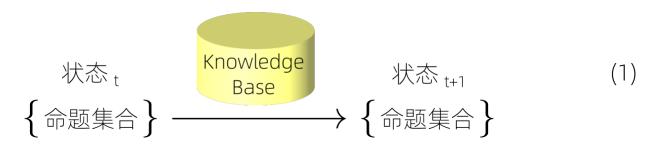
\bigcirc

逻辑与深度学习的关系

这是经典逻辑 AI 的最基本运作模式:



它其实包含了两个算法:

• matching (unification):

逻辑 rules 是包含变量的条件命题,

例如 $\forall x. 是 \land (x) \Rightarrow \Leftrightarrow \mathcal{R}(x).$

Unification 判定一条 rule 是否可以 apply 到某逻辑命题上,

例如:是人(苏格拉底)可以跟上式的左边 unify.

Matching 的结果是得到一推 instantiated (特例化,即不包含变量)的命题。

• forward- or backward-chaining (resolution):

由已知事实推导出新结论,或反过来,判断某给定的新结论是否成立。

例如: 是人(苏格拉底)⇒会死(苏格拉底) ∧是人(苏格拉底)

可以推出:会死(苏格拉底)。

深度学习的特点,就是将

状态
$$_{t+1}$$
 (2)

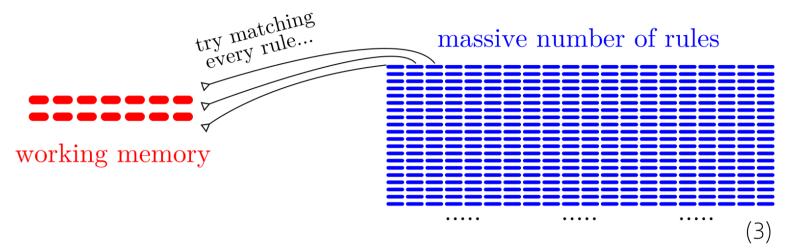
的逻辑推导过程,通通纳入进去一个非常复杂的非线性函数(= 深度神经网络)里面。这样做以后,上述的逻辑结构被"mingled"在一起,以至于很难分辨了。但也正是由于这种「大杂烩」,深度神经网络将一套复杂的组合算法压缩成数量不算太多的一层层的参数。它同时可以做 learning 和 inference这两个动作。这种简单粗暴的方法,其实非常有效率,要超越它的速度并不容易!

我们知道(或推测)一个智能系统 应该具有 符号逻辑的结构。这点知识可不可以用来 约束/加速 深度神经网络?答案似乎是有可能的。现时 state-of-the-art 处理 视觉的 CNN 和 处理文字的 BERT,它们都有内部结构,而不是fully-connected,而且 这内部结构 对应于 被处理的资料的结构。因此我们有理由相信,逻辑结构 可以用来约束 深度神经网络的结构,达到加速。



接下来我们详细一点看逻辑系统的结构:

Knowledge Base 里面有很多 rules,系统要将这些 rules 逐一 match with 系统状态 (= working memory) 里面的命题:

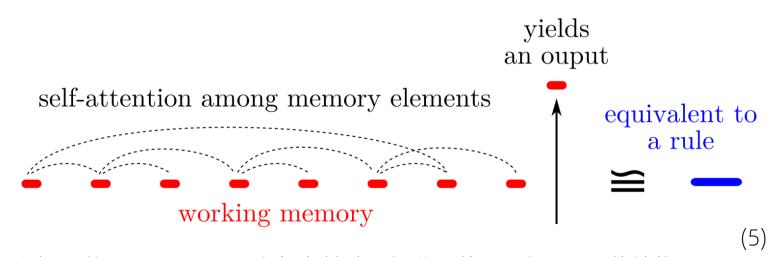


成功 matched 的 rules 可以导出新的结论,加进 working memory 的状态 里面。

这个复杂的操作,完全被一个神经网络取代。或者可以更抽象地说:

状态
$$_{t}$$
 \longrightarrow 状态 $_{t+1}$ (4)

而以 Transformer 来说,它是一种输入元之间的记忆体(这记忆就储存在 Q, K, V 矩阵里),而它 implicitly 做到了 rules 的作用:



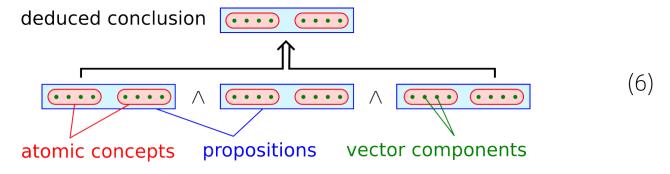
换句话说, Transformer 内部有某种(扭曲了的)逻辑 rules 的结构。那么很自然的问题就是:能否发掘更多逻辑/逻辑系统的结构?也就是说,公式(4)可以有怎样的代数结构约束?这个问题可以参考范畴逻辑的理论,还有经典 logic-based AI 系统的理论。

我们希望 勾画出公式 (4) 需要具备的代数约束,但暂时先用文字描述比较容易:

- 状态是 颗粒化 的,它是某集合的元素,元素之间可交换,也就是 Transformer 的 equivariance. (注意: Transformer 有 equivariance, 但 equivariance 未必一定要用 Transformer 实现)
- 深度结构:例如多层网络,每层是函数的复合 (composition). Transformer 也用了深度结构。
- 逻辑包括了命题层次和命题内部层次的颗粒化。后者是谓词 (predicate) 逻辑的结构,例如: loves(John, Mary),也可以简单地将它视为代数元之间的乘积,例如: John•loves•Mary, 后者也叫做"word". (不同类别的代数元之间不一定容许乘积,因此有 groupoid 的概念,但暂时来说这细节不重要。) 现时重点是如何将 这两层的颗粒化结构施加到深度神经网络上。
- 逻辑推导每步只产生一个新的结论(或其概率分布),然后这个新的结论,再加入到旧的状态中,作为一个命题集合的元素,而旧状态也要遗忘一些命题,否则需要无限记忆。这跟 Transformer 每次输出一列的tokens 有点不同(虽然我们不太肯定 Transformer tokens 究竟对应于命题还是谓词/原子概念)。
- 逻辑 rule 通常只跟某几个前提有关,其它前提是无关的,例如:眼睛好看 ∧ 鼻子好看 ∧ 嘴巴好看 ⇒ 帅,跟 有钱 或 穷 无关。Transformer 的 softmax 结构似乎也可以排除一些无关的 tokens 的影响。
- (可能还有其他的结构特征....)

逻辑与深度学习

根据我的理论,理想的逻辑形式是这样的(各种元素的数量纯粹示意):



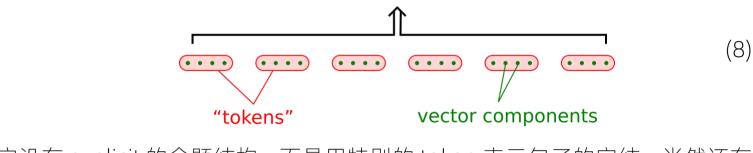
相比之下,我们目前可以写出来的 代数关系 $p \land q = p \land q$ 只表达了这种结构:

这就像问 $0...9 \times 0...9$ 跟00...99的分别(基本上没有分别,它们是isomorphic)。

类似地, { John, Mary } \times { apples, bananas, oranges } 跟 { John likes apples, ..., Mary likes bananas } 等 $2 \times 3 = 6$ 个命题 也是同构的。但前者可以被 \forall 或 \exists 量化,后者是 命题逻辑,不可以量化。

而 Transformer 处理 命题 和 概念 的方式是这样:

deduced conclusion (••••)



它没有 explicit 的命题结构,而是用特别的 token 表示句子的完结,当然还有 positional encoding 这些「伎俩」。所以,Transformer 是一种比较 $ad\ hoc$ 的设计,我们应该可以改进它。

The proposition structure P is made up of atoms a from a free monoid A, and the rule can be quantified by \forall and \exists over such atoms. "The rule can be quantified" means the mapping does not just send one compound element to another element, but that parts of the compound element can be varied and the output element would vary according to the rules of \forall or \exists as adjunctions to substitution maps.

The **Atomic Condition** says that the input and output of a rule are made up of atoms that can vary according to \forall and \exists as adjoint functors.

The **Atomic Condition** says that each $P_i=a_{i1}...a_{ik}$ where some a_{ij} can vary according to \forall and \exists as adjoint functors.

The **Atomic Condition** says that each $P_i = a_{i1}...a_{ik}$ where some a_{ij} can be copied (possibly with some transformation) to another location. And the transformation has to accord with \forall and \exists as adjunctions.

The **Atomic Condition** says that each $P_i = a_{i1}...a_{ik}$ where some a_{ij} can be copied (possibly with some transformation) to another location. And the transformation has to accord with \forall and \exists as adjunctions.

The **Atomic Condition** says that each $P_i = a_{i1}...a_{iK}$ where some $a_{ih} = T_\forall (a_{jk})$ and the transformation T_\forall has to accord with \forall as an adjunction to a substitution functor.

似乎要考虑 的群结构是如何影响 逻辑 rule (6) 的输入和输出? 类似这样:

$$q_1 \cdot q_2 \iff p_{11} \cdot p_{12} \wedge p_{21} \cdot p_{22} \tag{9}$$

它跟这样有什么不同:

$$Q \iff P_1 \land P_2 \quad ? \tag{10}$$

Q 是由两件东西组成,但这两件东西可以换成另外两件。但那就不再是同一个 rule. 如果要维持是同一个 rule,则 任何 \longrightarrow 元素不可以交换或更换。那是不是需要考虑所有的 rules?

是不是需要考虑所有的 rules? 或者可以考虑一个 ∀ rule,它包含很多个 instantiated rules. 而它的 substi-

tutions 总是以 群元素为 unit. 这一点可以怎样表示? 但其实当 rule 改变时,它似乎也是根据群元素的边界而改变的。似乎一定要

涉及大量的 rules 才可以表达 的对称性。

从经典 AI 的角度看,lattice of rules 结构 就是 algebra of atomic concepts 的结构。

的神经网络。

在 范畴逻辑 里面有 Beck-Chevalley 条件和 Frobenius 条件,或许是我们所需的对称性? 但细看之后,发觉还是不能解决问题..... For completeness,我还是描述一下,没兴趣的可以略过。

首先考虑比较容易明白的 Frobenius 条件。在逻辑上,它等于说:

$$\exists x. [\phi \land \psi(x)] \equiv \phi \land \exists x. \psi(x). \tag{11}$$

由于 经典逻辑 AI 普遍使用 ∀ 而忽略 ∃, 我将上式改写成:

$$\forall x. [\phi \lor \psi(x)] \equiv \phi \lor \forall x. \psi(x). \tag{12}$$
 但问题是, (12) 式的左边和右边, 其对应的神经网络 (6) 是一样的(看不出有

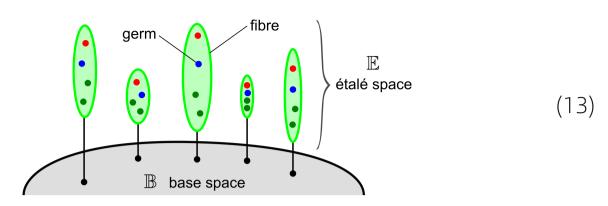
分别)。也就是说这个差别可能太 subtle 了, 它并不影响我们实际 implement

在逻辑里,任何变量 例如 x,y 等,必须被 \forall 或 \exists quantify,否则不成为合法的句子。所以 表达 谓词结构的对称性,也很可能要涉及 \forall 或 \exists .



以前说过,谓词逻辑 带来 **fibration** 或 **indexing** 结构。Beck-Chevalley 和 Frobenius 条件基本上是说, 这 纤维结构 是 "preserved by re-indexing functors".

这是 fibration 结构的示意图:



这整个结构 叫 bundle, 而 sheaf 是 bundle 加上某个特殊的 拓扑结构。

在 (A,f) 和 (B,g) 两个 bundle 之上可以定义 **fibred product** of A and B over I, 记作 $A \times_I B$:

$$\begin{array}{ccccc}
A \times B & \xrightarrow{q} & B \\
\downarrow & & \downarrow & \\
p & & \downarrow & \\
A & \xrightarrow{f} & I
\end{array}$$

$$(14)$$

其中 $h = f \circ p = g \circ q$. 这也是一个 pullback.

Beck-Chevalley 条件是说下面这幅图 commute:

$$K \times J \xrightarrow{u \times id} I \times J$$

$$\downarrow^{\pi} \qquad \qquad \downarrow^{\pi}$$

$$K \xrightarrow{u} I \qquad (15)$$

其中 π 就是代表 量词 \forall 或 \exists 的 投影,它们是 weakening map π^* 的伴随映射。

Beck-Chevalley 条件并不完全是空洞的;它有可能不成立。有一个反例是 Pitts提出的:考虑 $X \times Y$,其中 $X = Y = \mathbb{N} \cup \{\infty\}$ 亦即 自然数加上 ∞ 作为 top element;但 Y 是用 discrete order,亦即所有 order 都是 =. A 是 $X \times Y$ 上的关系: $A = \{(x,y) \in \mathbb{N} \times \mathbb{N} \mid x \leq y\}$.那么 $\exists y.(x,y) \in A$ 会是整个 X 集合。如果考虑 DCPO 范畴,我们要求 fibration of Scott-closed subsets (ordered by inclusion) over DCPO. $\exists y.A$ 的 Scott closure 的条件是它是一个 lower set closed under directed joins;而这个 Scott closure 条件似乎不成立,因而导致图 (15) 不 commute. (我对 Scott closure 的细节不太理解)

Lawvere 的工作将 Beck-Chevalley 条件变得更一般化:「简单」的 \forall 和 \exists 量词 是 weakening functor π^* (基于笛卡尔积)的伴随函子,但 Lawvere 将它扩充到任何 **substitution** functor u^* .