中国科学技术大学
University of Science and Technology of China

人工智能讲义

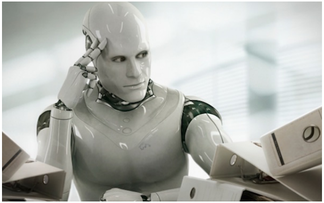
函数与复杂性

February 26, 2018

ustc

Outline

1. 函数
2. 函数描述的复杂性



什么是智能体?

智能体 ＝ 感知器 ＋ “环境-最佳应对” ＋ 执行器 ＋......
＝Observation ＋ Decision ＋ Action ＋ ......
＝ODA＋......

类比为：眼睛 ＋ 大脑 ＋ 手脚 ＋......

环境变量 X	最佳应对 A
x_1	a_1
x_2	a_2
$\dots$	$\dots$

Table: $a_i = f(x_i), i = 1, 2, \dots$

我们课程关注“大脑”

O：观测所得环境变量 X 的值 x_i
A：行动，行动 a_i
D：决策/大脑，用函数 f 来描述，给出面对环境 x_i 时的最佳行动 a_i^*

生产实践中：（不严谨的说法）

- 一种转换，从输入到输出的转换或映射机制；
- 可以是一个过程，如生产过程，将原材料转换为产品；可以是人的一个思维过程；
- 可以表示现实中几乎所有的“转换”。

在数学中：（严谨的说法）

- 传统定义：一般的，在一个变化过程中，有两个变量 x, y，如果给定一个 x 值，相应的就确定唯一的一个 y，那么就称 y 是 x 的函数，其中 x 是自变量，y 是因变量，x 的取值范围叫做这个函数的定义域，相应 y 的取值范围叫做函数的值域。
- 现代定义：设 A, B 是非空的集合，如果按照某种确定的对应关系 f，使对于集合 A 中的任意一个元素 x，在集合 B 中都有唯一确定的元素 y 和它对应，那么就称 f 为从集合 A 到集合 B 的一个函数，记作 $y = f(x), x \in A$ 或 $f(A) = \{y|f(x) = y, y \in B\}$ 。
- 近似的理解：两个集合 A, B 的元素之间的关系。

函数怎样表示最好呢？考虑：精确性、简洁性和使用方便性

- 解析表示：数学关系等式来描述函数，复杂的函数无法用解析式来表示；能用解析式简单描述出来的函数，表示两个集合（定义域和值域）元素之间有很强的关系，这种关系，是一种“知识”或“规律”。
- 列表法：将定义域的元素排在表的左边列，右边列给出对应的值域的值。能精确描述表示所有的函数；但是面对连续型定义域（或定义域包含无穷个元素）时，该方法无效；大多数函数的列表表示法面临表中的行（一行一个定义域的元素）太多的问题。
- 图像法：把一个函数的自变量 x 与对应的因变量 y 的值分别作为点的横坐标和纵坐标，在直角坐标系内描出它的对应点，所有这些点组成的图形叫做该函数的图象。这种表示函数关系的方法叫做图象法。这种方法的优点是通过函数图象可以直观、形象地把函数关系表示出来；缺点是从图象观察得到的数量关系是近似的。
- 语言叙述法：使用语言文字来描述函数的关系. 这种表示方法在现实中是最常见的。

函数是知识的一种表示方法

- 物理、化学、生物和社会科学等的研究，科学发现，寻找其中的“规律或知识”，并表述成 函数
- $s = vt, E = mc^2$，“一切微观粒子，包括电子和质子、中子，都有波粒二象性”，……
- 认识世界，总是针对某个“现象”，分析已知条件（集合 A）和结果 (集合 B) 之间的关系，寻找/理解/描述他们之间的“转换”机制，即认识某个“函数”。
- 所以，我们若用函数 f 来描述世上任一“变换”，那么认识世界，科学发现，就是所谓的“寻找/理解/定义/描述”对应的函数 f 的过程。
- 计算机人工智能中所谓“机器学习”，即让计算机自动来认识世界，寻找函数 f。
- 然而人工智能并非只有“寻找函数 f”，还有……

站在计算思维的角度看

- 在计算机科学中，用列表法来表示和定义函数;
- 在计算机科学中，可以用一段程序代码，把函数的输入到输出的映射“打印”出来，这段程序代码被视为“函数”;
- 计算机科学中，通常会把连续的，无穷个元素的集合等用有限集合来近似;（离散化，模拟 $\rightarrow$ 数字）;
- 用列表法来理解函数有助于对问题求解的时空复杂性进行分析。
- 这一思想将贯穿我们整个人工智能领域。

函数描述/定义：说明给定输入 x，输出 y 是什么

- 完全描述：对任意一个输入 $x \in \mathbb{X}$，都能给出相应的 $y \in \mathbb{Y}$。理论研究中，我们经常追寻对函数的完全描述，来掌握“全部”知识。
- 部分描述：并没有对所有可能输入都说明了其相对应的输出值。实际工程应用中，完全描述函数，很多时候是没有必要，或者不可行的，故常常给出部分描述即可。
- 符号说明：$|\cdot|$ 表示集合的大小。

完全描述

给定函数 $y = f(x)$，其中 $x \in \mathbb{X} = \{x_1, x_2, \ldots, x_l\}$，$l = |\mathbb{X}|$，可用如下表格来完全描述该函数，穷举法

x	$y = f(x)$
x_1	$y_1 = f(x_1)$
x_2	$y_2 = f(x_2)$
$\ldots$	$\ldots$
x_l	$y_l = f(x_l)$

部分描述

给定某些输入 $x \in \mathbb{X}$, 输出 $y \in \mathbb{Y}$, 可用如下表格来说明函数的部分对应关系，其中 $m < |\mathbb{X}|$

x	$y = f(x)$
x_1	$y_1 = p(x_1)$
x_2	$y_2 = p(x_2)$
$\ldots$	$\ldots$
x_m	$y_m = p(x_m)$

假设输入是一个向量，不妨设 $X=(X_1,X_2,\ldots,X_n)$，第 i 个分量的值域大小记为 $|X_i|$, 其第 j 个取值为 v_{ij} ，则如下表格完全表示一个函数:

X_1	X_2	$\ldots$	X_n	$f(X_1,X_2,\ldots,X_n)$
v_{11}	v_{21}	$\ldots$	v_{n1}	y_1
v_{11}	v_{21}	$\ldots$	v_{n2}	y_2
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$v_{1\|X_1\|}$	$v_{2\|X_2\|}$	$\ldots$	$v_{n\|X_n\|}$	$y_{\|X_1\|\|X_2\|\ldots\|X_n\|}$

Table: n 元函数的表示

思考：函数耗费的存储空间是多少?

- 随列数指数增加!

为描述/说明/定义一个函数，我们耗费了多少空间?

- 决定因素 1：列数 n
- 决定因素 2：每列的值域大小，$m_1, m_2, \ldots, m_n$
- 存储空间需求：$O(m_1 \times m_1 \times \ldots \times m_n)$

一个类似天气预报的现实问题有多复杂?

- 我们认为天气预报是个包括“建模”，“模型求解”和“模型使用”三步的过程：
- 建模：由人来确定有哪些列（包括列数），通常认为列越多，考虑的因素越多，预报准确性越高；重要的、相关的列越多，预报准确性越高；每个列的值精度越高(值域越大)，预报准确度越高；(模型被想象成表格)；
- 模型求解：填入已知样本数据到表格中，把剩余的未知的样本数据补上
- 模型应用：依据输入，选择表格的某一行，在此行中找到对应的最佳行动 y^*
- 模型求解和模型应用中，如果模型被想象成表格，时间复杂度？空间复杂度？如果行动是一个序列，其时空复杂度?

计算机求解问题，永恒的主题就是考虑算法的每个步骤和过程的时空复杂度。

AI 的主要研究问题

- 建模：人手工来做，凭借经验，受制于数据（主要是表格的列）的获取手段（仪器设备，如传感器等）；给定一个数据集，其实此时问题模型已经建好；能让机器自动做吗？复杂性在那儿？
- 模型求解：学习问题，获得最优的完整的表格描述，即函数 f 的表示；这就是人工智能中的学习问题；如何简化表格表示及获得对应的参数？深度学习。时空复杂性？
- 模型应用：搜索问题。找到表格的特定行，及环境的最佳应对行动；当要找表格的若干行，并连成序列，构成动作序列时，考虑到每个最优动作不一定构成最优动作序列，例如自动驾驶。2006 年以来的蒙特卡罗树搜索。时空复杂性？

奥卡姆剃刀原理：Ockham'sRazor

- 14 世纪逻辑学家、圣方济各会修士奥卡姆的威廉（William of Occam，约 1285 年至 1349 年）提出；
- "如无必要，勿增实体"；
- "简单有效原理"；
- "切勿浪费较多东西去做，用较少的东西，同样可以做好的事情。"
- 如果对于同一现象有两种不同的假说，我们应该采取比较简单的那一种；
- 吝啬定律 (Law of parsimony)，或者称为朴素原则；
- 牛顿提出的一个原则：如果某一原因既真又足以解释自然事物的特性，则我们不应当接受比这更多的原因；
- 莱布尼兹的"不可观测事物的同一性原理"；
- 管理企业制定决策时，应该尽量把复杂的事情简单化，剔除干扰，抓住主要矛盾，解决最根本的问题，才能让企业保持正确的方向；
- 爱因斯坦的一句著名的格言：万事万物应该都应尽可能简洁，但不能过于简单。

最小描述长度

- 1978 年由 Jorma Rissanen 引入；
- 卡姆剃刀形式化后的一种结果；
- 在给予假说的集合的情况下，能产生最多资料压缩效果的那个假说是最好的；
- 任一资料集/数据集都可以由一有限（譬如说，二进制制的）字母集内符号所成的字串来表示。" 最小描述长度原则背后的基本想法是：在任一给定的资料集内的任何规律性都可用来压缩。亦即在描述资料时，与逐字逐句来描述资料的方式相比，能使用比所需还少的符号"（Grünwald, 1998）；
- 我们希望选取到的假说能抓到资料中最多的规律，于是我们则寻找压缩效果最佳的假说。

柯尔莫哥洛夫复杂性

- 描述程序的复杂性, Kolmogorov-Chaitin complexity, stochastic complexity, 算法熵;
- 资料集/数据集可以看成是字符串;
- 程序代码可以看成是字符串;
- 一段语言可以看成是字符串;
- 资料集的字符串 s 可以找到内在规律后，用一个较短的程序压缩，执行该程序就会输出字符串 s;
- 所有的能输出字符串 s 的程序中，最短程序的长度称为“柯尔莫哥洛夫复杂性”，柯氏复杂性;
- 衡量描述一个对象所需要的信息量的一个尺度;
- 不同程序设计语言，同一功能的程序代码长度会不一样，一般给定程序设计语言来进行后续讨论。

评价标准

- 完备性：能否对任意输入，给出对应输出？
- 描述长度：
  - 表格描述具有最大复杂度；
  - 解析表示通常具有较小复杂度，解析式的参数个数可以作为解析式复杂度的度量，如多项式函数的系数向量；
  - 柯尔莫哥洛夫复杂性用于计算机科学中，用来比较函数的“程序代码描述”的复杂性。
- 准确性：函数近似表示或逼近时，需要用到。学习问题就是找函数的近似表示。

AI：从函数的观点来学习

- 如何获得函数 $f \Longrightarrow$ 机器学习
- 如何使用函数 $f \Longrightarrow$ 推理和问题求解

基于计算机的高级问题求解方法

- 高级数据结构：复杂问题从现实抽象成计算机的图结构描述（增加时空复杂度约束），表格的简化表示；
- 高级算法设计：复杂问题的求解算法，如图遍历（增加时空复杂度约束）
- 数据库技术的扩展：用数据库的语言来处理和解决部分问题；
- ……

统一到“函数”这个出发点