“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:

$$\text{“}\boxed{\text{Chinese sentence}}\text{”} \xrightarrow{\boldsymbol{F}} \text{“}\boxed{\text{English sentence}}\text{”} \tag{1}$$

$\boldsymbol{F}$ 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 $\boldsymbol{F}$**. 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 accelearating 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 $\boldsymbol{F}$:

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

2 Cartesian closure

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

For example, "eating dirty food causes stomach pains" is an NL sentence, it enters from into the mental state $\boldsymbol{x}$, as a **proposition**. But we want $\boldsymbol{x}$ to become part of KB. $\boldsymbol{F}$ 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 restriction of $\boldsymbol{F}$ to a specific input; Perhaps I could call such elements "micro-functions".

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

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

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

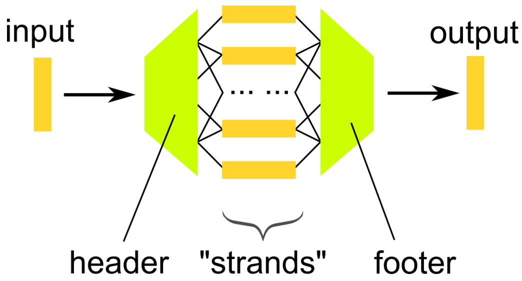
$$\boldsymbol{F} = \bigcup \boldsymbol{x}_i \tag{6}$$

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

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

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:



(8)

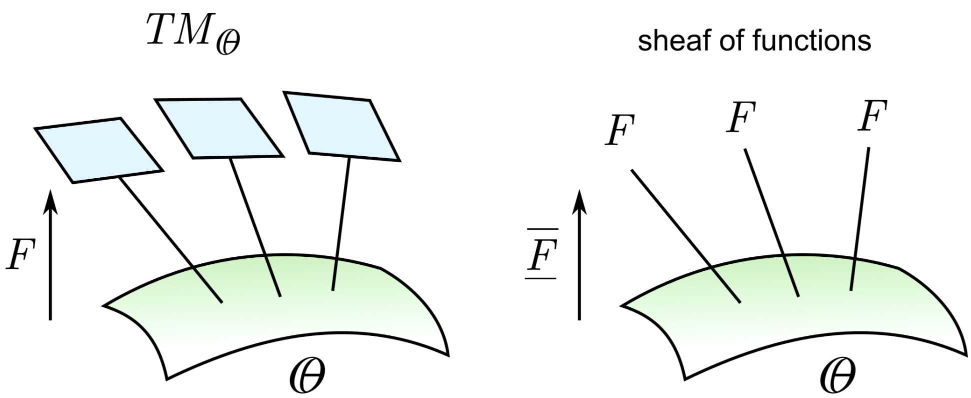
These "strands" are simpler neural networks, for instance with smaller widths and depths, so they can be described by shorter weight vectors. Precisely for this reason, a can be presented as input to its own neural network. We <u>cannot pass the entire network $\boldsymbol{F}$ to itself</u>, due to Cantor's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

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

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

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

$$\Theta \in \boldsymbol{\mathit{\Theta}} \tag{10}$$



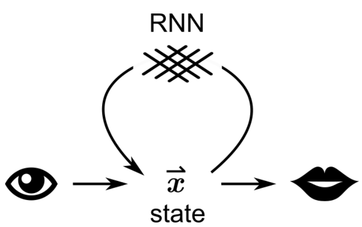
(11)

3 Overall architecture

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

(12)

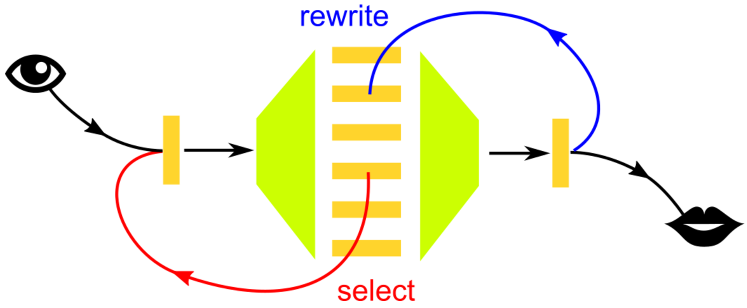
Our basic AGI architecture is:



(13)

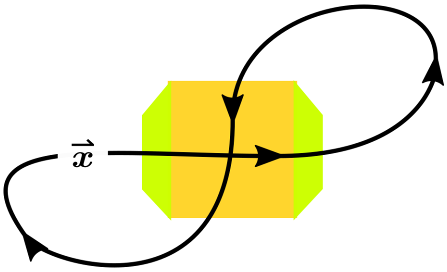
 = [deep] neural network, trained via **reinforcement learning**

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



(14)

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

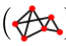


(15)

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

$\boldsymbol{x}_n \cup$  $\vdash \boldsymbol{x}_{n+1}$ (16)

4 Structure of memories

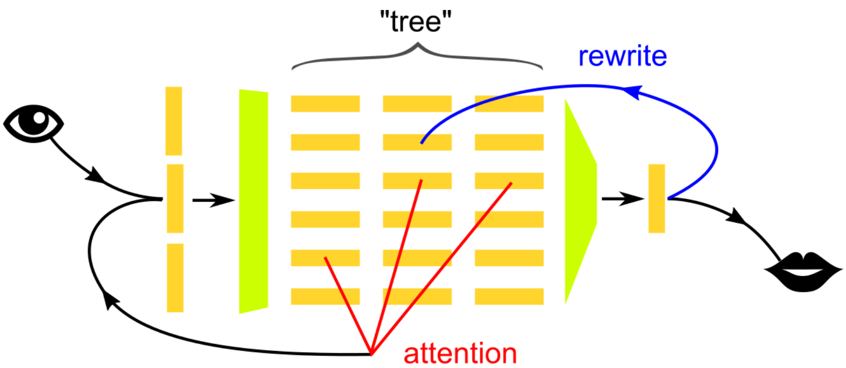
The “main memory” $\boldsymbol{F}$ can take the form of a tree (), graph (), or hyper-graph (), with increasing complexity.

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:



(17)

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`.