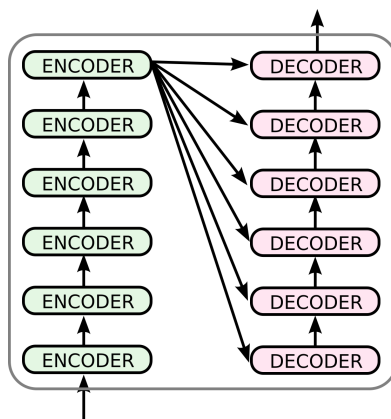


# White paper

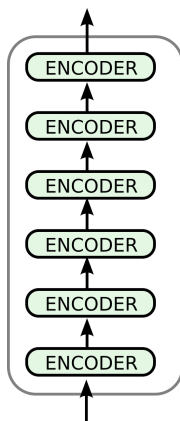
April 2, 2021

This is the original Transformer architecture:



(1)

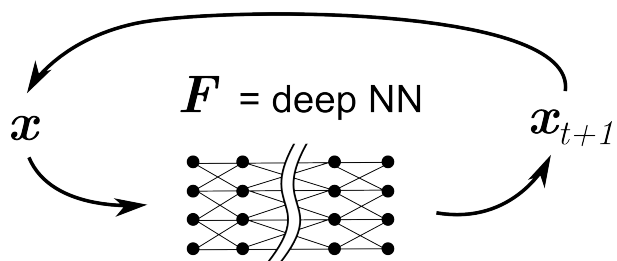
This is the “Encoder” part:



(2)

(A friend told me that BERT only uses the Encoder part of Transformer.)

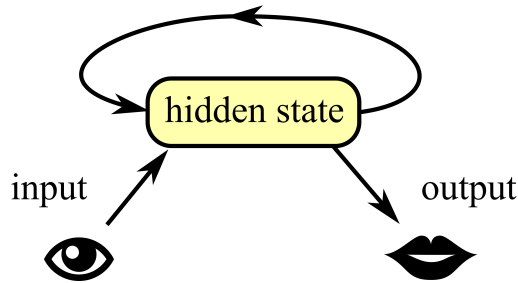
所不同的是，在 强化学习里，状态 is iteratively updated:



(3)

其中，input/output are reserved parts inside the state vector:

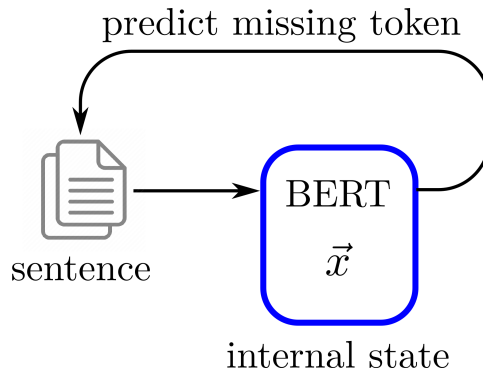
rewrite / update / transition function =  $F$



(4)

现在问题是我们想用 reinforcement learning 的架构，加到 BERT 上。

首先看看 BERT 是怎样训练的：



(5)

其实我也不太清楚怎样将 BERT 与 RL 结合，其中一篇论文是 **KG-A2C** (Knowledge-Graph Advantage Actor-Critic):

<https://openreview.net/forum?id=B1x6w0EtwH>

According to the paper, the update is done via an Advantage function  $A$ :

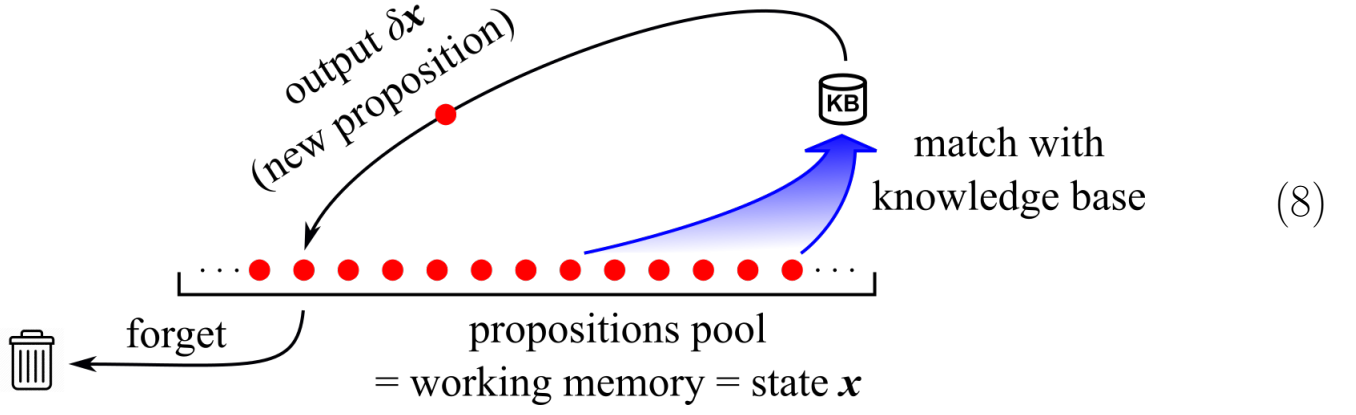
$$A(\mathbf{x}_t, a_t) = Q(\mathbf{x}_t, a_t) - V(\mathbf{x}_t) \quad (6)$$

$$Q(\mathbf{x}_t, a_t) = \mathbb{E}[R_t + \gamma V(\mathbf{x}_{t+1})] \quad (7)$$

where  $R$  is the reward.  $V$  is the value function and  $Q$  is the value function restricted to action  $a$ , as is standard in  $Q$ -learning. This part seems pretty standard for A2C.

如果不用 BERT 而用我的 symmetric neural network 方法，可能更易：

用 symmetric neural network 的话，系统的 状态 有 **逻辑命题** (red ●) 的结构：



在每个时间点  $t$ ，状态 以如下方式 **update**:

$$\mathbf{x}_{t+1} = \mathbf{x}_t \oplus \delta \mathbf{x} \ominus \text{forget}(\mathbf{x}_t) \quad (9)$$

暂时不会 implement forgetting mechanism, 我们只是用一个足够大的 state 装下整个 NL 句子的字 (每个 NL 字 用一个逻辑命题), **遗忘** 时间最早的那些命题。

在 policy gradient 方法下 reinforcement learning 的 **update** 是：

$$\theta := \theta + \eta \nabla_{\theta} J(\theta) \quad (10)$$

where

- $\theta$  = **parameter** of the policy  $\pi_{\theta}$
- $J$  = **value** function =  $\mathbb{E}[R(\tau)]$
- $R(\tau)$  = total **reward** of trajectory  $\tau$

where a **trajectory** is a sequence of (state, action):

$$\tau = s_0, a_0, s_1, a_1, \dots, s_T, a_T. \quad (11)$$

and the **policy** is a probability function:

$$\begin{aligned} \pi_{\theta}(\tau) &= \text{probability of trajectory } \tau \\ &= p(s_0)\pi_{\theta}(a_0|s_0)p(s_1|s_0, a_0) \cdot \dots \end{aligned} \quad (12)$$