

AGI 的一些基本概念

YKY 甄景贤

Independent researcher, Hong Kong
generic.intelligence@gmail.com

May 4, 2019

Talk summary

- 1 什么是 inductive bias ? 「没有免费午餐」
- 2 神经网络的力量来自什么 ?
- 3 Turing 机与逻辑的宇宙性
- 4 经典逻辑 AI 系统的基本结构

Neural network

- A neural network is a generic function with a large number of **parameters** called **weights**:

weight matrix for each layer total # of layers

$$\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}) = \sigma(W_1 \sigma(W_2 \dots \sigma(W_L \mathbf{x})))$$

- σ is the **sigmoid** function applied *component-wise* to the vector \mathbf{x} :

“Unreasonable” effectiveness of neural networks

- If \odot is replaced by polynomial, degree of the composite function increases **exponentially** as # layers increase

Intelligent agent

- The state vector \mathbf{x}_t of the neural network traces out a **trajectory** in configuration space, which is analogous to a “maze” with **rewards** (●) inside it:

(3)

- We regard the state \mathbf{x}_t as the **mental state** of an intelligent agent, the rewards are given externally by a teacher to reward intelligent behavior.

Hamiltonian control

- **Lagrangian** $L(\vec{x}) = \text{instantaneous reward}$ at state x :

$$J = \int L(\vec{x}) dt \quad (4)$$

- The **Hamiltonian** is defined as:

$$H = L + \frac{\partial J}{\partial \vec{x}} \vec{f} \quad (5)$$

- **Pontryagin maximum principle:**

$$H^* = \inf_u H \quad \text{or} \quad \nabla_{\vec{u}} H^* := \frac{\partial H^*}{\partial \vec{u}} = 0$$

Optimization over logic formulas

- The operation of the system is as follows:

(7)

- \vec{u} coincides with \vec{f} , its purpose is to **rewrite** \vec{x} :

$$\vec{f}(\vec{x}, \vec{u}) \equiv \vec{u}(\vec{x}) \quad (8)$$

Optimization over logic formulas (2)

- For example, the logic rule “love and not loved back \Rightarrow unhappy” performs the rewriting of the following sub-graph:

$$\mapsto \quad (9)$$

- This is the **state transition** $\vec{u} : \vec{x} \mapsto \vec{x}'$, which can also be regarded as the **logical inference** $\vec{u} : \vec{v} \vdash \vec{x}'$, where \vec{u} is the rewriting function or logic rule.

The problem with predicate logic

$$\forall x, y, z. \text{father}(x, y) \wedge \text{father}(y, z) \rightarrow \text{grandfather}(x, z) \quad (10)$$

- This involves **variable substitutions** which are troublesome to handle with neural networks.
(The difficulty seems to come from the cylindric-algebraic structure of predicate logic: if a formula have variables

Relation algebra

Given that:

$$\text{Father} \circ \text{Father} = \text{Grandfather} \quad (11)$$

we can deduce:

$$\text{john Father paul} \quad (12)$$

$$\text{paul Father pete} \quad (13)$$

$$\Rightarrow \text{john Father} \circ \text{Father pete} \quad (14)$$

$$\Rightarrow \text{john Grandfather pete} \quad (15)$$

via *direct* substitution of equal terms.

We're looking for Tensorflow developers to implement a prototype.

Thank you