

1. 人工智能算法简介

1. 学习目标

1. 了解人工智能能做什么事？
2. 学习准备知识: Python编程基础，线性代数，概率基础知识。

2. 线性代数基本概念

矢量

数学上看，矢量空间的元素，或矢量空间中的点，就是**矢量**。更具体地看，有序的值序列就构成了一个矢量。一般用小写黑体字母表示，如

$$\mathbf{x} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} = (2, 3)^T \quad (1)$$

是二维空间中的矢量。我们默认都采用笛卡尔坐标系。又如

$$\mathbf{y} = \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \end{pmatrix} \quad (2)$$

是三维空间中的矢量。矢量的元素用下标标识，

$$x_1 = 2, x_2 = 3. \quad (3)$$

注意:

1. 举例: 我们所在的三维空间可以近似看作是一个三维矢量空间。取定某点为原点，那么空间中每一点就是该空间中的一个矢量。所有的矢量就构成了这个三维空间本身。

2. 在学习Python时，区分列表[2,3], 元组(2,3)和矢量

$$\mathbf{x} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

矢量的基本操作

矢量的加法：

$$\mathbf{x} + \mathbf{z} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} + \begin{pmatrix} 1 \\ -2 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \quad (4)$$

一般地，对于两个二维矢量求和，结果为对应坐标相加

$$\mathbf{a} + \mathbf{b} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \end{pmatrix} \quad (5)$$

注意：

1. 只有维度相同的两个矢量才可以相加；
2. 对于高维空间中的矢量的求和，可以类推。

矢量的数乘

已知常数 λ 和 n 维矢量 \mathbf{x} 。数乘定义为

$$\lambda \mathbf{x} = \lambda \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \lambda x_1 \\ \lambda x_2 \\ \vdots \\ \lambda x_n \end{pmatrix}. \quad (6)$$

例如，有矢量

$$\mathbf{x} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad (7)$$

数乘为

$$3\mathbf{x} = 3 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix}; \quad (8)$$

矢量为

$$\mathbf{y} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad (9)$$

$$0.707\mathbf{x} = 0.707 \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.707 \\ 0.707 \end{pmatrix}; \quad (10)$$

说明：

1. 当常数为绝对值大于 1 的数时，数乘将矢量**伸长**，当常数为绝对值小于 1 的数时，数乘将矢量缩短。
2. 常数可以是**实数**，也可以是复数。（本课程中，大家可以不要考虑常数为复数的情形）
3. 作业：常数为**-1**表示对矢量做什么操作？

矢量的长度

例如,对于

$$\mathbf{x} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \quad (11)$$

$$|\mathbf{x}| = \sqrt{3^2 + 4^2} = 5 \quad (12)$$

通常，**向量长度**取为各元素的平方和之平方根。

矢量的点积

两个矢量的点积，又叫标量积，定义为两个矢量的对应元素的乘积。例如，对于二维矢量 \mathbf{x} 和 \mathbf{y} ，其点积

$$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2. \quad (13)$$

举例，

$$\mathbf{x} \cdot \mathbf{y} = \begin{pmatrix} 4 \\ 5 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 4 \times 1 + 5 \times 0 = 4 \quad (14)$$

注意：

1. 矢量的点积的结果为一个数，而不是矢量。
2. 在**书写**矢量点积的表达式中，点积符号(圆点)不可省略：

$$\mathbf{x} \cdot \mathbf{y} \neq \mathbf{xy} \quad (15)$$

3. 举例：一个人有年龄，性别，身高，体重，职业，收入，婚否，母语，国籍，宗教信仰。这些特征中的每一个都可以看成是一个维度。每个维度上的信息，可以用数字来编码。最后，对于这个人我们可以得到十维空间中的一个矢量！

$$V_{\text{person}} = \begin{pmatrix} 24 \\ F \\ 170 \\ 60 \\ OL \\ 5.5 \\ N \\ Chinese \\ CN \\ Tao \end{pmatrix} = \begin{pmatrix} 24 \\ 0 \\ 170 \\ 60 \\ 3 \\ 5.5 \\ 0 \\ 1 \\ 3 \\ 0 \end{pmatrix} \quad (16)$$

4. 同理，对于网页上的一篇文章，交易数据，甚至任何一个事件，我们都可以将它表示成一个矢量。例如，利用bag-of_words技术，我们可以把两篇文章分别表示成矢量，通过计算这两个矢量之差，或者两个矢量的点积，我们就可以判断两篇文章的相似度，从而可以有助于我们进行文章的归类，鉴别剽窃与否等等。

###

如果你想学习人工智能算法，那么你的准备知识应该包括一些编程知识，线性代数和对概率的理解。

接下来，我们要给大家简要介绍人工智能的能做什么事情。人工智能的范围非常广泛,从人工智能的历史，搜索算法的建立，设计游戏,解决游戏难题，到限制条件问题都值得学习。机器学习算法是人工智能里的核心。人工智能可广泛应用在自然语言处理，机器人学，机器视觉，语音分析，量化交易等等领域。用Python语言编程来解决人工智能问题是一个值得学习的技术。下面分别介绍一下各种常见算法。

3. 人工智能能做什么？

最基本的算法就是搜索。有许多中搜索方法可以使用比如盲目搜索（暴力搜索，无脑搜索，uninformed search），提示性搜索(又叫**启发性搜索**)，对抗搜索(游戏)等。第二类话题就是马科夫决策过程和强化学习。它们有一系列的应用，如自然语言处理，机器人，机器视觉等。现在我们一一讨论人工智能里的各个话题。

先来看理性智能体。我们研究人工智能的目的是设计智能体，它们可以**感知**其环境并且**作用**到环境上，从而实现其目标或者任务。一个智能体可以视为一个函数 $F(x)$ ，该函数从感知到的环境映射到一个作用在环境上的动作。理性智能体，就是做正确的事情的智能体。何为正确的事呢？就是智能体的表现达到最优，即所谓性能度量(performance measure)最大化。人工智能(AI)在给定的计算条件下，使得性能度量达到最大化。这就是AI的目的。要使得性能度量最大，可以从硬件和软件两方面优化改进，我们这里只讨论软件方面。

搜索智能体可以帮助我们已知点出发找到目标点。典型的例子是走迷宫，从某个给定起点和终点，找出一条路线使得我们能从起点到达终点。智能体会思考为了达到目的该如何做。

智能体要做的就是定义出到达目标点的动作或动作序列（路径）。一条路径会有不同的代价和深度（此处指的是通过该路径找到的解在搜索树中的深度）。最常见搜索方法可分为有两大类。盲目搜索并不用某领域的知识，它包括的技术有广度优先搜索，深度优先搜索，均匀代价搜索等。启发式搜索运用了一些如何更快地到达目标的经验法则或启发式信息，这类搜索法包括贪婪搜索法，A*搜索法，等等。搜索算法的例子包括八皇后问题。八皇后问题是指，我们在64个格子的国际象棋棋盘上适当地放置8个皇后，使得它们横向，纵向，对角向都不“共线”。

这就是要从约百万亿种可能的状态中，搜索出满足以上约束条件的状态来。另一个典型的搜索算法的例子就是路线搜索。给定包含一些城市的地图，地图可以用图结构来表示：城市用结点表示，城市之间的可能的路线用线表



示。两座城市之间的距离标记在它们的连线上。假设你想要从北京到马尔代夫。要达到此目的，你有不同的路线可走。搜索智能体的目的就是为了解探索这些可能性，并找出最好的路线。不同的路线需要不同的花费，花费可以用这些连线的长度(比如，单位为千米的数) 这里，搜索智能体的角色其实就是找到从北京到马尔代夫的路线，同时找出最好的路线。

AI的另一个重要话题是对抗搜索，或称游戏。实际上，已经有一些游戏可以用AI解决。我们看到国际象棋，象棋以及智力问答竞赛等等。基本思想就是设计智能体来玩游戏，并且与对手对抗。这里面的重要概念包括**mini max 算法**, **alpha beta pruning** and **随机游戏**。这里略过。

另一个例子:

checkers 例如，Chinook (一种计算机程序)结束了40年人类ended 40 years reign of human world champion Marion Tinsley in 1994.

Checkers 已经是已经解决了的问题。在下棋这一领域，计算机(或者Chinook)能打败任何人类。假设human is also playing an optimal play。对抗搜索的另一个名字就是游戏(games)。对抗(adversarial)意味着存在一个我们无法控制的对手。

对比: 游戏和搜索

游戏也是一个搜索问题，但是最优解不是一个引向目标的动作序列，而是一个帮助我们赢得比赛的策略。也就是说，如果对手做A操作，那么我们就做甲操作，对手做B操作，我们就相应地施行乙操作,如果对手做C操作，我们就做丙，等等。这种策略可以写成规则，好比下棋时候的口诀，但是它非常地枯燥和冗长。

机器学习

机器学习是当今AI的必然趋势。根据机器学习的杰出专家Tom Riccio所说，机器学习是关于如何创建能从数据和观测中提高我们的经验的计算机程序。你想要能教计算机如何学习并如何提高经验。这是机器学习之核心。

机器学习主要可分为监督学习，非监督学习，聚类，降维和强化学习等。

当你有标签时，那么你在做的就是监督学习。这些标签可以是任意连续的数值，也可以是离散的值（如文本）。如果标签只能取两个值，那么我们就在讨论二进制分类。我们想要建立的是一个函数，给定一个输入集合或是实例的描述，得出输出集合。例如，假设我们有银行顾客的信息：他们的年龄，性别，职位，工资等等。对每个顾客，我们有一个标签：是否有信用卡。又如，电子邮件是否是垃圾邮件也属于标签的例子。这样例子还有很多。因此，我们有所谓的正例子和负例子。它们可以分别用1, 0 表示。如何找出分开这两类特征的边界往往是这类问题的目标。这是监督学习的典型例子。因为已经告诉了我们谁是正例子，谁是负例子了，所以称之为监督学习。监督学习包括分类，k近邻预测，神经网络，线性回归，boosting等等方法。

第二类学习，就是**非监督学习**(unsupervised)。非监督学习中，我们不必为实例做标签。也就是说，我们的数据集中的实例是没有标签的。例如，我们有人口或顾客的数据，却没有任何类型的标签与之相联系。因此，这里要解决的问题就是：给定这些数据点，我们能否找出这些实例的数据点的聚类(clusters)吗？我们要寻找一个函数F用以把输入集合X映射到聚类的集合。这完全是非监督的算法。关于非监督学习，我们有很多不同的方法来实现。最主要的为K均值聚类。

约束问题

约束问题，其实是搜索问题，它不关注搜索的路径，而只关心**目标**，例如解一个游戏。一般地，问题用**变量**来表示，而不是用态来表示。约束问题(constraint satisfaction problem)的典型例子是数独游戏(Sudoku)。例如，在一个9行9列的格子盘中，横向一排格子我们称为一行，纵向的一排格子称为一列。要达到的目标是各行的数字都只能取从1到9的数字，各列亦如此，且在9个3X3的小九格的数字也只能是从1到9的不重复的数字。你可以表示这

个问题为一个CSP问题。每一个格子可视为一个变量。加在这些变量上的约束条件就是所有3X3格点内，所有行，所有列中的数字都只能是从1到9的不重复的数字。问题的解就是去寻找为变量赋值的方法，以满足这些约束条件。

逻辑智能体

人工智能的又一个重要领域是逻辑智能体。在**逻辑智能体**中，逻辑被用来构建我们所在的世界的模型。很早以前，布尔发明了命题逻辑(propositional logic), 并且，一阶逻辑有一个固定的语法。通过句子的逻辑表达式，我们用符号和连接词(connectives)来为世界建立模型。关于逻辑智能体的实例，读者可参考Prolog相关文章。

强化学习

在强化学习中，我们设计智能体在随机或特定环境中演化。智能体从强化或延迟奖励中学习。它是除监督学习和非监督学习之外的又一类学习。它是一类用于在输出结果为随机值的决策问题中的学习方法,它包含了一个能连续计划，学习并影响其环境的智能体。强化学习的驱动力是最大化奖励。在对抗，博弈，游戏，投资，交易等等领域都可以应用。

自然语言

人工智能可以用于自然语言处理，在这个领域我们关注计算机和人类语言之间的相互作用，我们关注视觉，感知于图像处理，并致力于建立AI机器视觉。这里的任务就是从任务中提取信息。这样的任务包括操纵，谈判，模式识别等。

操纵物理世界

最后，机器人学关注如何让人工智能体操纵物理世界。比如，实现自动驾驶，识别人脸，机器人的自我运动等。

今日总结

AI是一个宏大的学科，因为它包含很多方面。包括其组成的复杂性，语言和划分等。AI也是一个广阔的学科，对人类自身及社会有极其重大的影响。

What AI can do for us and we have seen several application is just already amazing. AI system, remember, do not have to model human or nature. But can act like or be inspired by human or nature. How human think is actually beyond the scope of this course and it's another branch of AI. And finally the aspect rationale agents, **the agent that do the right thing, is the central approach of the central idea in this course.**

Note here though that the rationality is not always possible

in complicated environment but we still will aim for to build a rationale agents.

A recurrent question in the media and

public opinion is whether AI is a threat to our human kind.

Indeed AI may raise some feeling of uncertainty about the future, and

whether living people wonder whether they will keep their jobs, and

whether AI systems would take over the world and exterminate the human race.

This comes from different kinds of peoples at all levels in the society

including Professor Stephen Hawkins who told the BBC that the development of

full artificial intelligence could spell the end of the human race.

So he's in a good position to see how

powerful are AI systems as he's using AI systems to communicate.

In my opinion AI is a flourishing and

exciting field where everyone can contribute.

We can contribute at the policy level, at the research level,

computing level, privacy preserving level, nature level and so on and so forth.

So we can be actors other than consumers of AI and

AI is a part of our lives anyway.

This is all what this course is about giving you the principle and

practice of artificial intelligence.