Paper: AGI from the perspectives of categorical logic and algebraic geometry

YKY

August 12, 2024

Contents

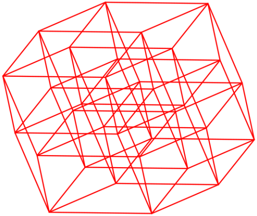
Part I

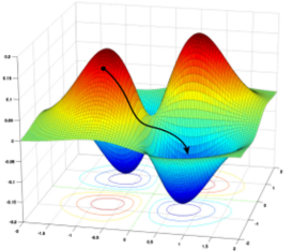
How does “No Free Lunch” guide us to accelerate AGI?

The hypothesis space, ie. the space of AGIs

- In deep learning, our hypothesis space Θ = neural networks = parameter space = weight space = $\mathbb{R}^n$ or $[0,1]^n$ if truncated.
- It looks like this (left):

$\Theta =$

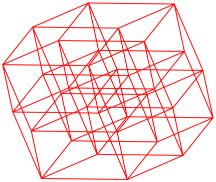




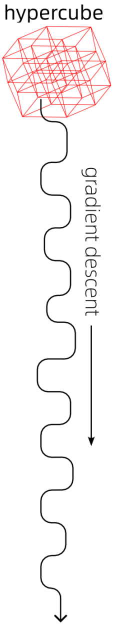
- Imagine a loss function as sitting above the hypercube; We seek to minimize it by **gradient descent**.
- When people visualize gradient descent, they tend to think of the above figure (right), but in practice, the "ups and downs" along one dimension is *dominated* (overshadowed) by the sheer number of dimensions.

What can "No Free Lunch" say about the space of AGIs?

- We believe a fair coin toss has probability 1/2 by making an **ignorance assumption** and applying the maximum entropy principle. The ignorance assumption says that we have no reason to believe that one side of the coin is more probable than the other.
- By a similar argument,



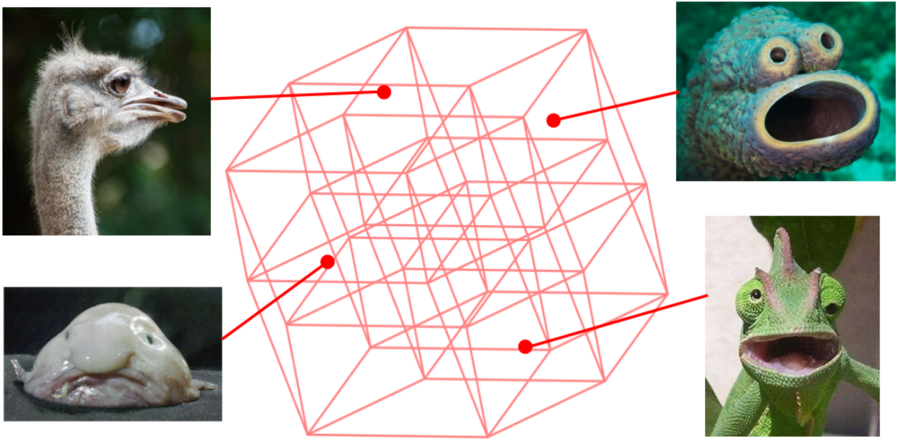
Unreadable effectiveness of gradient descent



- Gradient descent can keep going down *for a long time* as there are millions of dimensions.
- The "unreasonable" effectiveness of gradient descent in deep learning is currently our most powerful weapon to solve the AGI problem.
  Its explanation is likely very difficult, as it hinges on the P =? NP problem.

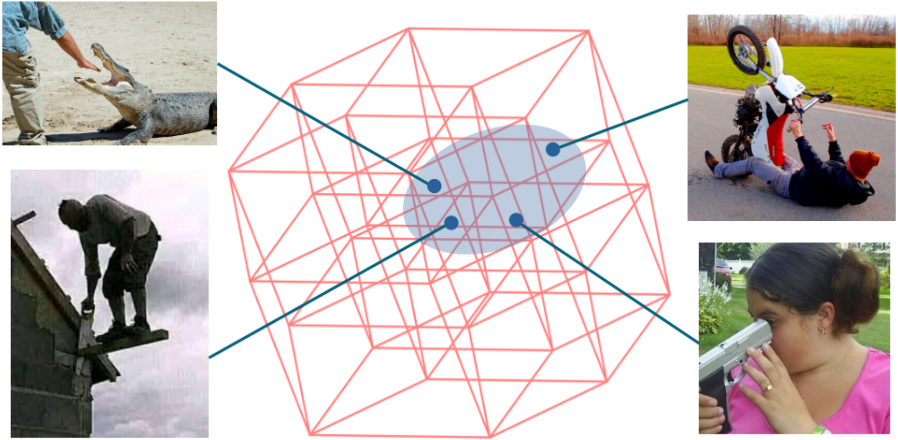
The hypothesis space has plenty of “dumb structures”

- It seems that dumb learning-machine structures (animals) are more common than smart ones



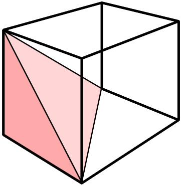
Even if we have found an intelligent structure, it may still contain lots of "dumb knowledge" in that sub-space

- It also seems that dumb knowledge is more common than smart knowledge



A motivating example: symmetric neural networks

- Permutation symmetry or commutativity ($ab = ba$) is the best-known symmetry in mathematics
- The input space, as a hypercube, is symmetric under exchange of vertices. The **fundamental domain** of this symmetry is one *corner* of the hypercube:

  (Note that this hypercube is the *input* space, not the mapping / hypothesis space)
- Building a symmetry into a neural network may seem a daunting task, but recent research shows that merely **projecting** all data points to the fundamental domain has the same effect as achieving symmetry [2].

A motivating example: symmetric neural networks

- If we restrict to binary logic, the total number of Boolean functions in n variables is 2^{2^n} whereas the number of **symmetric** Boolean functions is 2^{n+1}. So we clearly see an *exponential* reduction in the size of the hypothesis space.
- Tantalizingly, the Transformer also has this permutation symmetry, and researchers have proposed that Transformers are capable of performing *syntactic* manipulations similar to those in proofs of symbolic / predicate logic.
- We can further relax the restriction of binary logic to k-ary logic, ie, increase the number of lattice points in the hypercube, and the above analysis still holds qualitatively.

Part II

Categorical logic, and fibrations in particular

Curry-Howard isomorphism

Many readers are already familiar with this, so I'll just highlight some interesting points...

- The starting point of the Curry-Howard isomorphism is the following correspondence:

$$\boxed{\text{implication in logic}}\ A \to B \qquad A \xrightarrow{f} B\ \boxed{\text{function in type theory}}$$

- An example in type theory: "**currying**" means to convert a 2-argument function into a single-argument function which returns another function, ie. $f(x, y) = (g(x))(y)$.
  The types of f and g are equivalent, ie,
  $X \to (Y \to Z) \simeq (X \times Y) \to Z$.
  If we treat the above as logic it becomes:
  $X \to (Y \to Z) \equiv (X \wedge Y) \to Z$. One can verify this is a valid logic formula.
- This and other amazing coincidences led us to believe that CHI has some deep truth in it.

Categorical logic **clashes** with algebraic logic

- The late Joachim Lambek proposed a "trinity" between category theory, logic, and type theory [3]:

$$\begin{tikzcd} & \boxed{\text{category theory}} \arrow[dl, dash] \arrow[dr, dash] & \\ \boxed{\text{logic}} \arrow[rr, dash, "Curry-Howard"] & & \boxed{\text{type-theory}} \end{tikzcd} \tag{1}$$

- But the Curry-Howard approach seems, at least on the surface, *incompatible* with the **algebraic logic** approach, for example:

$$\begin{tikzcd} \overset{f : A \to B}{\boxed{\text{type theory}}} \arrow[dr, dash, squiggly, "incompatible"] & \\ \overset{a \to b}{\boxed{\text{logic}}} \arrow[u, "Curry-Howard"] \arrow[r] & \overset{a \oplus ab = 0}{\boxed{\text{algebraic logic}}} \end{tikzcd}$$

- I have been working in the Curry-Howard direction because it feels more "modern" but lately I began to think the algebraic logic approach may have more potential for computational acceleration.

Definition of a **fiber bundle**

A **fiber bundle** is a tuple $\xi = (E, p, B, F)$:

(i) E = **total space**

(ii) B = **base space**

(iii) F = a topological space called the **fiber** of ξ

(iv) $\begin{array}{c} E \\ \downarrow p \\ B \end{array}$ is a continuous surjective map, called the **projection**

(v) for each point $b \in B$, the inverse image $p^{-1}(b) = F_b$, called the **fiber over** b, is homeomorphic to F

(vi) B has an opening covering $\{U_a\}_{a \in A}$ such that for each $a \in A$, there is a homeomorphism: $\psi_a : U_a \times F \to p^{-1}(U_a)$.
If F is a discrete space, then the structure $F \hookrightarrow E \xrightarrow{p} B$ is called a **covering** of B.

The condition (vi) is just a re-phrase of (v) in the form of open sets, similar to the "gluing" together of charts in differential manifolds. So the essential condition is (v).

The "boring" cross product

token *token* *token*

is equivalent to $A \times A \times A$.

Part III

The "algebraic logic" approach

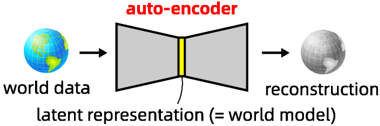
Algebraic logic

- Traditionally, algebra (and algebraic geometry) is concerned with systems of **polynomials** over the "classical" number fields $\mathbb{R}$ or $\mathbb{C}$. The polynomials are built upon addition and multiplication of numbers, that we're all familiar with.
- Propositional logic $(B, \wedge, \vee, \neg, \top, \bot)$ can be equated with **Boolean rings** $(B, \cdot, \oplus, -(), 0, 1)$, with ring multiplication identified as $\wedge$ and ring addition identified as symmetric difference ("XOR").
- As predicate logic involves more operators (eg. $\forall$ and $\exists$), the classical algebra of numbers seems inadequate to represent them.
- For example, **cylindric algebra**[1] has additional operations c_i (cylindrification) and Id_{ij} (diagonal).
- Notice that algebra does not appear in the "trinity" (1). The Curry-Howard approach is not directly related to algebraic logic.

[1] see Wikipedia for a quick intro

“Algebraic-logical” neural networks

- This is the basic setup of an auto-encoder which is the common setting for all current LLM models:

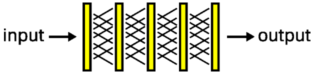


- The predictor / model can be construed as a logical **knowledge-base** (KB) making inferences:

$$\boxed{\text{input state}}\ \Gamma \ \vdash\!\!\xrightarrow{\text{KB}}\ \Delta\ \boxed{\text{new conclusion}}$$

$$\Gamma \xrightarrow[\text{LLM / GPT / etc}]{\text{deep NN}} \Delta$$

- A neural network is composed of multiple layers:



- We can replace the activation functions with **polynomials**, creating “**algebraic**” neural networks. Pushing **Curry-Howard** to its extreme, perhaps each layer of algebraic NN can be identified as a logical KB?

Algebraic-logical neural networks (2)

- Andréka-Németi [1] developed a **general framework** for studying algebraic logic, based on model-theoretic semantics. A logic is $\langle F, M, mng, \vdash, \Vdash \rangle$ where F = set of formulas, M = class of models or possible worlds, mng = meaning function: $F \times M \rightarrow$ Sets, $\vdash$ = syntactic provability, $\Vdash$ = semantic consequence, usually defined via mng.
- For example, in first-order logic, $mng(\psi)$ = set of all **evaluations** of variables $\vec{x}$ such that the formula $\psi(\vec{x})$ is satisfied in $\mathfrak{M} \in M$.
- For our purpose, let's start with a simple logic formula: $\forall x.P(x) \rightarrow Q(x)$. It can be implemented in **vector-matrix form**:

$$\begin{pmatrix} P \\ x \end{pmatrix} \begin{pmatrix} M \\ \mathrm{Id} \end{pmatrix} = \begin{pmatrix} Q \\ x \end{pmatrix}$$

where = element-wise activation function, M = transformation matrix.

- This is a bit tricky because our algebraic logic include terms that appear as input-output objects, as well as **maps** between such objects. But the Andréka-Németi framework can still be applied in essence.

Algebraic-logical neural networks (3)

- I just newly discovered this and need more time to explore.
- What we have here is very interesting: on the one hand we have a deep neural network capable of **gradient descent**; on the other hand it is also an algebraic logic.
- Our goal remains the same: to accelerate learning.

References

Thanks for watching ☺️

[1] H. Andréka, I. Németi, and I. Sain.
Universal Algebraic Logic: Dedicated to the Unity of Science.
Studies in Universal Logic. Springer Basel, 2021.

[2] Aslan, Platt, and Sheard. "Group invariant machine learning by fundamental domain projections". In:
Proceedings of Machine Learning Research 197 (2023), pp. 181–218.

[3] Lambek and Scott. Introduction to higher order categorical logic. Cambridge, 1986.