

“Introspection” in neural networks

甄景贤 (King-Yin Yan)

General.Intelligence@Gmail.com

July 20, 2017

Abstract.

0 Introduction

By “introspection” is meant the ability of an intelligent agent to **access** (read or write) the contents of its knowledge base.

Introspection 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,

1 Architecture

The architecture for **visual recognition** is:



Our basic architecture is:



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

⬢ = mental state / working memory

The main problems we need to solve:

- (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 achieve an ability that I call “learn by being told”?
This is related to the functional closure $\mathbb{X} \simeq \mathbb{X}^{\mathbb{X}}$ which gives a Cartesian-closed category
- (C) How to incorporate **episodic memory** into the basic architecture (2)?
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

$$\text{⌘} \iff \text{⌘}^{\text{KB}} \tag{3}$$

⌘^{KB} 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.

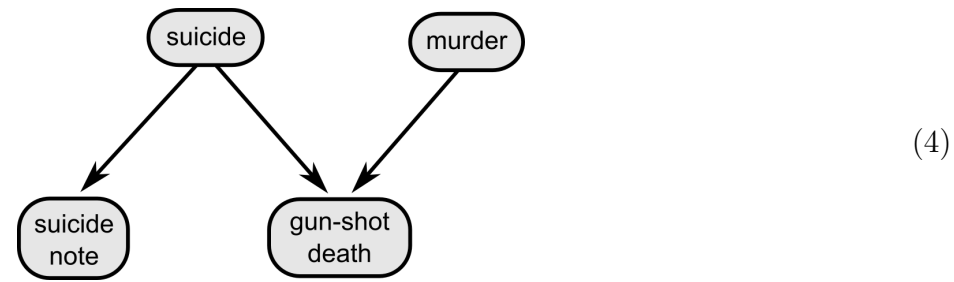
The success of deep learning in **vision** makes us believe that a deep network is capable of learning almost “any mapping”, unless the data exhibits even more complex structure, in which case we need RNN’s. Thus RNN seems able to learn arbitrary structures — hence “unreasonable effectiveness”.

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

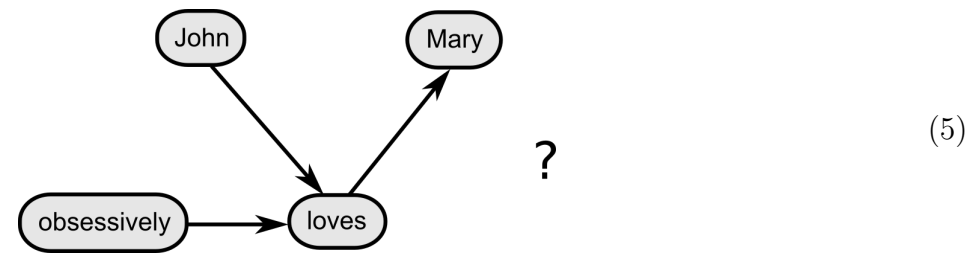
2 Structure of memories

2.1 Working memory

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



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



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

2.2 Episodic memory

Episodic memory = an even-bigger graph?

3 NN acting on graphs

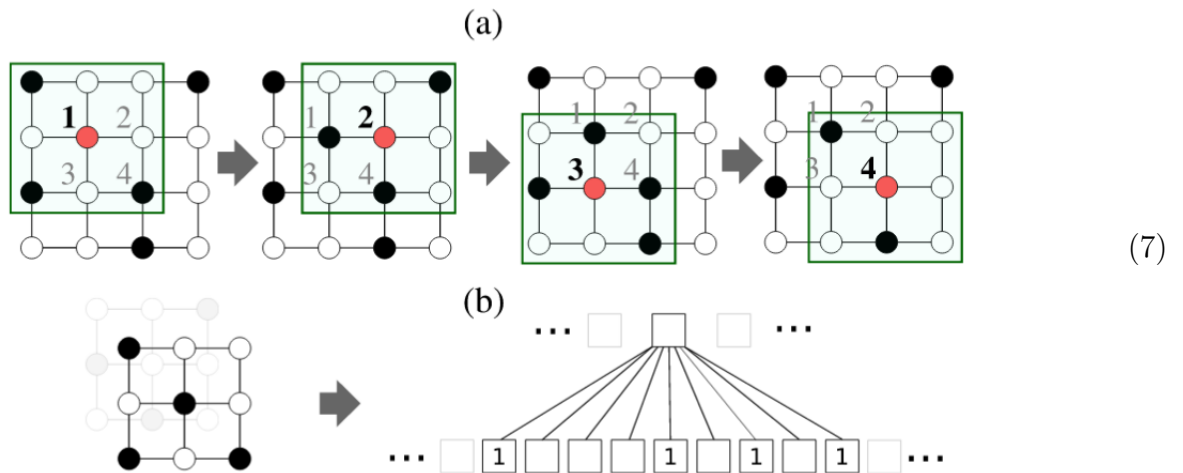
3.1 CNN

With this analogy:

$$\text{CNN for vision} \iff \text{CNN for graphs} \quad (6)$$


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:



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

4 Cartesian closure

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

$$x' = f(x) \quad (8)$$

where

$$f = \boxed{\text{KB}} = \text{restriction}$$

x = state

An individual logic rule is a restriction of f to a specific input.

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

$$\boxed{\text{KB}} = \bigcup f_i \quad (9)$$

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

$$\mathbf{f} = \bigcup \mathbf{x}_i \tag{10}$$

However, the problem is that the structure of \mathbf{f} (as the neural network $\otimes\otimes$) 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>.