

AGI 大统一理论

0 综述

- 大统一理论是在 **强化学习** 的框架下进行的，这是以 Richard Sutton 为代表人物 提出的理论框架。
- 在 强化学习里 最棘手的一个问题，就是如何 储存和计算 所有 **状态** 之上的 **概率分布**。对 AGI 来说，状态 = 思维空间。我们需要的是 所有可能的思维之上的概率分布，而这是 AGI 的一个硬性需求，无法避免。由于思维空间是高维的向量空间，它上面的概率分布是一个庞大的 mathematical object，很难在计算机上表示。如果用 神经网络 表示，则问题是如何对这个概率分布进行 **采样** (sampling)，在神经网络里，这是很困难的。
- **Hopfield 网络** 的权重 定义了一个 能量地势 (energy landscape)，它可以看成是一个 implicit 的 **概率分布**。透过 Hopfield 网络的 learning，可以改变这个概率分布。但这需要修改 Hopfield 网络的算法，将 能量 诠释成 概率，而这正是 **Boltzmann machine**，也称作 EBM (Energy-Based Models)。
- 根据 “Hopfield Network is All You Need” 论文¹，现代 Hopfield 网络的 state update rule 跟 **Transformer** 重合²。换句话说，每执行一次 Transformer，就会趋向 Hopfield 的能量最低点。
- Transformer 的 softmax 可以看成是 **大脑** 中某种 “winner-takes-all” 机制。从这个角度，可以类比大脑思考的机制，互相参考以获取更多灵感。
- 我最新的论文 提出，Transformer 具有 **逻辑结构**，可以在逻辑基础上建立 AGI。

¹感谢 Eric Zeng 给我推荐这篇论文。

²注意这是 state update rule 而不是 learning update rule. 前者 更改 Hopfield 网络的 激活 状态；后者 更改 Hopfield 网络的权重 / 记忆。

1 Hopfield 网络

1.1 经典 Hopfield 网络

我们的符号跟随 *Hopfield Network is All You Need*.

\mathbf{x}^i = 需要记忆的 **patterns** (有 N 个), x_s^i 是它的 s -th bit.

$\mathbf{X} = (\mathbf{x}^1, \dots, \mathbf{x}^N)$ 是所有 patterns 的矩阵。

ξ = 网络的 **状态**, $\xi_s = s$ -th 神经元的 激活状态。

连接权重 between s -th and t -th neurons:

Weights

$$T_{s,t} = \sum_i x_s^i x_t^i \tag{1}$$

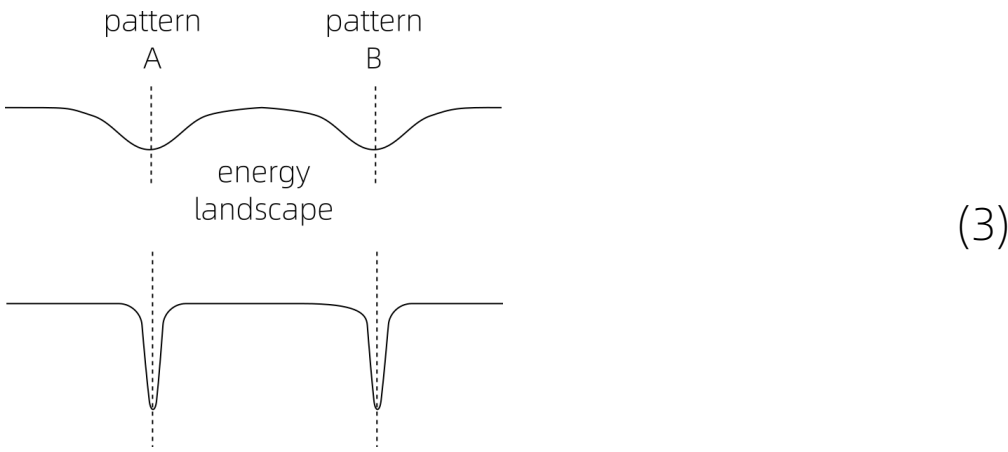
总能量 (Hamiltonian):

Energy

$$E = -\frac{1}{2} \sum_s \sum_{t \neq s} \xi_s T_{s,t} \xi_t \tag{2}$$

1.2 Modern Hopfield 网络

在经典 Hopfield 网络里，当 A 和 B 两个 patterns 太靠近的时候，它们会互相干扰，导致可以储存的 patterns 数量不大。现代 Hopfield 网络 改变 Hamiltonian 能量函数，令干扰减弱，可以储存数量更多的 patterns:



新的能量函数:

New Energy

$$E = - \sum_i F(\xi^T \mathbf{x}^i) \tag{4}$$

注: $\xi^T \mathbf{x}^i = \sum_s \xi_s x_s^i$, F = interaction function.

[Demircigil et al 2017] 提出 F 用 exponential 函数。

State update rule:

$$\xi^{\text{new}} = \mathbf{X} \text{ softmax}(\beta \mathbf{X}^T \xi) \tag{5}$$

1.3 Hopfield-Transformer 对应

传统 Transformer's state update rule:

$$\mathbf{Z} = \text{softmax} \left(\frac{1}{\sqrt{d_k}} \mathbf{Q} \mathbf{K}^T \right) \mathbf{V} \tag{6}$$

Modern Hopfield network's state update rule:

$$\mathbf{Z} = \text{softmax} \left(\beta \hat{\mathbf{R}} \hat{\mathbf{Y}}^T \right) \hat{\mathbf{Y}}^T \tag{7}$$

这里 $\hat{\mathbf{R}}, \hat{\mathbf{Y}}, \hat{\mathbf{X}}$ 是 $\mathbf{R}, \mathbf{Y}, \mathbf{X}$ 分别乘上了适当的 \mathbf{W} 's 矩阵，但为了数式的简洁，使用了代号。

Query patterns = $\mathbf{R} = (\mathbf{r}_1, \dots, \mathbf{r}_M)$ 是网络的**状态**。之所以有 M 个状态，是因为他们将 Transformer 的 M 个输入 **摊开**来，才构成一个大的 Hopfield 网。不这样做根本无法将 Hopfield 网和 Transformer 等同起来。

\mathbf{Y} 就是 Hopfield 记忆里的 patterns，它们担任 Transformer 里 **keys** 的角色。

在 Self-Attention 里， $\mathbf{R} = \mathbf{Y}$.

1.4 Boltzmann 机

注意：以下是 Boltzmann machine 跟 **经典** Hopfield network 的对应。

Let $O = (O_1, \dots, O_n)$ be the **state vector**.

$W = \{W_{s,t}\}$ are connection **weights**.

State update rule: i -th unit is set to 1 with probability

$$\frac{1}{1 + e^{-S_i/T}} \tag{8}$$

where T is a temperature.

如果用以上的 update rule，则 Hopfield 能量 变成 **概率分布**:

$$P(O) = P(O|W) = \frac{e^{-\mathcal{E}(O)/T}}{Z} \quad \text{Boltzmann distribution} \tag{9}$$

where partition function $Z = \sum_U e^{-\mathcal{E}(U)/T}$

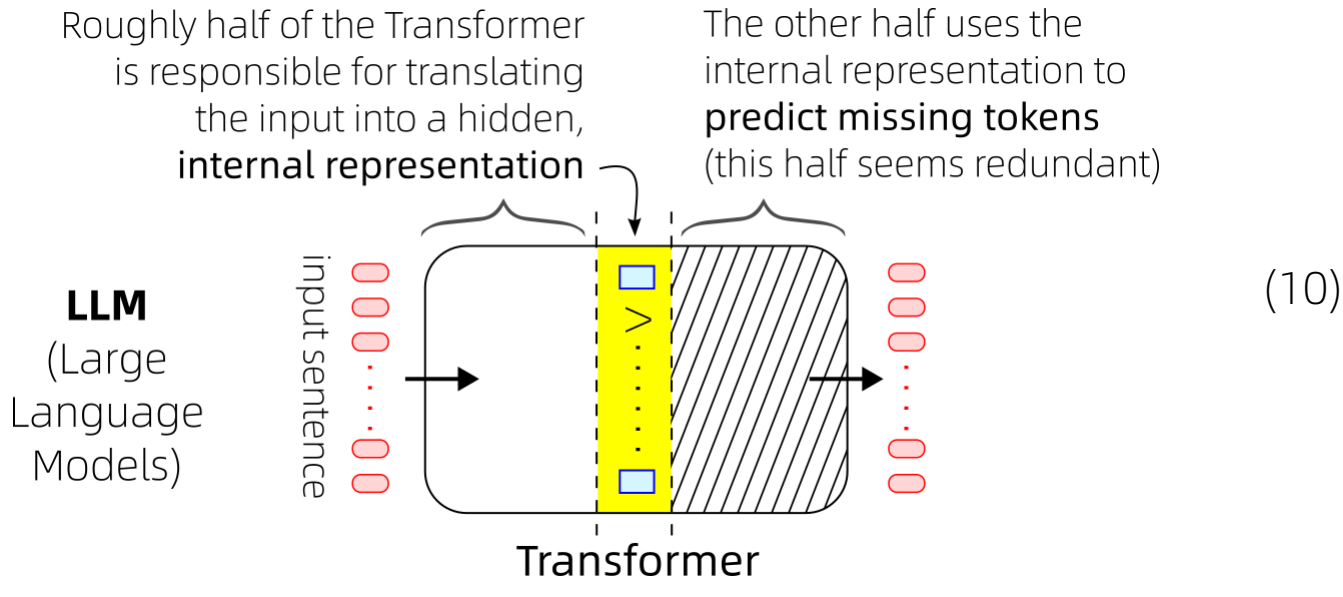
TO-DO: 求出 现代版的 Hopfield network 的 Boltzmann state update rule.

TO-DO: 求出 Hopfield-Boltzmann machine 的**learning** update rule. 但它似乎是根据 **记忆** 而 update 的??

2 与强化学习 结合

2.1 LLM 内部很可能存在冗余

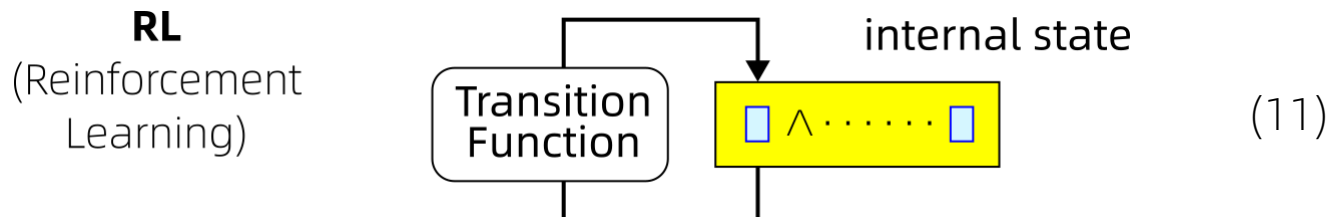
LLM (大型语言模型) 是根据 自回归 (auto-regression, 强迫模型的输出跟输入一样) 的原理训练。在这种模式下, Transformer 用大约一半的资源, 将输入句子转化为一个 具有高度抽象语义的 内在表示。它的后半部 利用这个抽象表示 去预测掩盖了的词语:



在一个智能系统里面, 其实我们最需要的是 中间的那个 hidden representation, 用来做「下游」的工作, 但不一定是预测词语, 所以后半部对智能系统来说, 构成了一种 **冗余**。

有两个做法: 一是接受这个冗余, 继续将整个 AGI 系统做出来。二是改变系统架构, 重新训练一个更合乎直观 的模型。

个人认为 强化学习 模型 更为直观, 可以省却很多理解上的「拐弯抹角」, 令 AGI 的设计更清晰:



这个 transition function 就是一个类似 Transformer 的函数。

2.2 Transformer → Hopfield → Boltzmann

在强化学习里, 系统需要 **储存** 并 **学习** 一个分布在所有状态或动作 (包括 思维动作, 即「思维空间」) 之上的 **概率分布**。由于思维空间是一个庞大的 高维向量空间, 在计算机上很难处理。(数学家们很轻松就写下这种空间, 但我们必需考虑实践的可行性。有人打趣说:「计算机学家就是赶时间的数学家」)

不能简单地用 Transformer 输出这个 概率分布。目前习惯是, Transformer 输出的 token, 会乘上一个矩阵, 让输出转换成一支很长的向量, 它代表 词典里每个词的概率分布。但这个 trick 转到 思维空间上就不管用了, 因为将所有的思维枚举出来 不切实际。

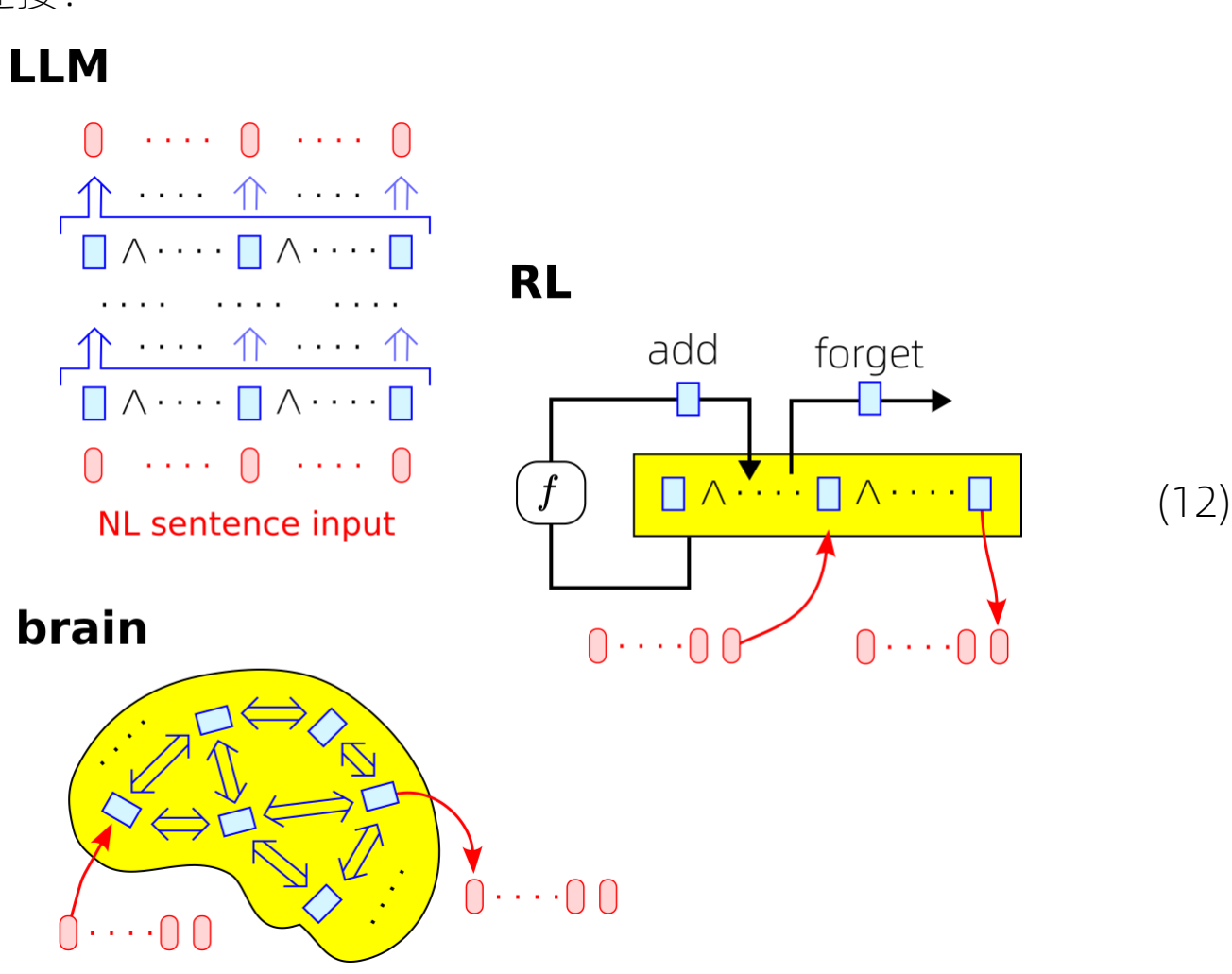
于是我们想到一个办法就是, 用「隐式」的方法表示这个概率分布。关键是一篇名为 “Hopfield Network is All You Need” 的论文。Hochreiter et al 论证, Transformer 是 Hopfield 网络的一个特例。

Hopfield 网络是一种 特别简单的 fully-connected 神经网络, 它具有 associative memory 的特性, 可以凭部分 pattern 回忆整个 pattern. 记忆由神经元之间的 **连接权重** 决定, 而这些权重定义了一个 **能量函数**。Hopfield 网络就是统计物理学里面的 Ising model 应用到了神经网络上。

Hopfield 网络的能量函数可以转变为 概率, 那就是 Boltzmann machine. 这个东西在深度学习里占有举足轻重的地位; Hinton、Bengio、LeCun 等人都深入地研究过它, 特别是因为它适用于 强化学习。它也叫作 EBM (energy-based models)。

Transformer 可以轻易地转变为 Hopfield 网络, 然后变成 EBM. 那么, 根据强化学习的 Bellman 方程, 必然可以找到它的 learning update rule, 于是可以建立 AGI 的最基本模型。

这里还有一点细节需要想明白: 在 RL 的「状态」里面有很多逻辑命题, 它们互相激活, 导致下一个结论命题的出现。这情况跟大脑里的 “areas” 互相激活 很相似。或许大脑 可以给我们一些启发, 如何更有效率地组织 大量 神经元之间的连接?



2.3 RL update rule

强化学习的 **Bellman update** 是根据 某个 output 的 reward, 例如我比较熟悉的 Q-Learning 的 temporal difference update:

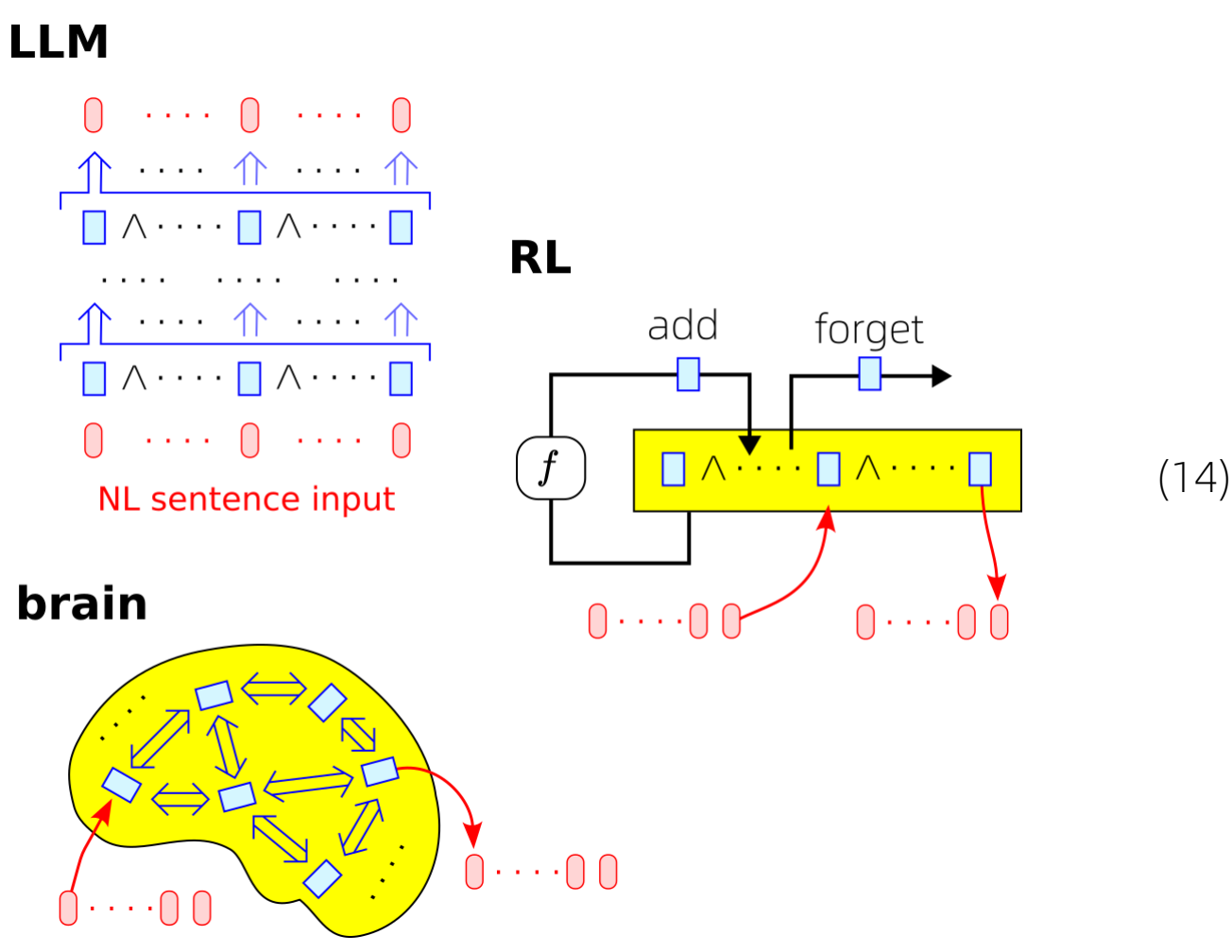
$$Q(s, a) += \eta [R + \gamma \Delta Q]$$

(13)

η = learning rate,
 γ = discount factor,
 s = state,
 a = action,

而 这个 Q 值可以看成是某种 **能量**。

下图中, 我们比较 LLM, RL 和 大脑。它们的 **状态** 有什么对应关系?




从 model-based RL 的角度看 (例如 PILCO), 需要 预测世界。但这不包括预测自己的思想。但基于感觉资料的思想 包括对世界的预测。模型就是对世界的思想。究竟 model-based 跟 model-free 有什么分别? 对 世界的预测 帮助寻找最佳的对策。但我说 对世界的预测 是 sensory-based inference 而已。

在状态之中 某些命题 update, 但其他命题 可以不变。

RL 里面, weights 决定 next state, utility 决定 next state, weights 决定 utility = energy = probability distribution over next states. 但 current state 的 utility value 似乎很难获得?

State 跟 action 的分别, 似乎就是 新命题 跟 命题集合 的分别。

在大脑中, 命题似乎有 位置的固定性。

 这部分暂时仍未想清楚....

3 大脑