神经网络中的「内省」
(“introspection” in neural networks)

甄景贤 (King-Yin Yan)

General.Intelligence@Gmail.com

July 20, 2017

Abstract. 在本文中「内省」是指智能系统直接读 / 写知识的能力，此能力在经典 logic-based AI 是免费做到的，但神经网络内的「知识」素来有「黑盒」的问题。解决办法是让神经网络直接作用在它自身的 weights 上。

0 Introduction

这篇文章说的「内省」的意思是指智能系统有能力读 / 写它内部的知识 [1]。例如说，一个比较蠢的智能系统可以用 sequence-to-sequence 的方式将中文翻译成英文：

$$\text{“}\boxed{\text{中文句子}}\text{”} \xrightarrow{\boldsymbol{F}} \text{“}\boxed{\text{英文句子}}\text{”} \tag{1}$$

$\boldsymbol{F}$ 代表系统的函数。但系统并不真的明白句子的意义，句子只是「水过鸭背」地流过系统。一个更聪明的系统是：句子可以**进入**到 $\boldsymbol{F}$ 里。我所说的「自省」就是这意思。

Introspection is useful in:

- learning by instructions, or “learn by being told”
  (a technique crucial to accelearating the learning of human knowledge)
- belief revision / truth maintenance
  (the most challenging and highest-level task in logic-based AI)

举例来说，小孩子的行为是由他内部的知识决定的，「知识决定行为」。

- 当小孩子看到一个成人做的动作，他会模仿那动作。

(2)

[1] 「内省」亦有 meta-reasoning 的意思，亦即除了**外在**的知识，系统还拥有关於系统自身状态的知识。本文中「内省」是指存取「普通知识」的能力。

- 或者小孩子听到一句说话：「不要吃污糟食物」，他明白了那句说话的意思而改变行为。

这两个例子都涉及到「感觉资料」进入 $\boldsymbol{F}$ 里面：

$$\boxed{\text{sensory input}} \hookrightarrow \boldsymbol{F} \tag{3}$$

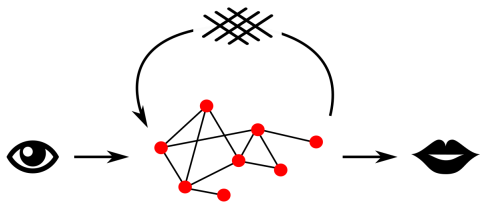
Introspection is related to the functional closure $\mathbb{X} \simeq \mathbb{X}^{\mathbb{X}}$ which gives a **Cartesian-closed category** (CCC).

1 Architecture

For reference, the architecture for **visual recognition** is:

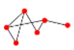
(4)

Our basic AGI architecture is:



(5)

 = [deep] neural network, trained via **reinforcement learning**

 = mental state / working memory

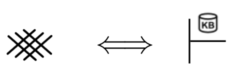
The main problems we need to solve for AGI:

(A) How to enable a neural network to act on a graph structure (that does not easily fit into a fixed-length vector)?

(B) How to solve the introspection problem?

(C) How to incorporate **episodic memory** into the basic architecture (5)?
Episodic memory may be essential for the learning of common-sense (eg. the need to process **stories**).

We can use a deep network to emulate logical inference:



(6)

 means to perform a **single step** of logical inference, ie, the **consequence operator**.

In the past, the learning of KB relied on **inductive logic learning**, based on combinatorial search, which was too slow. The new hope is for deep learning to learn this mapping in reasonable time.

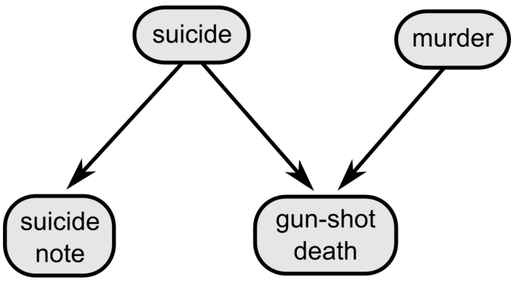
Deep learning 在 vision 中的成功，令我们相信它几乎可以 learn 出「任何 mapping」，除非那 mapping 具有 更深层 的结构；这时要用到 RNN。似乎 RNN 可以学习「任何结构」— “unreasonable effectiveness”。

An interesting idea is: would 2nd-order RNN’s have even more advantages?

2 Structure of memories

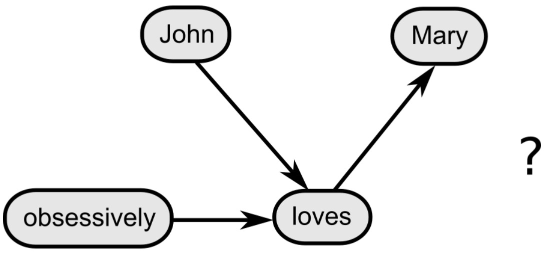
2.1 Working memory

At the proposition level, memory is organized as a **Bayesian network**, where each node is a proposition:



(7)

At the sub-propositional level, every proposition may be represented as an entity-relation graph, where each node is a **concept atom**:



(8)

but we are still unsure about the exact construction mechanism of sub-propostional graphs.

2.2 Episodic memory

Episodic memory = an even-bigger graph?

3 NN acting on graphs

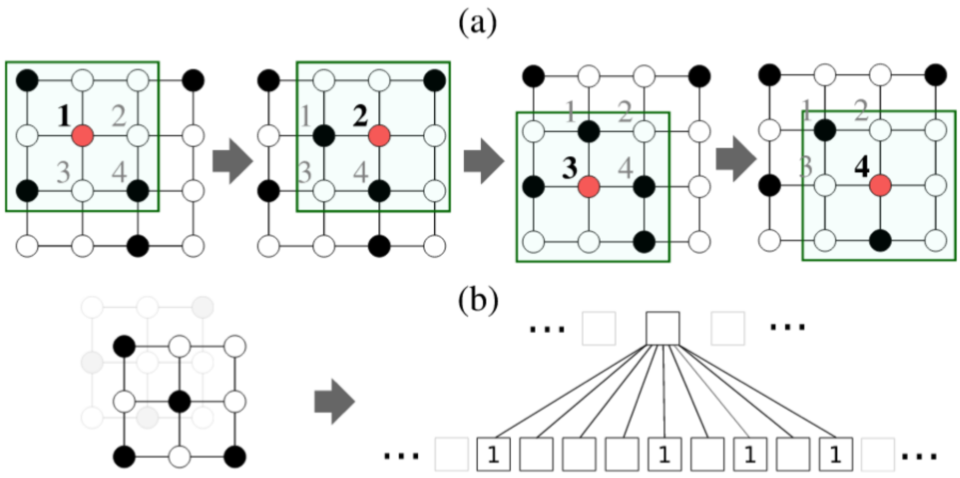
3.1 CNN

With this analogy:

$$\text{CNN for vision} \iff \text{CNN for graphs} \tag{9}$$

a new breed of algorithms have been developed, eg: [1] [2] [3]. For a nice introduction see the blog entry: `https://tkipf.github.io/graph-convolutional-networks/`.

As explained in [1], a CNN works as if a "receptive field" moves over an image:



(10)

and the idea is to let a similar receptive field traverse a graph.

4 Cartesian closure

举例来说，「吃了污糟的食物会肚痛」是一个句子，它经由 进入 mental state $\boldsymbol{x}$ ， 变成 proposition。但我们希望这逻辑命题变成 KB 的一部分。With

$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}) \tag{11}$$

where

$\boldsymbol{f} = \text{KB} =$

$\boldsymbol{x} = \text{state}$

An individual logic rule is a restriction of $\boldsymbol{f}$ to a specific input.

$\boldsymbol{f} \equiv \text{KB}$ is the sum of restrictions:

$$\text{KB} = \bigcup \boldsymbol{f}_i \tag{12}$$

Or roughly speaking, $\boldsymbol{f}$ is the sum total of objects like $\boldsymbol{x}$:

$$\boldsymbol{f} = \bigcup \boldsymbol{x}_i \tag{13}$$

However, the problem is that the structure of $\boldsymbol{f}$ (as the neural network) is too complicated to be expressed as a sum of restricted functions. This remains an unsolved problem.

Acknowledgements

Thanks to Jonathan Yan for suggesting to use CNN for graphs and showed me the relevant papers.

Bibliography

[1] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. 2016. URL `http://arxiv.org/abs/1605.05273`.

[2] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. 2016. URL `http://arxiv.org/abs/1606.09375`.

[3] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. 2016. URL `http://arxiv.org/abs/1609.02907`.