Introduction to Artificial Intelligence AI导论

EBU4203

- TB1: Introduction to AI, uncertainty in decision making, machine learning basics
  人工智能入门，决策中的不确定性，机器学习基础
- TB2: Deep learning and reinforcement learning
  深度学习和强化学习
- TB3: Practical AI Applications and Computer Vision
  实用人工智能应用与计算机视觉
- TB4: Natural Language Processing (NLP) and future trends in AI
  自然语言处理与人工智能的发展趋势

Introduce 简要介绍

assessment

- 1 x Class Test 3% 课堂小测
  - After teaching block 2
- 2 x Self-revision Online Quizzes 3% 网上自测
  - Open for a week
- Laboratory 14% 实验
  - Lab reports 实验报告
  - 一共三次,第一次不用写
- Final exam 80%
  - closed-book written exam 闭卷考试
  - Past papers will be put on QMPlus 过去的试卷将放在 QMPlus 上
  - Note: A minimum total mark of 40% is required to pass this module
    注意: 通过本模块最低总分为40%
- Coursework:
  - Note: There is a coursework hurdle of 30% (A minimum total coursework mark of 30% is required to pass this module)
    注: 有一个30% 的课程作业障碍(通过此模块需要最少30% 的课程作业总分)

Information

- Course website: 课程网站
  - Login to QMPlus
  - Course Area: EBU4203 (Introduction to AI)
  - Check it regularly, as it is possible there could be additional information e.g. messages, extra practice exercises, tutorials, etc.
    定期检查，因为可能会有额外的信息，如信息，额外的练习，教程等。
- Email
  - You are expected to check your QM email every week at least!
    你至少应该每周查看一次 QM 邮件！

Recommended Text book and references 推荐的教科书和参考资料

- ["1"] Russell, S., & Norvig, P. (2021). Artificial Intelligence: a modern approach, 4th US ed. University of California, Berkeley.
  ["1"]拉塞尔，S。 & 诺维格，P。(2021)。人工智能: 一种现代方法，美国第四版。加州大学伯克利分校。
- There are plenty of books available on this topic.
  有很多关于这个主题的书。

Few tips

- Attend every lecture, tutorial, lab and assessment sessions.
  参加每一个讲座，辅导，实验室和评估会议。
- Revise your lecture materials after every class.
  每节课后都要修改讲义。
- Make use of available materials, and read books and online materials.
  利用现有的资料，阅读书籍和网上资料。
- Be interactive during the class and tutorial sessions.
  在课堂和辅导课程中保持互动。
- Ask your lecturers/TAs and discuss with your classmates.
  询问你的讲师/助教，并与你的同学讨论。

mentimeter

interaction tools

week 1

- Part 1: Introduction to AI AI引入
- Part 2: Uncertainty in decision making 决策的不确定性
- Part 3: Machine learning basics 机器学习基础

Part 1: Introduction to AI 第一部分: 人工智能导论

- Definition and scope of AI 人工智能的定义和范围
- Motivation for exploring AI 探索人工智能的动机
- Brief history of AI 人工智能简史
- Branches and applications of AI 人工智能的分支与应用
- Ethical considerations in AI 人工智能的伦理思考

What is artificial intelligence?

Definition and scope of AI 人工智能的定义和范围

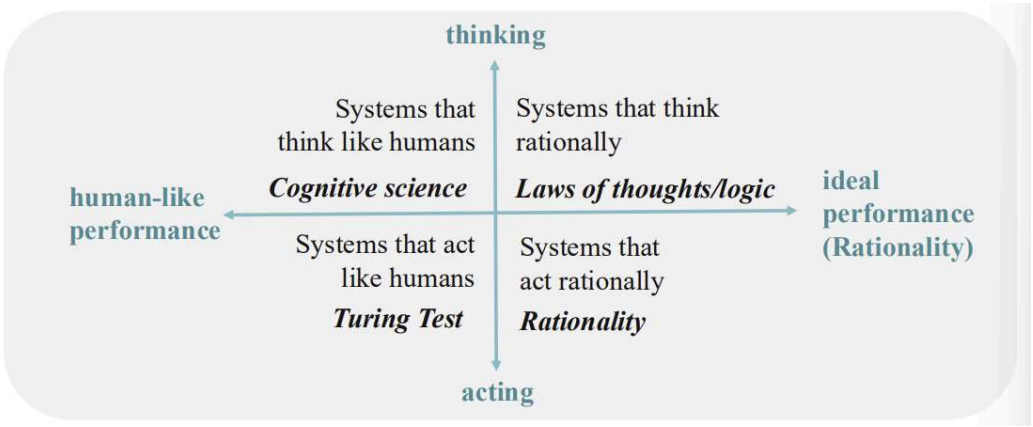
Alan Turing 图灵 —— 图灵测试 Turing Test

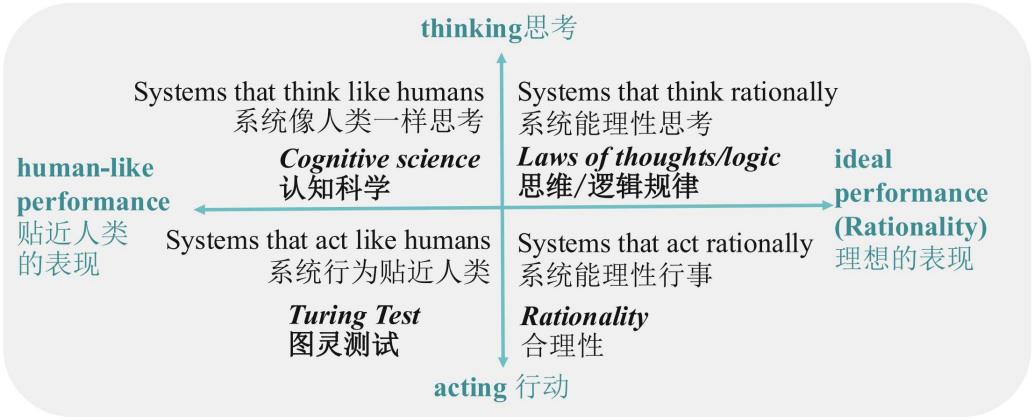
- The Turing Test aims to evaluate **whether a machine can exhibit intelligence comparable to that of a human.**
  图灵测试的目的是评估一台机器能否展现出与人类相当的智力。
- A **text conversation** between a judge, a human, and a machine, where the judge tries to determine whether he is conversing with a human or a machine.
  一种法官、人类和机器之间的文本对话，法官试图确定他是在与人类还是机器交谈。
- 重要性和局限
  - Significance:
    - The Turing Test serves as a method for assessing the level of artificial intelligence.
      图灵测试是评估人工智能水平的一种方法。
    - If a machine can pass the Turing Test, it indicates a certain level of intelligence and raises questions about AI capabilities.
      如果一台机器能够通过图灵测试，它表明了一定程度的智能，并提出了关于人工智能能力的问题。
  - Limitations：
    - The Turing Test focuses **solely on external behavior** and does not evaluate internal cognitive processes.
      图灵测试只关注外部行为，不评估内部认知过程。
    - It may be influenced by **subjective judgments** from the judge and other factors.
      它可能受到法官主观判断等因素的影响。
- IBM's Jeopardy Challenge: An intriguing step toward AI passing the Turing Test
  IBM 的危险挑战: 迈向人工智能通过图灵测试的有趣一步
  - In 2011, IBM's **supercomputer "Watson" won the Jeopardy Challenge**, becoming the first robot champion.
    2011年，IBM 的超级计算机"沃森"赢得了危险挑战赛，成为第一个机器人冠军。
  - Watson's performance in the Jeopardy Challenge: **Comprehending questions, analysing information, and selecting the most probable answer**.
    沃森在危险挑战中的表现: 理解问题，分析信息，选择最可能的答案。

- IBM's Jeopardy Challenge provided a demonstration of **a technological breakthrough related to the Turing Test**, proving the potential of machines to process natural language and reasoning, and driving further development in the field of artificial intelligence.
  IBM 的 Jeopardy Challenge 展示了与图灵测试相关的技术突破，证明了机器处理自然语言和推理的潜力，并推动了人工智能领域的进一步发展。

The Four potential goals or definitions of AI 人工智能的四个潜在目标或定义

- They differentiates computer systems on the basis of rationality and thinking vs. acting
  他们区分计算机系统的基础是理性和思考与行动:





- At its simplest form, artificial intelligence is a field, which **combines computer science and robust datasets**, to enable **problem-solving**.
  简单来说，人工智能是一个将计算机科学和强大的数据集结合起来的领域，它能够解决问题。
- It also encompasses sub-fields of **machine learning and deep learning**, which are frequently mentioned in conjunction with artificial intelligence.
  它还包括机器学习和深度学习的子领域，这些领域经常与人工智能一起被提及。
- These disciplines are comprised of AI algorithms which seek to **create expert systems which make predictions or classifications based on input data**.
  这些学科由人工智能算法组成，该算法寻求创建专家系统，根据输入数据进行预测或分类。

The scope of AI 人工智能的范围

- As we begin the new millennium
  千禧年后开始
  - science and technology are changing rapidly
    科学技术正在迅速变化
  - "old" sciences such as physics are relatively well-understood
    像物理这样的"古老"科学相对来说已经被广为人知了
  - computers are ubiquitous
    电脑无处不在
- Grand Challenges in Science and Technology
  科学技术面临的重大挑战
  - understanding the brain
    对脑科学的理解与研究
  - reasoning, cognition, creativity
    推理、认知、创造力

- creating intelligent machines
  创造智能机器

The Foundations of AI 人工智能的基础

How does AI work? 人工智能是如何工作的?

Motivation for exploring AI

Why AI Matters? 为什么人工智能很重要

1. Potential to Transform AI has the potential **to revolutionize various aspects** of our lives, work, and leisure activities.
   人工智能有可能彻底改变我们生活、工作和休闲活动的各个方面。

2. Business Automation AI has been effectively utilized in businesses **to automate tasks** that were previously performed by humans, such as customer service, lead generation, fraud detection, and quality control.
   人工智能已经被有效地应用于企业中，使以前由人类执行的任务自动化，例如客户服务、引导生成、欺诈检测和质量控制。

3. Superior Performance In many areas, **AI outperforms humans** in tasks, especially those that are **repetitive and detail-oriented**. AI tools can **quickly analyse large volumes of legal documents**, ensuring accurate and complete information.

在许多领域，人工智能在任务方面胜过人类，尤其是那些重复性和注重细节的任务。人工智能工具可以快速分析大量的法律文件，确保准确和完整的信息

4. Efficiency and Accuracy AI tools can **complete tasks quickly and with relatively few errors**, particularly in areas that require analysing extensive data sets. This enables businesses to gain insights into their operations that may have otherwise gone unnoticed.
人工智能工具可以快速完成任务，错误相对较少，特别是在需要分析大量数据集的领域。这使得企业能够深入了解他们的业务，否则可能会被忽视。

5. Generative AI Tools The growing population of generative AI tools holds great importance in fields like education, marketing, and product design. These tools offer **innovative solutions and creative outputs**.
越来越多的生成性人工智能工具在教育、市场营销和产品设计等领域具有重要意义。这些工具提供了创新的解决方案和创造性的产出。

AI opens the door to new opportunities 人工智能为新的机会打开了大门

- UBER
- Meta
- Microsoft
- Alphabet
- Apple

The advantages of AI 人工智能的优势

1. Good at detail-oriented jobs
擅长细节导向的工作

2. Saves labour and increases productivity
节省劳动力，提高生产力

3. Delivers consistent results
产生一致的结果

4. AI-powered virtual agents are always available
人工智能驱动的虚拟代理总是可用的

5. Reduced time for data-heavy tasks
减少数据量大的任务的时间

6. Can improve customer satisfaction through personalization
可以通过个性化提高客户满意度

AI is NOT everything (limitations) 人工智能不是一切(局限性)

1. Expensive
昂贵的

2. Requires deep technical expertise
需要深厚的专业技术

3. Limited supply of qualified workers to build AI tools
人工智能工具的合格工人供应有限

4. Reflects the biases of its training data, at scale.
在规模上反映了其训练数据的偏差。

5. Lack of ability to generalize from one task to another
缺乏从一项任务归纳到另一项任务的能力

6. Eliminates human jobs, increasing unemployment rates
减少人类工作，增加失业率

Brief history of AI 人工智能简史

Ancient Roots of Intelligent Artifacts 智能物品的古老根源

From Mythical Servants 来自神话仆人

- The concept of inanimate objects endowed with intelligence has been around since ancient times.
被赋予智慧的无生命物体的概念自古以来就存在。
    - Greek god Hephaestus and robot-like servants out of gold
    希腊神赫菲斯托斯和机器人般的仆人用金子做的
    - Engineers in ancient Egypt and statues of gods animated by priests
    古埃及的工程师和祭司制作的神像

To Symbolic Thinkers 对象征思想家的思考

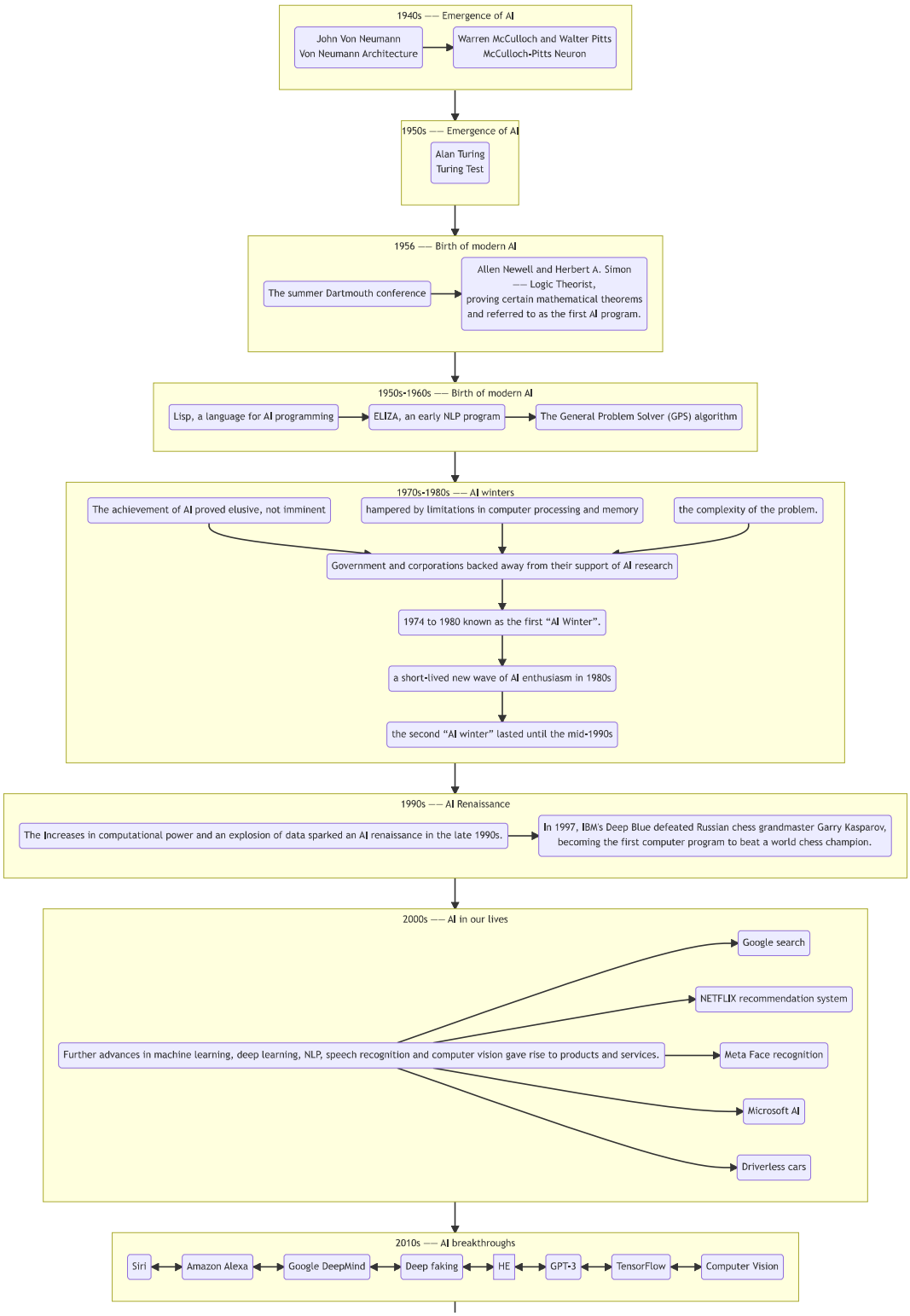
- They used the tools and logic of their times to describe human thought processes as symbols, laying the foundation for AI concepts such as general knowledge representation.
  他们利用当时的工具和逻辑将人类的思维过程描述为符号，为一般知识表示等人工智能概念奠定了基础。
- Aristotle 亚里士多德 Ramon Llull 拉蒙·柳利 René Descartes 勒内 · 笛卡尔 Thomas Bayes 托马斯·贝叶斯

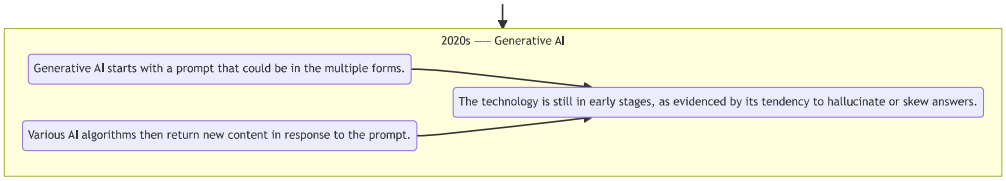
Pioneers of Programmable Machines 可编程机器的先驱

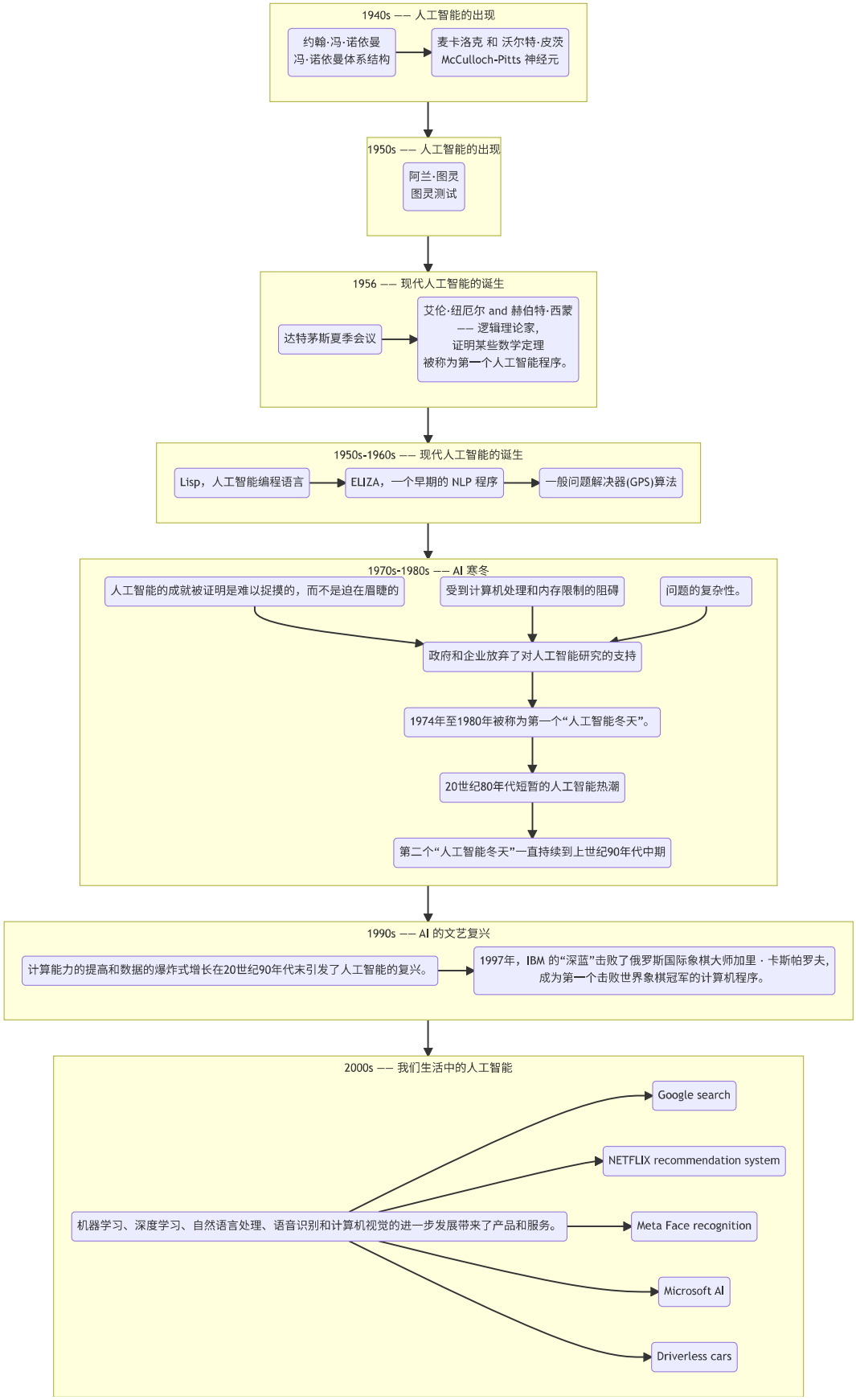
The foundational work that would give rise to the modern computer
产生现代计算机的基础工作

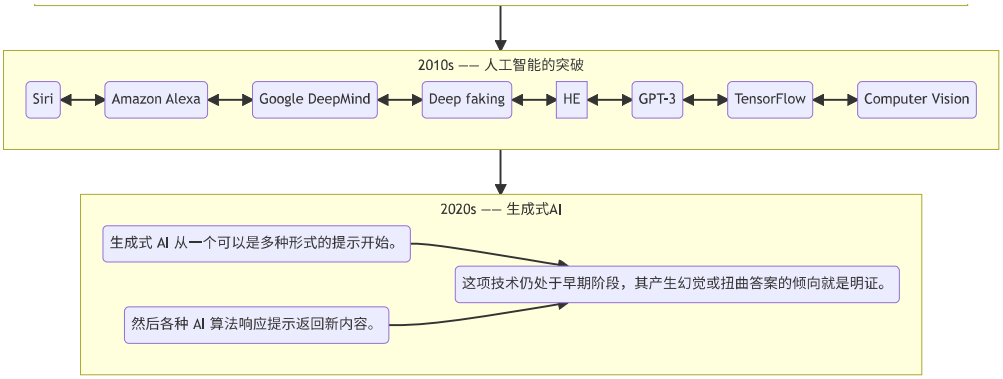
- the mill with a printing mechanism of the Analytical Engine 带有分析机打印机构的磨坊
- Babbage's difference engine 巴贝奇的差分引擎

Milestones in the Journey of AI


1940s —— Emergence of AI
John Von Neumann
Von Neumann Architecture
Warren McCulloch and Walter Pitts
McCulloch-Pitts Neuron
1950s —— Emergence of AI
Alan Turing
Turing Test
1956 —— Birth of modern AI
The summer Dartmouth conference
Allen Newell and Herbert A. Simon
—— Logic Theorist,
proving certain mathematical theorems
and referred to as the first AI program.
1950s-1960s —— Birth of modern AI
Lisp, a language for AI programming
ELIZA, an early NLP program
The General Problem Solver (GPS) algorithm
1970s-1980s —— AI winters
The achievement of AI proved elusive, not imminent
hampered by limitations in computer processing and memory
the complexity of the problem.
Government and corporations backed away from their support of AI research
1974 to 1980 known as the first “AI Winter”.
a short-lived new wave of AI enthusiasm in 1980s
the second “AI winter” lasted until the mid-1990s
1990s —— AI Renaissance
The increases in computational power and an explosion of data sparked an AI renaissance in the late 1990s.
In 1997, IBM's Deep Blue defeated Russian chess grandmaster Garry Kasparov, becoming the first computer program to beat a world chess champion.
2000s —— AI in our lives
Google search
NETFLIX recommendation system
Further advances in machine learning, deep learning, NLP, speech recognition and computer vision gave rise to products and services.
Meta Face recognition
Microsoft AI
Driverless cars
2010s —— AI breakthroughs
Siri
Amazon Alexa
Google DeepMind
Deep faking
HE
GPT-3
TensorFlow
Computer Vision


2020s —— Generative AI
Generative AI starts with a prompt that could be in the multiple forms.
Various AI algorithms then return new content in response to the prompt.
The technology is still in early stages, as evidenced by its tendency to hallucinate or skew answers.


1940s —— 人工智能的出现
约翰·冯·诺依曼
冯·诺依曼体系结构
麦卡洛克 和 沃尔特·皮茨
McCulloch-Pitts 神经元
1950s —— 人工智能的出现
阿兰·图灵
图灵测试
1956 —— 现代人工智能的诞生
达特茅斯夏季会议
艾伦·纽厄尔 and 赫伯特·西蒙
—— 逻辑理论家，
证明某些数学定理
被称为第一个人工智能程序。
1950s-1960s —— 现代人工智能的诞生
Lisp，人工智能编程语言
ELIZA，一个早期的 NLP 程序
一般问题解决器(GPS)算法
1970s-1980s —— AI 寒冬
人工智能的成就被证明是难以捉摸的，而不是迫在眉睫的
受到计算机处理和内存限制的阻碍
问题的复杂性。
政府和企业放弃了对人工智能研究的支持
1974年至1980年被称为第一个“人工智能冬天”。
20世纪80年代短暂的人工智能热潮
第二个“人工智能冬天”一直持续到上世纪90年代中期
1990s —— AI 的文艺复兴
计算能力的提高和数据的爆炸式增长在20世纪90年代末引发了人工智能的复兴。
1997年，IBM 的“深蓝”击败了俄罗斯国际象棋大师加里 · 卡斯帕罗夫，
成为第一个击败世界象棋冠军的计算机程序。
2000s —— 我们生活中的人工智能
机器学习、深度学习、自然语言处理、语音识别和计算机视觉的进一步发展带来了产品和服务。
Google search
NETFLIX recommendation system
Meta Face recognition
Microsoft AI
Driverless cars



Branches and applications of AI 人工智能的分支与应用

Weak AI vs. Strong AI

- Weak AI
  - also called Narrow AI or Artificial Narrow Intelligence (ANI)
  - is **AI trained and focused to perform specific tasks.**
  - Weak AI drives most of the AI that surrounds us today. 'Narrow' might be a more accurate descriptor for this type of AI as it is anything but weak; it enables some very robust applications.
    今天我们周围的大部分人工智能都是由弱人工智能驱动的。"窄"可能是一个更准确的描述这种类型的人工智能，因为它是任何东西，但弱，它使一些非常健壮的应用程序。
- Strong AI
  - made up of **Artificial General Intelligence (AGI)** and **Artificial Super Intelligence (ASI)**.
    由人工通用智能(AGI)和人工超级智能(ASI)组成。
  - AGI, or general AI, is a theoretical form of AI where a machine would have **an intelligence equal to humans**; it would have a **self-aware consciousness** that has the ability to solve problems, learn, and plan for the future.
    人工智能(AGI)是人工智能的一种理论形式，在这种形式中，机器拥有与人类相当的智能; 它具有自我意识，能够解决问题、学习和规划未来。
  - ASI—also known as superintelligence—would **surpass the intelligence and ability of the human brain**.
    人工智能——也被称为超级智能——将超越人类大脑的智力和能力。
  - While strong AI is still entirely theoretical with no practical examples in use today, that doesn't mean AI researchers aren't also exploring its development.
    虽然强大的人工智能仍然完全是理论上的，没有实际应用的例子，但这并不意味着人工智能研究人员没有探索它的发展。

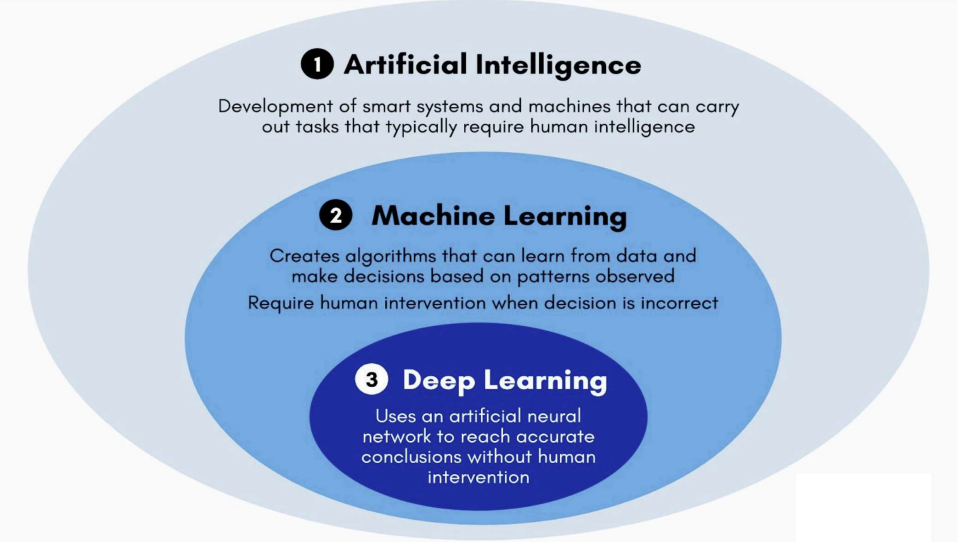
Four Types of AI

- Type 1: Reactive machines 类型1: 活性机器
  - have no memory 没有记忆
  - task-specific 只能执行特定任务
  - EXP.
    - An example is **Deep Blue**, the IBM chess program that beat Garry Kasparov in the 1990s.
      一个例子是深蓝(Deep Blue)，IBM 的国际象棋程序在上世纪90年代击败了加里•卡斯帕罗夫(Garry Kasparov)。
    - Deep Blue can identify pieces on a chessboard and make predictions, but because it has no memory, it **cannot use past experiences to inform future ones.**
      深蓝可以识别棋盘上的棋子并做出预测，但是因为它没有记忆，所以它不能用过去的经历来告诉未来的经历。
- Type 2: Limited memor 第2类: 记忆力有限
  - have memory 拥有记忆

- use past experiences to inform future decisions.
  利用过去的经验为将来的决策提供依据。
  Some of the decision-making functions in self-driving cars are designed this way.
  自动驾驶汽车的一些决策功能就是这样设计的。

- Type 3: Theory of mind 类型3: 心理理论
  - have the social intelligence to **understand emotions**
    具有理解情感的社会智慧
  - This type of AI will be able to infer human intentions and predict behavior, a necessary skill for AI systems to become integral members of human teams.
    这种类型的人工智能将能够推断人类的意图和预测行为，这是人工智能系统成为人类团队不可或缺的成员所必需的技能。
- Type 4: Self-awareness 类型4: 自我意识
  - have **a sense of self**, which gives them consciousness.
    有自我意识，这给了他们意识。
  - understand their own current stateThis type of AI does not yet exist.
    了解自己的现状这种类型的人工智能尚不存在。

Relationship between artificial intelligence, machine learning, and deep learning 人工智能、机器学习与深度学习的关系



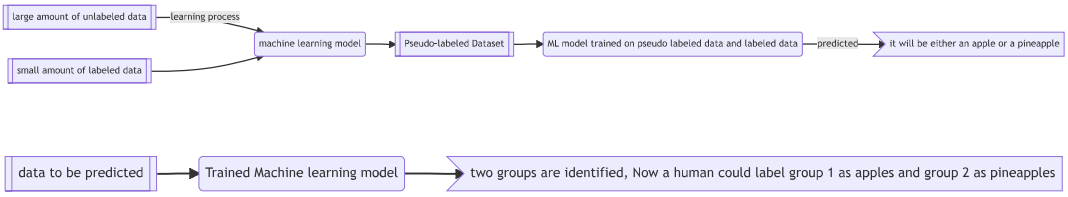
How machine learning works?

- ➢ Models 模型
  - Assumptions to be mapped to the learning problem
    映射到学习问题的假设
  - **(problem modelling, defining the assumption space)**
    (问题建模，定义假设空间)
- ➢ Strategies 策略
  - Criteria for learning/selecting the optimal model from the hypothesis space
    从假设空间学习/选择最优模型的准则
  - **(Determine objective function)**
    (确定目标函数)
- ➢ Algorithm 算法
  - Specific calculations for solving the optimal model based on the objective function
    基于目标函数求解最优模型的具体计算
  - **(solving for model parameters)**
    (模型参数求解)

Classification of models by data label

- Data Label
  - Supervised learning 监督学习 Supervised learning samples have labels (output targets); learns labelled interfaces from data (input-output mapping function), suitable for predictive data labelling
    监督式学习样本有标签(输出目标)，从数据中学习有标签的界面(输入输出映射功能)，适用于预测性数据标签

        - 分类 classfication
    - unsupervised learning 无监督学习 Unsupervised learning samples have no labelling; learns patterns from data, suitable for describing data
      非监督式学习样本没有标签，从数据中学习模式，适合描述数据
        - 聚类 clustering
    - Semi-supervised learning 半监督学习 （不算到三种里面，而是作为前两种的融合）
        - Starting point: labelled samples difficult to obtain, unlabelled samples relatively inexpensive.
          起始点: 标记样品难以获得，未标记样品相对便宜。
        - Idea: Assume that unlabelled samples are independently and identically distributed with labelled samples, i.e., contain important information about the distribution of the data
          想法: 假设未标记的样品与标记的样品分布独立且相同，即含有关于数据分布的重要信息



- Reinforcement Learning 强化学习 uses unlabelled data but can know whether it is getting closer or further away from the goal (rewarding feedback)
  使用未标记的数据，但可以知道它是否离目标越来越近或越来越远(奖励反馈)

- **Use cases of AI technology 人工智能技术的用例**

1. Automation: AI technologies paired with automation tools like robotic process automation (RPA) **automate repetitive, rules-based tasks, expanding task volume and types**.
   自动化: 人工智能技术配合自动化工具，如机器人过程自动化(RPA)自动化重复，基于规则的任务，扩大任务量和类型。
2. Machine Learning: Enables computers to **act without explicit programming**. Deep learning automates **predictive analytics**.
   机器学习: 使计算机不需要编程就能运行。深度学习使预测分析自动化。
3. Computer Vision (CV): Gives machines the ability to **see and analyse visual information** using cameras and digital signal processing.
   计算机视觉(CV) : 使机器能够看到和分析视觉信息使用相机和数字信号处理。
4. Natural Language Processing (NLP): **Processes human language by computer programs**, including tasks like translation, sentiment analysis, and speech recognition.
   自然语言处理(NLP) : 通过计算机程序处理人类语言，包括翻译、情感分析和语音识别等任务。
5. Robotics: Engineering field focused on designing and manufacturing robots for tasks challenging for humans or requiring consistent performance.
   机器人学: 工程领域专注于设计和制造机器人来完成对人类具有挑战性或需要一致性能的任务。
6. Self-Driving Cars: Utilize computer vision, image recognition, and deep learning to navigate roads and avoid obstacles.
   自动驾驶汽车: 利用计算机视觉、图像识别和深度学习来驾驶道路和避开障碍物。
7. Text, Image, and Audio Generation: Generative AI techniques create various media types based on text prompts, applied extensively across businesses for content creation.
   文本、图像和音频生成: 生成式人工智能技术基于文本提示创建各种媒体类型，广泛应用于企业内容创建。

AI applications

1. Healthcare: AI is used to improve diagnoses, mine patient data, and assist with administrative tasks like scheduling appointments.
   医疗保健: 人工智能用于改善诊断，挖掘患者数据，并协助行政任务，如安排预约。
2. Business: Machine learning and chatbots enhance customer service, while generative AI has the potential to revolutionize product design and disrupt business models.
   业务: 机器学习和聊天机器人提高了客户服务，而生成性人工智能有可能彻底改革产品设计和颠覆商业模式。
3. Education: AI automates grading, adapts to student needs, and provides additional support. It also aids in crafting course materials and changing the learning process.
   教育: 人工智能自动评分，适应学生的需要，并提供额外的支持。它还有助于精心制作课程材料和改变学习过程。
4. Finance: AI disrupts the financial industry through personal finance applications, automated trading, and the buying process for homes.
   金融: 人工智能通过个人理财应用、自动交易和购房过程扰乱了金融业。

5. Law: AI assists with legal processes such as document classification, data description, and outcome prediction.
   法律: 人工智能协助法律程序，如文档分类，数据描述和结果预测。
6. Entertainment and Media: AI is used for targeted advertising, content recommendation, script creation, automated journalism, and movie production.
   娱乐和媒体: 人工智能用于定向广告、内容推荐、剧本创作、自动化新闻和电影制作。
7. Software Coding and IT Processes: Generative AI tools aid in code generation, while AI automates IT processes like data entry and security measures.
   软件编码和 IT 过程: 生成 AI 工具帮助代码生成，而 AI 自动化 IT 过程，如数据输入和安全措施。
8. Security: AI is applied to cybersecurity for threat detection, anomaly detection, and behavior analytics.
   安全性: 人工智能应用于网络安全，用于威胁检测、异常检测和行为分析。
9. Manufacturing: Robots collaborate with human workers in tasks previously done separately, increasing efficiency and multitasking capabilities.
   制造业: 机器人与人类工人协作完成以前单独完成的任务，提高效率和多任务处理能力。
10. Banking: Chatbots and virtual assistants improve customer service and compliance with regulations, while AI aids in decision-making for loans and investments.
    银行业务: 聊天机器人和虚拟助理改善客户服务和遵守规定，而人工智能协助贷款和投资决策。
11. Transportation: AI manages traffic, predicts flight delays, enhances supply chain management, and promotes safer and more efficient transportation methods.
    运输: 人工智能管理交通，预测航班延误，加强供应链管理，促进更安全和更有效的运输方法。

Ethical considerations in AI 人工智能的伦理思考 （limitations）

Training Bias 含有歧视的训练

- AI systems can **perpetuate biases present in the training data**, which can lead to **unfair or discriminatory outcomes**.
  人工智能系统可能使培训数据中存在的偏见长期存在，从而导致不公平或歧视性的结果。
- Monitoring and addressing bias in machine learning algorithms is crucial to ensure fairness and avoid reinforcing existing inequalities.
  监测和处理机器学习算法中的偏差对于确保公平性和避免加剧现有的不平等是至关重要的。

Misuse 误用，滥用

- AI technology can be misused for malicious purposes
  人工智能技术可能被滥用于恶意目的
  - creating deepfakes
  - engaging in phishing attacks. 进行网络钓鱼攻击。
- Safeguarding against misuse requires careful regulation and security measures.
  防止滥用需要认真的监管和安全措施。

Interpretability 可解释性

- AI algorithms can be **difficult to interpret**. AI算法难以被数学解释
  - deep learning 深度学习
  - generative adversarial network (GAN) 生成式对抗网络
- This poses challenges in industries with regulatory compliance requirements, where **interpretability is necessary to meet legal obligations**.
  这对有守规要求的行业提出了挑战，因为在这些行业，解释性对于履行法律义务是必要的。

Job Displacement 工作被替代

- The automation enabled by AI can lead **to job losses and significant disruptions in the workforce.**
  人工智能带来的自动化可能导致失业和劳动力大量中断。
- Preparing for the impact on employment and addressing the need for upskilling and reskilling becomes crucial.
  为对就业的影响做好准备以及解决提高技能和重新提高技能的需要变得至关重要。

Legal Concerns 法律问题

- AI raises legal issues, including potential cases of **AI-generated libel and copyright infringement**.
  AI 提出了法律问题，包括可能出现的由AI引发的诽谤和盗版案件 。
- Developing appropriate legal frameworks and regulations to address these concerns is essential.
  必须制定适当的法律框架和条例来解决这些问题。

Data Privacy 数据隐私

- AI applications often **rely on vast amounts of sensitive data**, particularly in fields like banking, healthcare, and law.
  人工智能应用程序通常依赖于大量的敏感数据，特别是在银行、医疗保健和法律等领域。

- Ensuring proper data privacy protections and adhering to relevant regulations is crucial to safeguard individuals' privacy.
  确保适当的数据隐私保护和遵守相关法规对于保护个人隐私至关重要。

Address 解决方法

- responsible AI development 负责任的人工智能开发
- robust regulations 强有力的监管
- transparency 透明，透明性
- ongoing monitoring 持续监测
- stakeholder engagement 利益相关者参与
- Ethical considerations must be an integral part of the AI development process. 道德考虑必须是人工智能开发过程的一个部分。

Part 2: Uncertainty in decision making 决策的不确定性

- Logic and uncertainty 逻辑和不确定性
- Probability theory 概率论
- Random Variables 随机变量
- Bayes rule and conditional independence 贝叶斯规则和条件独立
- Bayes (belief) network 贝叶斯(信念)网络

Logic and uncertainty 逻辑和不确定性

```
- Aim
  - To familiarise with uncertainty quantifications
  - To understand probabilistic reasoning and Bayes rule
- Outcome
  - Appreciate uncertainties
  - Quantification and reasoning using Probability
  - Probabilistic reasoning
  - Brief uncertain reasoning using
  - Bayes Network
```

Major problem with logical-agent approaches 用逻辑去应用智能的主要问题

1. Agents almost never have access to the whole truth about theirenvironments
   智能几乎永远无法了解他们所处环境的全部真相
2. There are important questions for which there is no yes/no answer (even in simple terms)
   有些重要的问题没有是非回答(即使是简单的回答)
3. Therefore, an agent must reason under uncertainty.
   因此，智能必须在不确定条件下进行推理。
4. Uncertainty also arises because of an agent's incomplete or incorrect understanding of itsenvironment.
   不确定性的产生也是由于智能对其环境的不完全或不正确的理解。

Why application fails (when uncertainties are not considered appropriately) ? 实际应用为何失败

1. LAZINESS: **too much work** to list the complete set ofantecedents or consequents needed to ensure an **exceptionless rule** and **too hard** to use such rules.
   懒惰: 为了确保一个无例外的规则和太难使用这样的规则，需要列出一整套完整的前因后果，工作量太大。
2. THEORETICAL ignorance: Medical science has no completetheory for the domain.
   理论上的无知: 医学在这个领域没有完整的理论。
3. PRACTICAL ignorance: Even if we **know all the rules**,we might be **uncertain** about a particular patient because not **all the necessary tests** have **been or can be run**.
   实际无知: 即使我们知道所有的规则，我们也可能对某个特定的病人不确定，因为并非所有必要的检查都已经或可以运行。

Reasoning under uncertainty 不确定性推理

- A rational agent is one that makes rational decisions — to maximize its performance measure
  理性代理人是做出理性决策的人——为了最大限度地提高其绩效指标
- A rational decision depends on
  理性的决定取决于
  - the **relative** importance of various goals
    不同目标的相对重要性

- the **likelihood** they will be achieved
  实现这些目标的可能性
- the **degree** to which they will be achieved
  达到的程度

Types of uncertainty 不确定性类型

- Uncertainty in **prior knowledge**
  先验知识的不确定性
- Uncertainty in **actions**
  行动的不确定性
- Uncertainty in **perception**
  感知的不确定性

Uncertainty is a summary of all that is not explicitly considered in the agent's knowledge base.
不确定性是代理的知识库中没有明确考虑的所有不确定性的总结。

Handling uncertainty 不确定性处理

- **Default reasoning [Optimistic]**
  正向推理〔乐观〕
  an agent assumes normality, until there is evidence of thecontrary.
  除非有相反的证据，否则智能就会假装一切正常。
- **Worst-case reasoning [Pessimistic]**
  最坏情况推理〔悲观者〕
  The agent assumes the **worst case**, and chooses the actions that maximizes a **utility function** in this case.
  智能假设最坏的情况，并在这种情况下选择使效用函数最大化的操作。
  Disadvantages:
  ❑ not **worth the effort** to develop or explore such a scenario; 不值得努力发展或探索这种情况
  ❑ may **waste resources** preparing for highly unlikelycontingencies; 可能会浪费资源，为极不可能发生的突发事件做准备
  ❑ **restricted** way of handling an emergency. 处理紧急事件的有限方式
- **Probabilistic reasoning [Realist]**
  概率性推理 [现实]

Probability theory 概率论

Probabilistic reasoning 概率性推理

- The agent has **probabilistic beliefs**
  - pieces of knowledge with associated probabilities (**strengths**)
    具有相关概率(优势)的知识片段
  - chooses its actions to maximize the expected value of some **utility function**
    选择自己的行为来使某种效用函数的期望值最大化
- Rationale: The world is not divided between "normal" and"abnormal", nor is it adversarial. Possible situations have various **likelihoods/chance** (probabilities)
  理由: 这个世界没有"正常"和"不正常"之分，也没有对抗。可能的情况有各种各样的可能性/机会(概率)

Probabilistic reasoning and degrees of belief 概率推理和信任度

- The agent's knowledge can only provide a **degree of belief** in the relevant sentences
  代理人的知识只能提供对相关句子的一定程度的信任
- The agent cannot say whether a sentence is true, but only that is **true x%** of the times
  代理不能说出一个句子是否为真，但只能说出真的 x% 的次数
- The main tool for handling degrees of belief is **Probability Theory**
  处理信任度的主要工具是概率论
- The use of probability **summarizes** the **uncertainty** that stems from human's **laziness or ignorance** about the domain
  概率的使用概括了由于人类的懒惰或对领域的无知而产生的不确定性

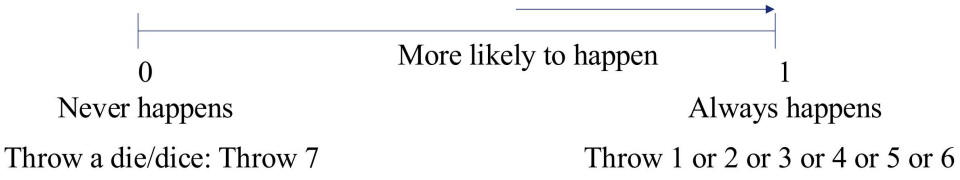
Probability theory & facts 概率论与事实

- Probability Theory makes the same ontological commitments as First-order Logic:
  概率论作出与一阶逻辑相同的本体论承诺:
  Every sentence φ is either true or false
  每个句子 φ 不是真就是假
- The **degree** of belief that φ is true is a number P between 0 and 1
  - $P(\varphi) = 1 \rightarrow \varphi$ is certainly true

- P(φ) = 0 → φ is certainly not true
- P(φ) = 0.65 → φ is true with a 65% chance

Probability facts 概率事实

- Let $\boldsymbol{A}$ be a propositional variable, a symbol denoting aproposition that is either true or false.
  设 a 是一个命题变量，一个表示命题是真或假的符号。
- $\boldsymbol{P}(\boldsymbol{A})$ denotes the probability that $\boldsymbol{A}$ is true in the absence of any other information.
  P (A)表示在没有任何其他信息的情况下 A 为真的概率。
- Similarly:
  - $P(\neg A)$ = probability that A is false (~ or NOT) $P(\neg A)$ = A 为假(或非假)的概率
  - $\boldsymbol{P}(\boldsymbol{A} \cap \boldsymbol{B})$ = probability that both $\boldsymbol{A}$ and $\boldsymbol{B}$ are true P (A ∩ B) = A 和 B 都为真的概率
  - $\boldsymbol{P}(\boldsymbol{A} \cup \boldsymbol{B})$ = probability that either $\boldsymbol{A}$ or $\boldsymbol{B}$ (or both) are true P (A ∪ B) = A 或 B (或两者)为真的概率
- Interpretation
  - If P is the probability of an event:$\boldsymbol{0} \leq \boldsymbol{P} \leq \boldsymbol{1}$
  - P = 0 means the event **cannot** occur
  - P = 1 means the event is **certain** to occur
  - The closer to 1, the **more likely** the event



    - A priori 先前信息
    - Relative frequency 相对频率
    - Subjective 主观

Recap: axioms of probability 概述: 概率公理

- Complementary events 互补事件
  - $P(A) + P(\neg A) = 1$ Hence, $P(A) = 1 - P(\neg A)$
- Combining events 复合事件
  - A or B; $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ [Union (dark blue and yellow)]
  - A and B; $P(A \cap B) = P(A) \times P(B)$ [Intersection]

Subjective/Bayesian Probability 主观/贝叶斯概率

1. Probabilities relate propositions to one's own state of knowledge
   概率将命题与自己的知识状态联系起来
2. Probabilities of propositions change with new evidence
   命题的概率随着新证据的出现而改变
3. This is analogous to logical entailment status KB |= φ(which changes with more knowledge), NOT truth!
   这类似于逻辑蕴含状态 KB | = φ (随着知识的增加而改变) ，不是真理!

- Therefore, Probability is an important reasoning for decisionmaking analysis!
  因此，概率论是决策分析的一个重要推理方法!

Unconditional & conditional probability 无条件及无条件概率

1. $\boldsymbol{P}(\boldsymbol{A})$ is the **unconditional (or prior) probability** of fact $\boldsymbol{A}$
   P (A)是事实 A 的无条件(或优先)概率
2. An agent can use the unconditional probability of $\boldsymbol{A}$ to reason about $\boldsymbol{A}$ in theabsence of further information
   在没有进一步信息的情况下，智能可以使用 A 的无条件概率来推理 A
3. If further evidence $\boldsymbol{B}$ becomes available, the agent must use the **conditional (or posterior) probability**: $P(\boldsymbol{A} \mid \boldsymbol{B})$
   如果进一步的证据 B 变得可用，智能必须使用条件(或后验)概率: $P(A \mid B)$
4. the probability of A **given that (all) the agent knows (is)** $\boldsymbol{B}$
   给定智能知道 B 的可能性

Note: $\boldsymbol{P}(\boldsymbol{A})$ can be thought as the conditional probability of $\boldsymbol{A}$ with respect to the empty evidence: $P(A) = P(A \mid B)$
注: 对于空证据，P (A)可以被认为是 A 的条件概率: $P(A) = P(A \mid B)$

Conditional probability 条件概率

- Definition：

$$\boldsymbol{P}(\boldsymbol{A} \cap \boldsymbol{B}) = \boldsymbol{P}(\boldsymbol{A} \mid \boldsymbol{B})\boldsymbol{P}(\boldsymbol{B})$$

  - Read $P(A \mid B)$: **Probability of $\boldsymbol{A}$ given that we know $\boldsymbol{B}$** $P(A)$ is called the **prior probability** of $\boldsymbol{A}$
    $P(A)$被称为A的先验概率 $P(A \mid B)$ is called the **posterior or conditional probability** of $\boldsymbol{A}$ given $\boldsymbol{B}$ $P(A \mid B)$被称为 a 给定的 b 的后面或条件概率

- Definition:

$$\boldsymbol{P}(\boldsymbol{B} \mid \boldsymbol{A}) = \boldsymbol{P}(\boldsymbol{A} \cap \boldsymbol{B})/\boldsymbol{P}(\boldsymbol{A})$$

  - "$B \mid A$" means "B given A" $P(B \mid A)$ is the probability that B will happen if A has already happened. $P(B \mid A)$是当 A 已经发生时 B 将发生的概率。

- Conditional probabilities are defined in terms of unconditional ones 条件概率是用无条件概率来定义的

- Whenever $\boldsymbol{P}(\boldsymbol{B}) > \boldsymbol{0}$,

$$\boldsymbol{P}(\boldsymbol{A} \mid \boldsymbol{B}) = \boldsymbol{P}(\boldsymbol{A} \cap \boldsymbol{B})/\boldsymbol{P}(\boldsymbol{B})$$

$$\boldsymbol{P}(\boldsymbol{A} \cap \boldsymbol{B}) = \boldsymbol{P}(\boldsymbol{A} \mid \boldsymbol{B})\boldsymbol{P}(\boldsymbol{B}) = \boldsymbol{P}(\boldsymbol{B} \mid \boldsymbol{A})\boldsymbol{P}(\boldsymbol{A})$$

- A and B are independent,then A 和 B 是独立的，则

$$P(A \mid B) = P(A)$$

$$P(B \mid A) = P(B)$$

$$P(A \cap B) = P(A)P(B)$$

- Another generalisation：另一个概括是:

$$P(A \cap B \cap C) = P(A \mid B, C)P(B \mid C)P(C)$$

Frequency Interpretation 频率解释

- Draw a ball from a bag containing n balls of the same size, red and s yellow. 从装有红色和黄色等大小的 n 个球的袋子中抽出一个球。

- The probability that the proposition A = "the ball is red" is true corresponds to the relative frequency with which we expect to draw a red ball 命题 A = "球是红色的"是真的的概率对应于我们期望画一个红色球的相对频率

$$P(A) = \frac{r}{n}$$

Random Variables 随机变量

Random Variables Definition

- A random variable is a variable rangingover a certain domain of $\boldsymbol{Values}$ 一个随机变量是一个变量范围超过一定的值域
- It is discrete if it ranges over a discrete (that is,countable) domain 如果它的范围超过一个离散(即可数)域，则它是离散的
- continuous if it ranges over the real numbers 在实数范围内是连续的
- We will only consider discrete randomvariables with finite domains 我们只考虑有限域的离散随机变量

> Note: Propositional variables can be seen as randomvariables over the Boolean domain 命题变量可以看作是布尔域上的随机变量

- A proposition that takes the value True with probability p and False with probability $1 - p$ is a random variable with distribution $(p, 1 - p)$
- If a bag contains balls having 3 possible colors – red, yellow, and blue – the color of a ball picked at random from the bag is a random variable with 3 possible values 一个取概率为 p 的 True 和概率为1-p 的 False 值的命题是一个具有分布的随机变量
- The (probability) distribution of a random variable X with n values $x_1, x_2, ..., x_n$ is: 随机变量 X与 n 值 $x_1, x_2, ..., x_n$的(概率)分布是:

$$(p_1, p_2, ..., p_n)$$

$$P(X = x_i) = p_i$$

$$\sum_{i=i,...,n} p_n = 1$$

Variable	Domain
Age	{1, 2, . . . , 120}
Weather	{sunnt, dry, cloudy, rain, snow}
Size	{amsll, medium, large}
Blonde	{true, false}

- The probability that a random variable X has value val is written as 随机变量 X 具有值 val 的概率写为

$$P(X = val)$$

> Note 1: $P(A = true)$ is written shortly as $P(a)$ while $P(A = false)$ iswritten as $P(\neg a)$ P (A = true)简写为 P (a) ，而 P (A = false)简写为 P (a) Note 2: Traditionally, in ProbabilityTheory variables are capitalized andconstant values are NOT.
> 传统上，在概率论中，变量是大写的，而常数值不是。

Probability distribution 概率分布

- If $\boldsymbol{X}$ is a random variable, we use the bold case $\boldsymbol{P}(\boldsymbol{X})$ to denote a vector ofvalues for the probabilities of each individual element that $\boldsymbol{X}$ can take. 如果 X 是一个随机变量，我们使用粗体大小写 P (X)来表示 X 可以接受的每个单独元素的概率的值向量。
- Example

$$P(Weather = sunny) = 0.6$$
$$P(Weather = rain) = 0.2$$
$$P(Weather = cloudy) = 0.18$$
$$P(Weather = snow) = 0.02$$
$$Then\ P(Weather) = \{0.6, 0.2, 0.18, 0.02\}$$

- $\boldsymbol{P}(\boldsymbol{Weather})$ is called a **probability distribution** for the random variable $\boldsymbol{P}(\boldsymbol{Weather})$ 被称为随机变量的**概率分布**

Expected value 期望值

- Random variable X with n values $x_1, x_2, ..., x_n$ and distribution $(p_1, p_2, ..., p_n)$
  随机变量 $X$with n值 $x_1, x_2, ..., x_n$和分布 $(p_1, p_2, ..., p_n)$
- Function U of X
  函数 U对于 X
- The **expected value** of U after doing A is
  在完成 A之后，U的**预期值**是

$$E[U] = \sum_{i=1,...,n} p_i U(x_i)$$

Joint Probability Distribution (JPD) 联合概率分布

- If $X_1, ..., X_n$ are random variables,

  $$P(X_1, ..., X_n)$$

  denotes their **joint probability distribution (JPD)**, an n-dimensional matrix specifying the probability of every possible combination of values for $X_1, ..., X_n$

  > 即多个随机变量的概率分布组合成一个概率分布

- All relevant probabilities about a vector $\{X_1, ..., X_n\}$ of random variables can becomputed from $P(X_1, ..., X_n)$
  关于随机变量向量 X_1， ...， X_n的所有相关概率都可以从 $P(X_1$， ...， $X_n)$中计算出来

  > 单个随机变量中各个事件概率和仍为1，矩阵中的概率要同时满足两个轴

- A JPD $P(X_1, ..., X_n)$ **provides complete information** about the probabilities of its random variables.
  联合概率分布 $P(X_1, ..., X_n)$提供关于其随机变量概率的完整信息。

- EXP

	Sky=sunny	Sky=cloudy	Sky=rain	Sky=snow	P(Wind)
W	0.3	0.15	0.17	0.01	0.63
-W	0.3	0.05	0.01	0.01	0.37

	Sky=sunny	Sky=cloudy	Sky=rain	Sky=snow	P(Wind)
P(Sky)	0.6	0.20	0.18	0.02	1.00

	$Toothache$	$\neg Toothache$
$Cavity$	0.04	0.06
$\neg Cavity$	0.01	0.89

- Limitation of Joint Probability Distribution 联合概率分布的局限
  - However, JPD's are often **hard to create** (incomplete knowledge of the domain).
    然而，联合概率分布通常很难创建(不完整的领域知识)。
  - Even when available, JPD **tables are very expensive**, or **impossible**, to store because of **their size**.
    即使在可用的情况下，由于联合概率分布表的大小，存储它们也是非常昂贵的，甚至是不可能的。
  - A JPD table for n random variables, each ranging over k distinct values, has k^n entries!
    用于 n随机变量的 联合概率分布表(每个变量的范围都超过 k不同的值)具有 k^n条目!
  - A better approach is to come up with conditional probabilities as needed and compute the others from them.
    一个更好的方法是根据需要提出条件概率，然后从中计算其他概率。

Bayes rule and conditional independence 贝叶斯规则和条件独立

Bayes Rule 贝叶斯规则

$$P(A \cap B) = P(A \mid B)P(B) = P(B \mid A)P(A)$$

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)}$$

- 通常是知道在B发生的情况下A发生的概率有多少，反过来难求
- 两个事件发生的各自概率也能被很好的统计

Bayes rule – another version 贝叶斯规则-另一个版本

- $$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} = \frac{P(B \mid A)P(A)}{P(A)P(B \mid A) + P(\neg A)P(B \mid \neg A)}$$
- $$P(B) = P(A)P(B \mid A) + P(\neg A)P(B \mid \neg A)$$
- 不需要知道$P(B)$

- EXP
  12% of the men and 4% of the women are taller than6 feet. Furthermore, 20% of the students in the classare women. Suppose that a randomly selected student is taller than 6 feet.Find the probability p that the student is a woman.

$$P(woman \mid tall) = \frac{P(tall \mid woman)P(woman)}{P(tall)} = \frac{P(tall \mid woman)P(woman)}{P(tall \mid woman)P(man) + P(tall \mid woman)P(man)}$$

$$= \frac{0.04 \times 0.2}{0.104} = 0.0769$$

Bayes theorem application 贝叶斯定理的应用

- Bayes Theorem has found numerous applications in manyfields, including Computer Science
  贝叶斯定理在许多领域都有广泛的应用，包括计算机科学
  - Bayesian Networks 贝叶斯网络
  - Bayesian Classifiers 贝叶斯分类机
  - spam filtering, web page classification (e.g. Yahoo□style hierarchies), object classification, etc.
    垃圾邮件过滤、网页分类(如 Yahoostyle 层次结构)、对象分类等。
- **Bayesian Machine Learning**: Bayesian Inference / Bayesian Decision Theory
  贝叶斯机器学习: 贝叶斯推断/贝叶斯决策理论

Conditional independence 条件独立性

- Two random variables A and B are (absolutely) independent if
  两个随机变量 A 和 B 是(绝对)独立的，如果

$$P(A, B) = P(A)P(B)$$

- Using product rule for A & B independent, we can show
  使用独立于 A & B 的乘积规则，我们可以知道

$$P(A, B) = P(A \mid B)P(B) = P(A)P(B)$$

$$Therefore\ P(A \mid B) = P(A)$$

- If n Boolean variables are independent, the full JPD is:
  如果 n 个布尔变量是独立的，则完整的 JPD 是:

$$P(X_1, ..., X_2) = \prod_i P(X_i)$$

  Full joint is generally specified by $2^n - 1$ numbers, but when independent only n numbers are needed.
  完全连接通常由 $2^n - 1$数字指定，但是当独立时只需要 n数字。

- Absolute independence is a very strong requirement, seldom met 绝对独立是一个非常强烈的要求，很少得到满足

- Conditional Independence - expressed as:

$$P(A \mid B, C) = P(A \mid C)$$

The chain rule for JPD JPD的链式法则

$$\begin{gathered} P(X_1, ..., X_n) \\ = P(X_1, ..., X_{n-1})P(N_n \mid X_1, ..., X_{n-1}) \\ = P(X_1, ..., X_{n-2})P(X_{n-1} \mid X_1, ..., X_{n-2})P(Xn \mid X_1, ..., X_{n-1}) \\ . \\ . \\ . \\ = \prod_i^n P(X_i \mid X_1, ..., X_{i-1}) \end{gathered}$$

Bayes (Belief) Network 贝叶斯(信念)网络

- Bayesian Networks are a successful example of probabilistic systems that exploit conditional independence to reason efficiently under uncertainty.
  贝叶斯网络是一个成功的例子，它利用条件独立系统在不确定情况下有效地进行推理。

- A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions.
  一个简单的，图形化的符号用于条件独立断言，因此也用于完整联合分布的紧凑规范。

- Syntax: 句法:

  - a set of nodes, one per random variable
    一组节点，每个随机变量一个
  - links mean parent "directly influences" child
    链接意味着父母"直接影响"孩子
  - a directed acyclic graph
    有向无环图
  - a conditional distribution (a table) for each node given its parents $P(X_i \mid parents(X_i))$
    给定其父节点 $P(X_i \mid 父节点(X_i))$的每个节点的条件分布(表)

- In the simplest case, conditional distribution represented as a conditional probability table (CPT)
  在最简单的情况下，条件分布表示为一个条件概率表(CPT)

A two node network & conditional probability 双节点网络及条件概率

- Node$A$is independent of Node B, so it is described by an unconditional probability $P(A)$
- $P(\neg A)$ is given by $1 - P(A)$
- Node B is conditionally dependent on A. It is described by four numbers, $P(B \mid \neg A), P(B \mid \neg A)$, $P(\neg B \mid A)$and $P(\neg B \mid \neg A)$.
- This can be expressed as 2 by 2 **conditional probability table (CPT)**.
- But$P(\neg B \mid A) = 1 = P(B \mid A)$ and $P(\neg B \mid \neg A) = 1 - P(B \mid \neg A)$.
- Therefore, only **two** independent numbers in CPT.

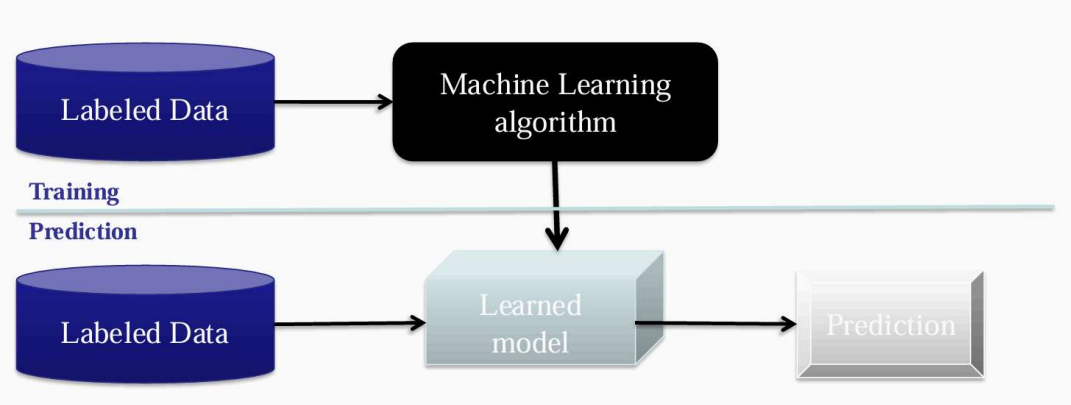
Part 3: Machine learning basics 第3部分: 机器学习基础

- Definition of Learning
- Three Types of Machine Learning
- Supervised Learning: Decision Trees

- Linear and non-linear classification methods

Machine Learning Basics 机器学习基础

- Artificial Intelligence is a scientific field concerned with the development of algorithms that allow computers to learn without being explicitly programmed
  人工智能是一门研究算法的科学，这种算法使得计算机不需要明确的编程就能学习
- Machine Learning is a branch of Artificial Intelligence, which focuses on methods that learn from data and make predictions on unseen data
  机器学习是人工智能的一个分支，主要研究从数据中学习和对未知数据进行预测的方法



Learning 学习

- Definition: "learning is a goal-directed process of a system that **improves the knowledge** or the **knowledge representation** of the system by exploring **experience** and prior **knowledge**"
  定义: "学习是一个以目标为导向的系统过程，它通过探索经验和先验知识来提高系统的知识或知识表示。"
- Acquisition of **new declarative knowledge**
  获取新的陈述性知识
- Development of motor and cognitive skills through **instruction** and **practice**
  通过指导和练习发展运动和认知技能
- Organization of new knowledge into general effective representation
  将新知识组织成一般有效的表示形式
- Discovery of new facts and theories through **observation** and **experimentation** 通过观察和实验发现新的事实和理论

Forms of Learning 学习形式

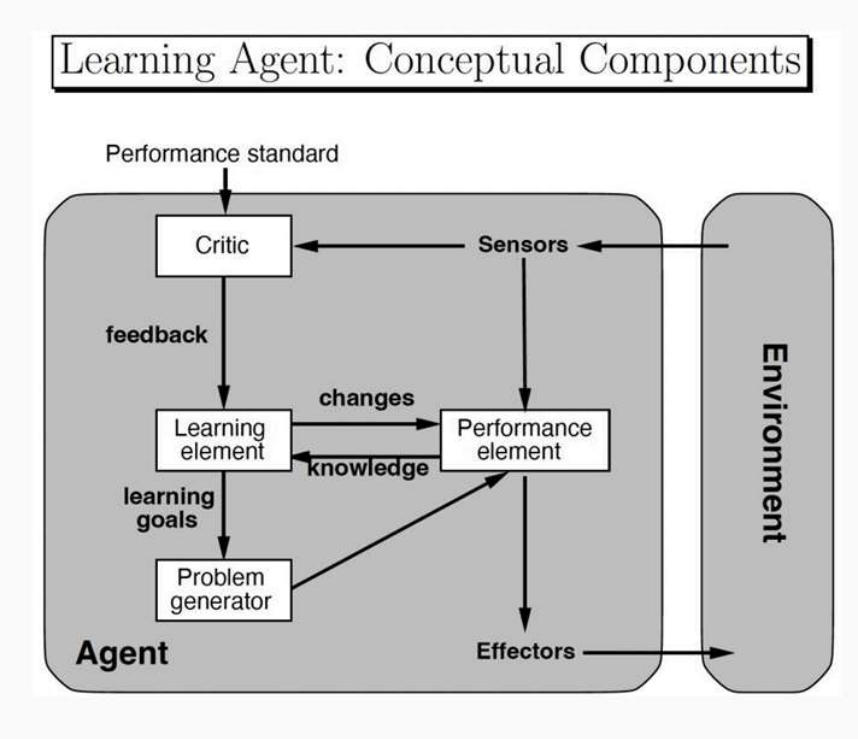
Any component of an agent can be improved by learning from data. The improvements, and the techniques used to make them, depend on four major factors:
代理的任何组件都可以通过从数据中学习来改进。这些改进以及制造它们的技术，取决于四个主要因素:

- **component**
- **prior knowledge**
- **representation**
- **feedback**

Components 组成

Representation and prior knowledge 表征和先验知识

- We have seen several examples of representations for agent components: Propositional and first-order logical sentences for the components in a logical agent;
  我们已经看到了代理组件表示的几个例子: 逻辑代理中组件的命题和一阶逻辑句;
- Bayesian networks for the inferential components of a decision-theoretic agent, and so on.
  贝叶斯网络用于推断分量的决策理论代理，等等。
- Markov Chain and Hidden Markov Models
  马尔可夫链与隐马尔可夫模型
- We say that learning a (possibly incorrect) general function or rule from specific input–output pairs is called inductive learning (more about this later).
  我们说从特定的输入输出对中学习一个(可能不正确的)一般函数或规则叫做归纳学习(稍后详述)。



Three Types of Machine Learning 机器学习的三种类型

Feedback to learn from 需要学习的反馈

- Three types of feedback that determine the three main types of learning:
  - Unsupervised learning: the agent learns patterns in the input even though no explicit feedback is supplied.
    非监督式学习: 即使没有提供明确的反馈，代理也会在输入中学习模式。（聚类 clustering）

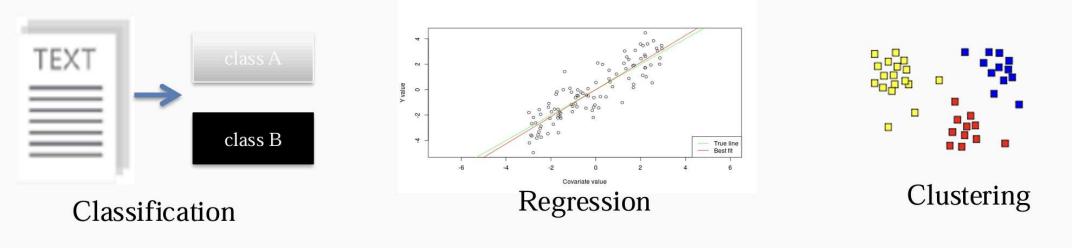
    > Unsupervised learning categories and techniques
    >
    > - Clustering
    >   - ok-means clustering
    >   - Mean-shift clustering Spectral clustering
    > - Density estimation
    >   - Gaussian mixture model (GMM)
    >   - Graphical models
    > - Dimensionality reduction
    >   - Principal component analysis (PCA)
    >   - Factor analysis

  - Supervised learning: the agent **observes** some example inputoutput pairs and learns a function that maps from input to output.
    监督式学习: 代理观察一些示例输入输出对，并学习一个从输入到输出的映射函数。

    > Supervised learning categories and techniques
    >
    > - Numerical classifier functions
    >   - Linear classifier, perceptron, logistic regression, support vector machines (SVM), neural networks
    > - Parametric (probabilistic) functions
    >   - Naïve Bayes, Gaussian discriminant analysis (GDA), hidden Markov models (HMM), probabilistic graphical models
    > - Non-parametric (instance-based) functions
    >   - k-nearest neighbors, kernel regression, kernel density estimation, local regression
    > - Symbolic functions
    >   - Decision trees, classification and regression trees (CART)

- Reinforcement learning: the agent learns from a series of reinforcements—rewards or punishments.
  强化学习: 代理人从一系列的增援中学习-奖励或惩罚。

- Summary of Machine Learning Types 机器学习类型综述
  - Supervised: learning with labeled data 监督: 使用标记数据学习
  - Unsupervised: discover patterns in unlabeled data 无监督: 在未标记的数据中发现模式
  - Reinforcement learning: learn to act based on feedback/reward 强化学习: 学会根据反馈/回报行事



Supervised Learning: Decision Trees 监督式学习: 决策树

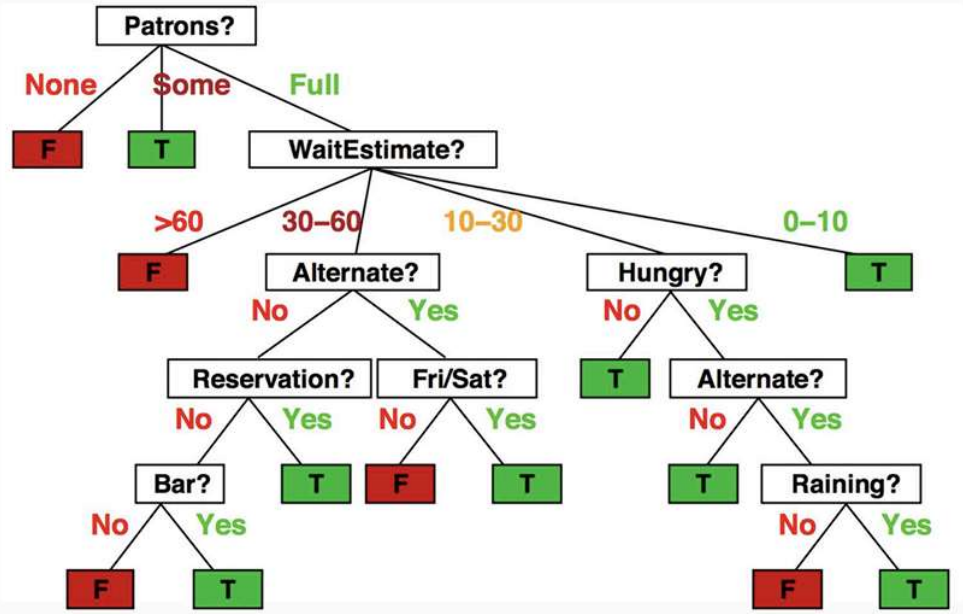
- A simple yet effective form of learning from examples
  一种简单而有效的从实例中学习的形式
- is a function that:
  - maps objects with a certain set of discrete attributes to discrete values based on the values of those attributes
    将具有某组离散属性的对象映射为基于这些属性的值的离散值
- It is representable as a tree in which
  它可以表示为一棵树，其中
  - every non-leaf node corresponds to a test on the value of one of the attributes
    每个非叶节点对应于一个属性值的测试
  - every leaf node specifies the value to be returned if that leaf is reached
    每个叶子节点指定到达该叶子时要返回的值
- A decision tree based on attributes $A_1, ..., A_n$ acts as classifier for objects that have those attributes
  基于属性 A_1， ...， A_n的决策树充当具有这些属性的对象的分类器

Decision Trees 决策树

- Decision trees make predictions by recursively splitting on different attributes according to a tree structure.
  决策树通过根据树结构对不同的属性进行递归分裂来进行预测。
- A decision tree with Boolean output defines a logical predicate 具有布尔输出的决策树定义了逻辑谓词

1.	Alternate: whether there is a suitable alternative restaurant nearby.
2.	Bar: whether the restaurant has a comfortable bar area to wait in.
3.	Fri/Sat: true on Fridays and Saturdays.
4.	Hungry: whether we are hungry.
5.	Patrons: how many people are in the restaurant (values are None, Some, and Full).
6.	Price: the restaurant's price range ($, $$, $$$).
7.	Raining: whether it is raining outside.
8.	Reservation: whether we made a reservation.
9.	Type: the kind of restaurant (French, Italian, Thai or Burger).
10.	WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$



Some terminology 一些术语

- The **goal predicate** is the predicate to be implemented by a decision tree.
  目标谓词是由决策树实现的谓词。
- The **training set** is the set of examples used to build the tree.
  训练集是用于构建树的示例集。
- A member of the training set is a **positive example** if it satisfies the goal predicate, it is a **negative example** if it does not.
  如果训练集的成员满足目标谓词，那么它就是一个正面例子; 如果不满足目标谓词，那么它就是一个负面例子。

A Good Decision Tree 一个好的决策树

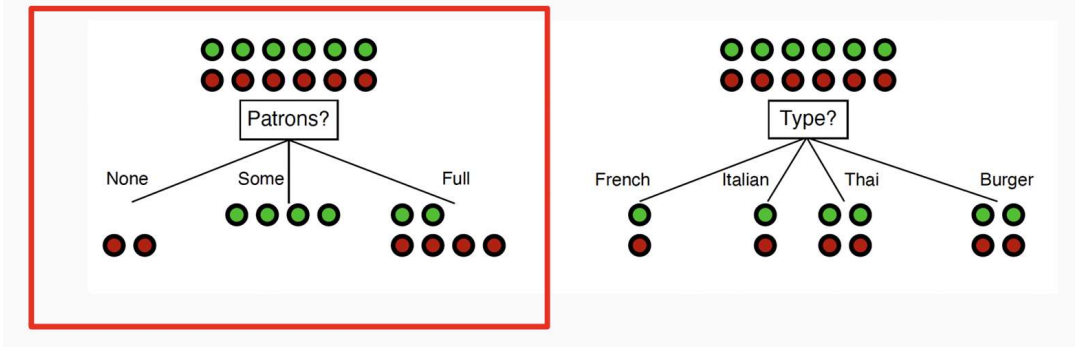
- extrapolates a **common pattern** from the examples
  从例子中推断出一个公共模式
- **correctly classifies all possible examples**, not just those in the training set
  正确分类所有可能的例子，而不仅仅是那些在训练集

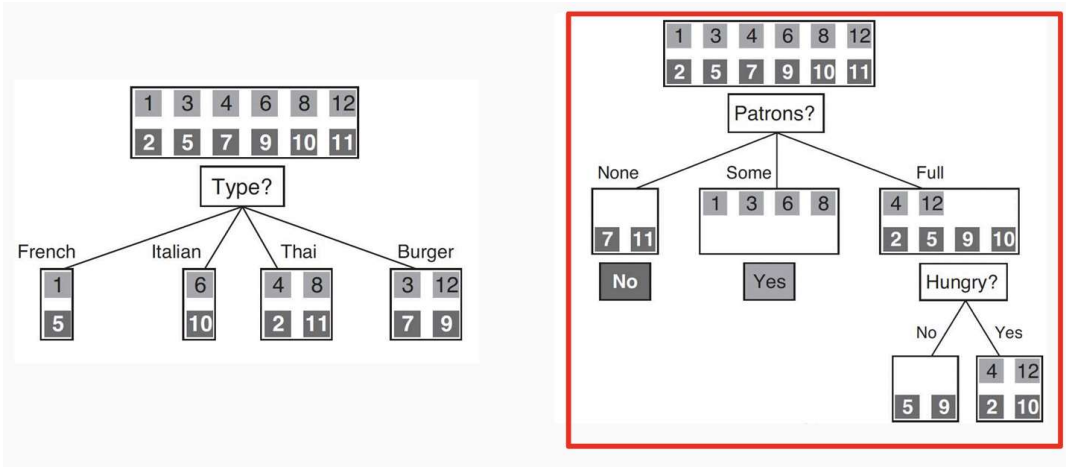
Choosing an attribute 选择属性

patrons is a better choice: it gives more information about the classification
patrons是一个更好的选择: 它提供了更多关于分类的信息 （训练数据即patrons）

Prefer more informative attributes leads to smaller trees 更喜欢信息更丰富的属性会导致更小的树

Main Idea: start building the tree by testing at its root an attribute that better splits the training set into homogeneous classes
主要思想: 通过测试一个能够更好地将训练集划分为同构类的属性，开始构建树





Choosing the best attribute 选择最好的属性

- 需要解决什么
  - What do we exactly mean by "best partitions the training set into homogeneous classes?"
    我们究竟是什么意思"最佳分区的训练集到同质类?"
  - What if every attribute splits the training set into non-homogeneous classes?
    如果每个属性都将训练集划分为非同构类会怎样?
  - Which one is better?
    哪个更好?
- 解决方法
  - **Information Theory** can help us **choosing**
    信息论可以帮助我们选择

Information theory 信息论

- Studies the mathematical laws governing systems designed to **communicate** or **manipulate** information.
  研究用于交流或操纵信息的系统的数学规律。
- It defines **quantitative** measures of information and the capacity of various systems to **transmit**, **store**, and **process** information.
  它定义了信息的定量度量以及各种系统传输、存储和处理信息的能力。
- it measures **the information content**, or **entropy**, of **messages/events**.
  它度量消息/事件的信息内容或熵。
- Information is measured in **bits**.
  信息是以位来衡量的。
- One bit represents the information we need to answer a yes/no question when we have no idea about the answer.
  一个位表示当我们不知道答案时回答是或否问题所需要的信息。

Information Content / entropy 信息内容/熵

If an event has n possible outcomes ($X = i$), each with prior probability $P(X = i)$, the **information content or entropy** H of the event's actual outcome is
如果一个事件有 n 个可能的结果 ($X = i$)，每个都有先验概率 $P(X = i)$，那么该事件实际结果的信息含量或熵 H 是

$$\mathrm{H}(X) = -\sum_{i=1}^{n} P(X = i) \log_2 P(X = i)$$

i.e., the average information content $-\log_2 P(X = i)$ of each possible outcome $X = i$ weighted by the outcome's probability
即，每个可能结果的平均信息内容 $-log_2 P(X = i)$由结果的概率加权

！！！熵越高数据分布和普适性越好，越有利于训练！！！

- Entropy is a measure of **disorder or uncertainty** 熵是对无序或不确定性的度量
- a measure of "Expected surprise"
  "意料之中的惊喜"的衡量标准
- The goal of machine learning model in general is to reduce uncertainty.
  机器学习模型的总体目标是降低不确定性。
- Measured in **bits**. 用位来衡量

"**Low Entropy**" 低熵

- Distribution of variable has many peaks and valleys
  变量的分布有多个峰谷
- Histograms has many low and highs
  直方图有许多低点和高点
- Value sampled are more predictable (low disorder/high level of purity)
  取样的值更可预测(低无序/高纯度)

"**High Entropy**" 高熵

- Variable has uniform like distribution
  变量具有均匀似分布
- Flat histogram
  平直直方图
- Value sampled are less predictable (high disorder/low level of purity)
  取样的数值不易预测(高无序度/低纯度)

Entropy Formula 熵公式

- entropy

$$\mathrm{H}(X) = -\sum_{i=1}^{n} P(X = i) \log_2 P(X = i)$$

- Conditional Entropy 条件熵

$$H(Y \mid X) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(y \mid x)$$

EXAMPLES：

- Entropy of fair coin toss 公平掷硬币的熵

$$H(P(h), P(t)) = H\left(\frac{1}{2}, \frac{1}{2}\right) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = \frac{1}{2} + \frac{1}{2} = 1 \text{ bit}$$

- Entropy of a loaded coin toss where $P(head) = 0.99$ 加载后抛硬币的熵，其中 $P(head) = 0.99$

$$H(P(h), P(t)) = H\left(\frac{99}{100}, \frac{1}{100}\right) = -0.99\log_2 0.99 - 0.01\log_2 0.01 \approx 0.08 \text{ bits}$$

- Entropy of a loaded coin toss with heads on both side 两边都有人头的加载硬币投掷的熵

$$H(P(h), P(t)) = H(1, 0) = -1\log_2 1 - 0\log_2 0 = 0 - 0 = 0 \text{ bits}$$

- What is the entropy of a group in which all examples belong to the same 一个群的熵是多少，其中所有的例子都属于同一个群

$$= H(1) = -1\log_2 1 = 0$$

  not a good training set for learning 不是很好的训练数据

- What is the entropy of a group in which all examples belong to the same class? 一个群中所有的例子都属于同一个类的熵是多少?

$$= H\left(\frac{1}{2}, \frac{1}{2}\right) = -0.5\log_2 0.5 - 0.5\log_2 0.5 = 1$$

  good training set for learning 良好的学习训练数据

Entropy of a decision tree 决策树的熵

- For decision trees, the event is question is whether the tree will return "yes" or "no" for a given input example e
  对于决策树，问题是对于给定的输入示例 e，树是否将返回" yes"或" no"
- Assume the training set E is a **representative sample** of the domain
  假设训练集 E 是域的一个代表性样本
- Then, the relative frequency of positive examples in E closely approximates the prior probability of a positive example
  然后，E 中正面例子的相对频率与正面例子的先验概率非常接近
- If E contains p positive examples and n negative examples, the probability distribution of answers by a correct decision tree is:
  如果 E 包含 p 正例子和 n 负例子，正确的决策树的答案概率分布是:

$$P(\text{yes}) = \frac{p}{p+n} \qquad P(\text{no}) = \frac{n}{p+n}$$

- Entropy of a correct decision tree: 正确决策树的熵:

$$H\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

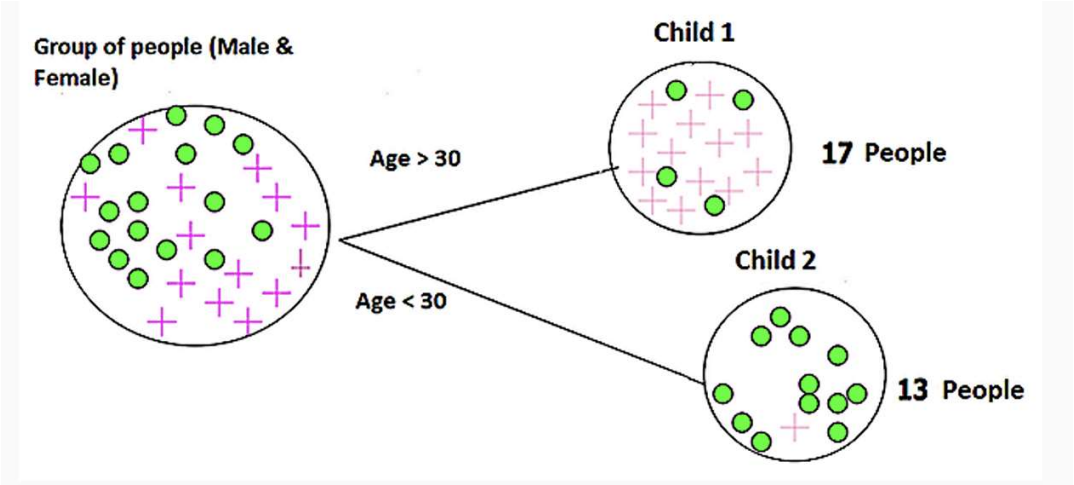
Information gain 信息增益

Measures the reduction in entropy or surprise by splitting a dataset according to a given value of a random variable.
测量按照一个随机变量的给定值将数据集分割后所引起的熵或意外减少程度。

$$I(X_n, Y) = H(Y) - H(Y \mid X_n)$$

n = number of splits N = 分割的次数

EXAMPLES



Find:

1. Entropy $H(People)$; $H(\text{People}) = -\left(\frac{14}{30}\log_2\frac{14}{30}\right) - \left(\frac{16}{30}\log_2\frac{16}{30}\right) = 0.996$
2. Entropy $H(Child1)$; $H(\text{Child}_1) = -\left(\frac{13}{17}\log_2\frac{13}{17}\right) - \left(\frac{4}{17}\log_2\frac{4}{17}\right) = 0.787$
3. Entropy $H(Child2)$; $H(\text{Child}_2) = -\left(\frac{1}{13}\log_2\frac{1}{13}\right) - \left(\frac{12}{13}\log_2\frac{12}{13}\right) = 0.391$
4. Information Gain I for 1) - 3). Weighted average entropy of children 儿童的加权平均数熵 $= -\left(\frac{17}{30}\cdot 0.787\right) - \left(\frac{13}{30}\cdot 0.391\right) = 0.615$
   Information Gain $I = 0.996 - 0.615 = 0.38$ for the split.

Decision Tree - Summary 决策树-总结

- At each level, one must choose: 在每个层次，必须做出一个选择
  - Which variable to split. 要拆分哪个变量。
  - Possibly where to split it. 可能在哪里分。
- Choose them based on how much information we would gain from the decision!
  根据我们能从决策中获得多少信息来选择它们!
  (choose attribute that gives the highest gain)!
  (选择获益最高的属性)！

Decision tree - limitations 决策树-局限性

- **Noise.** Two training examples may have identical values for all the attributes but be classified differently.
  噪声。两个训练例子可能对所有属性具有相同的值，但是分类不同。
- **Overfitting.** Irrelevant attributes may make spurious distinctions among training examples.
  过拟合。不相关的属性可能会在训练例子中造成虚假的区别。
- **Missing data.** The value of some attributes of some training examples may be missing.
  部分数据缺失。某些训练示例的某些属性的值可能缺少。
- **Multi-valued attributes.** The information gain of an attribute with many different values tends to be non-zero even when the attribute is irrelevant.
  多值属性。具有许多不同值的属性的信息增益往往是非零的，即使该属性是不相关的。
- **Continuous-valued attributes.** They must be discretized to be used.
  连续值属性。它们必须离散化才能使用。

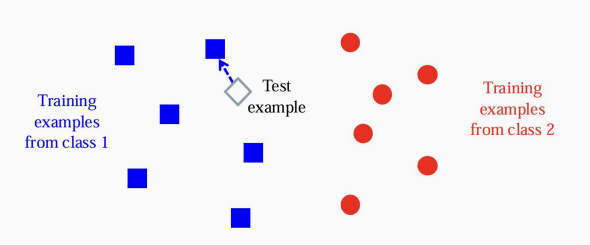
Linear and non-linear classification methods 线性和非线性分类方法

linear techniques 线性方法

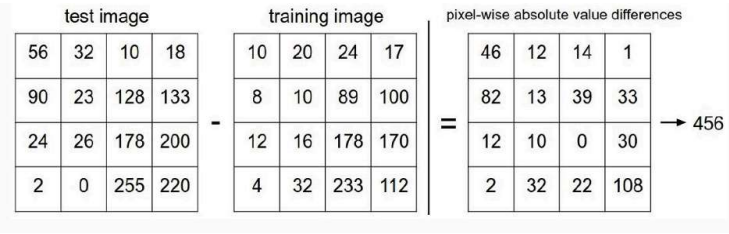
Nearest Neighbor Classifier 最近邻分类器

Nearest Neighbor – for each test data point, assign the class label of the nearest training data point
最近邻-对于每个测试数据点，分配最近训练数据点的类标签

- Adopt a distance function to find the nearest neighbor
  采用距离函数求最近邻
  - Calculate the distance to each data point in the training set, and assign the class of the nearest data point (minimum distance)
    计算到训练集中每个数据点的距离，并分配最近数据点的类(最小距离)
- It does not require learning a set of weights
  它不需要学习一组权重



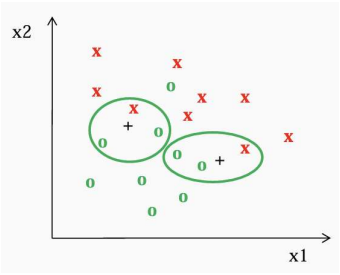
- For image classification, the distance between all pixels is calculated (e.g., using l_1 norm, or l_2 norm) 对于图像分类，计算所有像素之间的距离(例如，使用 l_1 标准或 l_2 标准)
- Disadvantages: 缺点:
  - The classifier **must remember** all training data and store it for future comparisons with the test data
    分类器必须记住所有的训练数据并存储它，以便将来与测试数据进行比较
  - Classifying a test image is **expensive** since it requires a comparison to all training images
    分类测试图像是昂贵的，因为它需要比较所有的训练图像



$$\ell_1 \text{ norm}$$
$$\text{(Manhattan distance)}$$
$$d_1(I_1, I_2) = \Sigma_p \, |I_1^p - I_2^p|$$

k-Nearest Neighbors Classifier k-近邻分类器

k-Nearest Neighbors approach considers multiple neighboring data points to classify a test data point
k近邻方法考虑多个相邻数据点对测试数据点进行分类



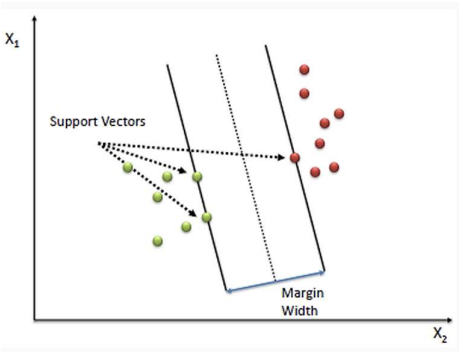
Linear Classifier 线性分类器

- Find a linear function f of the inputs x_i that separates the classes
  找到分隔类的输入的线性函数
- $f(x_i, W, b) = Wx_i + b$ W, b为参数
- Use pairs of inputs and labels to find the **weights matrix W** and the **bias vector b** The weights and biases are the **parameters** of the function f
  使用输入和标签对来寻找权重矩阵 W 和偏差向量 b 权重和偏差是函数 f 的参数
- Several methods have been used to find the optimal set of parameters of a linear classifier.
  A common method of choice is the **Perceptron algorithm**, where the parameters are updated until a minimal error is reached (single layer, does not use backpropagation)
  有几种方法已经被用来寻找线性分类器的最佳参数集。
  一个常见的选择方法是感知器算法，其中的参数被更新，直到达到最小的错误(单层，不使用反向传播)
- Linear classifier is a simple approach, but it is a building block of advanced classification algorithms, such as SVM and neural networks Earlier multi-layer neural networks were referred to as multi-layer perceptrons (MLPs)
  线性分类器是一种简单的方法，但它是先进的分类算法，如支持向量机和神经网络的一个组成部分。早期的多层神经网络被称为多层感知器(MLPs)
- The decision boundary is linear 决策边界是线性的
  - A straight line in 2D, a flat plane in 3D, a hyperplane in 3D and higher dimensional space
    二维的直线，三维的平面，三维的超平面和高维空间

Support Vector Machines 支持向量机

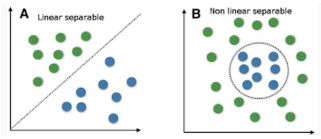
- How to find the best decision boundary?
  如何找到最好的决策边界?
  - All lines in the figure correctly separate the 2 classes
    图中的所有线条正确地将两个类分开
  - The line that is farthest from all training examples will have better generalization capabilities
    距离所有培训实例最远的线将具有更好的泛化能力
- SVM solves an optimization problem:
  SVM 解决了一个最佳化问题:
  - First, identify a decision boundary that correctly classifies the examples
    首先，确定一个正确分类示例的决策边界
  - Next, increase the geometric margin between the boundary and all examples
    接下来，增加边界和所有示例之间的几何边界
- The data points that define the **maximum margin width** are called **support vectors**
  定义最大边距宽度的数据点称为支持向量
- Find W and b by solving:

$$\min \frac{1}{2} \| w \|^2$$
$$\text{s.t. } y_i(w \cdot x_i + b) \geq 1, \quad \forall x_i$$



Linear vs Non-linear Techniques

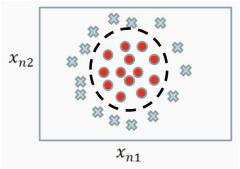
- techniques
  - Linear classification techniques 线性分类方法
    - Linear classifier 线性分类器
    - Perceptron 感知机
    - Logistic regression 逻辑回归
    - Linear SVM 线性支持向量机
    - Naïve Bayes 贝叶斯机
  - Non-linear classification techniques 非线性分类方法
    - k-nearest neighbors K-近邻
    - Non-linear SVM 非线性支持向量机
    - Neural networks 神经网络
    - Decision trees 决策树
    - Random forest 随机森林
- compare
  - For some tasks, input data can be linearly separable, and linear classifiers can be suitably applied
    对于某些任务，输入数据可以线性分离，适当应用线性分类器
  - For other tasks, linear classifiers may have difficulties to produce adequate decision boundaries
    对于其他任务，线性分类器可能难以产生足够的决策边界



Non-linear Techniques 非线性方法

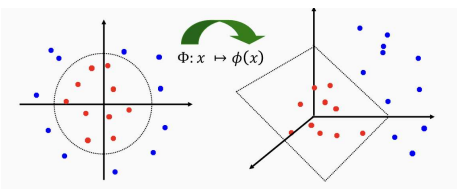
Non-linear classification 非线性分类

- Features Z_i are obtained as **non-linear functions** of the inputs X_i
  特征 Z_i作为输入 X_i的 **非线性函数** 获得
- It results in non-linear decision boundaries
  它导致非线性决策边界
- Can deal with non-linearly separable data
  可以处理非线性可分数据



Non-linear Support Vector Machines 非线性支持向量机

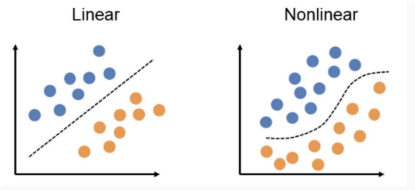
- The original input space is mapped to a higher-dimensional feature space where the training set is linearly separable
  将原始输入空间映射到训练集线性可分的高维特征空间
- Define a non-linear kernel function to calculate a non-linear decision boundary in the original feature space
  定义一个非线性核函数来计算原始特征空间中的非线性决策边界



Binary vs Multi-class Classification 二分类与多分类

- A classification problem with only 2 classes is referred to as binary classification. The output labels are 0 or 1.
  只有两个类的分类问题称为二进制分类，输出标签为0或1。

- A problem with 3 or more classes is referred to as multi-class classification
  有3个或更多类的问题称为多类分类
- Both the binary and multi-class classification problems can be linearly or nonlinearly separated
  二进制和多类分类问题都可以线性或非线性分离



No-Free-Lunch Theorem 不免费午餐定理

- The derived classification models for supervised learning are simplifications of the reality
  衍生出来的监督式学习分类模型是对现实的简化
  - The simplifications are based on certain assumptions.
    这些简化是基于某些假设。
  - The assumptions fail in some situations.
    这些假设在某些情况下会失败。
- In summary, No-Free-Lunch Theorem states:
  总而言之，"没有免费午餐"定理指出:
  - No single classifier works the best for all possible problems
    对于所有可能的问题，没有一个分类器是最好的
  - Since we need to make assumptions to generalize
    因为我们需要做一些假设来推广

Week 2: Deep Learning & Reinforcement Learning 第二周: 深度学习与强化学习

- Part 1: Deep Learning
  - Introduction to deep learning
  - Elements of neural networks and activation functions
  - Training NNs
  - Gradient descent
  - Regularization methods
  - NN architectures
- Part 2: Reinforcement Learning
  - Introduction to Reinforcement Learning
  - Markov Decision Processes (MDPs)
  - RL Techniques: From Q-learning to Actor-Critic
  - Applications of RL

Part 1: Deep Learning 第一部分: 深度学习

- Introduction to deep learning
- Elements of neural networks and activation functions
- Training NNs
- Gradient descent
- Regularization methods
- NN architectures

Introduction to deep learning 深度学习入门

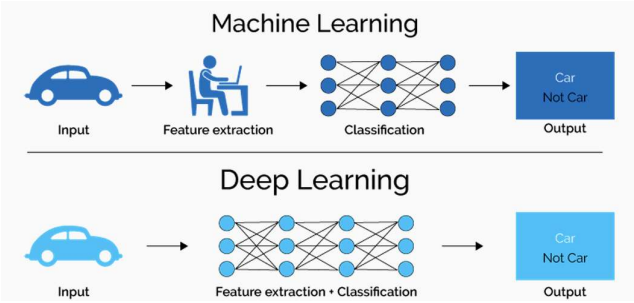
ML vs. Deep Learning 机器学习与深度学习

- Conventional machine learning methods **rely on human-designed feature representations**
  传统的机器学习方法依赖于人工设计的特征表示

- ML becomes just optimizing weights to best make a final prediction □
  机器学习只是优化权重，以最好地做出最终预测

- Deep learning (DL) is a machine learning subfield that uses multiple layers for learning data representations
  深度学习(DL)是一个机器学习子领域，它使用多个层次来学习数据表示

  - DL is exceptionally effective at **learning patterns**
    DL 在学习模式方面特别有效



- DL applies a multi-layer process for learning rich hierarchical features (i.e., data representations)
  DL 应用多层过程来学习丰富的层次特性(即数据表示)

  - Input image pixels → Edges → Textures → Parts → Objects

Why is DL Useful?

- DL provides a flexible, learnable framework for representing visual, text, linguistic information
  DL 为表示视觉、文本和语言信息提供了一个灵活的、可学习的框架

- Can learn in supervised and unsupervised manner
  可以在有监督和无监督的情况下学习

- an effective end-to-end learning system
  有效的端到端学习系统

- Requires large amounts of training data
  需要大量的训练数据

- Since about 2010, DL has outperformed other ML techniques
  自2010年以来，DL 已经超越了其他机器学习技术
  First in vision and speech, then NLP, and other applications
  首先是视觉和语言，然后是自然语言处理和其他应用

A biological neuron vs. artificial neuron 生物神经元对比人工神经元

Brains advantages with respect to digital computers:
与数字计算机相比，大脑的优势

- Massively parallel 大规模并行处理

- Fault-tolerant 容错

- Reliable 可靠的

- Graceful degradation 优雅降级

Representational Power 表象性

NNs with at least one hidden layer are **universal approximators**
具有至少一个隐层的神经网络是通用逼近器

(具有至少一个隐藏层的神经网络是通用逼近器。具体来说，对于任何连续函数 $h(x)$ 和任意的小误差 $\epsilon > 0$，总存在一个只有一个隐藏层的神经网络 $f(x)$，使得对于所有 x，都满足 $| h(x) - f(x) | < \epsilon$。)

NN can approximate any arbitrary complex continuous function
神经网络可以逼近任意复杂的连续函数

NNs use nonlinear mapping of the inputs x to the outputs f(x) to compute complex decision boundaries
神经网络使用输入 x 到输出 f (x)的非线性映射来计算复杂的决策边界

- reason of use deeper NNs:

  - The fact that deep NNs work better is an empirical observation
    事实上，深层神经网络工作得更好是一个经验观察

  - Mathematically, deep NNs have the same representational power as a one-layer NN
    从数学上讲，深层神经网络具有与单层神经网络相同的表示能力

Introduction to Neural Networks 神经网络导论

Handwritten digit recognition (MNIST dataset)
手写数字识别(MNIST 数据集)

- The intensity of each pixel is considered an input element
  每个像素的强度被认为是一个输入元素
- Output is the class of the digit
  输出是数字的类

对于手写数字识别，输入为一个图片矩阵，输出为从1到0的数字的概率（Each dimension represents the confidence of a digit
每个维表示一个数字的置信度）

Elements of neural networks and activation functions 神经网络元素和激活函数

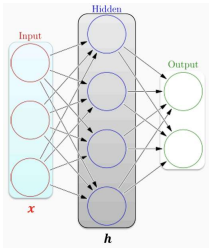
Elements of Neural Networks 神经网络要素

- NNs consist of hidden layers with neurons (i.e., computational units)
  神经网络由带有神经元(即计算单元)的隐层组成
- A single neuron maps a set of inputs into an output number, or $f: R^k \rightarrow R$
  单个神经元将一组输入映射到一个输出数字，即 $f: R^k \rightarrow R$
- neuron 神经元

$$z = a_1 w_1 + a_2 w_2 + \cdots + a_K w_K + b$$
$$a = \sigma(z)$$

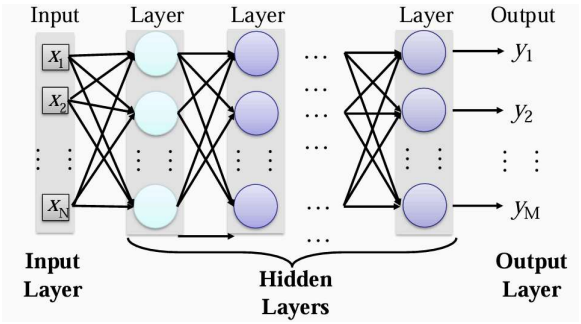
  a_i input 输入, w_k weights 权重, b bias 基础值, $\sigma(z)$ avtivation function 激活函数, a output输出.

- hidden layer（图片中间的层）



$hidden \ layer \ h=\sigma (W_1x+b_1)$

  - 图片中： 4 + 2 = 6 neurons (not counting inputs)
    [3 × 4] + [4 × 2] = 20 weights
    4 + 2 = 6 biases
    26 learnable parameters

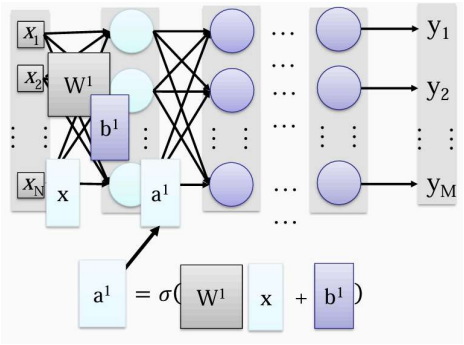
- Deep NNs have many **hidden layers**
  深层神经网络有许多隐藏层
  - **Fully-connected (dense)** layers (a.k.a. **Multi-Layer Perceptron or MLP**)
    完全连接(密集)层(又称多层感知器或 MLP)
  - Each neuron is connected to all neurons in the succeeding layer
    每个神经元连接到下一层的所有神经元
  - 

Matrix Operation 矩阵操作

- Matrix operations are helpful when working with multidimensional inputs and outputs
  矩阵运算在处理多维输入和输出时很有帮助
- $$\sigma(Wx + b) = a$$

$$\sigma\left(\begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0.98 \\ 0.12 \end{bmatrix}$$

- Multilayer NN, matrix calculations for the first layer
  多层神经网络，第一层的矩阵计算 Input vector x, weights matrix W^1, bias vector b^1, output vector a^1



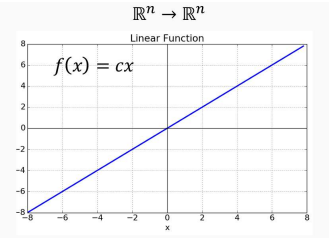
Activation Functions 激活函数

- **Non-linear activations** are needed to learn complex (non-linear) data representations
  学习复杂(非线性)数据表示需要非线性激活
  Otherwise, NNs would be just a linear function (such as $W_1 W_2 x = Wx$)
  否则，NN 将只是一个线性函数(例如 $W_1 W_2 x = Wx$)
- NNs with large number of layers (and neurons) can approximate more complex functions
  具有大量层(和神经元)的神经网络可以逼近更复杂的函数
  - Figure: more neurons improve representation (but, may overfit)
    图: 更多的神经元改善表征(但是，可能过度)

Activation: Linear Function 激活: 线性函数

- **Linear function** means that the output signal is proportional to the input signal to the neuron
  线性函数表示输出信号与神经元的输入信号成正比

$$f(x) = cx, \mathrm{R}^n \rightarrow \mathrm{R}^n$$



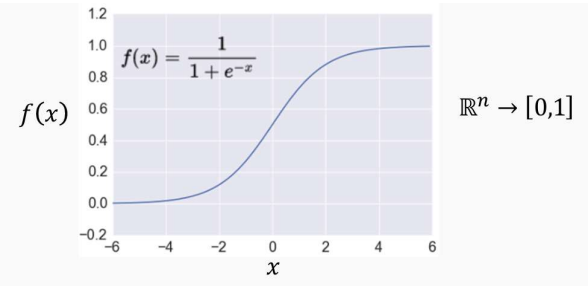
- - If the value of the constant c is 1, it is also called **identity activation function**
    如果常数 c 的值为1，它也被称为恒等式激活函数
  - This activation type is used in **regression problems**
    此激活类型用于回归问题

Activation: Sigmoid sigmoid函数

- **Sigmoid function** σ: takes a real-valued number and "squashes" it into the range between 0 and 1
  S形函数 σ: 取一个实值数，并将其"压缩"到0到1之间的范围内

$$f(x) = \frac{1}{1 + e^{-x}}, \mathrm{R}^n \rightarrow [0, 1]$$

  - The output can be interpreted as the firing rate of a biological neuron
    输出可以解释为生物神经元的放电速率
  - When the neuron's activation are 0 or 1, sigmoid neurons saturate 当神经元激活为0或1时，sigmoid神经元饱和
    - Gradients at these regions are almost zero (almost no signal will flow)
      这些区域的梯度几乎为零(几乎没有信号会流动)
  - Sigmoid activations are less common in modern NNs
    sigmoid激活在现代神经网络中不常见

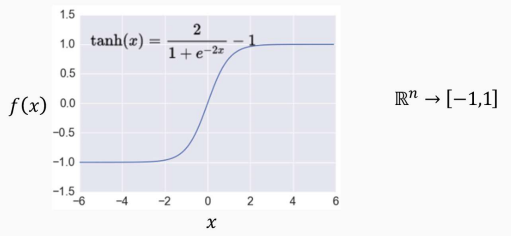


Activation: Tanh

- Tanh function: takes a real-valued number and “squashes” it into range between -1 and 1
  Tanh 函数: 获取一个实值数字，并将其“压缩”到 -1到1之间

$$\tanh(x) = \frac{2}{1+e^{-2x}} - 1, \mathbb{R}^n \rightarrow [-1, 1]$$

  - Like sigmoid, tanh neurons saturate 饱和
  - Unlike sigmoid, the output is zero-centered 与 sigmoid 不同，输出是以零为中心的
    - It is therefore preferred than sigmoid 比sigmoid更好
  - Tanh is a scaled sigmoid: $\tanh(x) = 2 \cdot \sigma(2x) - 1$



Activation: ReLU

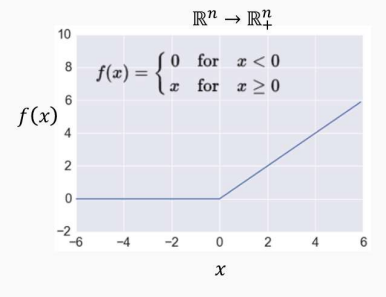
- ReLU (Rectified Linear Unit): takes a real-valued number and thresholds it at zero
  修正线性单位(ReLU) : 取一个实值数，阈值为零

$$f(x) = \max(0, x)$$

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

$$\mathbb{R}^n \rightarrow \mathbb{R}^n_+$$

- 应用
  - Most modern deep NNs use ReLU activations
    大多数现代深层神经网络使用 ReLU 激活
- 优势
  - ReLU is fast to compute (Compared to sigmoid, tanh ; Simply threshold a matrix at zero) ReLU 计算速度很快 (与 sigmoid 相比，tanh; 简单地阈值为零的矩阵)
  - Accelerates the convergence of gradient descent (Due to linear, non-saturating form) 加速梯度下降法的融合 (由于线性，非饱和形式)
  - Prevents the gradient vanishing problem
    防止渐变消失问题

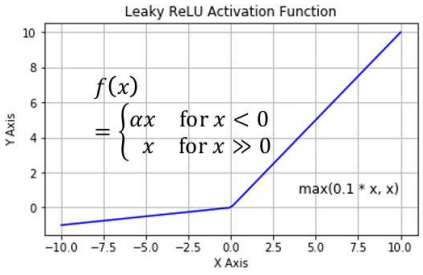


Activation: Leaky ReLU

- 普通ReLU的问题
  - The problem of ReLU activations: they can “die”
    ReLU 激活的问题: 它们可能“死亡”
  - ReLU could cause weights to update in a way that the gradients can become zero and the neuron will not activate again on any data
    ReLU 可能导致权重更新的方式，梯度可以成为零，神经元不会再次激活任何数据
- **Leaky ReLU** activation function is a variant of ReLU
  - Instead of the function being 0 when $x<0$, a leaky ReLU has a small negative slope (e.g., α = 0.01, or similar)
    当 x < 0时，函数不是0，而是一个小的负斜率(例如，α = 0.01，或类似)

$$f(x) = \begin{cases} ax & \text{for } x < 0 \\ x & \text{for } x \gg 0 \end{cases}$$

- 优势（解决问题）
  resolves the dying ReLU problem
- 应用
  - Most current works still use ReLU
    目前大部分的作品仍然使用 ReLU
  - With a proper setting of the learning rate, the problem of dying ReLU can be avoided
    通过合理设置学习速率，可以避免 RLU 死亡的问题



Activation: Softmax

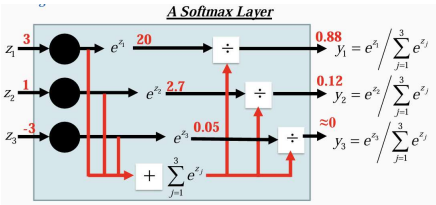
- The softmax layer applies softmax activations to output a probability value in the range [0, 1]
  Softmax 层应用 softmax 激活来输出范围[0,1]内的概率值
- 即将所有的输入规范成一个从0到1的概率值，且每一个概率的值加起来就是1
- The values z inputted to the softmax layer are referred to as **logits**
  输入到 softmax 层的值 z 称为 logits

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \qquad (\text{满足} \sum_{i=1}^{n} \sigma(z_i) = 1)$$

$\sigma(z_i)$ 表示第 i 个类别的Softmax输出。

z_i 是每个类别的输入值。

$\sum_{j=1}^{n} e^{z_j}$ 是所有类别输入值的指数和，用于归一化。



Training NNs 神经网络训练

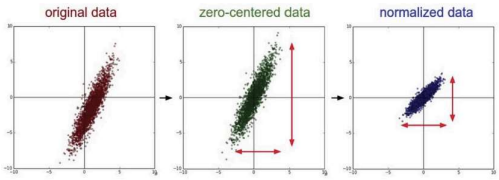
- The network **parameters** θ include the **weight matrices** and **bias vectors** from all layers
  网络 **参数** θ 包括来自所有层的 **权矩阵** 和 **偏向向量**

$$\theta = \{W^1, b^1, W^2, b^2, \ldots, W^L, b^L\}$$

  Often, the model parameters θ are referred to as weights
  通常，模型参数 θ 被称为权重

- Training a model **to learn a set of parameters** θ **that are optimal** (according to a criterion) is one of the greatest challenges in ML
  训练一个模型来学习一组最优的参数 θ (根据一个标准)是机器学习中最大的挑战之一
- **Data preprocessing -** helps **convergence** during training
  数据预处理-有助于在训练期间收敛
  - **Mean subtraction**, to obtain zero-centered data
    平均减法，得到零中心的数据
    - Subtract the mean for each individual data dimension (feature)
      减去每个独立数据维度(特性)的平均值
  - **Normalization** 规范化

    - Divide each feature by its standard deviation
      将每个特征按其标准差划分
        - To obtain standard deviation of 1 for each data dimension (feature)
          为每个数据维度(特征)取得1的标准差
    - Or, scale the data within the range [0,1] or [-1, 1]
      或者，在[0,1]或[-1,1]范围内缩放数据



- To train a NN, set the parameters θ such that for a training subset of images, the corresponding elements in the predicted output have maximum values
  为了训练神经网络，需要设置参数 θ，使得对于图像的训练子集，预测输出中的相应元素具有最大值
- Define a **loss function/objective function/cost function** $\mathrm{L}(\theta)$ that calculates the difference (error) between the model prediction and the true label
  定义一个损失函数/目标函数/成本函数 $\mathrm{L}(\theta)$，用于计算模型预测与真实标签之间的差异(误差)
- Find the optimal parameters θ^* that minimize the total loss $\mathrm{L}(\theta)$
  寻找最小化总损失 $\mathrm{L}(\theta)$ 的最佳参数 θ^*

> For a training set of N images, calculate the total loss overall all images:
> 对于 N 幅图像的训练集，计算所有图像的总损失: $$ \mathcal{L}(\theta) = \sum_{n=1}^N \mathcal{L}_n(\theta)
>
> $$

Loss Functions 损失函数

- Classification tasks 分类任务
    - Training examples 训练样本

$$\text{Pairs of } N \text{ inputs } x_i \text{ and ground-truth class labels } y_i$$
$$N \text{ 个输入 } x_i \text{ 与真实类别标签 } y_i \text{ 的配对}$$

    - Output Layer 输出层 Softmax Activations [maps to a probability distribution]
      Softmax 激活[映射到一个概率分布]

$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\top \mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\top \mathbf{w}_k}}$$

    - Loss function 损失函数 Cross-entropy 交叉熵

$$\mathrm{L}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} [y_k^{(i)} \log \hat{y}_k^{(i)} + (1 - y_k^{(i)}) \log (1 - \hat{y}_k^{(i)})]$$
$$\text{Ground-truth class labels （实际值） } y_i \text{ and model predicted class labels （模型预测值） } \hat{y}_i$$

- Regression tasks 回归任务
    - Training examples 训练样本

$$\text{Pairs of } N \text{ inputs } x_i \text{ and ground-truth output values } y_i$$
$$N \text{ 个输入 } x_i \text{ 与真实输出值 } y_i \text{ 的对}$$

    - Output Layer 输出层

$$\text{Linear (Identity) or Sigmoid Activation}$$
$$\text{线性或者sigmoid激活函数}$$

    - Loss function 损失函数
        - Mean Squared Error 均方误差

$$\mathrm{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2$$

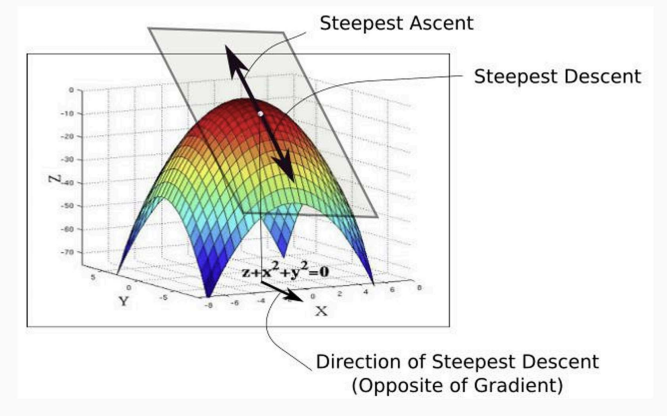
        - Mean Absolute Error 平均绝对误差

$$\mathrm{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left| y^{(i)} - \hat{y}^{(i)} \right|$$

Training NNs (2)

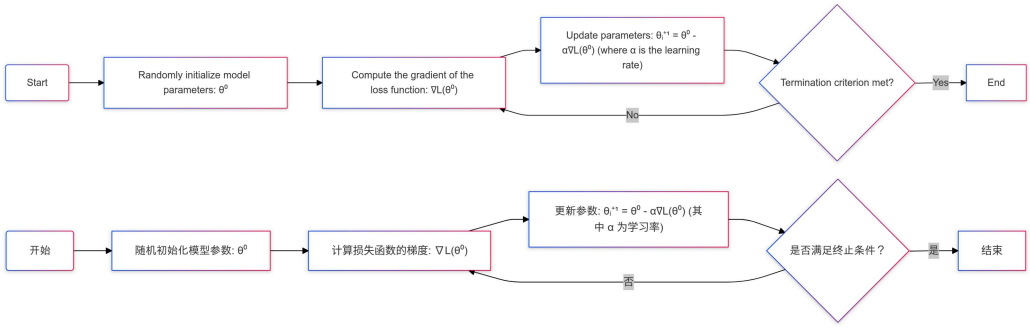
- Optimizing the loss function $\mathrm{L}(\theta)$ 优化损失函数

- **gradient descent (GD)** algorithm
  梯度下降法算法
  - GD applies iterative refinement of the network parameters θ
    GD 对网络参数θ进行迭代求精
  - GD uses the opposite direction of the **gradient** of the loss with respect to the NN parameters for updating θ ($\nabla \mathrm{L}(\theta) = [\frac{\partial \mathrm{L}}{\partial \theta_i}]$) GD 使用相对于神经网络参数的损失梯度的相反方向来更新 θ
  - The gradient of the loss function $\nabla \mathrm{L}(\theta)$ gives the direction of fastest increase of the loss function $\mathrm{L}(\theta)$ when the parameters θ are changed
    损失函数 $\nabla \mathrm{L}(\theta)$的梯度给出了当参数 θ改变时损失函数 $\mathrm{L}(\theta)$增长最快的方向



Gradient descent 梯度下降

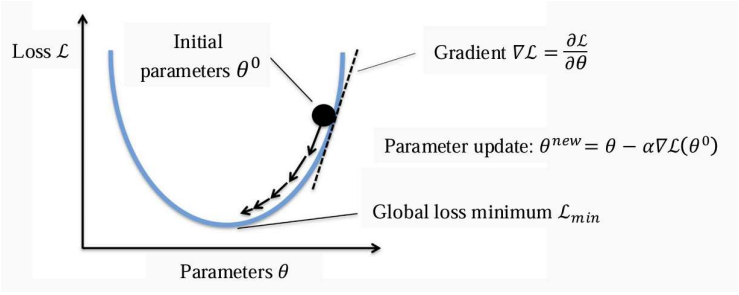
- Steps in the gradient descent algorithm:
  梯度下降法算法中的步骤:



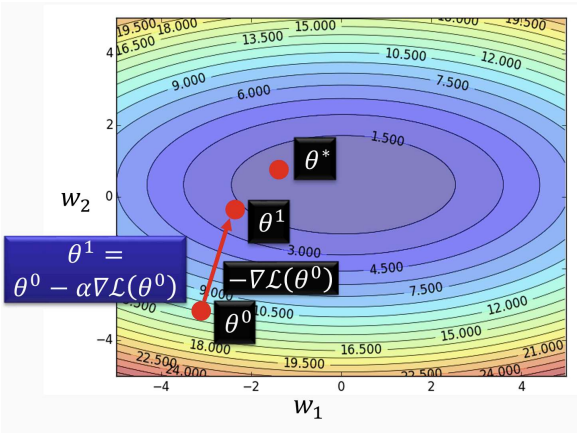
- 关键算法：

$$\theta^{\text{new}} = \theta^0 - \alpha \nabla \mathrm{L}(\theta^0)$$

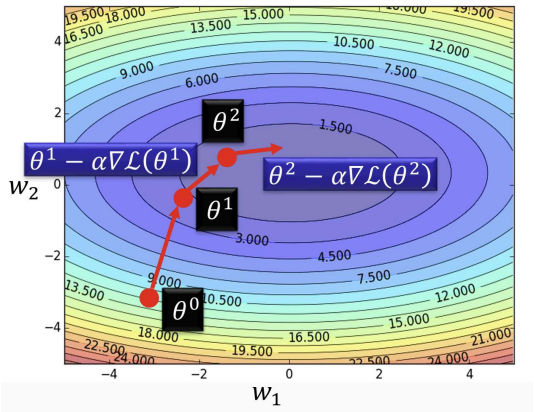
- 图解：



- Gradient Descent Algorithm 梯度下降算法



Eventually, we would reach a minimum …..



- Gradient descent algorithm stops when a local minimum of the loss surface is reached
  当损耗面达到局部最小时，梯度下降法算法停止
  - GD does not guarantee reaching a global minimum
    GD 并不能保证达到全球最低水平
  - However, empirical evidence suggests that GD works well for NNs
    但是，经验证据体现出梯度下降对于神经网络具有良好的效果