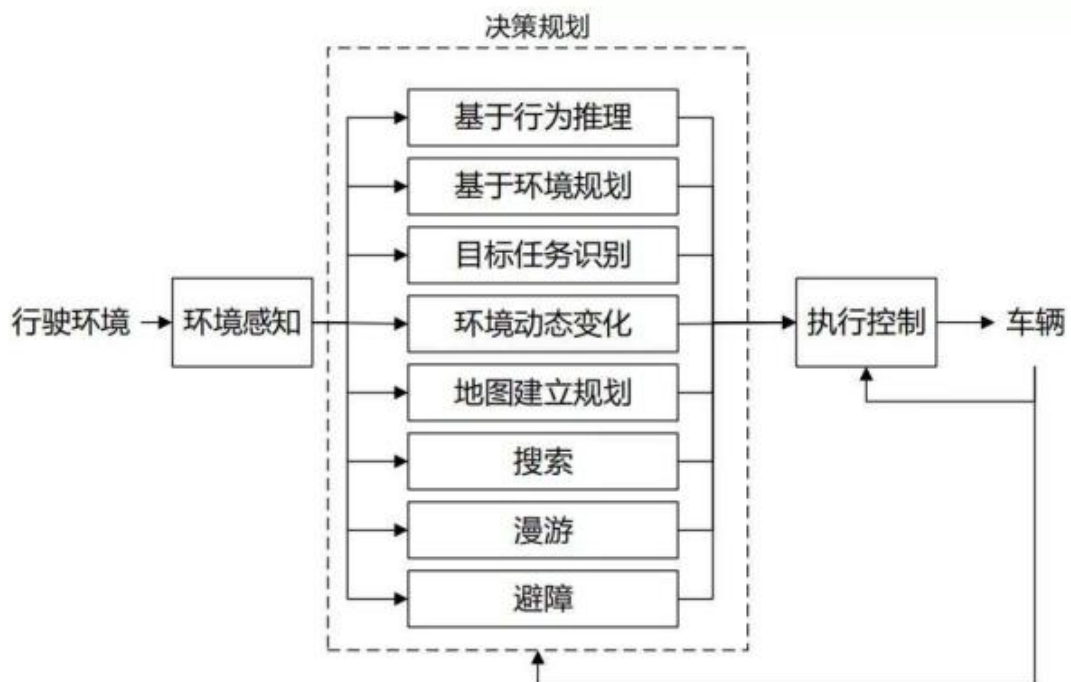
毫米波雷达：利用毫米波段电磁波进行测距、测速。

## 2. 多传感器融合

讲解多传感器融合的重要性，以及如何通过数据融合提高环境感知的准确性和鲁棒性。

# 四、无人驾驶系统

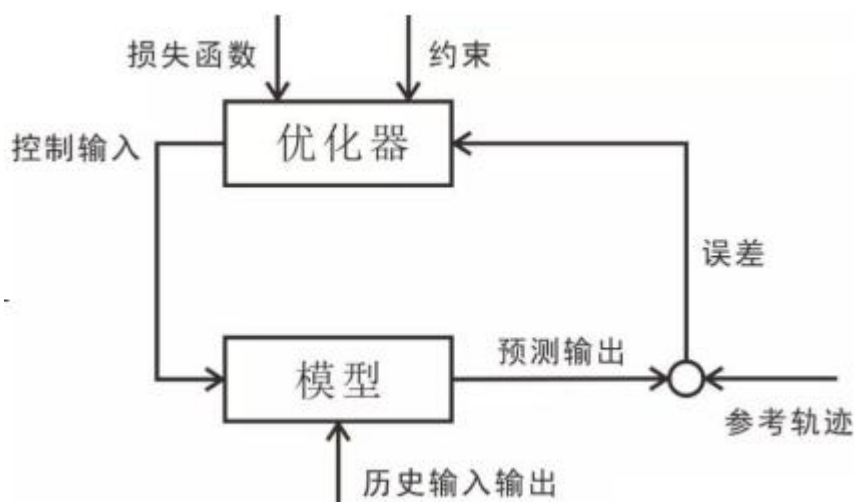
## 1. 决策系统



## 2. 执行系统

执行系统作为无人车系统的最底层，其任务是将规划好的动作实现，所以控制模块的评价指标即为控制的精准度。控制器通过比较车辆的测量结果和我们预期的状态输出控制动作，这一过程被称为反馈控制（Feedback Control）。

反馈控制被广泛的应用于自动化控制领域，其中最典型的反馈控制器当属PID控制器（Proportional-Integral-Derivative Controller）。



### 3. 人工智能在无人驾驶中的应用

人工智能技术的不断开发为无人驾驶汽车提供了源动力，同时，无人驾驶汽车的快速发展也为人工智能的深入发掘提出了更高深的要求。如何更好地实现系统感知、信息处理和指令执行等，是车辆无人驾驶的关键所在，因此需要应用到深度学习、机器视觉、数据处理等人工智能技术。



## 五、无人驾驶面临的挑战

技术领域难关重重。虽然人工智能技术的发展必将日趋成熟,但作为上路行驶的智能汽车在某方面仍旧存在一些技术问题需要解决,例如,如何分辨、区分不同路标存在难度,如施工路段、临时限行、内部道路等。在行为决策方面,对于路上飘起的树叶、纸屑、塑料袋等非必要环境会不会进行误报,而引起紧急

刹车。对于临时限行、交警指挥、突发情况等突发路况信息,如何进行采集、识别与规划。如何确保复杂环境下对交通环境感知无盲区和决策最优化。同样,自动驾驶的判断、数据,需要依赖于高带宽的网络。

安全领域备受关注。人工智能的车辆无人驾驶技术是以互联网为载体进行工作的。它必须借助于互联网实时更新交通状况、上传和接收数据,这也决定了AI技术和无人驾驶车辆对互联网的依赖程度越来越高。但是,分析当前的网络状况却并不安全,网络攻击事件屡屡发生,如果无人驾驶车辆在技术上被黑客攻击或被别有用心者利用时,系统反应的延迟甚至疲软崩溃都有可能造成重大的安全隐患,这是它将成为危害社会和公民的最直接有效的工具。

伦理领域矛盾突出。每次科技的应用都会给传统道德带来挑战,人工智能在自动驾驶中的应用也让人们陷入道德判断的困境中。如当智能汽车躲避突发情况刹车失灵时,是撞向动物还是行人?是撞向行人确保驾乘者安全还是撞向建筑物确保行人安全?自动驾驶汽车的到来,让人们不得不面对这样的伦理冲突的困境,并坏得不做出抉择。

课程名称	人工智能技术	教案编号	13
教材名称	人工智能导论	专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第十三课 数据挖掘		
授课题目	数据挖掘原理及其应用		
授课学时	2节（ ）；3节（ <input checked="" type="checkbox"/> ）；其它（ ）		
课 型	理论（ <input checked="" type="checkbox"/> ）；实验（ ）；见习（ ）；实训（ ）；其它（ ）		
教学目的	<ol style="list-style-type: none"> <li>1. 理解数据挖掘的定义。</li> <li>2. 掌握数据挖掘的步骤。</li> <li>3. 了解数据挖掘的经典算法。</li> </ol>		
教学重点	<ol style="list-style-type: none"> <li>1. 数据挖掘定义：明确数据挖掘是从大量数据中提取有用信息和知识的过程。</li> <li>2. 数据挖掘步骤：掌握数据挖掘的一般流程，包括问题定义、数据收集、预处理、挖掘、评估和解释。</li> <li>3. 经典算法原理：理解决策树、支持向量机、朴素贝叶斯等算法的基本原理及其在数据挖掘中的应用。</li> </ol>		
教学难点	<ol style="list-style-type: none"> <li>1. 算法原理深入理解：学生可能难以深入理解决策树、支持向量机等算法的数学原理和计算过程。</li> <li>2. 数据预处理技巧：数据预处理是数据挖掘的关键步骤，学生可能难以掌握数据清洗、转换和特征选择等技巧。</li> <li>3. 算法选择与应用：在面对具体问题时，如何选择合适的算法并应用到实际场景中，对学生来说是一个挑战。</li> </ol>		
课程思政	<ol style="list-style-type: none"> <li>1. 数据安全与隐私保护：强调在数据挖掘过程中，必须严格遵守数据安全和隐私保护的原则，不得泄露敏感信息。</li> <li>2. 科学精神与求真务实：培养学生严谨的科学态度，追求真实、准确的数据挖掘结果，避免主观臆断和误导性结论。</li> </ol>		
教学方法	讲授（ <input checked="" type="checkbox"/> ）；讨论（ ）；指导（ ）；示教（ <input checked="" type="checkbox"/> ）；其它（ ）		
电子教案	有（ <input checked="" type="checkbox"/> ）	Microsoft PowerPoint（ <input checked="" type="checkbox"/> ）；Author ware（ ）；其它（ ）	
	无（ ）		
教学资源	多媒体（ <input checked="" type="checkbox"/> ）；模型（ ）；标本（ ）；实物（ ）；音像（ ）；其它（ <input checked="" type="checkbox"/> ）		
教学过程 时间安排	3学时		
思考题	数据挖掘定义理解：简述数据挖掘的定义，并举例说明其在石油化工行业中的一个应用场景。		

## 第十三课：数据挖掘（3 学时）

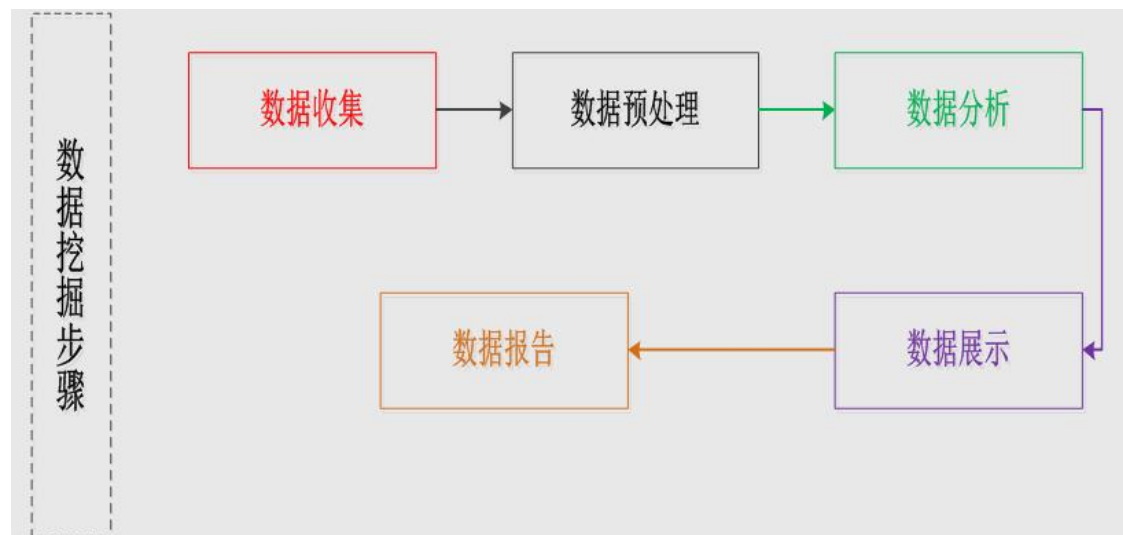
授课老师：吴晓畅

### 一、数据挖掘技术介绍

数据挖掘则是机器学习中的一个重要分支，即从大量的数据中挖掘出有价值、关键的信息来帮助决策。从商业的角度上看，数据挖掘其实是一类深层次的数据分析方法。数据分析本身已经有很多年的历史，只不过在过去数据收集和分析的目的是用于科学研究，另外，由于当时计算能力的限制，对大数据量进行分析的复杂数据分析方法受到很大限制。

数据挖掘是人工智能领域非常重要的一个分支，通常也称为数据库中的知识发现(Knowledge Discovery in Database, KDD)，数据发掘在零售业、制造业、财务金融保险、通讯及医疗服务等相关领域都有非常广泛的应用。并且，很多公司运用数据挖掘技术取得了很多成功的案例，比如，电子商务网站(淘宝、京东、苏宁等等)对网站数据进行分析，识别用户的行为模式，对用户进行个性化推荐，优化网站设计。

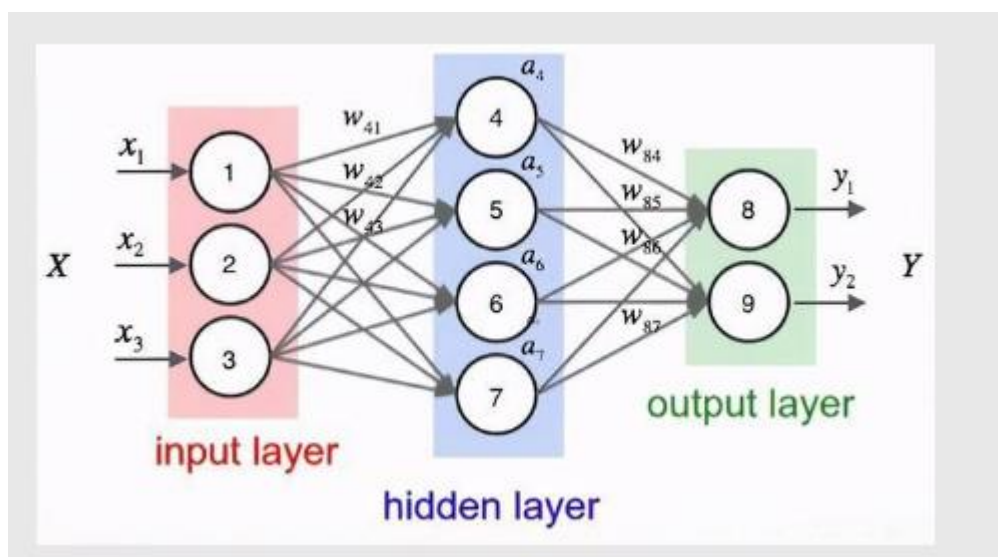
### 二、数据挖掘步骤



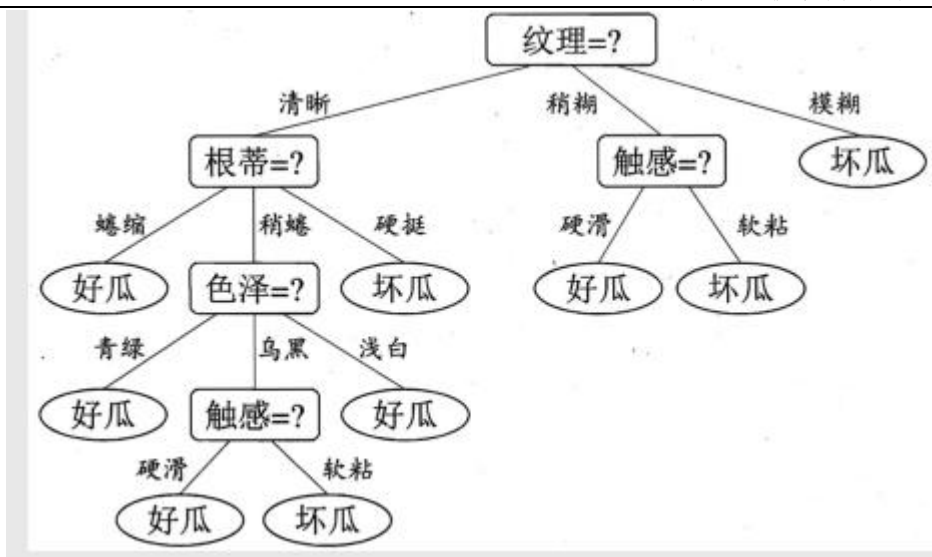
### 三、数据挖掘常用方法



神经网络算法。神经网络技术起源于上世纪五六十年代，拥有输入层、隐藏层以及输出层。图10-15显示了基本的神经网络结构。由基本的神经网络可以组合、变换得到各种各样更加复杂的深度网络，比如卷积神经网络、循环神经网络、深度信念网络、生成对抗网络。



决策树算法。决策树算法起源于E. B. Hunt等人于1966年发表的论文“experiments in Induction”，改论文详细介绍了决策树的构建、应用的全过程。之后，（罗斯·昆兰）等人提出了ID3决策树算法，使得决策树算法的相关研究进入高潮，并由此衍生出C4.5、CART等相关决策树算法。顾名思义，决策树就是一棵树，一颗决策树包含一个根节点、若干个内部结点和若干个叶结点。



#### 四、数据挖掘技术应用

传统的成绩管理系统可以对学生各科成绩进行统计，但是一门课程的成绩与不同科目之间的联系以及与上课老师讲授的方法，平时表现等这些隐藏信息可以用数据挖掘的方法建立模型来得到，并通过数据展示来看到这些因素是如何对成绩产生影响。通过数据挖掘从学生的成绩中找到隐含在其中的有效信息，这样既可以帮助老师了解学生的学习情况，又可以帮助学生了解学习重点，达到教学相长的目的。

姓名	语文	数学	姓名	语文	数学	姓名	语文	数学	姓名	语文	数学
吴宇航	75	97	徐健康	88	86	肖彩霞	72	60	陈雯	74	63
王刚	78	94	谢雨欣	91	85	陈雨杰	75	72	尹娜娜	88	77
胡双妹	76	88	王博	85	77	邓天	74	82	吴罕	77	76
谢文昌	87	79	刘星磊	90	84	陈飞	67	77	宋文峰	75	87
潘俊宇	88	93	李莉萍	89	75	谢明江	69	84	熊莉莉	78	66
汤建琴	80	84	陈丽君	80	36	左慈玲	88	76	陈林罕	43	62
吴佳豪	72	87	王洁	85	79	王佳佳	69	58	潘思宇	77	90
邓媛芬	82	94	谷雪	74	80	陈欢欢	70	29	王二妹	20	45
刘洪明	82	91	陈怡琦	75	77	邹玲玲	75	60	段小勇	91	87
邹滨宇	79	80	沈静	80	65	王月秋	48	72	潘虹鸣	73	65
李雨莲	83	68	孙佳	82	90	刘鉴锋	64	95	王盟	84	73
张艺茹	85	70	张文涛	81	68	邓洪英	83	62	欧力滨	60	49
桂田东	80	92	徐会敏	78	67	孙铁英	72	77	徐龙	73	17

课程名称	人工智能技术		教案编号	14
教材名称	人工智能导论		专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第十四课 项目实战1			
授课题目	基于聚类分析的石油化工生产过程故障预判系统			
授课学时	2节 ( ) ; 3节 ( <input checked="" type="checkbox"/> ) ; 其它 ( )			
课 型	理论 ( <input checked="" type="checkbox"/> ) ; 实验 ( ) ; 见习 ( ) ; 实训 ( ) ; 其它 ( )			
教学目的	<ol style="list-style-type: none"> <li>1. 了解基于聚类分析的石油化工生产过程故障预判系统。</li> <li>2. 理解聚类分析的基本概念和其在故障预判中的应用。</li> <li>3. 熟悉数据集和数据预处理的过程。</li> </ol>			
教学重点	<ol style="list-style-type: none"> <li>1. 数据挖掘定义：明确数据挖掘是从大量数据中提取有用信息和知识的过程。</li> <li>2. 数据挖掘步骤：掌握数据挖掘的一般流程，包括问题定义、数据收集、预处理、挖掘、评估和解释。</li> <li>3. 经典算法原理：理解决策树、支持向量机、朴素贝叶斯等算法的基本原理及其在数据挖掘中的应用。</li> </ol>			
教学难点	<ol style="list-style-type: none"> <li>1. 算法原理深入理解：学生可能难以深入理解决策树、支持向量机等算法的数学原理和计算过程。</li> <li>2. 数据预处理技巧：数据预处理是数据挖掘的关键步骤，学生可能难以掌握数据清洗、转换和特征选择等技巧。</li> <li>3. 算法选择与应用：在面对具体问题时，如何选择合适的算法并应用到实际场景中，对学生来说是一个挑战。</li> </ol>			
课程思政	<ol style="list-style-type: none"> <li>3. 数据安全与隐私保护：强调在数据挖掘过程中，必须严格遵守数据安全和隐私保护的原则，不得泄露敏感信息。</li> <li>4. 科学精神与求真务实：培养学生严谨的科学态度，追求真实、准确的数据挖掘结果，避免主观臆断和误导性结论。</li> </ol>			
教学方法	讲授 ( <input checked="" type="checkbox"/> ) ; 讨论 ( ) ; 指导 ( ) ; 示教 ( <input checked="" type="checkbox"/> ) ; 其它 ( )			
电子教案	有 ( <input checked="" type="checkbox"/> )	Microsoft PowerPoint ( <input checked="" type="checkbox"/> ) ; Author ware ( ) ; 其它 ( )		
	无 ( )			
教学资源	多媒体 ( <input checked="" type="checkbox"/> ) ; 模型 ( ) ; 标本 ( ) ; 实物 ( ) ; 音像 ( ) ; 其它 ( <input checked="" type="checkbox"/> )			
教学过程 时间安排	3学时			
思考题				

## 第十四课：项目实战1（3 学时）

授课老师：吴晓畅

### 一、课程导入

#### 1. 引入话题

简要回顾上节课数据挖掘在石油化工中的应用，引出聚类分析作为数据挖掘中重要技术之一，在石油化工生产过程故障预判中的独特作用。

提问：“在石油化工生产过程中，如何利用数据提前发现潜在故障，避免生产事故？”

#### 2. 课程目标

明确本节课的学习目标，包括了解基于聚类分析的故障预判系统、理解聚类分析的基本概念及其在故障预判中的应用、熟悉数据集和数据预处理过程。

### 二、基于聚类分析的故障预判系统概述

#### 1. 系统作用

介绍基于聚类分析的故障预判系统如何通过分析生产过程中的数据，识别出异常模式，从而提前预警潜在故障。

#### 2. 工作流程

阐述系统的工作流程，包括数据收集、数据预处理、聚类分析、结果评估与预警等步骤。

#### 3. 关键技术

强调聚类分析在系统中的核心地位，以及数据预处理、特征选择等关键技术对系统性能的影响。

### 三、聚类分析基本概念

#### 1. 定义与目的

定义聚类分析，解释其目的是将数据集中的对象分成若干组，使得同一组内的对象相似度较高，不同组之间的对象相似度较低。

## 2. 常用算法

K-means算法：介绍其基本原理、步骤和优缺点。

层次聚类：讲解层次聚类的两种主要类型（凝聚式和分裂式）及其应用场景。

DBSCAN算法：阐述DBSCAN算法如何基于密度进行聚类，以及其在处理噪声数据时的优势。

## 3. 算法选择

讨论在选择聚类算法时需要考虑的因素，如数据集的大小、形状、噪声水平等。

# 四、数据集介绍与数据预处理

## 1. 数据集介绍

介绍石油化工生产过程中可能收集到的数据类型，如温度、压力、流量等，以及这些数据在故障预判中的作用。

## 2. 数据预处理

数据清洗：去除噪声数据、处理缺失值等。

数据标准化：将数据缩放到同一尺度，避免某些特征因量纲不同而对聚类结果产生过大影响。

特征选择：根据聚类目的选择合适的特征，减少计算复杂度，提高聚类效果。

课程名称	人工智能技术		教案编号	15
教材名称	人工智能导论		专业班级	应用化工技术251/252、新能源材料应用技术251
授课章节	第十五课 项目实战2			
授课题目	基于聚类分析的石油化工生产过程故障预判系统			
授课学时	2节 ( ) ; 3节 ( √ ) ; 其它 ( )			
课 型	理论 ( √ ) ; 实验 ( ) ; 见习 ( ) ; 实训 ( ) ; 其它 ( )			
教学目的	<ol style="list-style-type: none"> <li>1. 掌握使用Python实现K-means算法的方法。</li> <li>2. 使用预处理后的石油化工生产过程数据集进行K-means聚类分析。</li> <li>3. 熟练掌握聚类结果的可视化方法。</li> <li>4. 分析聚类结果，识别可能的故障</li> </ol>			
教学重点	<ol style="list-style-type: none"> <li>1. K-means算法实现：掌握使用Python实现K-means算法的核心代码和步骤。</li> <li>2. 聚类结果可视化：熟练使用Matplotlib或Seaborn等工具对聚类结果进行可视化展示。</li> <li>3. 故障识别与分析：通过聚类结果分析数据分布特征，识别石油化工生产过程中可能的异常或故障模式。</li> </ol>			
教学难点	<ol style="list-style-type: none"> <li>1. K-means算法参数调优：如何选择合适的K值（聚类数量）以及初始质心，避免陷入局部最优解。</li> <li>2. 数据预处理对聚类效果的影响：理解数据标准化、缺失值处理等预处理步骤对聚类结果的重要性。</li> <li>3. 聚类结果的实际意义解读：如何将聚类结果与石油化工生产过程中的实际故障现象关联起来，进行故障定位和原因分析。</li> </ol>			
课程思政	<ol style="list-style-type: none"> <li>1. 严谨的科学态度：强调在聚类分析中参数选择和结果解读的严谨性，培养学生实事求是的科学精神。</li> <li>2. 安全生产责任意识：通过故障预判系统的应用案例，引导学生认识人工智能技术在保障石油化工生产安全中的重要作用，增强安全生产责任感。</li> <li>3. 技术创新与产业应用：鼓励学生思考如何将聚类分析等人工智能技术进一步应用于石油化工行业的其他领域，推动产业技术创新。</li> </ol>			
教学方法	讲授 ( √ ) ; 讨论 ( ) ; 指导 ( ) ; 示教 ( √ ) ; 其它 ( )			
电子教案	有 ( √ )	Microsoft PowerPoint ( √ ) ; Author ware ( ) ; 其它 ( )		
	无 ( )			
教学资源	多媒体 ( √ ) ; 模型 ( ) ; 标本 ( ) ; 实物 ( ) ; 音像 ( ) ; 其它 ( √ )			
教学过程 时间安排	3学时			
思考题				

## 第十五课：项目实战2（3 学时）

### 授课老师：吴晓畅

### 一、课程导入

#### 1. 回顾与引入

简要回顾上节课关于聚类分析的基本概念及在石油化工领域的应用背景。

提问：“在石油化工生产过程中，如何利用K-means算法进行故障预判？”

引导学生思考本节课的主题。

#### 2. 课程目标说明

明确本节课的学习目标，包括掌握Python实现K-means算法、使用预处理后的数据集进行聚类分析、可视化聚类结果以及分析聚类结果识别故障。

### 二、K-means算法的基本原理与实现

#### 1. 基本原理讲解

定义K-means算法：一种无监督学习算法，用于将数据集划分为K个簇，使得同一簇内的数据点相似度较高，不同簇之间的数据点相似度较低。

算法步骤：初始化K个质心、分配数据点到最近的质心、更新质心位置、重复直到收敛。

#### 2. Python实现演示

使用Python的scikit-learn库演示K-means算法的实现过程。

代码示例：包括数据导入、模型初始化、模型训练、聚类结果获取等步骤。

#### 3. 参数选择讨论

讨论K值的选择方法，如肘部法则、轮廓系数等。

强调初始质心选择对聚类结果的影响，以及如何避免局部最优解。

### 三、聚类结果的可视化方法

#### 1. 可视化工具介绍

介绍使用Matplotlib、Seaborn等库进行聚类结果可视化的方法。

## 2. 可视化实践

演示如何使用Python代码绘制聚类结果的散点图、热力图等。

强调可视化在聚类结果分析中的重要性，如帮助直观理解数据分布、发现异常值等。

## 四、聚类结果分析与故障识别

### 1. 聚类结果分析

引导学生分析聚类结果，如簇的大小、形状、密度等。

讨论不同簇之间的差异和相似性，以及这些差异和相似性可能代表的实际意义。

### 2. 故障识别

结合石油化工生产过程的实际情况，讨论如何通过聚类结果识别可能的故障。

例如，如果某个簇的数据点明显偏离其他簇，可能代表该时间段或该设备存在异常。

## 第十六课：期末考核（3 学时）

授课老师：吴晓畅

### 一、数据集准备

提供一份预处理后的石油化工生产过程数据集（含温度、压力、流量等关键参数）。

数据集应包含正常生产数据及模拟的故障数据，以便学生分析。

### 二、环境搭建

确保机房或在线编程环境已安装Python、scikit-learn、Matplotlib、Seaborn等必要库。

提供详细的开发环境配置指南，确保学生能顺利开始编程。

### 三、考核说明文档

编写考核说明文档，明确考核要求、评分标准、提交方式等。

文档中应包含数据集描述、任务要求（如实现K-means聚类、可视化结果、分析故障等）。