On AGI architecture

December 19, 2019 mid-term report

YKY

Independent researcher, Hong Kong

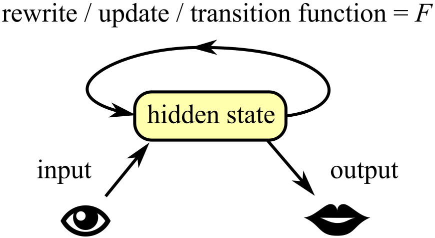
generic.intelligence@gmail.com

Table of contents

Hello friends ☺️
I have made some intermediate progress recently, which I share below.
I am also looking for collaborators.

The simplest AGI architecture

- The simplest AGI architecture consists of a single recurrent loop:

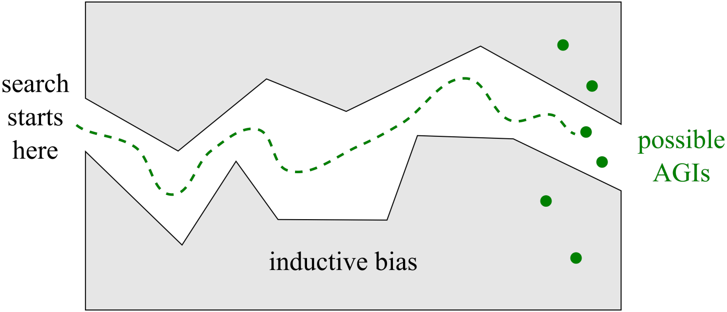


(1)

- It operates under reinforcement learning, maximizing rewards by the Bellman optimality condition
- The transition function F can be implemented by a neural network
- According to No Free Lunch theorem, problem with this architecture is lack of inductive bias, learning is too slow

Some musings on No Free Lunch (NFL)

- According to NFL, there is no such things as "good" or "bad" inductive bias
- As long as it accelerates learning, and still accomodates AGI, it is good bias:

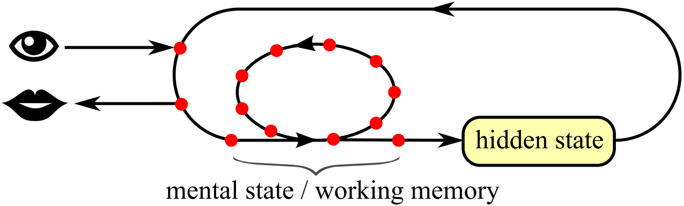


(2)

- For example, the neural network F can be made sparse while preserving deepness
- Yet, I proposed earlier to use logic as bias. Is that redundant? 😖

"Double loop" architecture

- Assume working memory consists of disparate propositions (•), residing in an inner loop. As this loop is iterated, the propositions are condensed into the hidden state. Hence the "double-loop" architecture

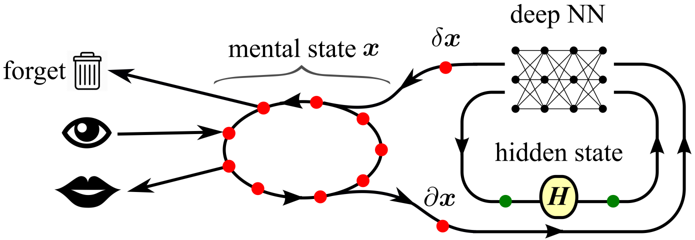


(3)

- This same architecture may be shared by the human brain, as its structure is simple and thus could have been evolved
- As far as I know, BERT also contains an implicit recurrence, where words in a sentence are condensed into a hidden state, from which target words are generated one by one
- So it seems that if we modify BERT to be a "double loop", we can get an AGI:

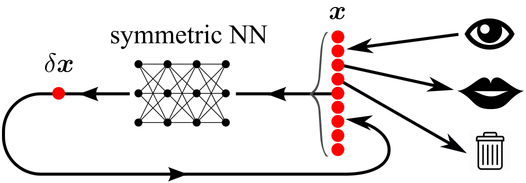
$$\boxed{\text{words / concepts}} \xrightarrow{\text{loop 1}} \boxed{\text{sentences / propositions}} \xrightarrow{\text{loop 2}} \boxed{\text{mental state}} \tag{4}$$

- Diagram (3) is a bit inaccurate, here is a more detailed diagram:



(5)

- In contrast, one can use a symmetric NN to make the architecture even simpler:



(6)

- But it's not easy to decide whether (5) or (6) learns faster 😖

Connection between reinforcement learning & quantum mechanics

- The optimal condition for reinforcement learning is the Bellman equation:

$$\boxed{\text{Bellman}} \qquad S_t(x) = \max_u \{L(x,u) + \gamma S_{t+1}(x)\} \tag{7}$$

- The differential version of Bellman equation is the Hamilton-Jacobi equation in classical analytic mechanics (This has been recognized in 1970's by Kalman, Pontryagin and others):

$$\boxed{\text{Hamilton-Jacobi}} \qquad \frac{\partial S(x,t)}{\partial t} = -H \tag{8}$$

- The Hamiltionian H arises when trying to maximize the Lagrangian L using Lagrangian multipliers. Such multipliers have the interpretation of momentum.

$$L = \text{KE} - \text{PE} \quad , \quad H = \text{KE} + \text{PE} \tag{9}$$

  where KE = kenetic energy, PE = potential energy.
- Up to now, we changed a discrete equation to a continuos differential equation, but that has not yielded any advantage

- Recently I independently discovered an exact way to go from the classical Hamilton-Jacobi equation to the Schrödinger equation via the substitution $\Psi = e^{-i\hbar S}$:

$$\boxed{\text{Hamilton-Jacobi}} \quad \frac{\partial S}{\partial t} = -H \quad \Rightarrow \quad i\hbar \frac{\partial \Psi}{\partial t} = H\Psi \quad \boxed{\text{Schrödinger}} \tag{10}$$

- We have always been told in textbooks that such a process of quantization can only be achieved heuristically, but some friend informed me that [Field 2010] has derived this result.
- From the perpective of AI, this means that solving the reinforcement learning problem is equivalent to solving the Schrödinger equation in Hilbert space!
- Moreover, the Schrödinger equation can be transformed into the diffusion / heat equation via the introduction of imaginary time:

$$\boxed{\text{wave eqn.}} \quad \frac{\partial \Psi}{\partial t} + i\Delta\Psi = 0 \quad \leftrightsquigarrow \quad \frac{\partial u}{\partial t} + \Delta u = 0 \quad \boxed{\text{heat eqn.}} \tag{11}$$

- Yet, if this is to be applicable to AGI, we need to discretize the state space (which becomes a graph), and use the discrete Laplacian Δ or discrete Schrödinger operator to act on graphs
- Impressive as it may sound, this may be of low practical value 😖

Topos theory

A topos is a category in which one can "do logic". The idea originated in 1950's when Lawvere tried to re-formulate the foundation of mathematics / set theory in the new language of category theory.

In a topos, 3 operations are allowed between objects:

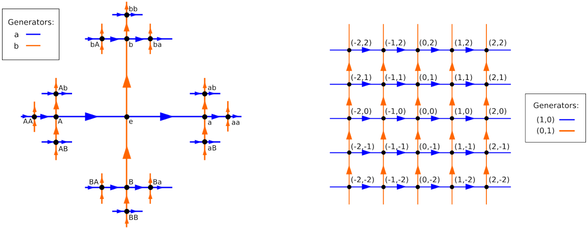
- Cartesian product $A \times B$ (corresponding to $A \wedge B$ in propositional logic)
- exponentiation $A \to B = B^A$ (corresponding to $A \Rightarrow B$ in propositional logic)
- subobject classifier $A \hookrightarrow B$ (corresponding to the subset notion $A \subseteq B$)

The significance of topos theory is that it distills which mathematical structures are able to carry on logic operations, e.g. relation graphs, algebras, etc.

In my theory, the neural network F implements $U \xrightarrow{F} V$ which is the exponentiation V^U, where U, V are vector spaces. This requires, at least, to embed the structure of $A \times B$ into vector space. However $A \times B = B \times A$ is commutative, which led me to consider Abelian group theory....

Permutation invariance

- Recently a nice idea has been proposed to embed Word2Vec into Poincaré disc / hyperbolic space [Nickel and Kiela 2017]
- Can we similarly embed logic structures (as vectors) into hyperbolic space?
- Cayley graph of the free group F_2 is a tree, but Cayley graph of the Abelian free group is grid-like:



(12)

- Generally, the free group F_n's Cayley graph can be embedded into the (planar) hyperbolic disc, but the Abelianization of $F_n = F_n^{\text{Ab}} \cong \mathbb{Z}^n = \mathbb{Z} \times \ldots \mathbb{Z}$ is a n-dimensional grid, which seems impossible to embed on a plane.
- Even considering representation theory, all irreducible representations of Abelian groups are 1-dimensional. The representation of F_n^{Ab} is precisely the direct sum of n copies of dim-1 representations. Useless!

- Seems that $\mathbb{Z}^n$ cannot be embedded into lower dimensions, unless we use some kind of fractal structure. However, fractals are exactly the realm that is out-of-reach of the universal approximating power of neural networks!
- Another strategy is to create symmetric neural networks whose output is invariant when the input is permuted. This could be achieved by “weight-sharing”.
- For this to work, the activation function must be polynomials.
- A drawback of this approach is when # layers grow, # constraints also grow exponentially, making it hard to build deep (many-layer) networks
- An advantage is that the resulting NN is very sparse in terms of # weights. This is a form of inductive bias.
- As of this writing, I am yet exploring another approach that is inspired by Google’s BERT. More on this later.

Field, JH (2010). “Derivation of the Schrödinger equation from the Hamilton-Jacobi equation in Feynman’s path integral formulation of quantum mechanics”. In: *European Journal of Physics*.

Nickel, Maximillian and Douwe Kiela (2017). “Poincaré embeddings for learning hierarchical representations”. In: *Advances in neural information processing systems*, pp. 6338–6347.

Thanks for watching ☺️