#### AGI 的一些基本概念

#### YKY 甄景贤

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

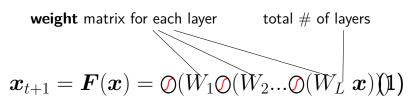
May 4, 2019

## 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
- 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:



• *(*) is the **sigmoid** function applied *component-wise* to the vector *x*:

# "Unreasonable" effectiveness of neural networks

 If ① is replaced by polynomial, degree of the composite function increases
 exponentially as # layers increase

### Intelligent agent

• The state vector  $x_t$  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  $x_t$  as the **mental** state of an intelligent agent, the rewards are given externally by a teacher to reward intelligent behavior.

#### Hamiltonian control

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

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

The Hamiltonian is defined as:

$$H = L + \frac{\partial J}{\partial \vec{x}} \vec{f} \tag{5}$$

Pontryagin maximum principle:

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

YKY 甄景贤

### Optimization over logic formulas

• The operation of the system is as follows:

(7)

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

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

# Optimization over logic formulas (2)

 For example, the logic rule "'love and not loved back ⇒ unhappy" performs the rewriting of the following sub-graph:

$$ightarrow$$
 (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

$$\forall x, y, z. \; \mathsf{father}(x, y) \land \mathsf{father}(y, z) \to \mathsf{grandfather}(x)$$
 (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
(KY 甄景贤 China AGI group M

#### Relation algebra

#### Given that:

Father 
$$\circ$$
 Father = Grandfather (11)

we can deduce.

john Father paul(12)paul Father pete(13)
$$\Rightarrow$$
 john Father  $\circ$  Father pete(14) $\Rightarrow$  john Grandfather pete(15)

via *direct* substitution of equal terms.

 $\Rightarrow$  john Grandfather pete

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

### Thank you