AGI via Combining Logic and Deep Learning

AGI 2021 Conference

YKY

October 12, 2021

Table of contents

Part I

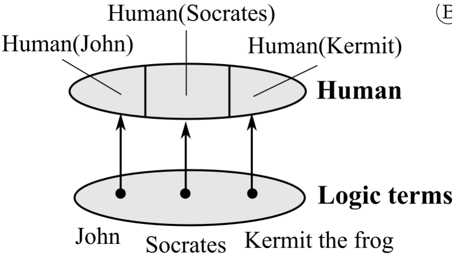
Categorical Logic

Categorical Logic

- Quantifier as adjunctions

Fibrations

- Fibration

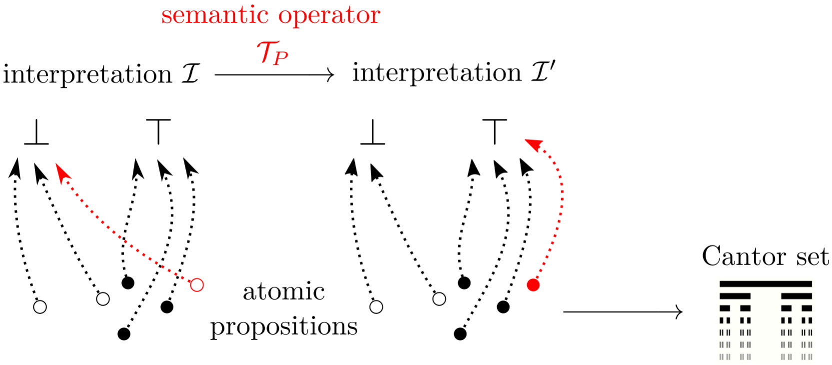


Part II

Model-based vs Syntax-based

Hitzler's Core Method

- The **semantic operator** $\mathcal{T}_P$ updates an interpretation to another interpretation:



Hitzler's Core Method

- The **level mapping** ι takes an interpretation to a real number:

$$\begin{tikzcd}
\mathcal{I} \arrow[r, "\mathcal{T}_P"] \arrow[d, "\iota"] & \mathcal{I} \arrow[d, "\iota"] \\
\mathcal{K} \arrow[r, "f_P"] & \mathcal{K}
\end{tikzcd}
\qquad
\iota(\mathcal{I}) = \sum_{\text{atom} \in \mathcal{I}} b^{-\ell(\text{atom})} \in \mathbb{R}$$

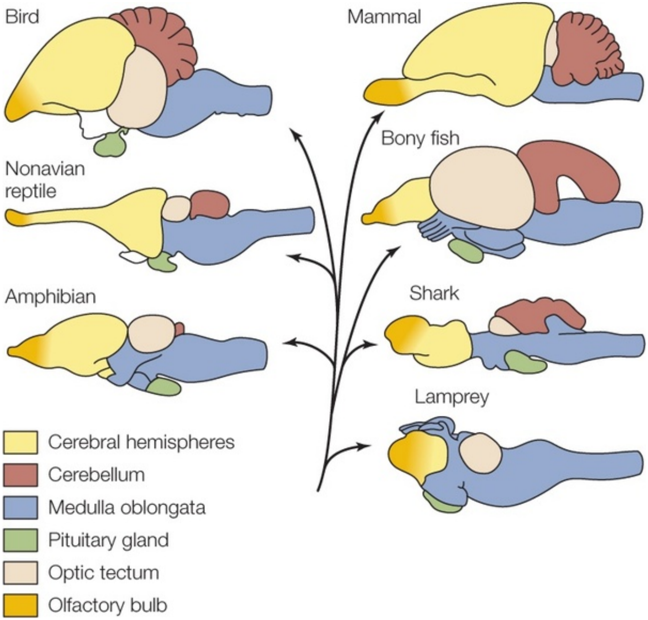
approximated by
neural network

Part III

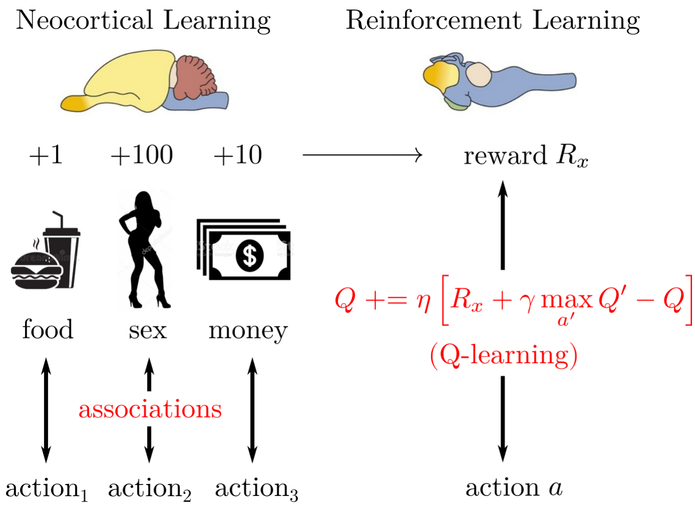
Reinforcement Learning

This is a lamprey

Evolution of the Neocortex



Neocortex vs Reinforcement Learning

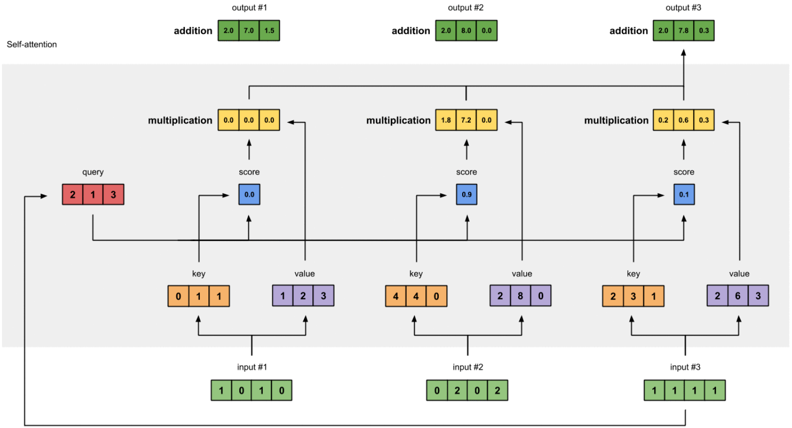


Part IV

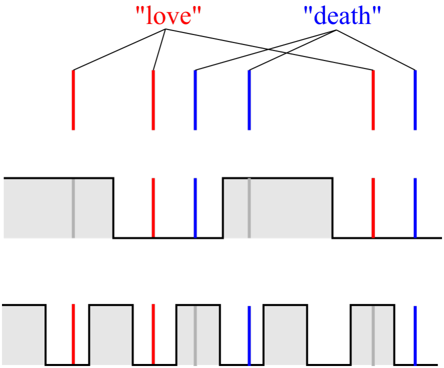
BERT / GPT

Equivariance of the Transformer

- Many people already know this:
  the Transformer module is **equivariant** in its inputs and outputs:



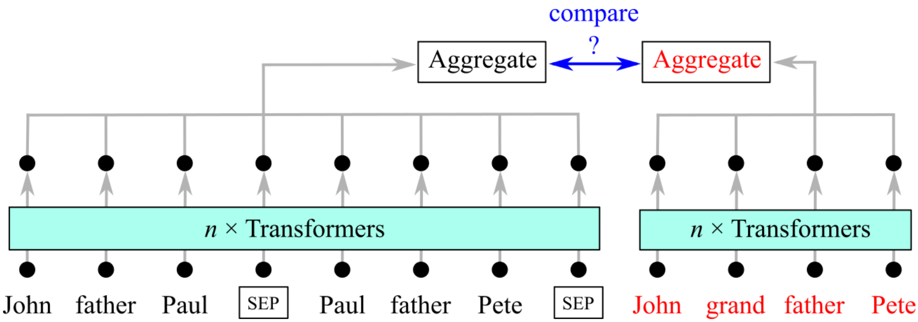
Additivity of Word Embeddings



- This suggests that word embeddings can be added together to give **composite** meanings

BERT's Hidden Representation

- This may be a close description of how BERT performs question-answering or logic deduction:



Part V

BERT + Knowledge Graphs

Memory Formation

- Memory

Memory Recall

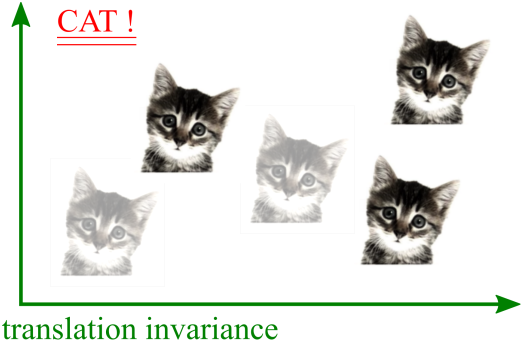
- Memory retrieval

Part VI

Old Stuffs

The success of CNN in computer vision

- In geometry, vision is said to possess the property of **translation invariance**:



(1)

- **Convolution** is an operation invariant under translation:

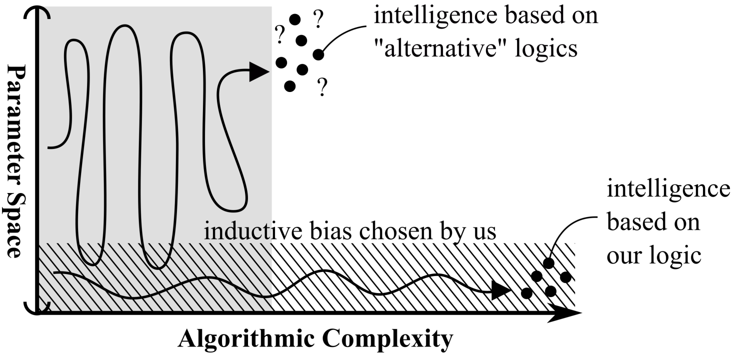
$$(T_x \circ f) * g = T_x \circ (f * g) \tag{2}$$

- Yann LeCun *et al* exploited the symmetry of CNNs to accelerate learning, successfully solved the visual recognition problem

(3)

Symmetry and inductive bias

- In mathematics, symmetry often simplifies computation, which is why mathematicians love to study symmetries
- In machine learning, one introduces **inductive bias** to narrow down the **search space**:



(4)

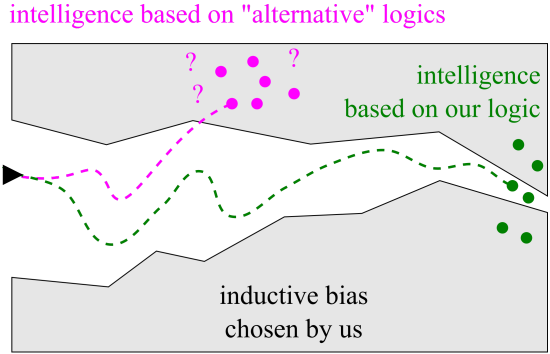
- Oftentimes, if inductive bias is chosen correctly, solution is found quickly, otherwise problem becomes **intractable**

Richard Sutton's view

- In contrast, Sutton expressed the view that AI can be solved merely by **increasing computing power**, under the reinforcement learning framework

(5)

- Our choice is just one out of many possible **forms** of logic:



(6)

- This is not only a theoretical issue;
  Indeed, AI labs around the world had begun the search for AGI with various strategies!

Doubts about logicism

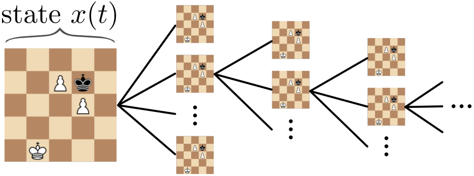
- Admittedly, the human brain does **not** think like symbolic logic
- By symbolic logic, we mean any "linear" (sequential) representation
- In contrast, **pictures** and **music** are *multi*-dimensional
- The brain may use representations that are not sequential or symbolic
- However, logic is a very efficient representation (from a computer perspective) even though it deviates from the biological brain
- Neuroscience is too difficult to crack, thus not a practical route to AGI
- Human cognition may be much closer to logic than we've thought. If we're given a description: "woman finds new lover, murders husband":

(7)

We may not know: Is the woman blonde? What is she wearing? In other words, our "mental model" is devoid of details and it may not possess any more information than just a sequence of symbols.

Reinforcement learning

- Think of the "**state**" in reinforcement learning like a **board positon** in a chess game:



(8)

- Reinforcement learning seeks to maximize the total **rewards** accrued over a (possibly infinite) **time horizon**:

$$\text{maximize } S = \int_0^\infty L \, dt \tag{9}$$

- Such maximization gives the AI **intelligence** because it is often beneficial to **delay** rewards, eg: to plot a clever chess move
- But reinforcement learning is a **brute force** approach; we need to give the model some additional **inductive bias**, eg, in the form of **logical structure**

Structure of logic

- The idea is: introduce symmetries of **logic** into deep learning to solve the AGI problem
- Because human cognition has logical structure, this inductive bias may help us find a solution to AGI faster
- Logic is a complicated structure, but its simplest symmetry is the **commutativity** (or permutation invariance) of **propositions**:

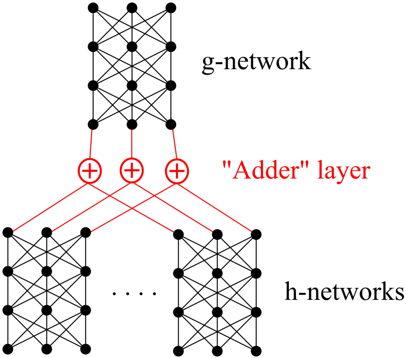
$$\begin{aligned} A \wedge B \quad &\equiv \quad B \wedge A \\ \text{it's raining} \wedge \text{lovesick} \quad &\equiv \quad \text{lovesick} \wedge \text{it's raining} \end{aligned} \tag{10}$$

- Its importance may be analogous to translation invariance in vision
- The significance of commutativity is: it **decomposes** the AI system's **mental state** into individual **propositions**

Symmetric neural networks

- Permutation invariance can be handled by **symmetric** neural networks
- I wasted 2 years trying to solve this problem, and then found out it had been solved 3 years before: [PointNet 2017] and [DeepSets 2017] and their mastery of mathematics is significantly above me!
- Any symmetric function can be represented by the following form (a special case of the Kolmogorov-Arnold representation of functions):

$$f(x, y, ...) = g(h(x) + h(y) + ...) \tag{11}$$



(12)

- Sym NN gives a powerful boost in efficiency $\propto n!$ where $n = \#\text{inputs}$
- The code for Sym NN is just a few lines of Tensorflow:

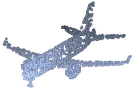
```
h = Dense(3, activation='tanh')
ys = []
for i in range(9):
    ys.append( h(xs[i]) )
y = Keras.stack(ys, axis=1)
Adder = Lambda(lambda x: Keras.sum(x, axis=1))
y = Adder(y)
g = Dense(3)
output = g(y)
```

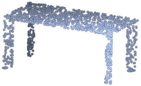
(13)

- Very easy to adopt this to existing models such as BERT and reinforcement learning
- I have successfully tested it on the game of TicTacToe: https://github.com/Cybernetic1/policy-gradient

For example: symmetric NN for object recognition

- Imagine objects represented as **point clouds**:

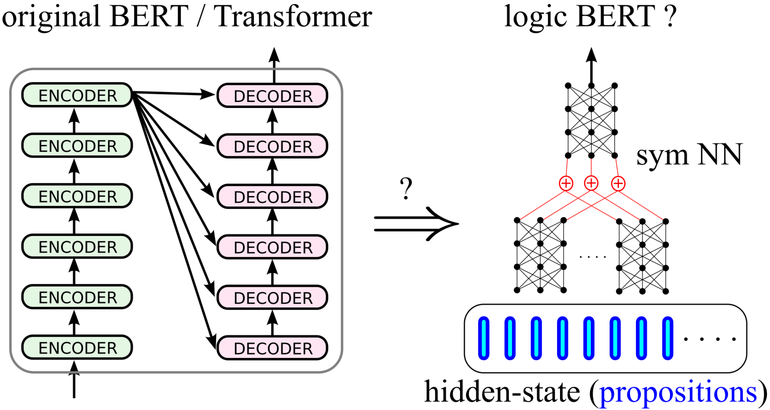
 $\xmapsto{f}$ "aeroplane"

 $\xmapsto{f}$ "desk"

- It does not matter **in what order** the points are in a sequence; the function $f(x_1, ..., x_n)$ is symmetric in its arguments (the points)
- Permutation invariance is **essential** for this to work

Logicalization of BERT

- Similarly, we can convert BERT's hidden state into a set of propositions, by replacing the original **decoder** with a sym NN:

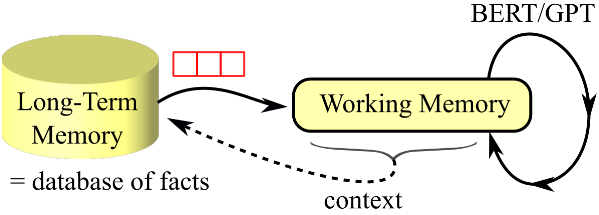


(14)

- The original **encoder** can be retained. As the **decoder** imposes symmetry on the hidden state, error propagation is expected to cause its representation to change
- Of course, this remains to be proven by experiment ☺

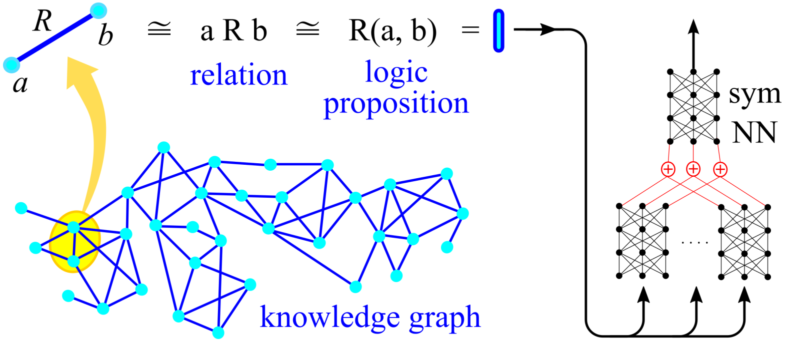
Advantages of logical AI (1)

- With logic, it becomes easy to design cognitive architectures, eg: long-term memory module



(15)

- Integrate seamlessly with **knowledge graphs**
- graphs are made up of edges,
  edges = relations between nodes = propositions:



(16)

References

Thanks for watching ☺️

Illustration credits:

- Translation invariance, from Udacity Course 730, Deep Learning (L3 Convolutional Neural Networks ▷ Convolutional Networks)