

My AGI theory explained

YKY 甄景贤

Independent researcher, Hong Kong

generic.intelligence@gmail.com

May 12, 2019

Talk summary

- 1 What is inductive bias? “No free lunch” theorem
- 2 What gives neural networks their power?
- 3 Turing machines and universal logic
- 4 Structure of classical AI systems

Section 1

What is inductive bias? “No free lunch” theorem

The goal of machine learning

- The goal of machine learning is to search in a **space** of learning machines, those machines that satisfy certain criteria
- For example, among the neural networks of a certain size and shape, find the weights that satisfy an **objective function**

AI Winter

- Generally speaking, bottleneck problem of AI = **search space too large**, thus learning too slow
- Historically, “AI Winter” occurred because **logic-based** AI learning suffers from combinatorial explosion, and we lacked workable **heuristics** to tackle it

Inductive bias

- Every learning algorithm has its **inductive bias**
- In other words, some regions of the search space would not be searched
- Thus bias makes learning faster
- But if bias is too strong, the space containing the solution would be cut off
“Throw the baby out with the water”

“No free lunch” theorem

- When search space has no *a priori* structure, any inductive bias must be **good at some problems while bad at others**; the meaning of “no free lunch”
- For example, vision has the invariance of 3D Euclidean geometry, thus the human visual cortex may have inductive bias for this invariance
- Or, human cognition has “logical” structure, using this inductive bias may accelerate machine learning of human intelligence

Kolmogorov complexity

- is **incomputable**, but **approximable**
- The **semantic distance metric** between logic propositions is related to it, where one logic deduction step corresponds to 1 unit of semantic distance
- Find a set of logic rules, that **explains** the world, and not deduce false facts, and # of rules cannot be too large — these requirements *implicitly* approximate Kolmogorov complexity

Section 2

What gives neural networks
their power?

Structure of a neural network

- 1 neuron is a **dot product** followed by a **non-linearity**:

$$\sigma \langle \mathbf{x}, \mathbf{w} \rangle \quad (1)$$

- The non-linearity can take various forms,
eg:

$$\sigma(\xi) = \frac{1}{1 + e^{-\xi}} \quad (2)$$

Structure of a neural network

- 1 **layer** of neurons is a matrix multiplication:

$$\mathcal{O}(W \cdot x) \quad (3)$$

- A neuron network is the function composition $(f \circ f)$ of many **layers**:

$$[\mathcal{O}W]^L x \quad (4)$$

Properties of neural networks

- A neural network is a function with many **parameters**
- It is a **universal function approximator** [Cybenko 1989]
- Its proof can be traced to Weierstrauss's approximation theorem (1885): any continuous function can be uniformly approximated by polynomials
- But the proof is **independent of depth**

Power of NNs comes from depth

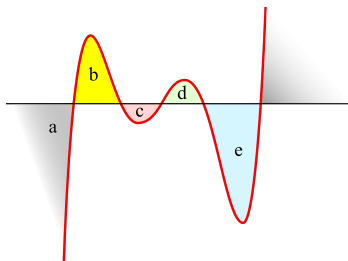
- Suppose $\odot Wx$ is a *cubic* polynomial
- Adding each layer is equivalent to:

$$(\text{polynomial} \circ \text{polynomial}) \quad (5)$$

- Thus, the resulting polynomial has total degree $= 3^L$
- In other words, total degree grows **exponentially**

Power of NNs comes from depth

- 代数基本定理：多项式 次数 = 曲线 跨过 $x = 0$ 多少次



(6)

- 高维：曲面对分类空间分割成多少块

Power of NNs comes from depth

- 这和 VC-dimension 道理一样 [Vapnik–Chervonenkis 1971]
- VC-dimension = 函数 能将 空间 分割成多少块
- 多层 神经网络 的 VC-dim 是 $O(N \log N)$ 其中 N 是 网络参数 的总个数，但证明用的是不连续的 阀函数
- 我估计 VC-dim 会是 指数增长的，但未
有证明

Power of NNs comes from depth

- VC-dim **指数增长** 的意义，表示神经网络能代表一些非常复杂的函数家族
- 而神经网络的参数个数相对地很少，可以在电脑上实现

Revelation from convolution networks

- Yann LeCun 在 1989 发明了 ConvNet, 触发了 机器视觉 的革命, 最近得了 Turing 奖



Revelation from convolution networks

- CNN 将普通 NN 的点积用卷积代替:

$$\boxed{\text{点积}} \quad \mathcal{O}\langle \boldsymbol{x}, \boldsymbol{w} \rangle \rightsquigarrow \mathcal{O}(f * g) \quad \boxed{\text{卷积}} \quad (7)$$

- 而卷积具有**平移不变性**, 有利于视觉:

$$T_x(f) * g = T_x(f * g) \quad (8)$$

- 这是一种归纳偏好, 令学习更快

Revelation from convolution networks

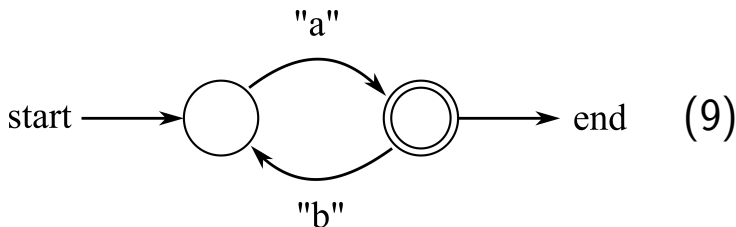
- 其实 视觉 需要的是 仿射 (affine) 不变性，它包括 平移、旋转、放大缩小 等
- 但 似乎 单单是 平移不变性 所带来的 学习加速，已足以令 CNN 在 2012 年超越了人类水平
- 可见，归纳偏好 在 深度学习 里 仍然是很有用的

Section 3

Turing machines and universal logic

Finite state machines

- 有限自动机 通常用一些 tuple 定义 (从略), 例如:



- 这个自动机**接受** “a”, “aba”, “ababa...” 等字符串

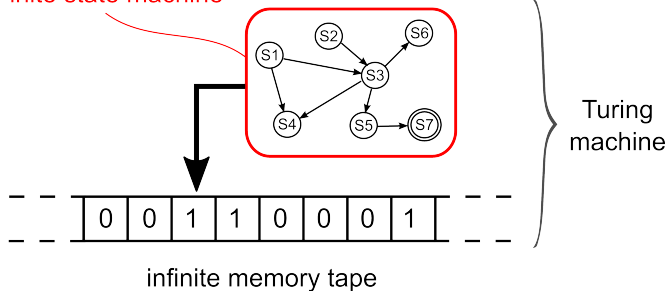
Finite state machines

- 有限自动机 可以接受 $a^m b^n$ 这种字串, m 和 n 不同
- 但它不能接受 $a^k b^k$ 这种字串, 因为它里面没有办法「记住」 k 是多少次
- 有限自动机 能辨认的 语言, 称作 regular languages
- Noam Chomsky 在 1950s 定义, 他是计算机科学家 + 语言学家, 现在主要谈政治, 从左派角度批评美国资本主义

Turing machines

Turing 机 = 有限自动机 + 无限长的**记忆磁带** (每个 state 可以 读 / 写 一个字符)

Finite state machine



(10)

Turing machines

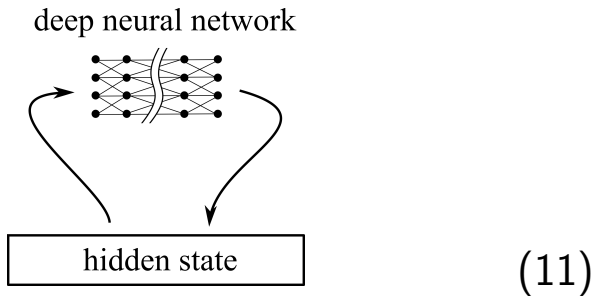
- 亦即是说：有限自动机 + 无限读写带
= 可以计算任何函数，此即
Church-Turing 假设
- Turing 机 等价於 λ -calculus、
combinatory logic、cellular automata、
game of life、recurrent 神经网络、等

Alan Turing (1912-1954)

- Turing 是一个 远超越於时代 的人
- 他考虑过 神经网络 作为学习机器
- 也考虑过 进化算法 (evolutionary algorithms)
- 而当时 1940s 还未有电脑 — 电脑是他发明的!
- 他求出所有可计算函数的形式, 从而将 AI 的问题 限制 在一框框内

Recurrent neural networks

其实一个 RNN 可以看成类似 (10) 的结构：



重点是 hidden state 可以储存计算的**中途结果**，令 RNN 也变成 Turing 机

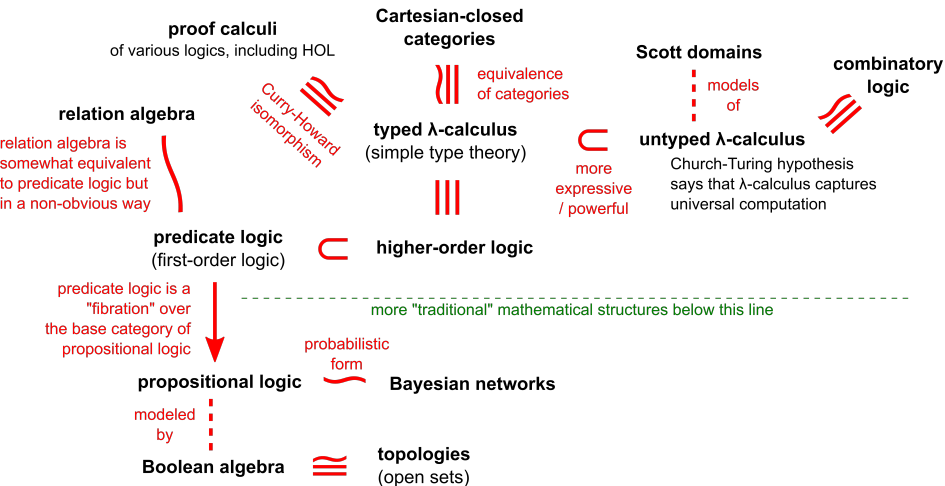
Section 4

Structure of classical AI systems

John McCarthy (1927-2011)

- 「AI 之父」
- 1956 年 在 Dartmouth 第一次举行「人工智能」会议
- 开创了 使用**数理逻辑** 作为 AI 的 **知识表述** (knowledge representation)
- 晚年研究 **改写系统** (term rewriting systems), 是一种更 **广义** 的逻辑

The world of logical structures

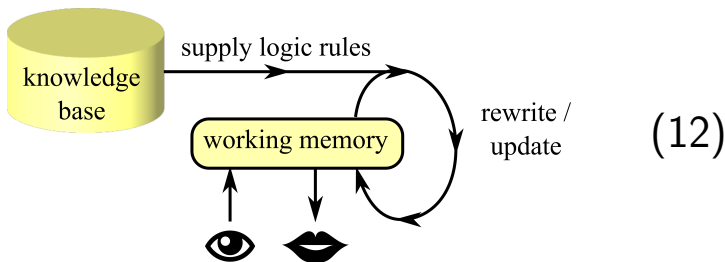


Propositional vs predicate logic

- 最重要是 搞清 命题逻辑 和 一阶 谓词逻辑 的区别
- 命题逻辑: $P_1 = \text{「昨天下雨」}$
 $P_2 = \text{「今天下雨」}$
命题 没有 内部结构
- 谓词逻辑: $P_3 = \text{下雨 (北京, 前天)}$
- 谓词 有 代入 (substitution) 的复杂性

Architecture of logic-based AI systems

这个架构很重要，就像蒸汽机时代的 Carnot cycle（卡诺循环）：



What is a logic rule?

- 举例：爱一个人 但他不爱你 则失恋

$$\heartsuit(x, y) \wedge \neg \heartsuit(y, x) \Rightarrow \odot(x) \quad (13)$$

- 这是一条 rule, 变量 x, y 需要代入适当的个体, 例如 $\{x \setminus \text{John}, y \setminus \text{Mary}\}$
- 寻找 代入 的算法叫 matching 或 unify

SOAR cognitive architecture

- SOAR 是一个著名的 **认知架构**
- 基本上 它根据 working memory 寻找 可以发动的 rules，类似图 (12)
- 它用 **Rete** 算法 快速地搜寻 可用的 rules，这是经典 AI 里的重要算法（*rete* 在拉丁文的意思是「网状」）
- 鉴於中国 AI 的后起之秀，视野不够广阔，故补充一些基础知识

My theory

- 我的理论里，rules 和 matching 机制，都纳入到神经网络里
- 神经网络这件武器，优点是逼近很复杂的 mappings，它是现时最强的机器学习方法
- 我将逻辑结构松弛 (relax)，务求做到足够的归纳偏好即可
- 这理论中最关键的元件，是 symmetric 神经网络

Commutativity of logic conjunctions

- \wedge 的交换律 可能是 逻辑 中最重要的规律:

$$\text{肚饿} \wedge \text{没钱} \Leftrightarrow \text{没钱} \wedge \text{肚饿} \quad (14)$$

- 也可以理解为: 从 前提 推导出 结论, 前提中 命题的**次序** 应该没有关系, 甚至可以夹杂无关的命题
- 交换律 抽象了 逻辑命题 的结构, 类似 抽象代数 中, 交换群 (又叫 阿贝尔群, 集合 $(A, +)$ 满足交换性)

Symmetric neural networks

- 卷积神经网络 具有 平移不变性
- 类似地，**对称**神经网络 具有 交换不变性 (permutation invariance)
- 它可以透过 weight-sharing 来实现，类似 卷积层 的权重共享
- SymNet : 逻辑 \approx ConvNet : 视觉

Will China build its own AGI?

- 日本在 1980 年代研发第 5 代电脑的失败，可以作为借鉴
- 地球的资源有限，科技发展往往是国与国之间竞争的结果
- 在美国有歧视中国的人，中国内部也有拖后腿、倒向外国的人，但在美国也有帮过我的朋友
- 所以我比较支持建立全球化的 AGI 项目

多谢收看 😊