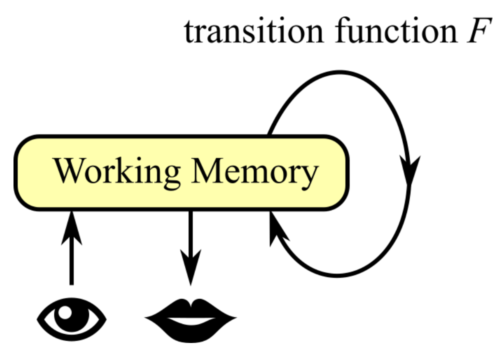
①

AGI practical framework

In classical logic-based AI, inductive learning means searching for a theory T (= set of logic rules) that "explains" or implies positive examples but not negative ones:

$$T \vdash e^+, \quad T \nvdash e^- \tag{1}$$

While logic learning is powerful, it relies on combinatorial search and was too inefficient, which caused "AI Winter".
The advent of deep learning may change this. The following is our basic RL (reinforcement learning) setup[a]:

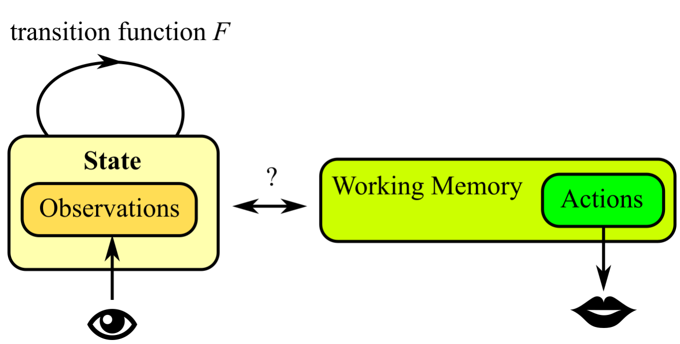


(3)

WM (working memory) is updated by a transition function. If this transition function is trained in a closed loop, it may be able to "explain" (in the sense of predicting) the input data.

---

[a]The above diagram is somewhat inaccurate as there are subtle differences between the notions of "state" and Working Memory. In RL, state usually refers to the external environment through observations. WM is slightly different in the sense that an internal belief may be wrong about the environment – eg, mistaking seeing something that doesn't exist. There is on-going research as to the relation between RL and WM. I have not fully resolved this issue.



(2)

I want to argue that the function F can be trained to perform like the set of logic rules T.

0 Introduction

The "standard model" is a way of thinking, that may help us better understand the general theory of AGI systems.
The essence of the standard model is just to identify a Working Memory as the "state" of the AGI system.
One benefit of our theory is that it relates Transformers / BERT / GPT to AGI systems. These language models are phenomenally intelligent, yet many people criticize them as not "truly" intelligent. The standard model suggests that they are indeed linked to AGI.

1 Reinforcement learning

This is the simplest form of a dynamical system:

$$\begin{tikzcd} \boxed{\text{state } x} \arrow[loop above, "\text{transition function } F"] \end{tikzcd} = \text{"working memory"} \tag{4}$$

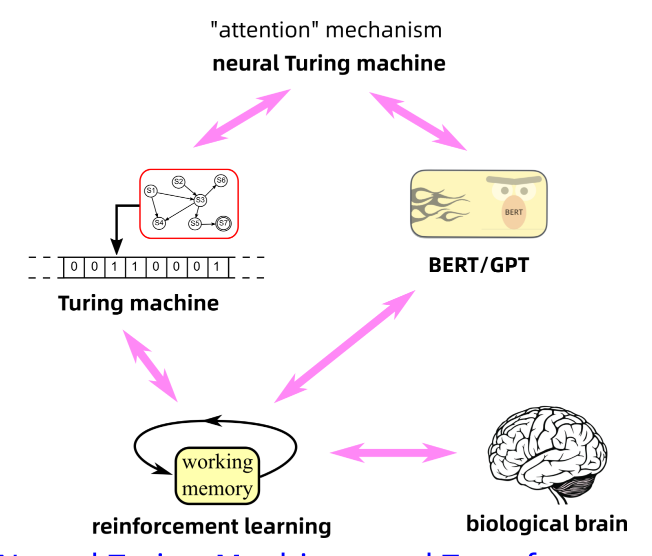
When we add a "control" or "action" variable a to it, it becomes the most basic control system:

$$\begin{tikzcd} \boxed{x} \arrow[loop above, "F(x,a)"] \end{tikzcd} \tag{5}$$

which is the setting for Dynamic Programming or Reinforcement Learning. The optimal solution for such systems is governed by the Hamilton-Jacobi-Bellman equation * :

$$V^*_t = \max_a \mathbb{E}[R_{t+1} + V^*_{t+1}] \tag{6}$$

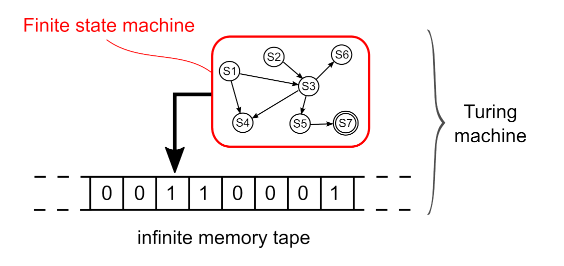
TO-DO: It would be worthwhile to find the brain mechanism that approximates reinforcement learning and use it to help the design of AGI.
Recently, Yann LeCun's Energy-Based Models offers a way to circumvent the problem of learning probability distributions over actions, when the action space is hugely high-dimensional. This seems to be an important step towards AGI systems.
I call this the "standard model" because of the extreme simplicity of this setup, and that I don't know of other alternative models that deviate much from it.
The following diagram shows how the standard model relates to several other important areas, so we can reap profits from their interactions:



(7)

2 Neural Turing Machines and Transformers

The attention mechanism was first proposed in the "Neural Turing Machine" paper by Graves et al [2014].
Recall that a Turing machine is a Finite State Machine augmented with a Memory Tape:

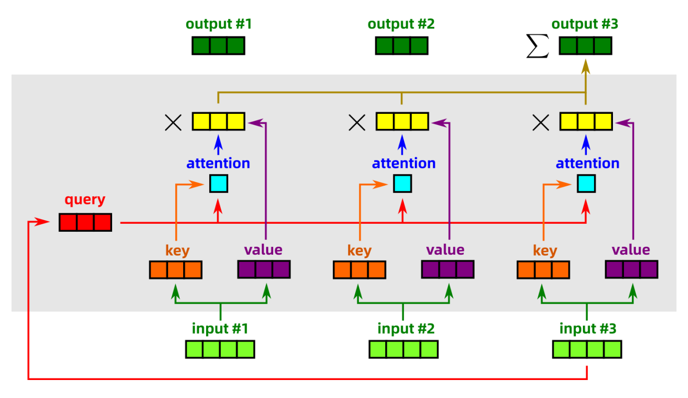


(8)

In Neural Turing Machines, Graves et al proposed the attention mechanism for an RNN "Controller" (playing the role of the Finite State Machine) to read and write from a Memory Matrix (the tape), using a content-based addressing method.
The Memory Matrix M consists of N items, each of constant size. The discreteness of the address would introduce discontinuities in gradients of the output, hence we need an Attention Vector to focus on a specific location in the memory matrix M.
The Attention Vector $\vec{a}$ is calculated via the following formula, familiar to students of the Transformer:

$$\vec{a} = \text{soft}\max_i \{\mathcal{D}(K, M_i)\} \tag{9}$$

where $D()$ is a similarity measure between the key K and memory item M_i.
The key K is emitted by the Controller as the value that it is looking for.
This then evolved into the Self Attention mechanism used in all Transformers. Now let us refresh with this diagram illustrating Self-Attention (redrawn from a blog article on the web):



(10)

The research done by Olah et al, in their 2021 paper A Mathematical Framework for Transformer Circuits, is very helpful towards understanding Transformers and Self-Attention.
For example when we say "all men are mortal":

$$\forall x.\ \text{Human}(x) \Rightarrow \text{Mortal}(x) \tag{11}$$

any object instantiated as x (eg. "Socrates") would have to be copied from the LHS to the RHS.

3 Relation to the biological brain

There are two distinct aspects in the brain:

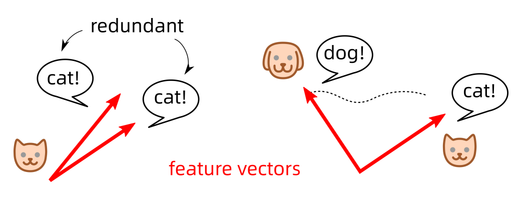
- Short-term or Working Memory is the electric activation of neuronal populations.
- Long-term memory is stored as synaptic strengths, established by synaptic formation and strengthening. The transfer from STM to LTM is called memory consolidation.

One theory has it that the prefrontal cortex maintains a number of "thoughts" with sub-populations or, perhaps, with micro-columns. These activated sub-populations are in competition with each other, through lateral inhibition. The thought(s) that win are the thoughts we retain – they "make sense".

3.1 How does symbolic logic emerge in the brain?

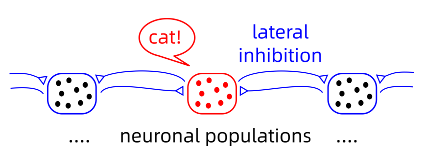
Disentangled features

If a room of people see a cat enter the room, one person will say "There's a cat in the room!" but afterwards it would be redundant for others to say exactly the same thing. Likewise, in a neural network, if two output features both identify "cat" then they are redundant, a waste of resources. So it is more efficient for one feature vector to move away to a new location in feature space:



(12)

The result is the emergence of disentangled features. There is now a lot of research papers on this topic; Personally I first learned of this from Marta Garnelo and Murray Shanahan's paper Shanahan2019. We can think of this as a first step of symbolization, in which objects are recognized by symbols.
In the cortex, neuronal populations are organized into "columns", with lateral inhibition among themselves. When one population is activated, it suppresses the activation of nearby populations. This is likely to be the mechanism that enables disentangled features to emerge:

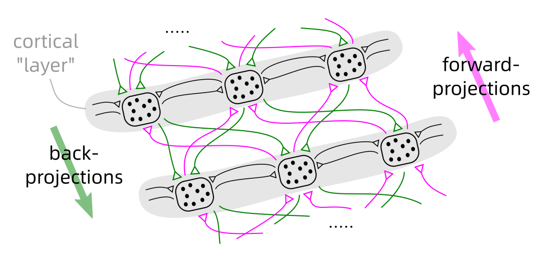


(13)

It is remarkable that the softmax in the Transformer / Self-Attention seems to be an abstract implementation of this winner-takes-all selection mechanism.

Bi-directional connections in the cortex

Moreover, the cortex is organized into layers with widespread recurrent (ie, forward and backward) connections †:



(14)

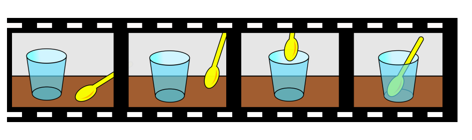
This bi-directional architecture may be applicable to AGI architecture (see also §4 on abductive reasoning), possibly replacing the current unidirectional model of feed-forward networks and the back-propagation algorithm.

Alternative to back-propagation?

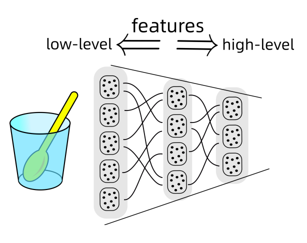
As is well-known, the brain does not use back-prop. The bi-directional innervation is a very significant brain architectural feature that has not yet been incorporated into current deep learning techniques.
In order to find an alternative to back-prop, we need to ask: What is the essence of deep learning? I think the answer lies in two words, "hierarchical" and "learned". As a counter example, decision trees are hierarchical structures that are learned, but the learning algorithm is too slow because it uses combinatorial search (reminiscent of NP hardness).
But the brain must have a roughly equally powerful learning mechanism as back-prop. A likely candidate is resonance. In figure (14) we have a hierarchically connected cortical structure. What we need is some sort of "infinitesimal" learning rule.

Hierarchy of features

If we consider relations between objects, for example, "spoon inside a glass", this too can emerge out of disentanglement of features, because it is a very economical / efficient representation of a complex scene:

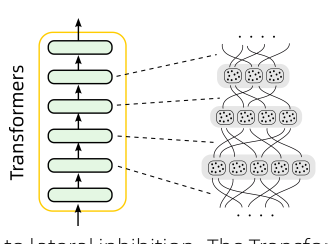

(15)

Every human can recognize this as "putting a spoon into a glass", a symbolic representation. Many researchers may have under-estimated how much the brain uses symbolic reasoning, and my proposal is that AGI can be based entirely on it.
One remaining question is how to represent symbolic data in a "neural" manner. A general form of symbolic data may be as a tree. Taking inspiration from the cortex (14), we may perhaps represent the tree / symbolic data as hierarchically organized neural feature vectors:



(16)

Remember that in the Transformer, symbols are organized as sequences, for example: "spoon · inside · glass." It may be desirable for AGI to have multiple levels of features, such as "spoon" and "glass" on a lower level, and "inside" on a higher level.
Juxtaposed side by side, the Transformer and the cortex seem to have many similarities:

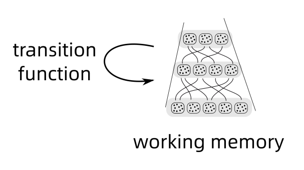


(17)

Softmax corresponds to lateral inhibition. The Transformer has many layers because it unfolds along the time axis the training of a recurrent network – part of the reason why the Transformer is very efficient. Each hidden layer of the Transformer can be construed as a "stage" of logical inference:

$$\text{input} \vdash \text{stage}_1 \vdash \text{stage}_2 \vdash .... \vdash \text{output}. \tag{18}$$

Also recall that our reinforcement learning model consists of just the state and its transition function:

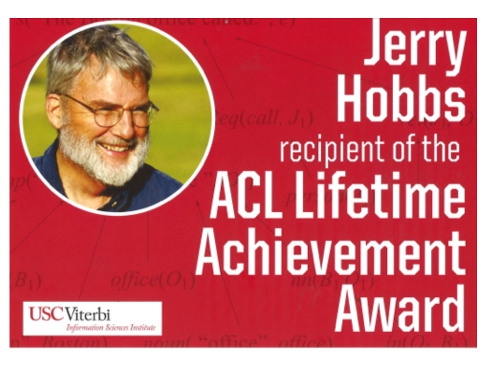


(19)

Based on this understanding, we need to figure out how to design the next version of Transformer and incorporate it into our AGI architecture....

4 Abductive reasoning

Abduction has been relatively neglected in AGI research, which had focused on forward inference. Recently there is a call to study this important aspect.
In logic, abduction means finding the explanation for some known facts. An explanation E is simply some propositions that imply the known fact F, ie, $E \Rightarrow F$.
For example, why do we think a certain actress, say Marilyn Monroe, is "sexy" ** ? That's because we recognize she has some features (visual or otherwise, no need to enumerate them explicitly) that we consider sexy. So, $E_1 \wedge E_2 \wedge .... \Rightarrow$ Sexy. Those conditions imply she is sexy, and they are the explanation for her sexiness.
Why is abduction important? For example, when a waitress says "The Ham Sandwich left a big tip", Ham Sandwich here refers to the customer who ordered it (an example of metonymy). The AI knows the plain facts such as that someone ordered a ham sandwich, and then it abduces that the most likely interpretation of the phrase "Ham Sandwich" is as the person associated with it. This is the basis of Abductive Interpretation of Natural Language proposed by Jerry Hobbs:



(20)

So abductive reasoning is basically just bidirectional inference.
When a system has both forward and backward connections, it forms a loop and its dynamics is likely to produce "resonance". This harks back to the ART (Adaptive Resonance Theory) proposed by Grossberg and Carpenter beginning in the 1980s.
Such resonance behavior can be viewed as the system seeking to minimize an energy, ie, trying to find the "best explanation" to a set of facts.
This is also corroborated by neuroscientific evidence: areas in the cerebral cortex are replete with both forward- as well as back-projections, as depicted in diagram (14). We can further abstract this with the following diagram, where F and G are not functions but optimization constraints:

$$\begin{tikzcd} \vec{y} \arrow[d, bend left, "G"] \\ \vec{x} \arrow[u, bend left, "F"] \end{tikzcd} \tag{21}$$

If the input $\vec{x}$ produces the output $\vec{y}$ after some iterations, then it is likely that the output $\vec{y}$ would produce $\vec{x}$ in the inverse direction. In other words, we have a neural mechanism that implements a function f and its inverse f^{-1}. The significance of this (from the learning point of view) is that we only need to learn the function f and we get f^{-1} for free.
In logic, if forward inference is denoted as $\vdash_{KB}$, where KB is a set of logic rules, then abduction is $(\vdash_{KB})^{-1}$. Abductive interpretation is basically a constraint-satisfaction process that uses inference rules in both directions.

5 Dealing with assumptions