White paper

甄景贤 (King-Yin Yan)

General.Intelligence@Gmail.com

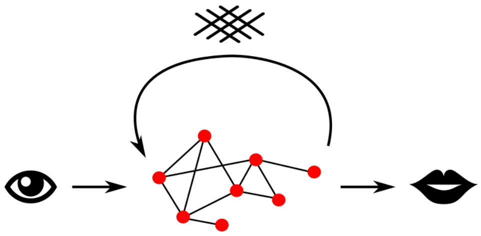
Abstract. This paper describes a cognitive architecture for common-sense reasoning.

0 Introduction

The architecture for **visual recognition** is:

(1)

Our basic architecture is:



(2)

 = [deep] neural network, trained via reinforcement learning

 = mental state / working memory

The main problems we need to solve:

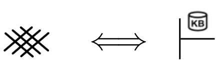
(A) How to enable a neural network to act on a graph structure (that does not easily fit into a fixed-length vector)?

(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**).

The advantage brought by deep learning is that it is capable of factorizing **intermediate features** out of "multifarious" data, classifying them into a relatively smaller number of classes. This capability is absent in traditional AI. For example, in decision trees, once classes are split they are processed *separately*. Whereas neural networks have layers like continually "stretching noodles", they can do what is called **representation learning** in the hidden layers. This technology was unavailable before.

The reason why I believe we can build AGI, is because we can use a deep network to emulate the following function



(3)

$\vdash^{\text{KB}}$ 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.

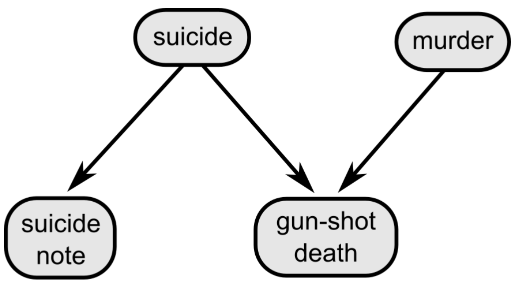
The success of deep learning in **vision** makes us believe that a deep network is capable of learning almost “any mapping”, unless the data exhibits even more complex structure, in which case we need RNN’s. Thus RNN seems able to learn arbitrary structures — hence “unreasonable effectiveness”.

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

1 Structure of memories

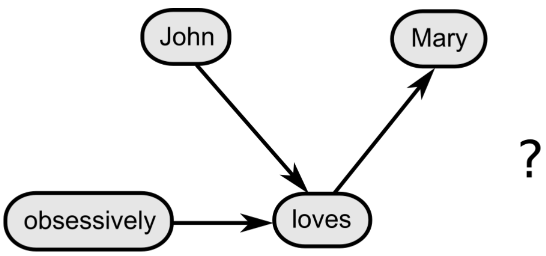
1.1 Working memory

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



(4)

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



(5)

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

1.2 Episodic memory

Episodic memory = an even-bigger graph?

2 NN acting on graphs

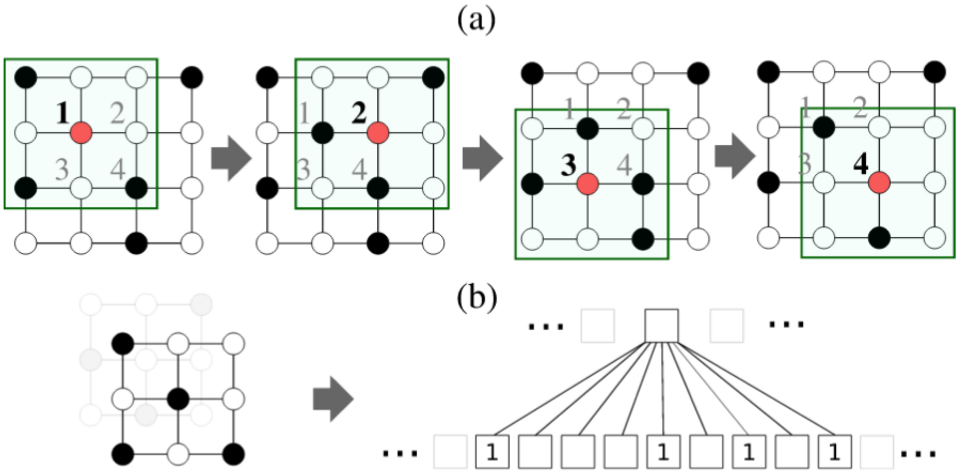
2.1 CNN

With this analogy:

$$\text{CNN for vision} \iff \text{CNN for graphs} \tag{6}$$

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:



(7)

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

3 Cartesian closure

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 . With

$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}) \tag{8}$$

where

$\boldsymbol{f}$ =  = 

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

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

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

 $$= \bigcup \boldsymbol{f}_i \tag{9}$$

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

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

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.