

# “Introspection” in neural networks

甄景贤 (King-Yin Yan)

General.Intelligence@Gmail.com

July 25, 2017

**Abstract.** In this paper, “introspection” refers to the ability of an intelligent agent to access its own knowledge. This ability comes for free in classical logic-based AI, but neural networks are notorious for the “black-box” problem. The solution is to have the network act on its own weights.

## 0 Introduction

By “introspection” is meant the ability of an intelligent agent to **access** (read or write) the contents of its knowledge base. For example, a dumber agent may use the “sequence-to-sequence” technique to translate Chinese sentences into English:

$$\boxed{\text{“Chinese sentence”}} \xrightarrow{F} \boxed{\text{“English sentence”}} \quad (1)$$

$F$  is the system’s **function**. But the system does not truly understand the sentences’ meaning; The sentences simply “pass through” the system. A more intelligent agent would allow sentences to **go into**  $F$ . This is what I mean by “introspection”.

“Introspection” also connotes **meta-reasoning**, which means that, in addition to **extrinsic knowledge**, the system also possesses knowledge about **its own states**. In this paper, we only concern ourselves with the agent’s access to extrinsic knowledge.

## 1 Applications

Introspection (in the present paper’s sense) is useful in:

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

For example, a child’s behavior is determined by his internal knowledge; “Knowledge determines action”.

- When a toddler watches an adult's gesture, he tries to imitate that gesture:



(2)


- Or when a child hears a saying: “don’t eat dirty food”, he understands the words and changes his behavior.

Both examples involve putting “sensory data” into  $\mathbf{F}$ :

$$\boxed{\text{sensory data}} \hookrightarrow \mathbf{F} \quad (3)$$

## 2 Cartesian closure

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

For example, “eating dirty food causes stomach pains” is an NL sentence, it enters from  into the mental state  $\mathbf{x}$ , as a **proposition**. But we want  $\mathbf{x}$  to become part of  $\boxed{\mathbf{KB}}$ .  $\mathbf{F}$  is the state-transition function:

$$\mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n) \quad (4)$$

where

$$\begin{aligned} \mathbf{F} &= \boxed{\mathbf{KB}} = \text{XXXX} \\ \mathbf{x} &= \text{state} \end{aligned}$$

An individual logic rule is a restriction of  $\mathbf{F}$  to a specific input; Perhaps I could call such elements “micro-functions”.

$\mathbf{F} \equiv \boxed{\mathbf{KB}}$  is the “union” of micro-functions:

$$\boxed{\mathbf{KB}} = \bigcup f_i \quad (5)$$

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

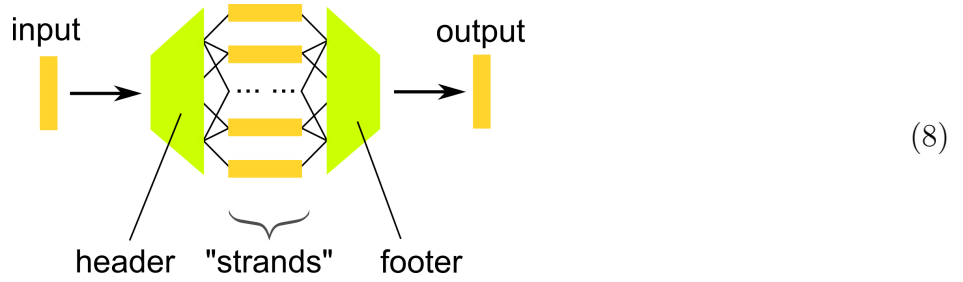
$$\mathbf{F} = \bigcup \mathbf{x}_i \quad (6)$$

But  $\mathbf{F}$  is a neural network; Its general form is:

$$\boxed{\text{output}} \quad \mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n) = \textcircled{1} W^1 \textcircled{2} W^2 \dots \textcircled{L} W^L \mathbf{x}_n \quad (7)$$

$L$  = total number of layers. Because all the layers of non-linearities are “entwined together”, apparently we cannot “decompose” a neural network. That is, until the author hears of David Ha *et al*’s

PathNet [1] idea, which is a big network consisting of smaller neural-network modules. Inspired by that, I propose to construct a “threaded” neural network:



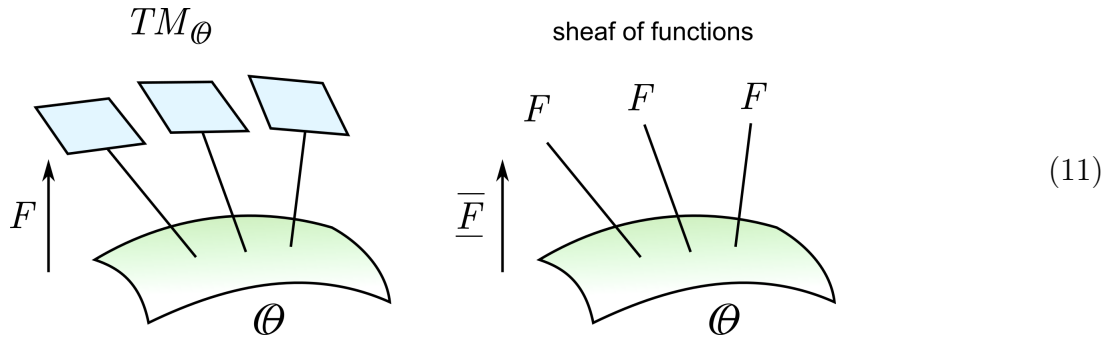
These “strands” are simpler neural networks, for instance with smaller widths and depths, so they can be described by shorter weight vectors. Precisely for this reason, a can be presented as input to its own neural network. We cannot pass the entire network  $\mathbf{F}$  to itself, due to Cantor’s theorem, which says  $\mathbb{X} = \mathbb{X}^{\mathbb{X}}$  is impossible.

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

$$\mathbf{F} = \overline{\mathbf{F}} \circ \bigcup \mathbf{f}_i \circ \underline{\mathbf{F}} \quad (9)$$

Each roughly corresponds to a single **proposition** in logic-based AI. Such propositions may be conditional or plain statements.

$$\Theta \in \mathcal{\Theta} \quad (10)$$



### 3 Overall architecture

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

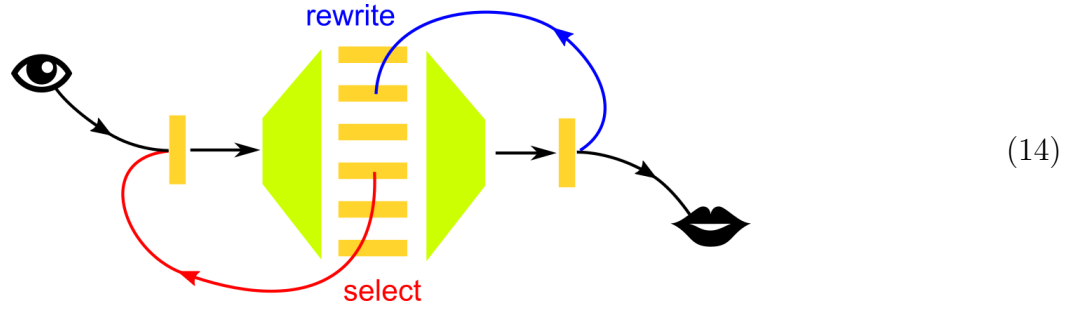


Our basic AGI architecture is:

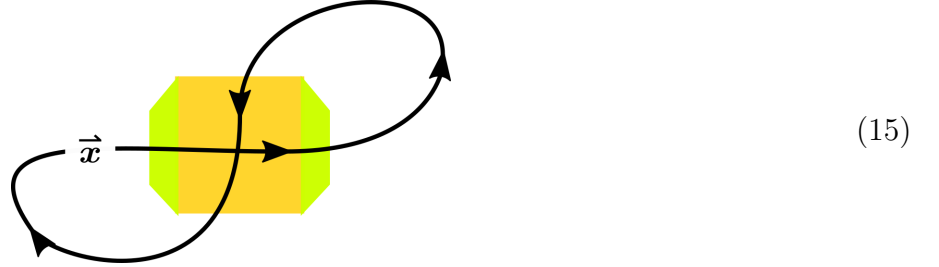


= [deep] neural network, trained via **reinforcement learning**

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



Viewing the “information flow” in a simplified way, we notice a “second” pass through the network’s internal weights:



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

$$x_n \cup \boxed{\text{KB}} \vdash x_{n+1} \quad (16)$$

## 4 Structure of memories

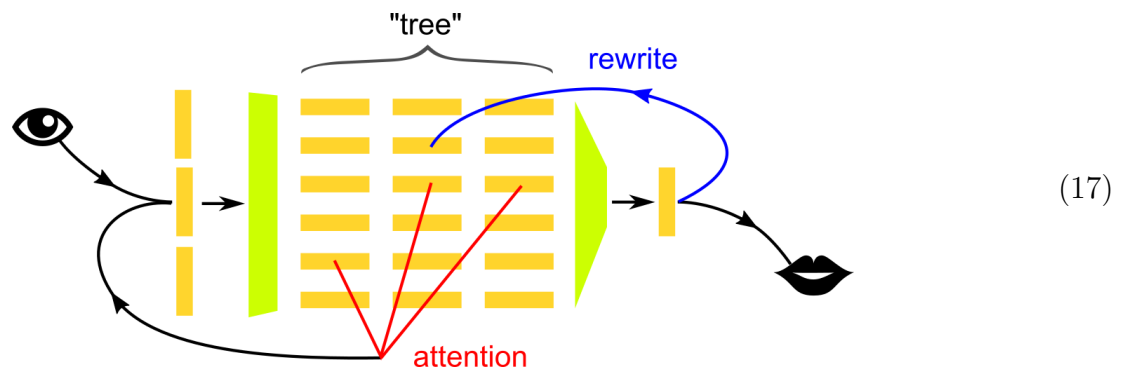
The “main memory”  $F$  can take the form of a tree () , graph () , or hyper-graph () , with increasing complexity.


The **mental state**  $x$ , or “working memory”, can also assume the above-mentioned forms.

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

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:



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 **consequence 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>.