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

甄景贤 (King-Yin Yan)

General.Intelligence@Gmail.com

July 21, 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。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「内省」就是这意思。

「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。但本文中「内省」是指存取「普通知识」的能力。

1 Applications

Introspection (in the present paper’s sense) 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)

- 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将「感觉资料」放进 $\boldsymbol{F}$ 里面：

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

2 Cartesian closure

Introspection requires the functional closure $\mathbb{X} \simeq \mathbb{X}^{\mathbb{X}}$ which yields a **Cartesian-closed category** (CCC).

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。$\boldsymbol{F}$ is the state-transition function:

$$\boldsymbol{x}_{n+1} = \boldsymbol{F}(\boldsymbol{x}_n) \tag{4}$$

where

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

An individual logic rule is a <u>restriction</u> of $\boldsymbol{F}$ to a specific input; Perhaps I could call such elements "micro-functions".

$\boldsymbol{F} \equiv$ KB is the <u>"union" of micro-functions</u>:

$$\text{KB} = \bigcup \boldsymbol{f}_i \tag{5}$$

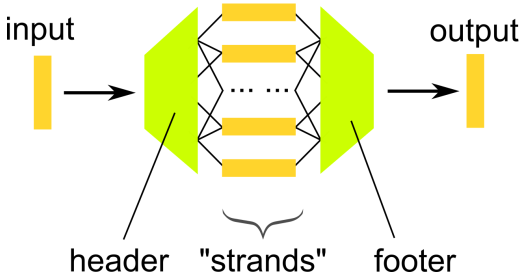
Or, in a vague sense, $\boldsymbol{F}$ is the sum total of objects like $\boldsymbol{x}$:

$$\boldsymbol{F} = \bigcup \boldsymbol{x}_i \tag{6}$$

但 $\boldsymbol{F}$ 是一个神经网络，它的一般形式是：

$$\boxed{\text{output}}\ \boldsymbol{x}_{n+1} = \boldsymbol{F}(\boldsymbol{x}_n) = \overset{1}{W} \quad \overset{2}{W} \quad ..... \quad \overset{L}{W} \quad \boldsymbol{x}_n \tag{7}$$

L = total number of layers. 由於各层的非线性「纠缠在一起」，表面上无法将神经网络「分解」。直到笔者受了 David Ha *et al* 提出的 PathNet [1] 理论所启发，PathNet 是由一些较小的神经网络 modules 组成，所以或许可以建构如下形式的「丝状神经网络」：



(8)

这些「丝条 」可以是简单的神经网络，例如每个的宽度或深度很小，因而可以用较短的 weights vector 描述。正是因为这原因，一个 本身可以作为神经网络的输入。但整个神经网络 $\boldsymbol{F}$ <u>无法输入自己</u>，因为根据 Cantor's theorem，$\mathbb{X} = \mathbb{X}^{\mathbb{X}}$ 是不可能的。

Let $\overline{\boldsymbol{F}}$ = header, $\underline{\boldsymbol{F}}$ = footer, $\boldsymbol{f}_i$ = strands, then:

$$\boldsymbol{F} = \overline{\boldsymbol{F}} \circ \bigcup \boldsymbol{f}_i \circ \underline{\boldsymbol{F}} \tag{9}$$

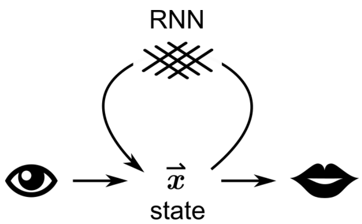
每个 大约对应於逻辑上的一个**命题**（proposition, 可以是条件命题或普通命题）。

3 Overall architecture

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

(10)

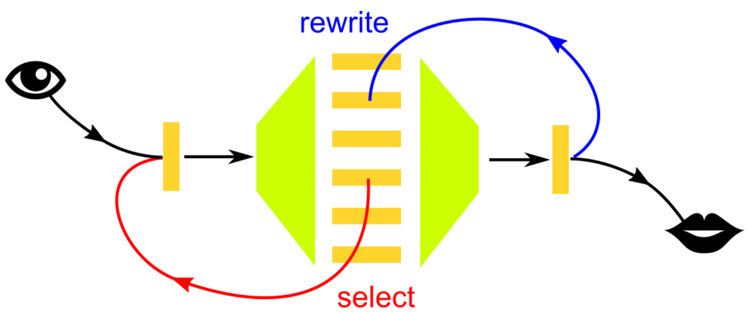
Our basic AGI architecture is:



(11)

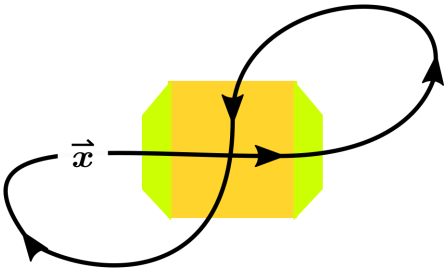
 = [deep] neural network, trained via **reinforcement learning**

The overall **recurrent** setup operates like this:



(12)

Viewing the "information flow" in a simplified way, we notice a "second" pass through the network's internal weights:



(13)

This mode of operation has always been standard in logic-based systems. The is the KB. The vertical pass represents reading/writing information to/from KB. The horizontal pass represents using the KB for logical inference (thinking), ie:

$$\boldsymbol{x}_n \cup \text{KB} \vdash \boldsymbol{x}_{n+1} \tag{14}$$

4 Structure of memories

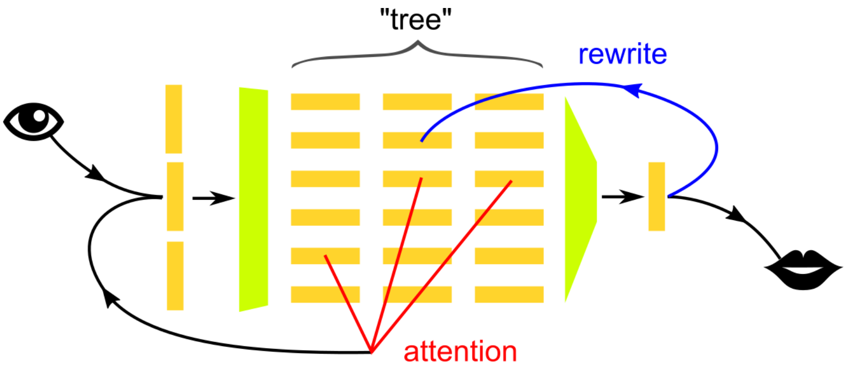
The "main memory" $\boldsymbol{F}$ can take the form of a tree (), graph (), or hyper-graph (), with increasing complexity.

The **mental state** $\boldsymbol{x}$, or "working memory", can also assume the above-mentioned forms.

Currently I am not sure whether to place **episodic memory** inside $\boldsymbol{F}$ or as a separate module outside $\boldsymbol{F}$.

We need to organize the 's in the form of , or , in such a way that the resulting structure is also a neural network, or more generally a mathematical **function** in Hilbert space.

But there is one simple way: Basically, a deep network is automatically "tree-like" because of its many layers (**levels**) of weights organized hierarchically. Thus we can build a network like this:



(15)

The attention mechanism selects a number of 's to be the **current state** or "working memory". Notice that the input size is bigger than the output size, which reflects the structure of the logical **consequece operator** $\vdash$.

Acknowledgements

Thanks to David Ha for his PathNet idea.

Bibliography

[1] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *CoRR*, abs/1701.08734, 2017. URL `http://arxiv.org/abs/1701.08734`.