筌者所以在鱼，得鱼而忘筌；蹄者所以在兔，得兔而忘蹄；言者所以在意，得意而忘言。

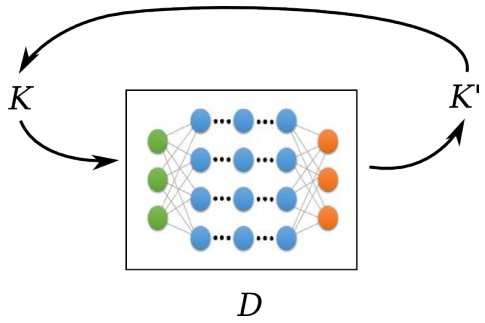
《庄子 · 外物》

Genifer 5.2 理论笔记

YKY (甄景贤)

July 31, 2015

上次的模型：



K 可以直接输出一些 words，words 的序列构成句子。

这做法太简单，因为我假设自然语言是一些 sequences，但自然语言的结构不是序列那么简单，例如「未吃过饭」包含「吃过饭」的序列，但语义是相反的。所以自然语言必须要在语义 (semantic) 层面上处理，而不是语法 (syntactic) 层面上。自然语言的复杂性是不会轻易消失的。

1 Language map

在 Memory Networks [1] 里，他们用了一个 language map，将自然语言句子转换成 internal representation of knowledge in $\mathbf{K}$，$\mathbf{K} \ni K$ 是知识状态的空间。可以叫这个 map 做 language map：

$$\begin{aligned} \mathcal{L}: \quad \mathbf{S} \quad &\to \mathbf{K} \\ \mathcal{L}: \quad \text{句子} \quad &\mapsto K \end{aligned}$$

（但其实句子只是知识状态 K 的一部分。）

於是我们的 RNN model 变成这样：

$$\text{句子输入} \xrightarrow{\mathcal{L}} \mathbf{K} \xrightarrow{\mathcal{L}^{-1}} \text{句子输出}, \quad \mathbf{K} \overset{D}{\circlearrowright}$$

有了这个 language map 是很方便的，但它有几个问题：

- 需要用到 NLP parsing，很麻烦，如果能省略比较好
- 这个 map 基本上 fix 了 internal knowledge representation 的形式。但我的直观（也可能错误）觉得 knowledge representation 应该是「不知道的」比较好，它应该是由学习 D 的过程「诱导」出来的，换句话说：the knowledge representation format should be *induced* from the learning of D. 因为以前在经典逻辑系统中，K = KB (knowledge base) 是一些命题的集合，换句话说 $K = \bigsqcup S_i$，S_i 是句子或「命题」。当时的做法是将 KB 组织成 hierarchical 结构，方便搜寻。但我觉得如果 $\mathbf{K}$ 的结构是这样的话，跟原问题的情况太相似，一切都太「有秩序」，可能不是神经网络最有效的用法。
- 自然语言是需要慢慢「吸收」或「理解」(comprehend) 的，但这过程在 Memory Networks 的模型里忽略了。将自然句子转换成 logical 形式（即 internal representation），几乎是不需时间的 transliteration 过程。如果输入一本《世界历史》的原文，不消一秒便可以转换成 internal representation，但不能说 Genifer 已经「理解」了全书的内容吧？

2 理解 / 慢吸收

所以 $\mathcal{L}$ 不是一个普通 map 而是一个很复杂的算法。

对新输入知识的「慢吸收」包括这些运作：

- consequences （找出新命题的所有可能推论结果，至少在 n 步之内）
- consistency （新信念不和旧信念抵触）
- explanations （新信念可以被已有知识解释）

「理解」的过程可以看成是：经过 n 次的推导 $\mathcal{D}$，K 变成 K'：

$$K \xmapsto{D^n} K'$$

而 K' 有我们想要的特性（即「后果、和谐、解释」）。问题就是怎样测试 K' 有这些特性？特别困难的原因是 $K' \in \mathbf{K}$ 的 representation 是不透明的。

K' 的特性可以怎样测试？

似乎唯一的方法是透过这样的查询：

$$K \xmapsto{D^n} Q\,?$$

但 Q 也是 $\in \mathbf{K}$，而 $\mathbf{K}$ 的结构是不透明的。

如果有 $\mathcal{L}$ 的话，直接将问题用自然语言问，变成 $Q = \mathcal{L}(\text{句子})$，就可以查询 K。换句话说，$\mathcal{L}$ 的好处是容许 direct access to K，将知识状态的内容，直接用自然语言读出来（或写入去）。但现实中 $\mathcal{L}$ 不应该存在。

假设有 $\mathcal{L}$，容易做到以下两种学习方式：

2.1 Inductive learning

举例：

小明是香港人 $\wedge$ 小明戴眼镜
小强是香港人 $\wedge$ 小强戴眼镜
小雄是香港人 $\wedge$ 小雄戴眼镜
小娟是香港人 $\wedge$ 小娟戴眼镜
结论：所有香港人都戴眼镜

用逻辑表示这法则就是：

$$\forall X.\ \text{Hong-Kong}(X) \rightarrow \text{wear-glasses}(X)$$

但在 RNN 里，这法则是暗含在 D 之中。例如如果 $k =$「大强是香港人」，那么 D 作用在 K 上最终会得出 $K' =$「大强戴眼镜」。

换句话说，已知 K_0，K^*，学习 D：

$$K_0 \xrightarrow{D} .... K^*$$

因为我们可以用 $K = \mathcal{L}$(自然语言句子) 来计算 K_0 和 K^*，所以这个算法是可行的。

2.2 Reinforcement learning of D

$$K \xrightarrow{D} .... K'$$

which can recieve ”right” or ”wrong” rewards.

This is analogous to:

$$\text{state} \xrightarrow{\text{action}} \text{state'}$$

3 Speech generation （发言）

发言有两种模式：

- 查询某句子 Q 是不是真的
- 查询关於某题目 I 的内容，eg: ”Tell me about your mother”

整个 Genifer 系统包含 RL + RNN 两部分。

4 RNN

RNN (D) 是一个 feedforward NN，只是它的输出再回溃到输入。

它可以执行 3 个运作：

4.1 Deduction

Deduction 只需要 forward propagation。（实际上 deduction 可能没有什么用，重要的是 querying。）

4.2 Learning

Learning 是通过 back-prop，我们要求的是从 K_0 开始：

$$K_0 \xrightarrow{D} .... K_\infty$$
$$K_\infty \geq K^* \tag{1}$$

但这里需要用到 $\geq$ 关系，下述。

目的是学习 D，令误差 $\mathcal{E}$ 最少。

梯度 $\frac{\partial \mathcal{E}}{\partial W}$ 的计算应该是可行的；这里 W 是 RNN 的 weights。

但怎样计算 $\mathcal{E} = K_\infty - K^*$ ？有个严重问题是：我们要知道 K^* 的知识的表达方式，但如果 D 是个 black box，K^* 的 representation 就不是透明的。

Joseph 找到一篇 paper，叫 Memory Networks，来自 Facebook AI research [1]。他们也是用 RNN 来学习 Q& A，但它有一个 component I = input feature map，负责将句子变成 internal feature representation，它需要涉及传统 NLP 的 parsing。我们可不可以不做这步？☹

问题是如果输出是 words，我们必须比较句子，亦即是 K 的**序列**。似乎可以用 convolution 方法：记 $S := K_0, K_1, K_2, ...$ 为输出的 sequence，$S^* := K_0^*, K_1^*, ..., K_m^*$ 为想要的答案 sequence，则误差可以定义成：

$$\mathcal{E} := S * S^*$$

其中 $*$ 是 convolution。（但我不是 100% 肯定这个用法是否正确。）

4.3 Querying

$$\begin{aligned} K_0 &\xrightarrow{D} .... K_\infty \\ K_\infty &\geq ? \ K^* \end{aligned} \tag{2}$$

传统逻辑的做法是，找 K_n （n 个推导步骤之后的结果），然后试试 K_n 包不包含 K^*。但通常更有效率是反向地由结论 K^* 开始寻找。

可以看成是这个问题：

$$\begin{aligned} \text{solve} \quad D^n(\mathbf{x}) &> K^* \\ K_0 &\geq \mathbf{x} \end{aligned} \tag{3}$$

其中 $\mathbf{x}$ 是变量。我们要求 $>$ 是大於某个 threshold ϵ。这是一条 iterative equation，似乎还有希望。

上次说过如果 D 是 monotonous，即 $\forall \mathbf{x} \ D(\mathbf{x}) \geq \mathbf{x}$，可能有帮助。

4.4 $\geq$ 关系

$\geq$ 是逻辑中的「generalize」关系，它有两种模式：

- 人会死 $\geq$ Socrates 会死
- 人会死 $\geq$ (人会死 $\wedge$ 月亮是圆的)

在 (topological) vector space 理论里，我们可以定义 vector 之间的 $\mathbf{v}_1 \geq \mathbf{v}_2$，方法是选取任何一个 cone （锥形）$C$：

$$\mathbf{v}_1 \geq \mathbf{v}_2 \Leftrightarrow (\mathbf{v}_1 - \mathbf{v}_2) \in C$$

例如如果在平面上，C 可以是右上角的 quadrant。

我在考虑：我们可不可以选取任何一个在 $\mathbf{K}$ 空间中的 cone 来定义 $\geq$，然后让 RNN 自己学习 $\geq$ 的逻辑结构（例如 动物 $\geq$ 猫、$A \geq A \wedge B$）?

References

[1] Weston, Chopra, and Bordes. Memory networks. *ICLR (also arXiv)*, 2015.