

半面授校级硕士

人工智能和知识工程



tech 科学技术大学

半面授校级硕士 人工智能和知识工程

模式: 混合式 (在线+临床实践)

时间: 12个月

学位: TECH 科技大学

网络访问: www.techitute.com/cn/information-technology/hybrid-professional-master-degree/hybrid-professional-master-degree-artificial-intelligence-knowledge-engineering

目录

01 介绍	02 为什么要选择这个半面授校 级硕士?	03 目标	04 能力
4	8	12	18
	05 教学规划	06 临床实习	07 我在哪里可以进行实习?
	22	34	40
		08 方法	09 学位
		44	52

01 介绍

尽管人工工程仍处于非常早期的阶段,但事实是,它的多种应用,以及取决于数字化新技术发展速度的广泛发展空间,预示着一场可与当时互联网产生的革命相媲美的革命。这就是为什么拥有该领域以及知识工程的专业学位对于任何想要详细了解这门科学的来龙去脉并掌握使之成为可能的工具和软件的计算机科学家来说都是安全的选择。为此,TECH开发了非常完整的计划,该计划在12个月的高强度培训中结合了理论和实践,旨在使毕业生完成课程,成为该领域的专家。



“

注册这个半面授校级硕士,认真准备人工智能和知识工程的革命,从现在开始掌握其工具”

几十年来,人类将他们的知识和能力传递给技术系统的兴趣一直是复杂系统的前身,例如构成人工智能和知识工程的系统。尽管这是一个具有指数级增长幅度的领域,但事实是,今天可以找到像人一样思考或行动的设备,使决策、解决问题或学习等活动自动化。这方面的一个例子,相当普遍的是手机或虚拟语音助手(如Siri或Alexa)或面部检测。

它的多种应用,以及这门科学发展带来的可能性,使其成为未来几十年的基本技术。因此,为了使毕业生能够自主和集中地专注于这个不断扩大的领域,TECH开发了非常完整的半面授校级硕士。它是由计算机工程专家设计的教学计划,包括该领域最具创新性的方面,深入研究计算机科学家必须掌握的每个部分,以成功管理计算机项目的管理,自动学习,智能系统或高级算法设计。

所有这一切,通过 1,500 小时的 100% 在线理论培训,不仅包括该行业最详尽和最具活力的议程,还包括高质量和不同格式的其他材料,以便你可以深入研究你认为最重要和相关的每个部分。这还不是全部,因为在克服这段时间后,毕业生将有可能在一个著名的中心实际实习3周,在那里他可以积极参与当时正在开发的活动,以及工作和学习真正的人工智能专业人士在该领域拥有广泛而无边的职业生涯。

这个**人工智能和知识工程半面授校级硕士**包含市场上最完整和最新的课程。主要特点是:

- 由IT专业人员在项目管理,分析和多智能体系统设计方面提出的100多个案例的开发
- 它的图形,示意图和非常实用的内容,收集了有关专业实践必不可少的计算机学科的科学和辅助信息
- 基于遗传算法的人工智能及其多种应用的详尽知识
- 使用现有的主要软件和工具创建基于语言的智能本体系统
- 这将由理论讲座、向专家提问、关于争议性问题的讨论论坛和个人反思工作来补充
- 可以从任何有互联网连接的固定或便携式设备上获取内容
- 此外,你可以在最好的IT公司之一实习



在短短12个月内,你将掌握人工工程的理论和实践”

“

TECH每年与成千上万的学生合作,他们在完成这样的课程后,设法找到了引导他们取得职业成功的道路。你也想得到它吗?”

在这个硕士提案中,具有专业性质和半面授模式,该计划旨在更新计算机专业人员,他们在专门从事人工智能和知识工程的工程部门发展其职能,并且符合高水平资格的要求。内容基于该部门的最新证据,并以教学方式将理论知识整合到计算机实践中,理论实践要素将促进知识的更新,并允许在项目管理和指导方面做出决策。

由于他们采用最新教育技术制作的多媒体内容,他们将允许专业人员享受情境和情境学习,即模拟环境,该环境将提供身临其境的培训,以便在真实情况下进行训练。该计划的设计侧重于基于问题的学习,你必须尝试解决整个过程中出现的不同专业实践情况。为此,你将获得由公认专家制作的创新交互式视频系统的协助。

一个程序,通过它你将获得有关算法的最详尽的知识,掌握树,Heaps, Grafos和Greedy,尽管它们很复杂。

理论课程将100%在线进行,因此你可以组织自己并设置自己的时间表。



02

为什么要选择这个半面授校级硕士？

人工智能正在全面发展,因此,有必要拥有具有扎实理论知识的专业人员,毫无疑问,必须在实践中部署这些知识以创建智能系统。因此,在这个领域,了解最新的技术与用于其直接应用的编程语言一样重要。出于这个原因,TECH创造了这个开创性的学位,它将算法设计,多智能体系统和仿生计算等领域的最新更新与在该领域一家享有盛誉的公司的实际停留相结合。





“

TECH在学术全景中为你提供独特的半面授校级硕士,让你在最好的专家的帮助下在人工智能领域专业成长”

1. 与最新的可用技术保持同步

近年来,新技术无疑彻底改变了人工智能和知识工程领域,为他们提供了算法编程软件,推动了它们的发展。出于这个原因,TECH创建了这个半面授校级硕士,它使学生更接近该领域最前卫的技术。

2. 汲取最佳专家的专业知识

这个半面授校级硕士有一个优秀的专家教学人员团队,他们将在理论阶段陪伴所有学生,使他们能够成功实现目标。在实践阶段也保持了一个目标,即在此过程中,毕业生将接受人工智能和知识工程方面真正专家的辅导。

3. 进入一流的实习环境

为了实现为学生提供优质教学的格言,TECH对教学人员和进行实习的公司进行了细致的挑选过程。这保证了学生获得高水平的大学学位,在那里他们将被人工智能和知识工程领域最好的专家所包围。





4. 将最好的理论与最先进的实践相结合

这个半面授校级硕士旨在让学生更接近最新和相关的知识,摆脱长时间的学习,专注于他们职业发展的关键概念。因此,TECH通过该计划提供了一种新的学习模式,旨在向学生展示专家在创建人工智能和知识工程系统时使用的方法,技术和工具。

5. 拓展知识的疆域

TECH不仅提供了在国内且在国际进行这种实践培训的可能性。因此,学生将能够打开一系列选择,使他们能够在新兴领域专业成长,此外,还可以与该领域的真正专家并肩工作。

“

你将在你选择的中心有一个完全的实践沉浸”

03 目标

围绕人工智能和知识工程出现的未来可能性，以及缺乏详尽和密集的学位，不仅适应劳动力市场的需求，而且适应学生的需求，是促使TECH及其团队开发该计划的原因。因此，其目标是为毕业生提供掌握这门科学每个部分所需的所有信息。通过这种方式，通过对其来龙去脉的深入了解和积极实践，你将能够面对与该领域及其多种应用相关的任何项目，并完全保证成功。





“

如果你很清楚你想专攻人工智能, 这个半面授校级硕士将为你做好准备, 帮助你达到你最雄心勃勃和最苛刻的职业目标”



总体目标

- 该计划的总体目标是通过获得有关计算领域和计算机结构的广泛知识, 在科学和技术上培养毕业生进行计算机工程的实践。此外, 它的目的也是让专业人员获得软件专家的技能, 以及掌握该主题所必需的数学、统计和物理技能

“

让自己在一个中心的指导下
获得与成千上万的学生一样,
由于他们的学位专业, 找到了
他们所有问题的答案”





具体目标

模块 1. 编程基础知识

- ◆ 了解计算机的基本结构、软件和通用编程语言
- ◆ 学习设计和解释算法, 这是开发计算机程序的必要基础
- ◆ 理解计算机程序的基础要素, 如不同类型的数据、运算符、表达式、语句、I/O和控制语句
- ◆ 了解通用编程语言中不同的数据结构, 包括静态和动态的, 并掌握文件处理的基本知识
- ◆ 学习计算机程序中的不同测试技术, 以及与良好的源代码一起生成良好的文档的重要性
- ◆ 学习C++编程语言的基本概念, 这是世界上使用最广泛的语言之一

模块 2. 数据结构

- ◆ 学习C++语言编程的基础知识, 包括类、变量、条件表达式和对象
- ◆ 理解抽象数据类型、线性数据结构类型、简单和复杂的分层数据结构及其C++实现
- ◆ 了解除常规数据结构以外的高级数据结构的操作
- ◆ 学习优先级堆和队列有关的理论和实践
- ◆ 学习Hash表的操作, 如抽象数据类型和函数
- ◆ 了解图论, 以及有关图的高级算法和概念

模块 3. 算法和复杂性

- 学习算法设计的主要策略, 以及算法计算的不同方法和措施
- 学习软件开发中使用的主要排序算法
- 理解不同算法对树、Heaps和图的操作
- 了解Greedy算法的工作原理, 策略和在主要已知问题中的应用实例我们将学习在图上使用Greedy算法
- 我们将学习最小路径搜索的主要策略, 以及领域的基本问题和解决这些问题的算法
- 了解Backtracking技术及主要用途, 以及替代技术

模块 4. 高级算法设计

- 深化高级算法设计, 分析递归算法和分割与征服算法, 以及进行摊销分析
- 理解动态编程的概念和NP问题的算法
- 了解组合优化的工作方式, 以及不同的随机化算法和并行算法
- 认识并了解不同本地搜索方法的操作以及与候选物件
- 学习正规程序验证和迭代程序验证的机制, 包括一阶逻辑和Hoare的正规系统
- 了解一些主要的数值方法是如何工作的, 如二分法、牛顿拉斐尔法和正割法

模块 5. 计算逻辑

- 学习计算逻辑的基础知识, 它的用途和使用的理由
- 知道命题逻辑中形式化和演绎的不同策略, 包括自然推理、公理演绎和自然演绎, 以及命题微积分的原始规则
- 掌握命题逻辑的高级知识, 深入了解其语义和这种逻辑的主要应用, 如逻辑电路
- 既要了解谓词自然演绎的微积分, 又要了解谓词逻辑的形式化和演绎策略
- 理解自然语言的基础及其演绎机制
- 介绍使用PROLOG语言的逻辑编程

模块 6. 人工智能和知识工程

- 奠定人工智能和知识工程的基础, 简单了解人工智能的历史, 直到今天
- 了解人工智能中搜索的基础概念, 包括有信息和无信息的搜索
- 了解人工智能如何在游戏中发挥作用
- 学习神经网络的基这个概念和遗传算法的使用
- 获得适当的机制来代表知识, 特别是考虑到语义网
- 理解专家系统和决策支持系统的运作

模块 7. 智能系统

- ◆ 学习所有与代理理论和代理架构有关的概念及其推理过程
- ◆ 吸收信息和知识概念背后的理论和实践, 以及表现知识的方式
- ◆ 了解与本体相关的理论, 以及学习本体语言和创建本体的软件
- ◆ 学习不同的知识表示模型, 例如词汇表, 分类法, 同义词库和思维导图等
- ◆ 了解语义推理器、基于知识的系统和专家系统的工作原理
- ◆ 了解语义网的功能, 它的当前和未来状态, 以及基于语义网的应用程序

模块 8. 机器学习和数据挖掘

- ◆ 介绍知识发现的过程和机器学习的基本概念
- ◆ 学习数据探索和预处理的方法, 以及基于决策树的不同算法
- ◆ 了解贝叶斯方法的工作原理、回归和连续响应方法
- ◆ 通过学习使用混淆矩阵和数字评价、卡帕统计和ROC曲线, 了解不同的分类规则和对分类器的评价
- ◆ 获取一系列与文本挖掘和自然语言处理 (NLP) 和聚类相关的基本知识
- ◆ 加深神经网络的知识, 从简单的神经网络到循环神经网络

模块 9. 多智能体系统和计算感知

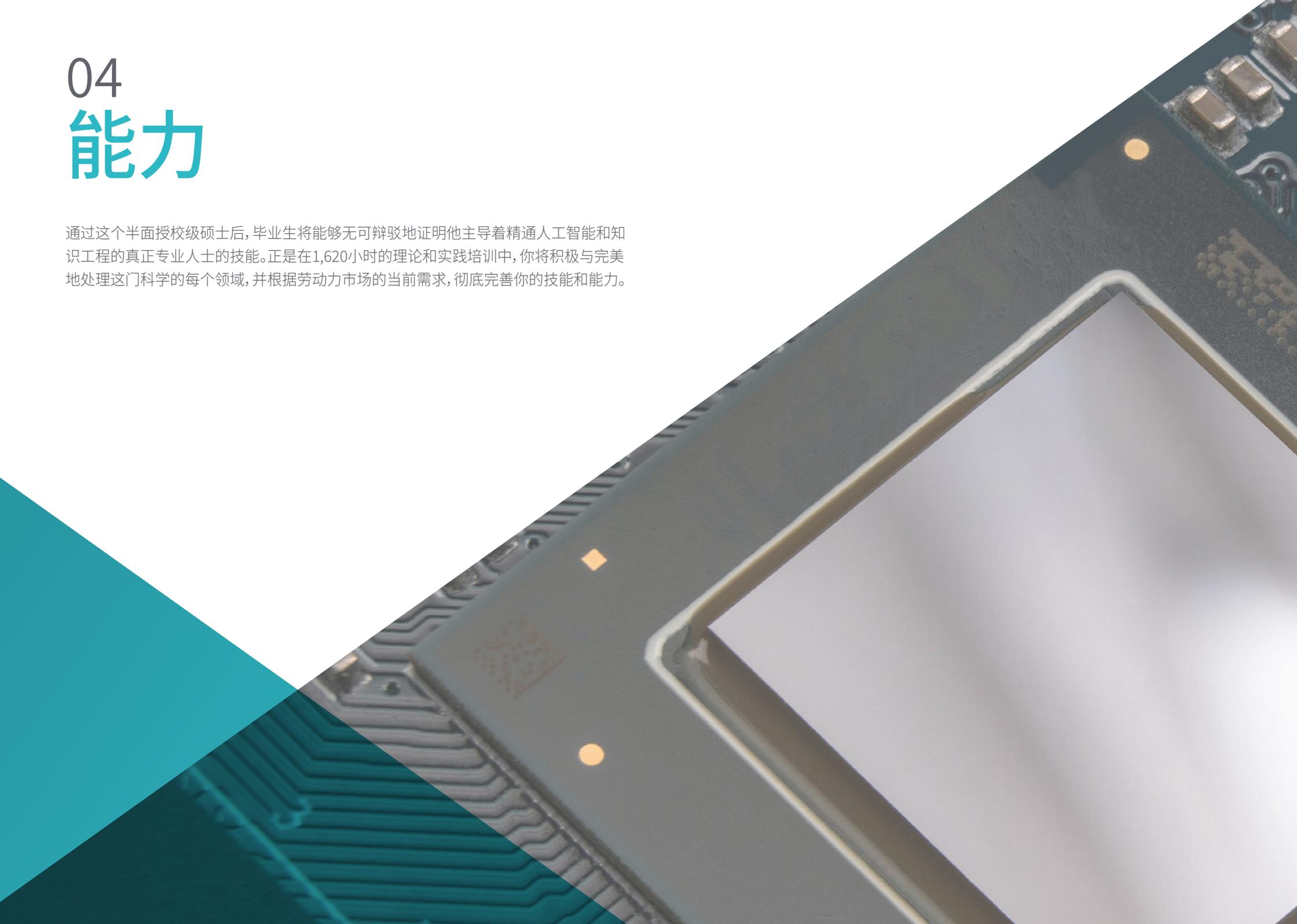
- ◆ 了解与代理和多代理系统相关的基本和高级概念
- ◆ 研究FIPA代理的标准, 同时考虑代理之间的通信, 其管理和体系结构等问题
- ◆ 深化JADE平台 (Java代理开发框架) 的学习, 学习在其中编程基本和高级概念, 包括通信主题和代理发现
- ◆ 为自然语言处理奠定基础, 例如自动语音识别和计算语言学
- ◆ 深入了解计算机视觉、数字图像分析、转换和分割的操作

模块 10. 仿生智能计算

- ◆ 介绍仿生智能计算的概念, 并了解不同类型的社会适应算法和遗传算法的功能
- ◆ 深入研究进化计算的不同模型, 了解其策略、编程、算法和基于分布估计的模型
- ◆ 了解遗传算法的空间探索-开发的主要策略
- ◆ 理解应用于学习问题和多目标问题的进化编程的工作原理
- ◆ 学习与神经网络有关的基础概念, 了解它们如何在应用于医学研究、经济学和计算机视觉等不同领域的真实使用案例中发挥作用

04 能力

通过这个半面授校级硕士后,毕业生将能够无可辩驳地证明他主导着精通人工智能和知识工程的真正专业人士的技能。正是在1,620小时的理论和实践培训中,你将积极与完美地处理这门科学的每个领域,并根据劳动力市场的当前需求,彻底完善你的技能和能力。



“

一个安全的赌注, 通过
详尽的知识和实际实践
来完善你的专业技能”



总体能力

- 获得计算机工程专业实践的 necessary 技能, 并了解所有必要的因素, 以质量和应对能力执行它
- 掌握与人工智能和知识工程相关的主要管理和项目创建工具

“

这个半面授校级硕士将带你提高你在创建人工智能的高级算法方面的技能”





具体能力

- ◆ 在考虑到人工智能发展的所有因素的情况下开发人工智能领域的编程
- ◆ 了解C++编程中的数据结构
- ◆ 设计基本和高级算法
- ◆ 了解计算逻辑并将其应用于项目设计
- ◆ 了解人工智能, 其用途和发展, 并实施自己的项目
- ◆ 了解它们是什么, 它们如何工作以及如何与智能系统合作
- ◆ 掌握机器学习的基本概念
- ◆ 了解JADE、FIPA、计算机视觉和其他多代理系统
- ◆ 了解生物智能计算算法及其使用策略

05 教学规划

为了始终提供最完整的学位, TECH开发了该半面授校级硕士课程, 同时考虑到精通计算机工程的专家团队的标准。通过这种方式, 可以根据该部门的即时新知形成一个课程大纲, 其中还包括数小时的高质量附加材料, 并以不同的格式呈现。这些是详细的视频, 研究文章, 动态摘要, 真实案例的模拟和补充阅读材料, 毕业生将能够在他认为与他的职业发展最相关或最有兴趣的每个方面进行深化。



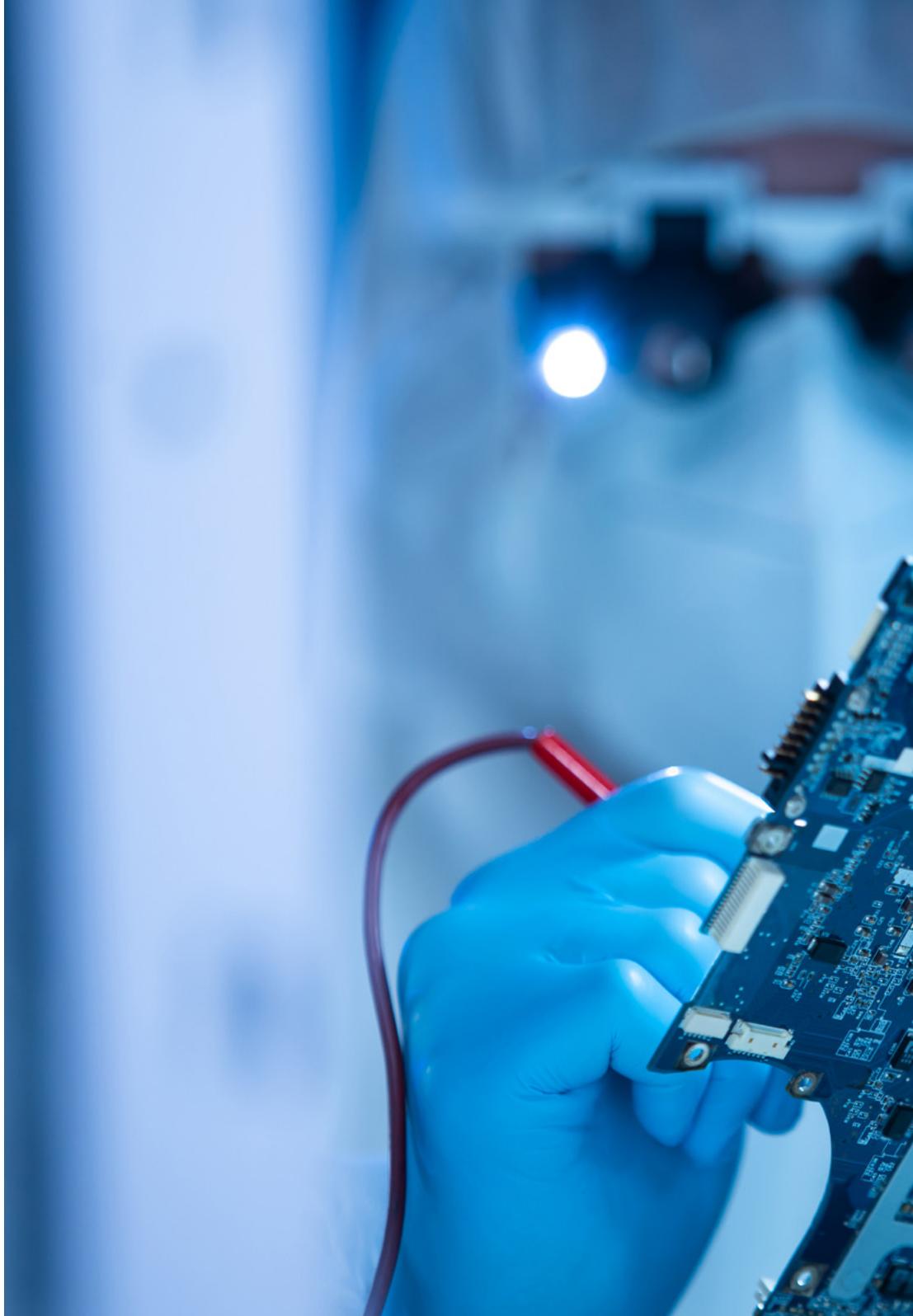


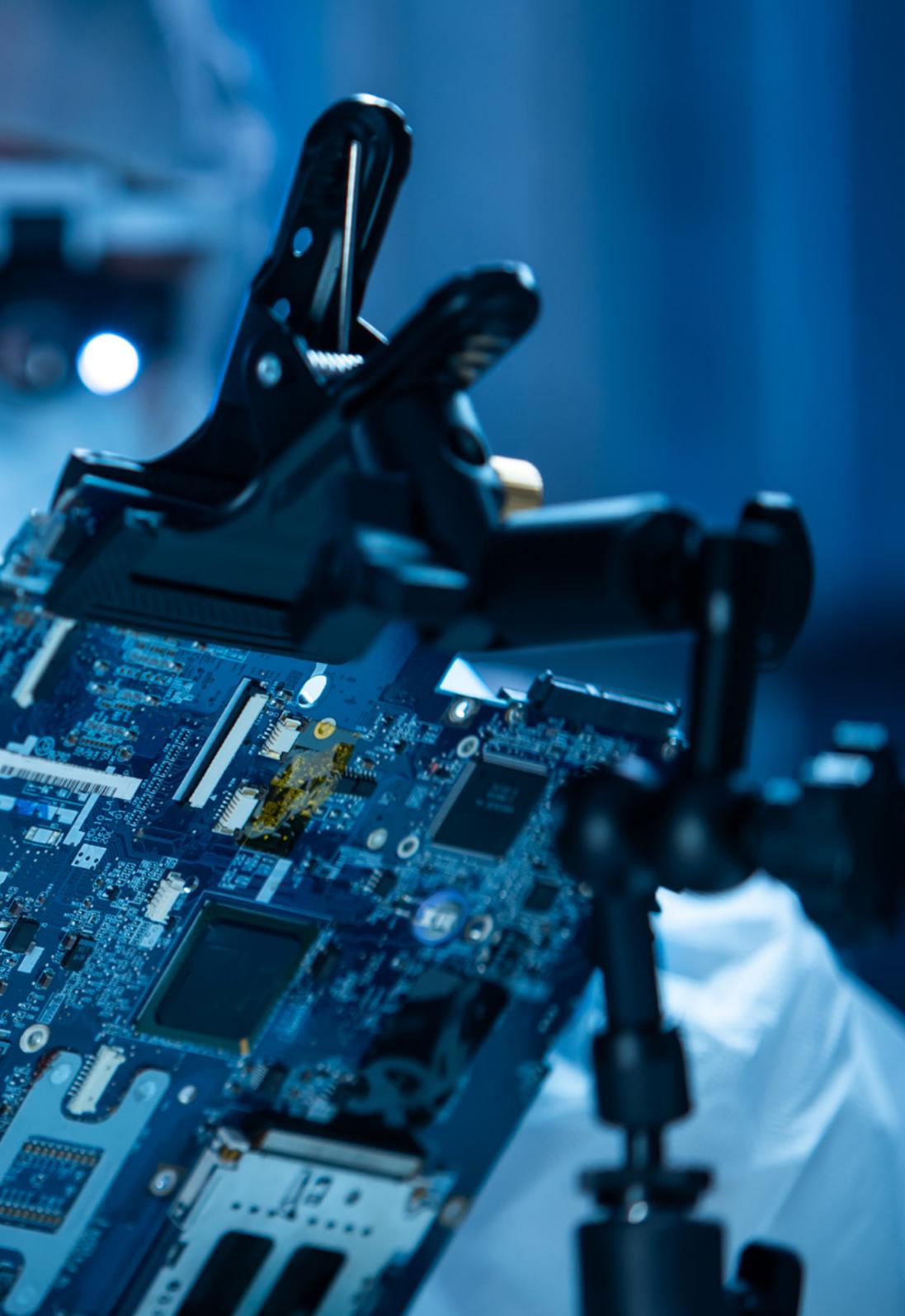
“

一个适应你、你的需求、你的兴趣和你的要求的计划。一个半面授校级硕士，毫无疑问，这将标志着你职业生涯的分水岭”

模块 1. 编程基础知识

- 1.1. 编程简介
 - 1.1.1. 计算机的基本结构
 - 1.1.2. 软件
 - 1.1.3. 编程语言
 - 1.1.4. 计算机应用程序的生命周期
- 1.2. 算法设计
 - 1.2.1. 问题的解决
 - 1.2.2. 描述性技术
 - 1.2.3. 算法的元素和结构
- 1.3. 程序的要素
 - 1.3.1. C++语言的起源和特点
 - 1.3.2. 开发环境
 - 1.3.3. 方案概念
 - 1.3.4. 基础数据类型
 - 1.3.5. 运算符
 - 1.3.6. 表达方式
 - 1.3.7. 句子
 - 1.3.8. 输入和输出数据
- 1.4. 控制语句
 - 1.4.1. 句子
 - 1.4.2. 分叉
 - 1.4.3. 循环
- 1.5. 抽象和模块化:函数
 - 1.5.1. 模块化设计
 - 1.5.2. 功能和效用的概念
 - 1.5.3. 函数的定义
 - 1.5.4. 函数调用的执行流程
 - 1.5.5. 原型化一个函数
 - 1.5.6. 结果返回
 - 1.5.7. 调用函数:参数
 - 1.5.8. 通过引用和值传递参数
 - 1.5.9. 标识符范围





- 1.6. 静态数据结构
 - 1.6.1. Arrays
 - 1.6.2. 矩阵。多面体
 - 1.6.3. 搜索和排序
 - 1.6.4. 锁链。字符串的 I/O 函数
 - 1.6.5. 结构。结合
 - 1.6.6. 新数据类型
- 1.7. 动态数据结构:指针
 - 1.7.1. 概念。指针定义
 - 1.7.2. 运算符和指针操作
 - 1.7.3. 指针Arrays
 - 1.7.4. 指针和arrays
 - 1.7.5. 指向字符串的指针
 - 1.7.6. 结构体指针
 - 1.7.7. 多重间接
 - 1.7.8. 指向函数的指针
 - 1.7.9. 将函数、结构和数组作为函数参数传递
- 1.8. 文件
 - 1.8.1. 基本概念
 - 1.8.2. 文件操作
 - 1.8.3. 文件类型
 - 1.8.4. 文件的组织
 - 1.8.5. C++ 文件简介
 - 1.8.6. 文件管理
- 1.9. 递归
 - 1.9.1. 递归的定义
 - 1.9.2. 递归类型
 - 1.9.3. 优点和缺点
 - 1.9.4. 考虑因素
 - 1.9.5. 递归-迭代转换
 - 1.9.6. 递归栈

- 1.10. 测试和文档
 - 1.10.1. 程序测试
 - 1.10.2. 白盒测试
 - 1.10.3. 黑盒测试
 - 1.10.4. 测试工具
 - 1.10.5. 程序文档

模块 2. 数据结构

- 2.1. 服务端编程简介:
 - 2.1.1. 类、构造函数、方法和属性
 - 2.1.2. 变量
 - 2.1.3. 条件表达式和循环
 - 2.1.4. 物品
- 2.2. 抽象数据类型 (ADT)
 - 2.2.1. 数据类型
 - 2.2.2. 基础结构和 TAD
 - 2.2.3. 向量和 arrays
- 2.3. 线性数据结构
 - 2.3.1. TAD 清单。定义
 - 2.3.2. 链表和双向链表
 - 2.3.3. 有序列表
 - 2.3.4. C++ 的列表
 - 2.3.5. TAD 堆栈
 - 2.3.6. TAD 队列
 - 2.3.7. C++ 中的堆栈和队列
- 2.4. 分层数据结构
 - 2.4.1. TAD 树
 - 2.4.2. 游览
 - 2.4.3. n叉树
 - 2.4.4. 二叉树
 - 2.4.5. 二叉搜索树
- 2.5. 分层数据结构:复杂树
 - 2.5.1. 完美平衡或最小高度的树
 - 2.5.2. 多路径树
 - 2.5.3. 参考书目
- 2.6. 堆和优先队列
 - 2.6.1. TAD 土墩
 - 2.6.2. TAD 优先队列
- 2.7. 哈希表
 - 2.7.1. TAD 哈希表
 - 2.7.2. 哈希函数
 - 2.7.3. 哈希表中的哈希函数
 - 2.7.4. 再分散
 - 2.7.5. 开放式哈希表
- 2.8. 图表
 - 2.8.1. ADT 图
 - 2.8.2. 图表类型
 - 2.8.3. 图形表示和基本操作
 - 2.8.4. 图形设计
- 2.9. 图上的算法和高级概念
 - 2.9.1. 图表问题
 - 2.9.2. 路径算法
 - 2.9.3. 搜索或遍历算法
 - 2.9.4. 其他算法
- 2.10. 其他数据结构
 - 2.10.1. 套组
 - 2.10.2. 平行 Arrays
 - 2.10.3. 符号表
 - 2.10.4. 尝试

模块 3. 算法和复杂性

- 3.1. 算法设计策略简介
 - 3.1.1. 递归
 - 3.1.2. 分而治之
 - 3.1.3. 其他策略
- 3.2. 算法的效率与分析
 - 3.2.1. 效率措施
 - 3.2.2. 测量输入的大小
 - 3.2.3. 测量执行时间
 - 3.2.4. 最坏情况、最好情况和中间情况
 - 3.2.5. 渐近符号
 - 3.2.6. 非递归算法的数学分析准则
 - 3.2.7. 递归算法的数学分析
 - 3.2.8. 算法的实证分析
- 3.3. 排序算法
 - 3.3.1. 协调概念
 - 3.3.2. 冒泡排序
 - 3.3.3. 选择排序
 - 3.3.4. 插入排序
 - 3.3.5. 合并排序(merge_sort)
 - 3.3.6. 快速排序(Quicksort)
- 3.4. 带树的算法
 - 3.4.1. 树的概念
 - 3.4.2. 二叉树
 - 3.4.3. 树游览
 - 3.4.4. 表示表达
 - 3.4.5. 有序二叉树
 - 3.4.6. 平衡二叉树
- 3.5. heaps算法
 - 3.5.1. Heaps
 - 3.5.2. heapsort序算法
 - 3.5.3. 优先队列

- 3.6. 图形算法
 - 3.6.1. 代表
 - 3.6.2. 行程宽度
 - 3.6.3. 深度游览
 - 3.6.4. 拓扑排序
- 3.7. Greedy算法
 - 3.7.1. Greedy策略
 - 3.7.2. Greedy策略元素
 - 3.7.3. 货币兑换
 - 3.7.4. 旅人的问题
 - 3.7.5. 背包问题
- 3.8. 搜索最小路径
 - 3.8.1. 最短路径的问题
 - 3.8.2. 负弧和循环
 - 3.8.3. 迪克斯特拉的算法
- 3.9. 图上的Greedy算法
 - 3.9.1. 最小生成树
 - 3.9.2. Prim 算法
 - 3.9.3. Kruskal 算法
 - 3.9.4. 复杂性分析
- 3.10. 溯源
 - 3.10.1. 回溯
 - 3.10.2. 替代技术

模块 4. 高级算法设计

- 4.1. 递归算法与类型分治分析
 - 4.1.1. 齐次和非齐次递归方程的逼近和求解
 - 4.1.2. 分而治之战略概述
- 4.2. 摊销分析
 - 4.2.1. 补充分析
 - 4.2.2. 会计方法
 - 4.2.3. 潜在方法

- 4.3. NP问题的动态规划和算法
 - 4.3.1. 动态规划的特点
 - 4.3.2. 返回;回溯
 - 4.3.3. 分枝和修剪
- 4.4. 组合优化
 - 4.4.1. 问题的表示
 - 4.4.2. 一维优化
- 4.5. 随机化算法
 - 4.5.1. 随机化算法示例
 - 4.5.2. 布冯定理
 - 4.5.3. 蒙特卡罗算法
 - 4.5.4. 维加斯算法
- 4.6. 本地搜索和候选
 - 4.6.1. Garcient Ascent
 - 4.6.2. Hill Climbing
 - 4.6.3. Simulated Annealing
 - 4.6.4. Tabu Search
 - 4.6.5. 搜索候选
- 4.7. 程序的正式验证
 - 4.7.1. 功能抽象规范
 - 4.7.2. 一阶逻辑的语言
 - 4.7.3. 霍尔的正式系统
- 4.8. 迭代程序的验证
 - 4.8.1. 霍尔形式系统规则
 - 4.8.2. 迭代不变的概念
- 4.9. 数值方法
 - 4.9.1. 二分法
 - 4.9.2. NewtonRaphson方法
 - 4.9.3. 割线法
- 4.10. 并行算法
 - 4.10.1. 并行二元运算
 - 4.10.2. 带图的并行操作
 - 4.10.3. 分而治之的并行
 - 4.10.4. 动态规划中的并行性

模块 5. 计算逻辑

- 5.1. 逻辑的合理性
 - 5.1.1. 研究逻辑的目的
 - 5.1.2. 逻辑是为何作用?
 - 5.1.3. 推理的组成部分和类型
 - 5.1.4. 逻辑计算的组成部分
 - 5.1.5. 语义学
 - 5.1.6. 逻辑存在的理由
 - 5.1.7. 如何检查一个逻辑是否充分?
- 5.2. 语句的自然推导计算
 - 5.2.1. 正式的语言
 - 5.2.2. 归纳机制
- 5.3. 命题逻辑的形式化和演绎策略
 - 5.3.1. 正规化战略
 - 5.3.2. 自然推理
 - 5.3.3. 法律和规则
 - 5.3.4. 公理演绎法和自然演绎法
 - 5.3.5. 自然演绎的微积分
 - 5.3.6. 命题微积分的原始规则
- 5.4. 命题逻辑的语义学
 - 5.4.1. 真值表
 - 5.4.2. 等效性
 - 5.4.3. 废话和矛盾
 - 5.4.4. 命题句子的验证
 - 5.4.5. 通过真值表的方式进行验证
 - 5.4.6. 使用语义树进行验证
 - 5.4.7. 通过反驳进行验证
- 5.5. 命题逻辑的应用:逻辑电路
 - 5.5.1. 基本闸门
 - 5.5.2. 迴路
 - 5.5.3. 迴路的数学模型
 - 5.5.4. 最小化
 - 5.5.5. 第二种典型形式和最小形式的和的乘积
 - 5.5.6. 其他闸门

- 5.6. 自然谓词演绎微积分
 - 5.6.1. 正式的语言
 - 5.6.2. 归纳机制
 - 5.7. 谓词逻辑的形式化策略
 - 5.7.1. 谓词逻辑的形式化介绍
 - 5.7.2. 带有量词的形式化策略
 - 5.8. 谓词逻辑的演绎策略
 - 5.8.1. 遗漏的原因
 - 5.8.2. 新规则的提出
 - 5.8.3. 谓词逻辑是一种自然演绎微积分
 - 5.9. 谓词逻辑的应用:逻辑编程介绍
 - 5.9.1. 非正式介绍
 - 5.9.2. Prolog的要素
 - 5.9.3. 重新评估和停产
 - 5.10. 集合论、谓词逻辑及其语义学
 - 5.10.1. 归纳集理论
 - 5.10.2. 谓词语义学简介
- 模块 6. 人工智能和知识工程**
- 6.1. 人工智能和知识工程简介
 - 6.1.1. 人工智能的简史
 - 6.1.2. 今天的人工智能
 - 6.1.3. 知识工程
 - 6.2. 搜索
 - 6.2.1. 常见的搜索概念
 - 6.2.2. 不知情的搜索
 - 6.2.3. 知情的搜索
 - 6.3. 布尔可满足性、约束可满足性和自动规划
 - 6.3.1. 布尔可满足性
 - 6.3.2. 约束可满足性问题
 - 6.3.3. 自动规划和PDDL
 - 6.3.4. 作为启发式搜索的规划
 - 6.3.5. 与SAT一起规划
 - 6.4. 游戏中的人工智能
 - 6.4.1. 博弈论
 - 6.4.2. 最小值和Alpha-Beta修剪
 - 6.4.3. 模拟场景:蒙特卡洛
 - 6.5. 有监督和无监督的学习
 - 6.5.1. 自动学习简介
 - 6.5.2. 分类
 - 6.5.3. 回归
 - 6.5.4. 结果验证
 - 6.5.5. 聚类(Clustering)
 - 6.6. 神经网络
 - 6.6.1. 生物学基础
 - 6.6.2. 计算模型
 - 6.6.3. 有监督和无监督的神经网络
 - 6.6.4. 简单的感知器
 - 6.6.5. 多层感知器
 - 6.7. 遗传算法
 - 6.7.1. 历史
 - 6.7.2. 生物学基础
 - 6.7.3. 问题编码
 - 6.7.4. 最初的人口生成
 - 6.7.5. 主要算法和遗传算子
 - 6.7.6. 对个人的评价:健身
 - 6.8. 术语表、词汇表、分类法
 - 6.8.1. 词汇
 - 6.8.2. 分类法
 - 6.8.3. 叙词表
 - 6.8.4. 本体论
 - 6.9. 知识表示:语义网
 - 6.9.1. 语义网
 - 6.9.2. 规格RDF、RDFS和OWL
 - 6.9.3. 推论/推理
 - 6.9.4. 关联数据

- 6.10. 专家系统和DSS
 - 6.10.1. 专家系统
 - 6.10.2. 摄影的支持系统

模块 7. 智能系统

- 7.1. 智能体理论
 - 7.1.1. 概念的历史
 - 7.1.2. 智能体定义
 - 7.1.3. 人工智能中的智能体
 - 7.1.4. 软件工程中的智能体
- 7.2. 智能体架构
 - 7.2.1. 智能体推理过程
 - 7.2.2. 反应性智能体
 - 7.2.3. 推论的智能体
 - 7.2.4. 混合性智能体
 - 7.2.5. 比较
- 7.3. 信息和知识
 - 7.3.1. 数据、信息和知识之间的区别
 - 7.3.2. 数据质量评估
 - 7.3.3. 数据采集方法
 - 7.3.4. 信息获取方式
 - 7.3.5. 知识获取方式
- 7.4. 知识表述
 - 7.4.1. 知识表示的重要性
 - 7.4.2. 通过其角色定义知识表示
 - 7.4.3. 知识表示的特征
- 7.5. 本体论
 - 7.5.1. 元数据介绍
 - 7.5.2. 本体论的哲学概念
 - 7.5.3. 本体论的计算概念
 - 7.5.4. 领域本体和更高层次的本体
 - 7.5.5. 如何建立本体论

- 7.6. 本体语言和本体构建软件
 - 7.6.1. RDF、Turtle 和 N3 三元组
 - 7.6.2. RDF模式
 - 7.6.3. OWL
 - 7.6.4. SPARQL
 - 7.6.5. 介绍用于创建本体的不同工具
 - 7.6.6. Protégé安装和使用
- 7.7. 语义网
 - 7.7.1. 语义网的现状和未来
 - 7.7.2. 语义网应用
- 7.8. 其他知识表示模式
 - 7.8.1. 词汇
 - 7.8.2. 全球视野
 - 7.8.3. 分类法
 - 7.8.4. 叙词表
 - 7.8.5. 大众分类法
 - 7.8.6. 比较
 - 7.8.7. 思维导图
- 7.9. 知识表征的评估和整合
 - 7.9.1. 零阶逻辑
 - 7.9.2. 一阶逻辑
 - 7.9.3. 描述性逻辑
 - 7.9.4. 不同类型逻辑之间的关系
 - 7.9.5. Prolog: 基于一阶逻辑的编程
- 7.10. 语义推理器、基于知识的系统和专家系统
 - 7.10.1. 推理概念
 - 7.10.2. 推理机的应用
 - 7.10.3. 基于知识的系统
 - 7.10.4. MYCIN, 专家系统的历史
 - 7.10.5. 专家系统的元素和架构
 - 7.10.6. 专家系统的创建

模块 8. 机器学习和数据挖掘

- 8.1. 介绍知识发现过程和机器学习的基本概念
 - 8.1.1. 知识发现过程的关键概念
 - 8.1.2. 知识发现过程的历史视角
 - 8.1.3. 知识发现过程的各个阶段
 - 8.1.4. 知识发现过程中使用的技术
 - 8.1.5. 好的机器学习模型的特点
 - 8.1.6. 机器学习信息的类型
 - 8.1.7. 基这个的学习概念
 - 8.1.8. 无监督学习的基这个概念
- 8.2. 数据探索和预处理
 - 8.2.1. 数据处理
 - 8.2.2. 数据分析流程中的数据处理
 - 8.2.3. 数据类型
 - 8.2.4. 数据转换
 - 8.2.5. 连续变量的可视化和探索
 - 8.2.6. 分类变量的显示和探索
 - 8.2.7. 相关性措施
 - 8.2.8. 最常见的图形表示法
 - 8.2.9. 多变量分析和降维介绍
- 8.3. 决策树
 - 8.3.1. ID3算法
 - 8.3.2. C4.5算法
 - 8.3.3. 过度训练和修剪
 - 8.3.4. 结果分析
- 8.4. 对分类器的评估
 - 8.4.1. 混淆矩阵
 - 8.4.2. 数值评价矩阵
 - 8.4.3. Kappa统计学
 - 8.4.4. ROC曲线
- 8.5. 分类规则
 - 8.5.1. 规则评价措施
 - 8.5.2. 图形表示法简介
 - 8.5.3. 顺序叠加算法
- 8.6. 神经网络
 - 8.6.1. 基本概念
 - 8.6.2. 简单的神经网络
 - 8.6.3. 反向传播算法
 - 8.6.4. 递归神经网络简介
- 8.7. 贝叶斯方法
 - 8.7.1. 概率的基本概念
 - 8.7.2. 贝叶斯定理
 - 8.7.3. 奈何贝叶斯
 - 8.7.4. 贝叶斯网络简介
- 8.8. 回归和连续反应模型
 - 8.8.1. 简单线性回归
 - 8.8.2. 多重线性回归
 - 8.8.3. 逻辑回归
 - 8.8.4. 回归树
 - 8.8.5. 支持向量机(SVM)简介
 - 8.8.6. 拟合度测量
- 8.9. 聚类
 - 8.9.1. 基本概念
 - 8.9.2. 分层聚类
 - 8.9.3. 概率论的方法
 - 8.9.4. EM算法
 - 8.9.5. B-立方体法
 - 8.9.6. 隐式方法
- 8.10. 文本挖掘和自然语言处理(NLP)
 - 8.10.1. 基本概念
 - 8.10.2. 语料库的创建
 - 8.10.3. 描述性分析
 - 8.10.4. 情感分析简介

模块 9. 多智能体系统和计算感知

- 9.1. 智能体和多智能体系统
 - 9.1.1. 智能体的概念
 - 9.1.2. 架构
 - 9.1.3. 沟通和协调
 - 9.1.4. 编程语言和工具
 - 9.1.5. 智能体应用
 - 9.1.6. Foundation for Intelligent Physical Agents (FIPA)
- 9.2. 智能体的标准:FIPA
 - 9.2.1. 智能体之间的沟通
 - 9.2.2. 智能体管理
 - 9.2.3. 抽象建筑
 - 9.2.4. 其他规格
- 9.3. JADE平台
 - 9.3.1. 根据JADE的软件代理
 - 9.3.2. 架构
 - 9.3.3. 安装和执行
 - 9.3.4. JADE软件包
- 9.4. 用JADE进行基本编程
 - 9.4.1. 管理控制台
 - 9.4.2. 基本智能体的创建
- 9.5. 用JADE进行高级编程
 - 9.5.1. 高级代理创建
 - 9.5.2. 智能体之间的沟通
 - 9.5.3. 智能体的发现
- 9.6. 人工视觉
 - 9.6.1. 数字图像处理和分析
 - 9.6.2. 图像分析和机器视觉
 - 9.6.3. 图像处理和人类视觉
 - 9.6.4. 图像采集系统
 - 9.6.5. 图像的形成和感知

- 9.7. 数字图像分析
 - 9.7.1. 图像分析过程的各个阶段
 - 9.7.2. 预处理
 - 9.7.3. 基本操作
 - 9.7.4. 空间过滤
- 9.8. 数字图像转换和图像分割
 - 9.8.1. Fourier变换
 - 9.8.2. 频率滤波
 - 9.8.3. 基本概念
 - 9.8.4. 阈值处理
 - 9.8.5. 轮廓检测
- 9.9. 阈值处理
 - 9.9.1. 特征提取
 - 9.9.2. 分类算法
- 9.10. 自然语言处理
 - 9.10.1. 自动语音识别
 - 9.10.2. 计算语言学

模块 10. 仿生智能计算

- 10.1. 仿生智能计算简介
 - 10.1.1. 仿生智能计算简介
- 10.2. 社会适应算法
 - 10.2.1. 基于蚁群的仿生智能计算
 - 10.2.2. 蚁群算法的变体
 - 10.2.3. 粒子云计算
- 10.3. 遗传算法
 - 10.3.1. 一般结构
 - 10.3.2. 主要运算符的应用
- 10.4. 遗传算法的空间探索-开发策略
 - 10.4.1. CHC算法
 - 10.4.2. 多模式问题

- 10.5. 进化计算模型(一)
 - 10.5.1. 进化策略
 - 10.5.2. 进化编程
 - 10.5.3. 基于差分进化的算法
- 10.6. 进化计算模型(二)
 - 10.6.1. 基于分布估计 (EDA) 的演化模型
 - 10.6.2. 遗传编程
- 10.7. 进化规划应用于学习问题
 - 10.7.1. 基于规则的学习
 - 10.7.2. 实例选择问题中的进化方法
- 10.8. 多目标问题
 - 10.8.1. 支配的概念
 - 10.8.2. 进化算法在多目标问题中的应用
- 10.9. 神经网络(一)
 - 10.9.1. 神经网络简介
 - 10.9.2. 神经网络的实际例子
- 10.10. 神经网络(二)
 - 10.10.1. 神经网络在医学研究中的用例
 - 10.10.2. 神经网络在经济学中的使用案例
 - 10.10.3. 神经网络在计算机视觉中的使用案例

“

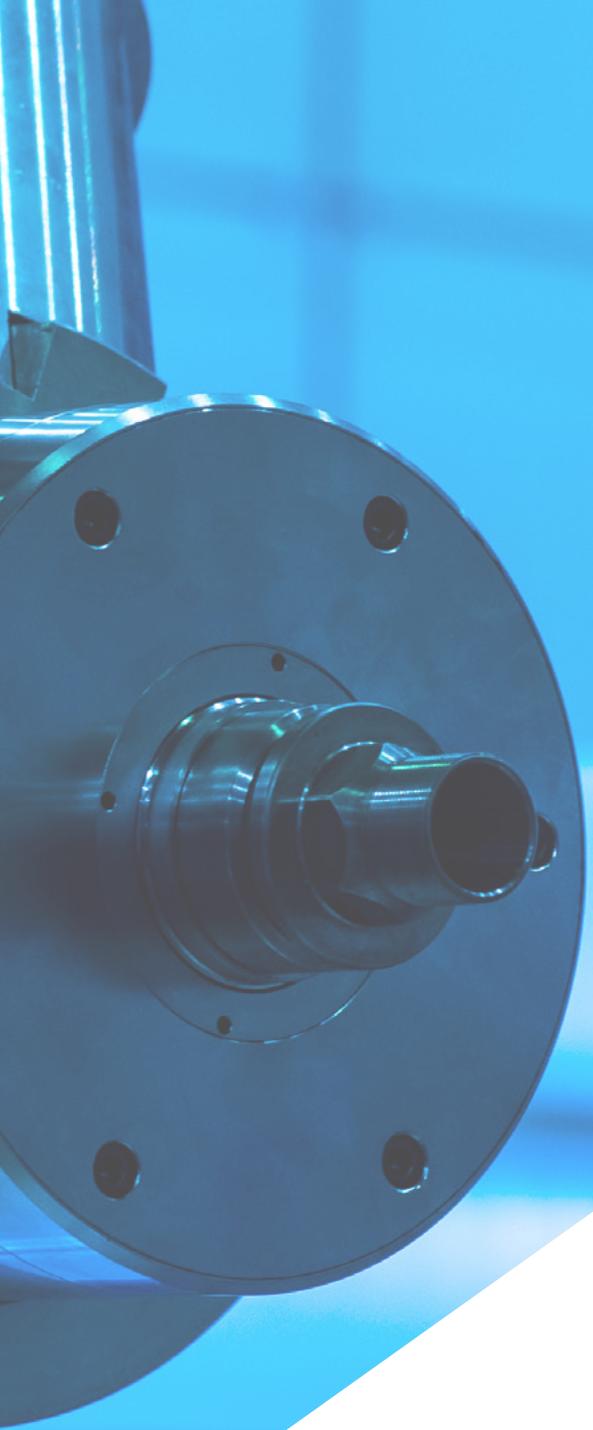
你将通过自动调度和 PDDL (如启发式搜索或 SAT) 深入研究布尔值满足性和约束”

06

临床实习

虽然理论课程对于毕业生知识的发展将是一个充满活力且非常有益的经历,但毫无疑问,这个面授校级硕士的优势在于,在计算机科学部门的参考中心进行为期3周的实践期。记忆所学知识的最佳方法是通过在大公司的自主工作,此外,学生将与专家合作,以及从他们的策略中学习,并获得可以包含在他们的履历中的非凡的工作经验。





“

这是120小时的实践培训, 你将能够掌控当前人工工程环境中最先进的计算机工具”

在TECH及其专家团队决定执行该计划的那一刻,他们这样做是为了为毕业生提供另一个机会,让他继续专业成长,同时扩大他未来的可能性。为此,这一次他们决定在一个享有盛誉的中心创造一种非常实用的体验,该中心开发了3周,也就是120小时内分配,其中学生必须在连续8小时从周一到周五去公司。

在本培训提案中,这些活动旨在发展和提高必要的技能,以提供与人工智能和知识工程相关的活动,并且面向从事该行业的特定禀赋,具有很高的工作表现。

因此,这是一个独特而无与伦比的机会,可以与专家团队一起努力提高你的技能和才能,他们将确保你获得最广泛和最专业的知识。因此,你将能够塑造你的专业形象,并根据你的实践实施最具创新性、最复杂和最有效的编程和计算策略,使其质量适应市场的当前需求。

实践教学将在学生的积极参与下进行,执行每个能力领域的活动和程序(学习学习和学习做),在教师和其他培训合作伙伴的陪伴和指导下,促进团队合作和 multidisciplinary 整合作为高级计算机实践的横向能力(学习成为和学习联系)。

下述程序将成为培训实践部分的基础,其实现将取决于中心的可用性及其工作量,拟议的活动如下:



你想使用机器学习和数据挖掘系统吗?有了这个教学计划,你将学会完美地处理主要的探索和预处理软件"



模块	实践活动
编程和数据结构化	不同类型的设计算法
	处理编程语言的动态和静态数据结构
	在计算机程序中使用测试技术
	在C++中实现不同类型的数据结构
	处理更高级的数据结构
算法设计	利用哈希表
	在常见编程问题中使用Greedy算法
	算法管理中的回溯和其他替代技术
	创建特定算法以涵盖所执行项目的特定问题
	设计高级算法, 利用有效的分析来完成此类任务
	执行正式的程序检查
人工智能和知识工程	使用组合技术优化算法
	在不同情况下利用人工智能
	在创建人工智能时使用遗传算法
	根据所需上下文对人工智能进行编程
	在智能系统中使用特定语言和软件创建本体
	在智能系统中设计代理体系结构
创建机器学习、数据挖掘和多智能体系统	管理基于专家知识的系统和语义网
	使用简单和循环神经网络进行开发
	使用基于决策树的算法处理数据预处理
	使用数值评估和混淆矩阵对分类器进行分类和评估
	根据多智能体系统自己的架构对其进行管理
	使用JADE编程和开发多智能体系统

责任保险

这个教育机构的主要关注点是保证受训者和公司实际培训过程中所需要的其他合作人员的安全。在致力于实现这一目标的措施中,包括对整个教学过程中可能发生的任何事件作出反应。

为此,“TECH将购买一份民事责任保险,以覆盖在实习中心逗留期间可能出现的任何意外情况”。

这份受训人员的责任保险将有广泛的覆盖面,并将在实习期开始前投保。这样一来,专业人员就不必担心必须处理突发情况,而且在中心的实践课程结束前都会得到保障。



实践培训的一般条件

该计划的实习协议的一般条件将如下。

1. 辅导: 在半面授校级硕士期间, 学生将被分配到两名辅导员, 他们将全程陪伴学生, 解决可能出现的任何疑惑和问题。一方面, 将有一位属于工作安置中心的专业导师, 他将随时指导和支持学生。另一方面, 也会有一名学术导师, 其任务是在整个过程中协调和帮助学生, 解决他们的疑惑, 并为他们可能需要的东西提供便利。通过这种方式, 专业人员将一直陪同, 并能够咨询任何可能出现的疑问, 包括实践和学术方面的疑问。

2. 时间: 实习计划将有连续三周的实践培训时间, 分布在每周五天, 每天8小时。出勤的日子和时间表将由中心负责, 并适当提前通知专业人员, 提前足够的时间以方便其组织。

3. 不出席: 如果在半面授校级硕士程开始的当天没有出现, 学生将失去同样的权利, 没有报销或更改日期的可能性。在没有正当/医疗理由的情况下缺席超过两天, 将导致学生辞去实习, 因此, 自动终止实习。在实习过程中可能出现的任何问题都必须及时和紧急地报告给学术导师。

4. 证书: 通过半面授校级硕士的学生将收到一份证书, 认可他们在有关中心的逗留。

5. 雇佣关系: 半面授校级硕士不构成任何形式的雇佣关系。

6. 以前的学习经历: 一些中心可能要求提供以前的学习证明, 以便参加半面授校级硕士。在这些情况下, 有必要向TECH实习部出示该证明, 以确认所选中心的分配。

7. 不包括: 半面授校级硕士不包括本条件中未描述的任何内容。因此, 它不包括住宿、前往实习城市的交通、签证或任何其他未描述的服务。

然而, 学生可以向他们的学术导师咨询这方面的任何疑问或建议。他/她将提供所有必要的信息以方便办理手续。

07

我在哪里可以进行实习?

每年,TECH都会在国际舞台上选择数百家公司,以便他们的毕业生参加一个实践期,保证他们不仅是一系列无数的活动,而且是帮助他们充分发展为专业人士的经验。这就是为什么像这样的计划是成为大型公司的一部分的最佳机会,你可以在其中使用最前卫的技术和最有效的策略,能够实施你的实践和课程的真正专家的技能。





“

你将从这次实习中获得的经验将成为选拔过程中的独特人才, 这要归功于你将实习的公司的声望”

tech 42 | 我在哪里可以做实习?

学生可以在以下中心参加这个半面授校级硕士的实习:



计算机从业人员

Grupo Fórmula

国家 城市
墨西哥 墨西哥城

地址: Cda. San Isidro 44, Reforma Soc,
Miguel Hidalgo, 11650 Ciudad de México, CDMX

多媒体通信领域的领先公司
和内容生成

相关的实践培训:
-平面设计师
-人员管理



“

现在就报名, 通过一个全面的课程在你的工作领域取得进步, 让你把你所学到的一切付诸实践”

08 方法

这个培训计划提供了一种不同的学习方式。我们的方法是通过循环的学习模式发展起来的: **Re-learning**。

这个教学系统被世界上一些最著名的医学院所采用,并被**新英格兰医学杂志**等权威出版物认为是最有效的教学系统之一。





发现 Re-learning, 这个系统放弃了传统的线性学习, 带你体验循环教学系统: 这种学习方式已经证明了其巨大的有效性, 尤其是在需要记忆的科目中”

案例研究, 了解所有内容的背景

我们的方案提供了一种革命性的技能和知识发展方法。我们的目标是在一个不断变化, 竞争激烈和高要求的环境中加强能力建设。

“

和TECH, 你可以体验到一种正在动摇世界各地传统大学基础的学习方式”



你将进入一个以重复为基础的学习系统, 在整个教学大纲中采用自然和渐进式教学。

一种创新并不同的学习方法

该技术课程是一个密集的教学计划，从零开始，提出了该领域在国内和国际上最苛刻的挑战和决定。由于这种方法，个人和职业成长得到了促进，向成功迈出了决定性的一步。案例法是构成这一内容的技术基础，确保遵循当前经济、社会和职业现实。

“我们的课程使你准备好在不确定的环境中面对新的挑战，并取得事业上的成功”

在世界顶级计算机科学学校存在的时间里，案例法一直是最广泛使用的学习系统。1912年开发的案例法是为了让法律学生不仅在理论内容的基础上学习法律，案例法向他们展示真实的复杂情况，让他们就如何解决这些问题作出明智的决定和价值判断。1924年，它被确立为哈佛大学的一种标准教学方法。

在特定情况下，专业人士应该怎么做？这就是我们在案例法中面对的问题，这是一种以行动为导向的学习方法。在整个课程中，学生将面对多个真实的案例。他们必须整合所有的知识，研究、论证和捍卫他们的想法和决定。

学生将通过合作活动和真实案例，学习如何解决真实商业环境中的复杂情况。



Re-learning 方法

TECH有效地将案例研究方法与基于循环的100%在线学习系统相结合,在每节课中结合了个不同的教学元素。

我们用最好的100%在线教学方法加强案例研究: Re-learning。

在2019年,我们取得了世界上所有西班牙语在线大学中最好的学习成绩。

在TECH,你将用一种旨在培训未来管理人员的尖端方法进行学习。这种处于世界教育学前沿的方法被称为 Re-learning。

我校是唯一获准使用这一成功方法的西班牙语大学。2019年,我们成功地提高了学生的整体满意度(教学质量,材料质量,课程结构,目标.....),与西班牙语最佳在线大学的指标相匹配。



在我们的方案中,学习不是一个线性的过程,而是以螺旋式的方式发生(学习,解除学习,忘记和重新学习)。因此,我们将这些元素中的每一个都结合起来。这种方法已经培养了超过65万名大学毕业生,在生物化学,遗传学,外科,国际法,管理技能,体育科学,哲学,法律,工程,新闻,历史,金融市场和工具等不同领域取得了前所未有的成功。所有这些都是在一个高要求的环境中进行的,大学学生的社会经济状况很好,平均年龄为43.5岁。

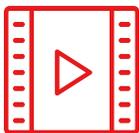
Re-learning 将使你的学习事半功倍,表现更出色,使你更多地参与到训练中,培养批判精神,捍卫论点和对比意见:直接等同于成功。

从神经科学领域的最新科学证据来看,我们不仅知道如何组织信息,想法,图像和记忆,而且知道我们学到东西的地方和背景,这是我们记住并将其储存在海马体的根本原因,并能将其保留在长期记忆中。

通过这种方式,在所谓的神经认知背景依赖的电子学习中,我们课程的不同元素与学员发展其专业实践的背景相联系。



该方案提供了最好的教育材料,为专业人士做了充分准备:



学习材料

所有的教学内容都是由教授该课程的专家专门为该课程创作的,因此,教学的发展是具体的。

然后,这些内容被应用于视听格式,创造了TECH在线工作方法。所有这些,都是用最新的技术,提供最高质量的材料,供学生使用。



大师课程

有科学证据表明第三方专家观察的有用性。

向专家学习可以加强知识和记忆,并为未来的困难决策建立信心。



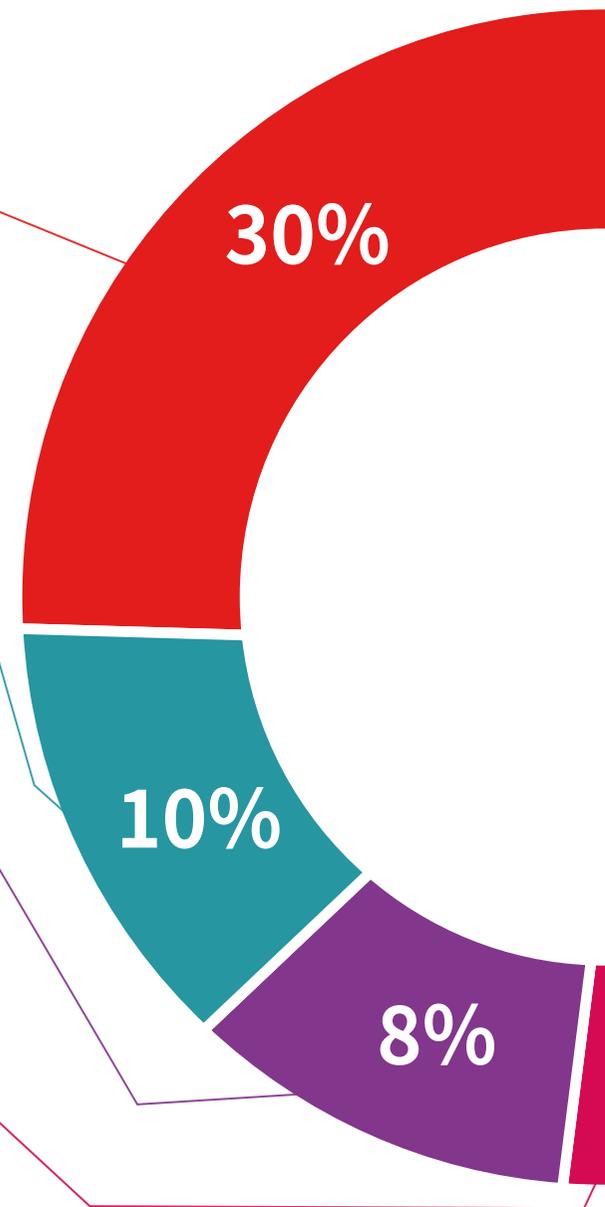
技能和能力的实践

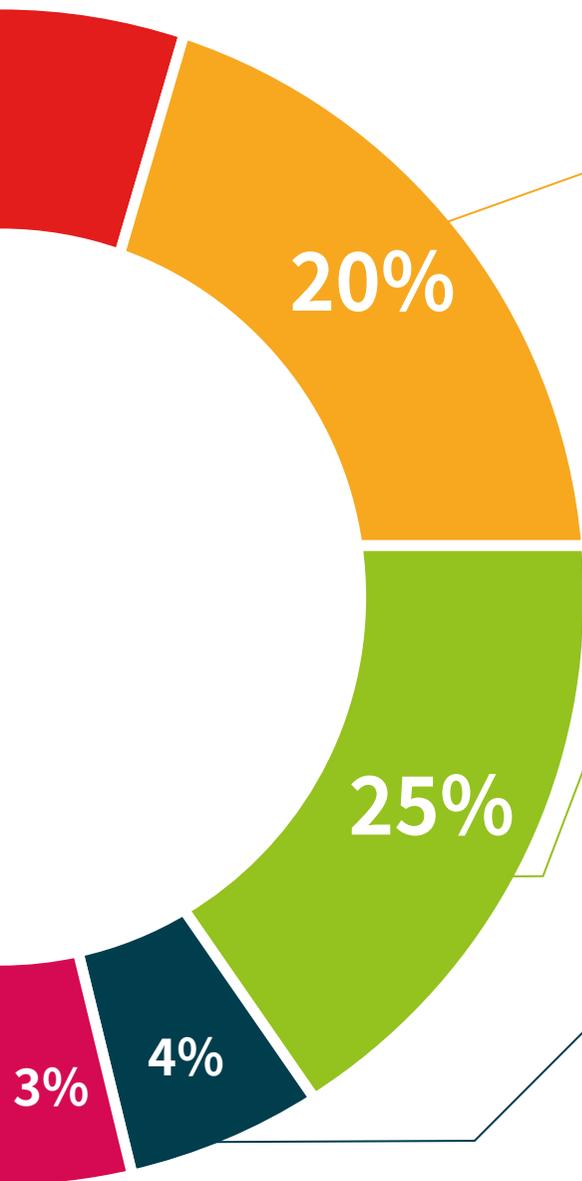
你将开展活动以发展每个学科领域的具体能力和技能。在我们所处的全球化框架内,我们提供实践和氛围帮你取得成为专家所需的技能和能力。



延伸阅读

最近的文章,共识文件和国际准则等。在TECH的虚拟图书馆里,学生可以获得他们完成培训所需的一切。





案例研究

他们将完成专门为这个学位选择的最佳案例研究。由国际上最好的专家介绍,分析和辅导案例。



互动式总结

TECH团队以有吸引力和动态的方式将内容呈现在多媒体中,其中包括音频,视频,图像,图表和概念图,以强化知识。
这个用于展示多媒体内容的独特教育系统被微软授予“欧洲成功案例”称号。



测试和循环测试

在整个课程中,通过评估和自我评估活动和练习,定期评估和重新评估学习者的知识:通过这种方式,学习者可以看到他/她是如何实现其目标的。



09 学位

人工智能和知识工程半面授校级硕士除了保证最严格和最新的培训外,还可以获得由TECH科技大学颁发的半面授校级硕士学位证书。





“

成功地完成这个学位,省去
出门或办理文件的麻烦”

这个人工智能和知识工程半面授校级硕士包含了市场上最完整和最新的课程。
评估通过后, 学生将通过邮寄收到TECH科技大学颁发的相应的半面授校级硕士学位。

TECH科技大学颁发的证书将表达在半面授校级硕士获得的资格, 并将满足工作交流, 竞争性考试和专业职业评估委员会的普遍要求。

学位: 人工智能和知识工程半面授校级硕士

模式: 在线

时长: 12个月



健康 信心 未来 人 导师
教育 信息 教学
保证 资格认证 学习
机构 社区 科技 承诺 创新
个性化的关注 现在
知识 网页 培训
网上教室 发展 语言

tech 科学技术大学

半面授校级硕士
人工智能和知识工程

模式: 混合式(在线+临床实践)

时间: 12个月

学位: TECH 科技大学

半面授校级硕士

人工智能和知识工程

