



Recurrent Neural Network (Part 1)

Introduction

Slot Filling

根据用户说出的话，自动讲时间、地址等关键信息填写到对应的槽上，并过滤掉无效的词语。

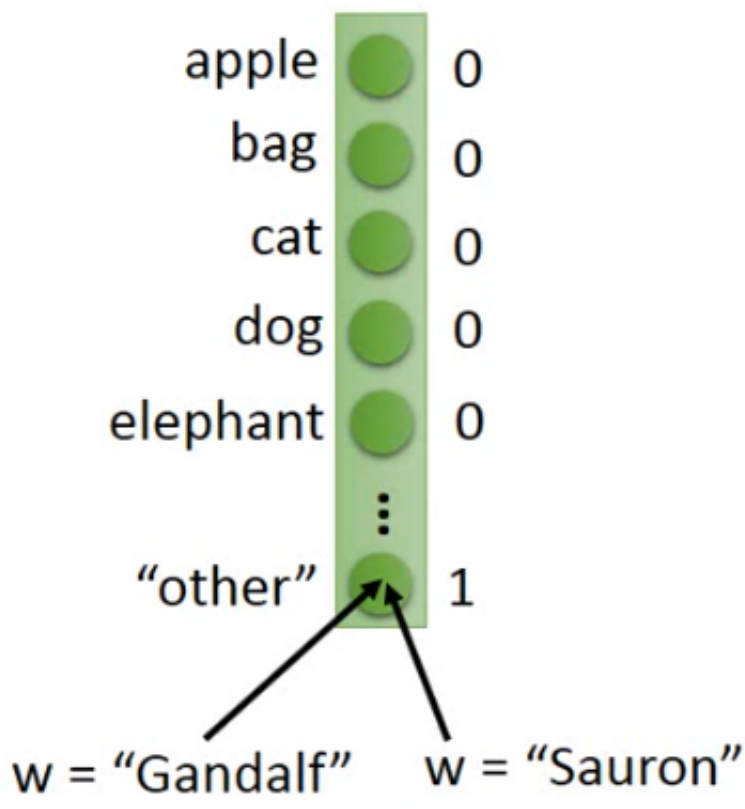
比如要知道 Taipei 属于dest这个slot，November 2nd 属于time这个slot。

那么如何解决Slot Filling的问题呢？当然，我们可以用一个feedforward来解决，但是将一个词汇丢进network之前需要将其用一个vector来表示。

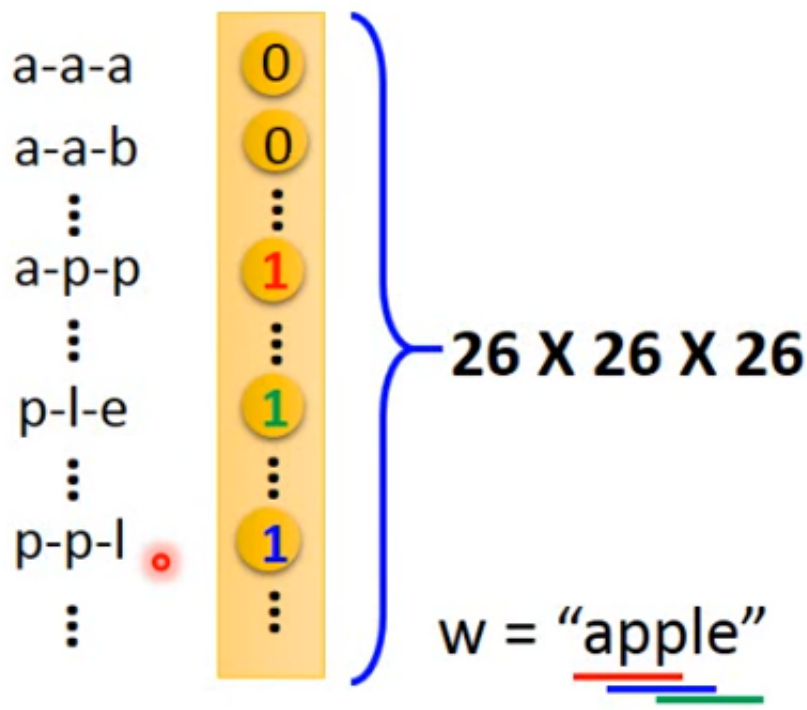
- 1-of-N encoding
- Beyond 1-of-N encoding 多加一个dimension "Other"
- Word hashing

Beyond 1-of-N encoding

Dimension for "Other"



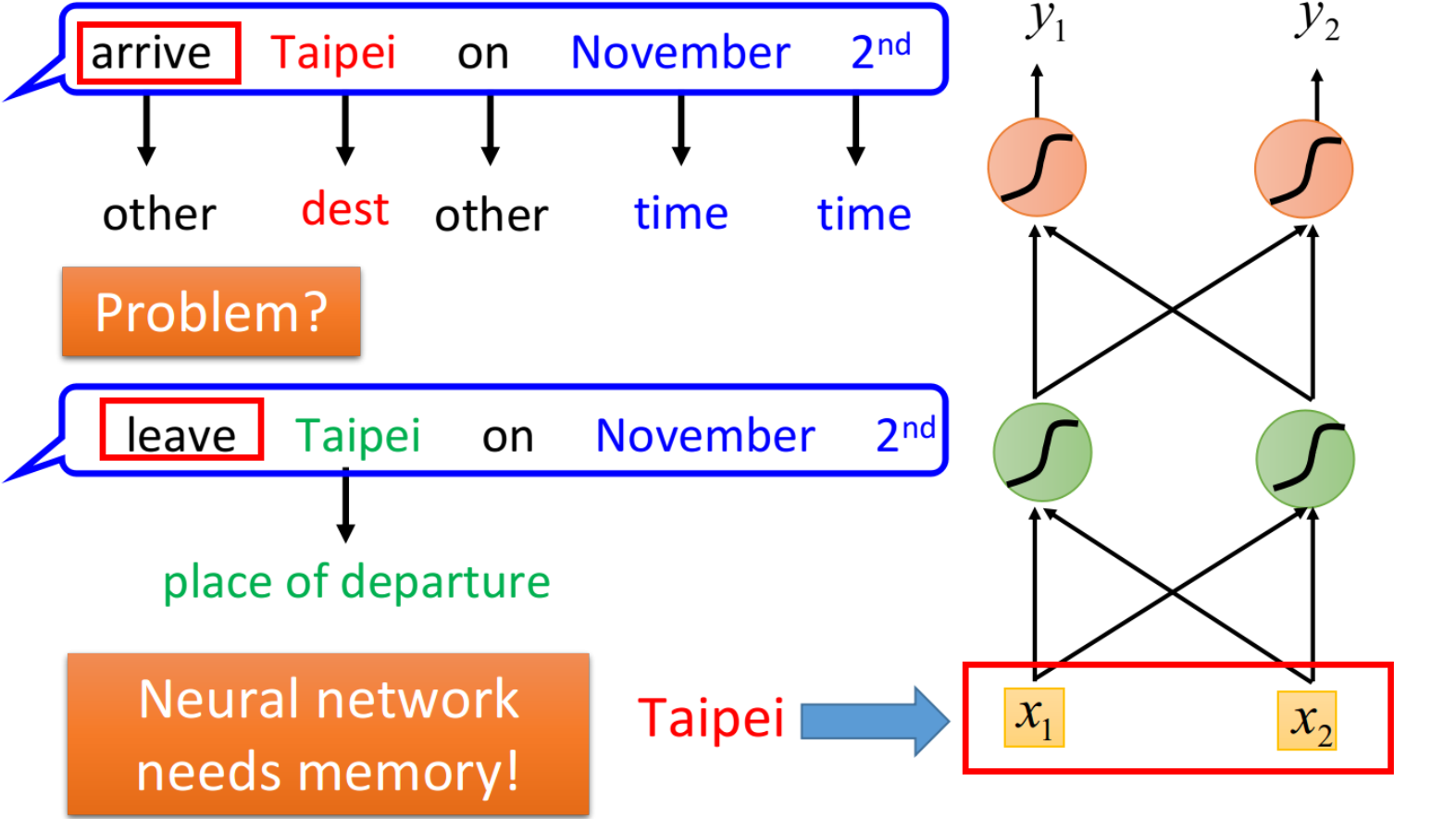
Word hashing



当我们丢进词汇“Taipei”的vector之后，output就是Taipei属于dest或者time of departure 这两个slot的几率。但是光这样还是不够的，当我们输入下图蓝框中的两段话时，对于NN来说，两个“Taipei”是一样的，它没办法区分“Taipei”是应该属于dest还是place of departure。

这时候我们就希望NN是有记忆的，它可以根据上下文来判断“Taipei”是属于哪个slot。

Example Application

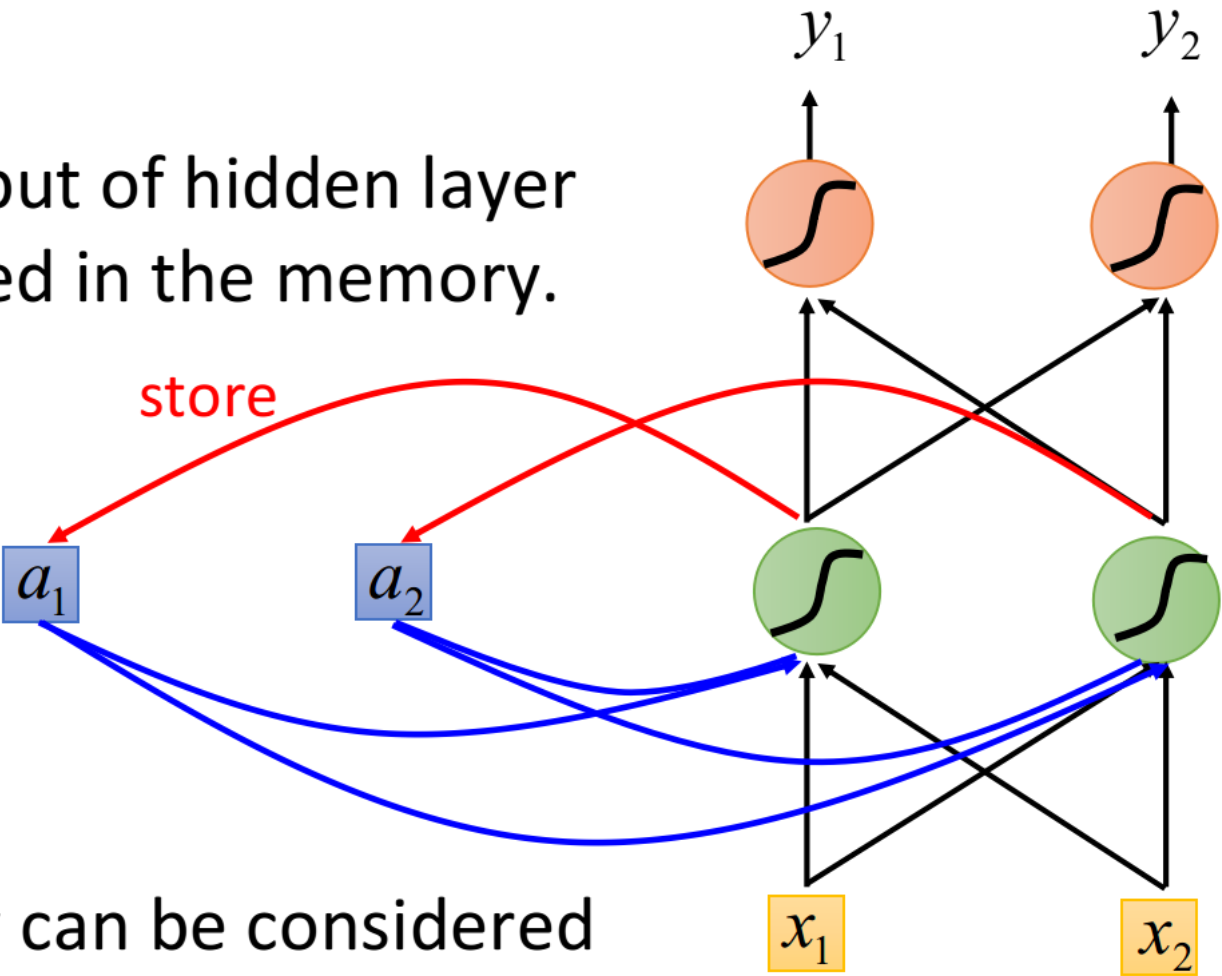


这种有记忆的神经网络就叫做Recurrent Neural Network(RNN)。

将hidden layer里的output存到memory(下图蓝色方框)中。当下次有input输入的时候，就不会再单纯的考虑新输入的input，还会考虑 a_1 a_2 的值。memory中的初始值为0。

Recurrent Neural Network (RNN)

The output of hidden layer are stored in the memory.



Memory can be considered as another input.

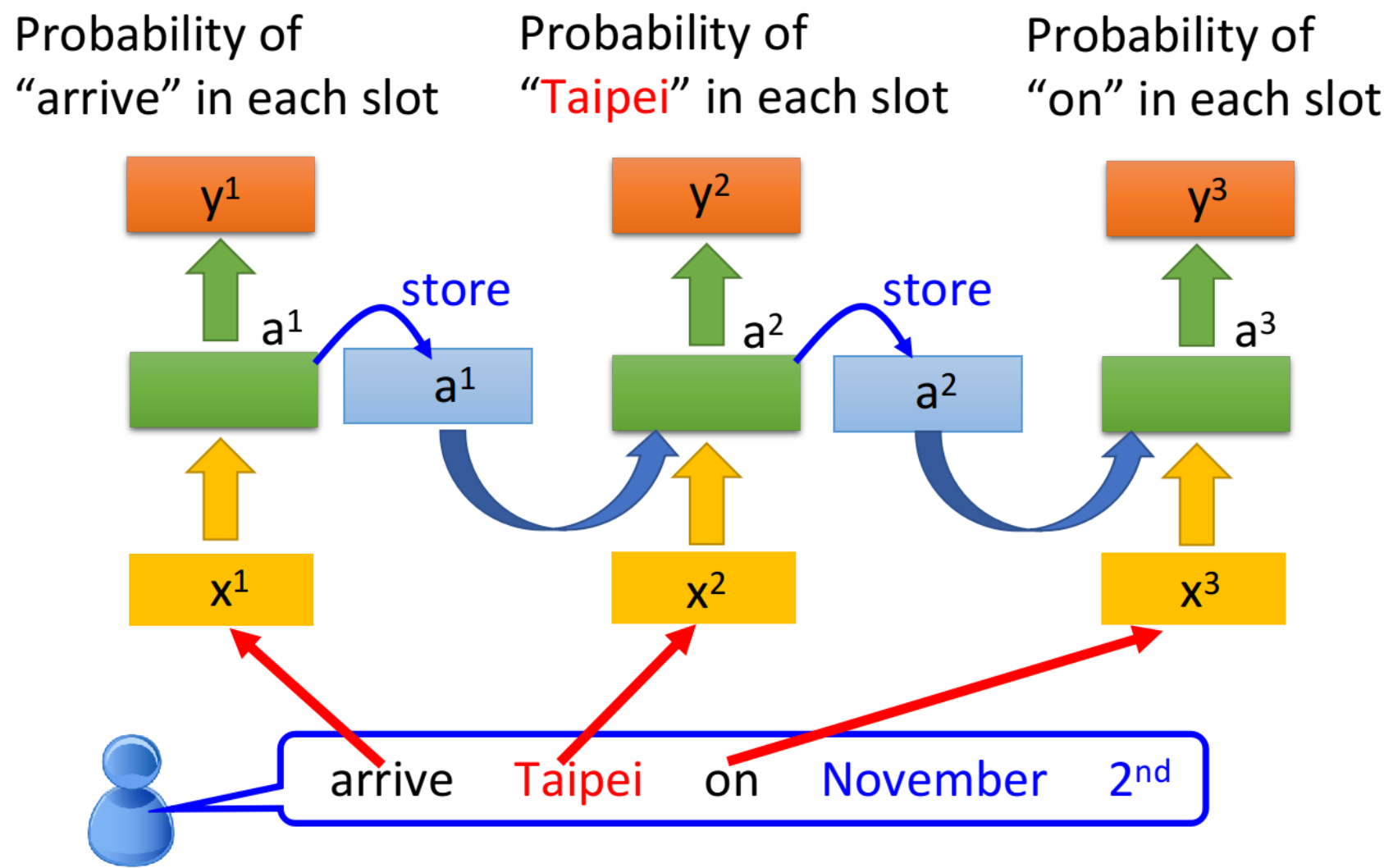
注意：改变input的顺序会改变output。因为input的顺序不同，导致memory中存储的元素不同，因此最终输出结果也会改变。

Slot Filling with RNN

- 先输入“arrive”的vector x^1 ，得到hidden layer生成的 a^1 ，将 a^1 存储在memory，然后根据 a^1 生成 y^1 表示“arrive” 属于每个slot的概率
- 再输入“Taipei”的vector x^2 ，得到hidden layer生成的 a^2 ，将 a^2 存储在memory，然后根据 a^2 生成 y^2 表示“Taipei” 属于每个slot的概率
- 再以此类推，输入“on”.....

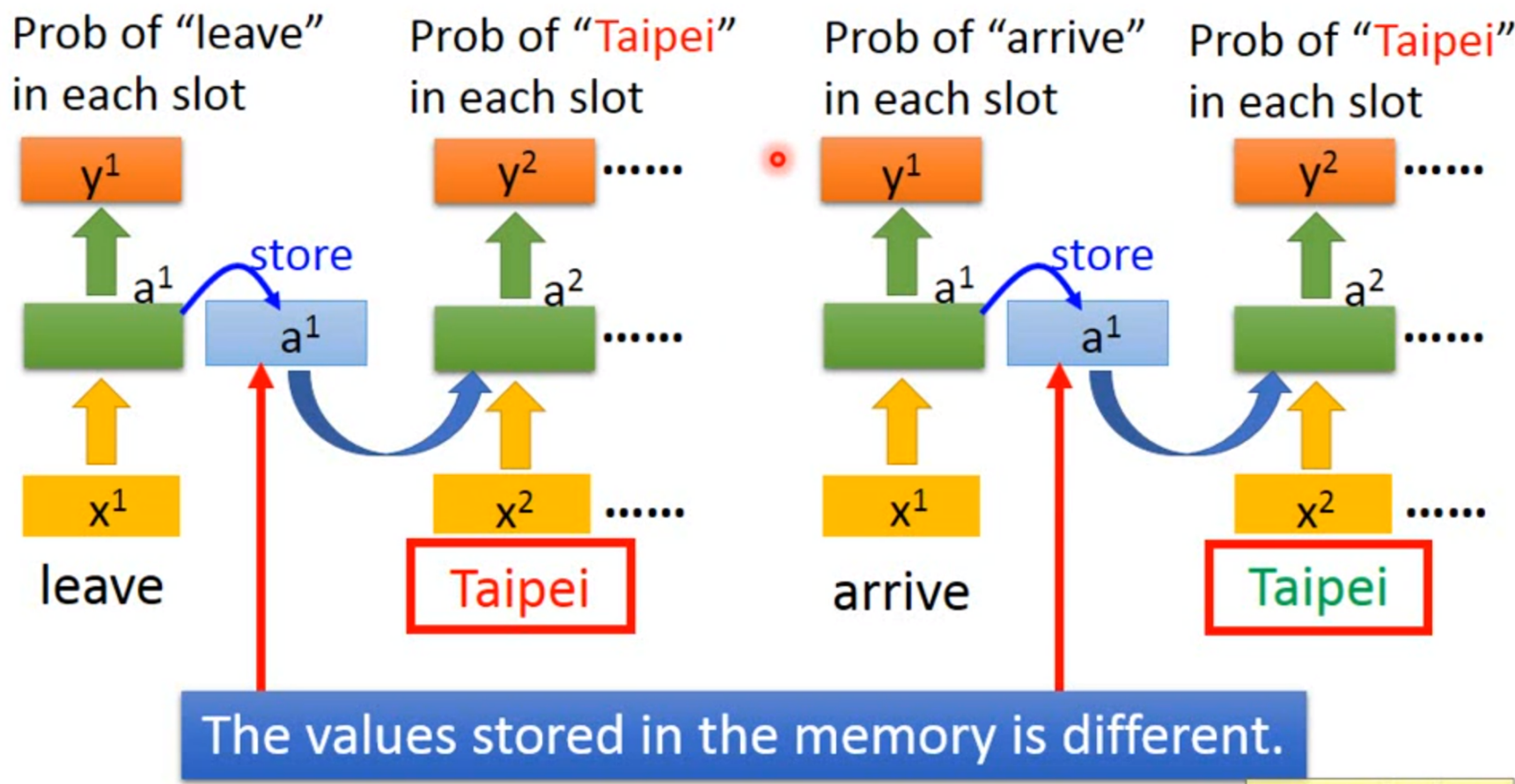
RNN

The same network is used again and again.



注意：上面不是三个network，是同一个network在三个不同的时间点被使用了三次。

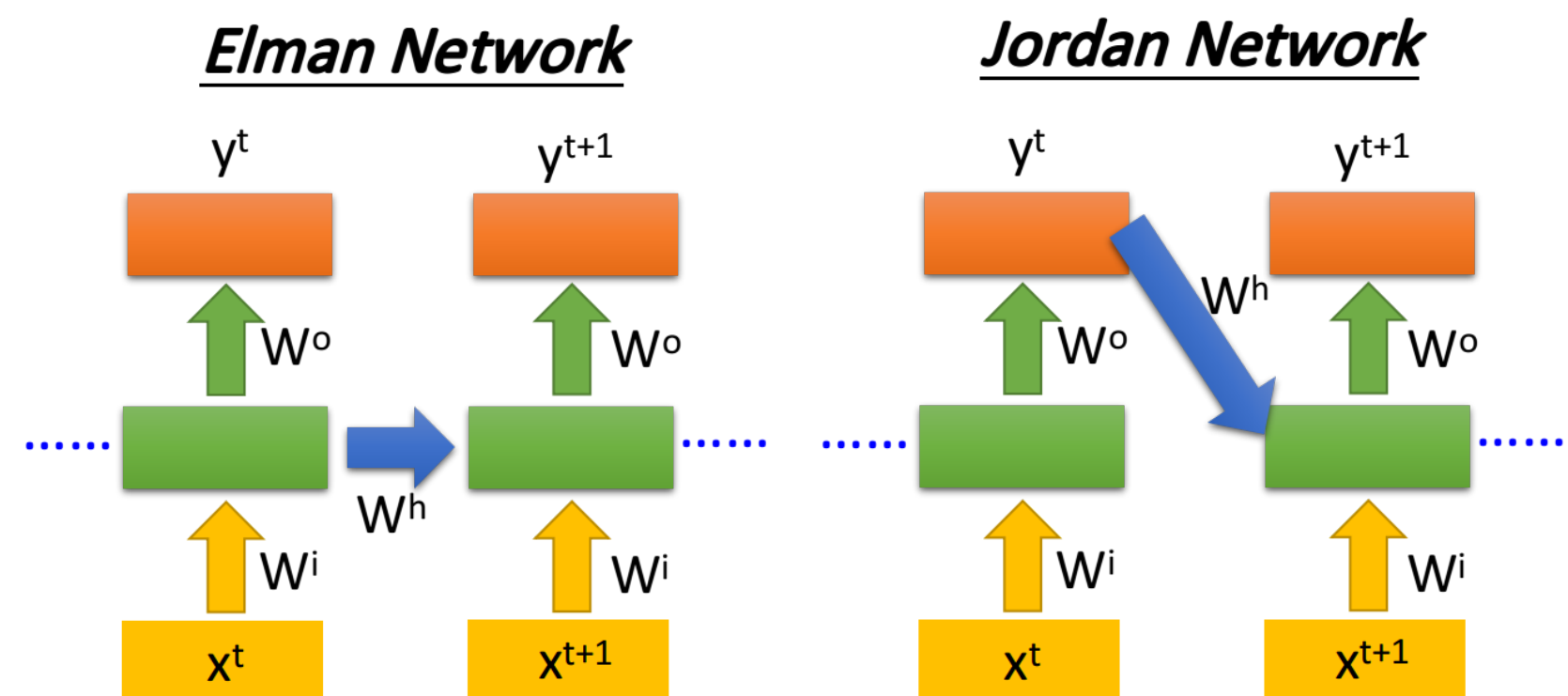
所以这时候在RNN中，即使input x^2 都是“Taipei”，但是因为 a^1 中保存的元素不一样("leave","arrive")，最后的输出也不一样。



- Elman Network:将hidden layer的值存起来，在下一个时间点读出来
- Jordan Network: 将整个output的值存起来，在下一个时间点读出来

因为hidden layer没有target，所以我们并不知道存在memory中的会是什么。所以一般来说Jordan Network的效果更好。

Elman Network & Jordan Network

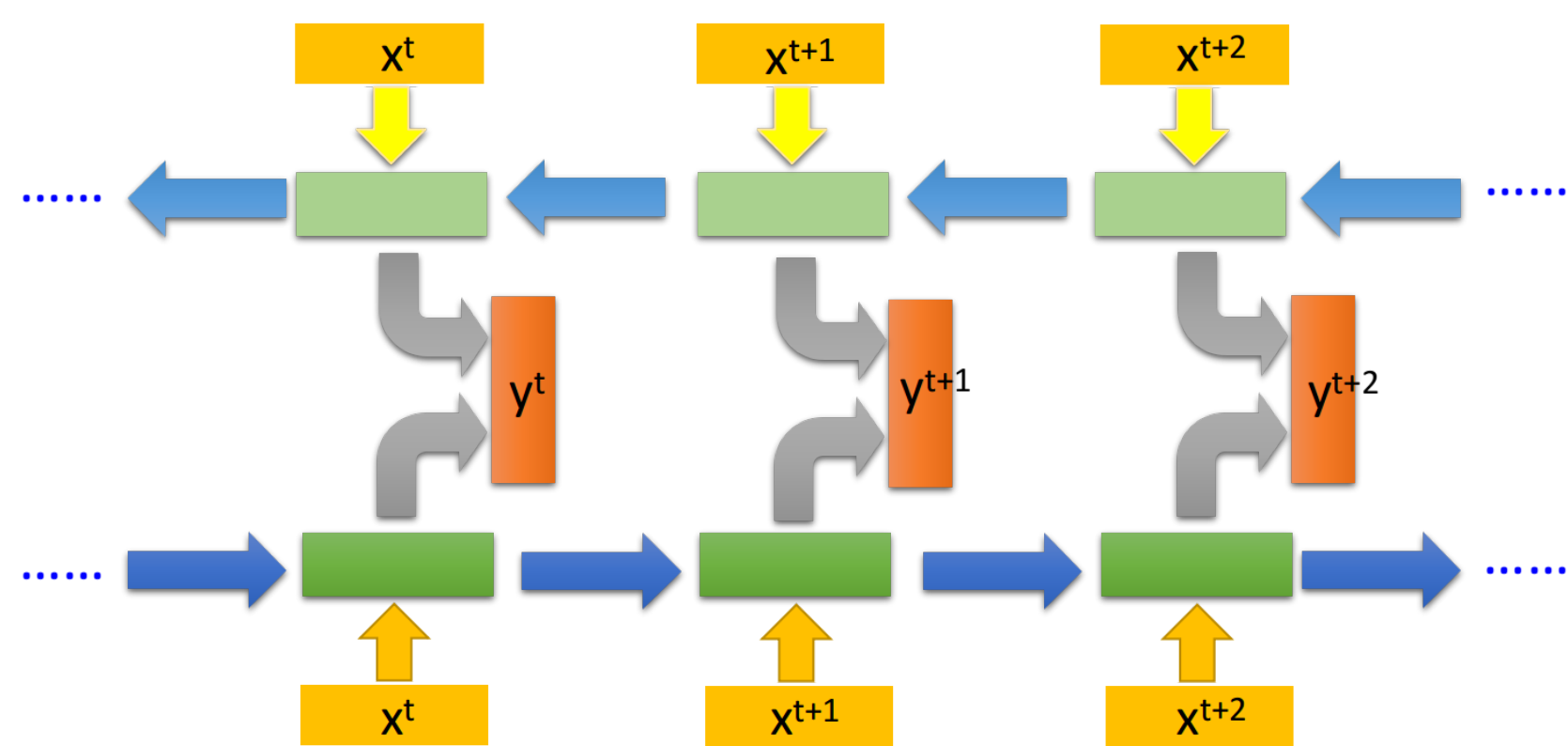


Bidirectional RNN (双向)

读取方向也可以反过来，所以可以同时train正向和逆向的RNN。

这样在产生output的时候，network可以看到更广的范围。如果我们只有正向的RNN，输出 y^{t+1} 的时候，只看过从 x^1 到 x^{t+1} 的input。但是如果是双向的RNN，那么还可以看到从 x^{t+1} 到句尾 x^n 的input。也就是能看到全文。

Bidirectional RNN

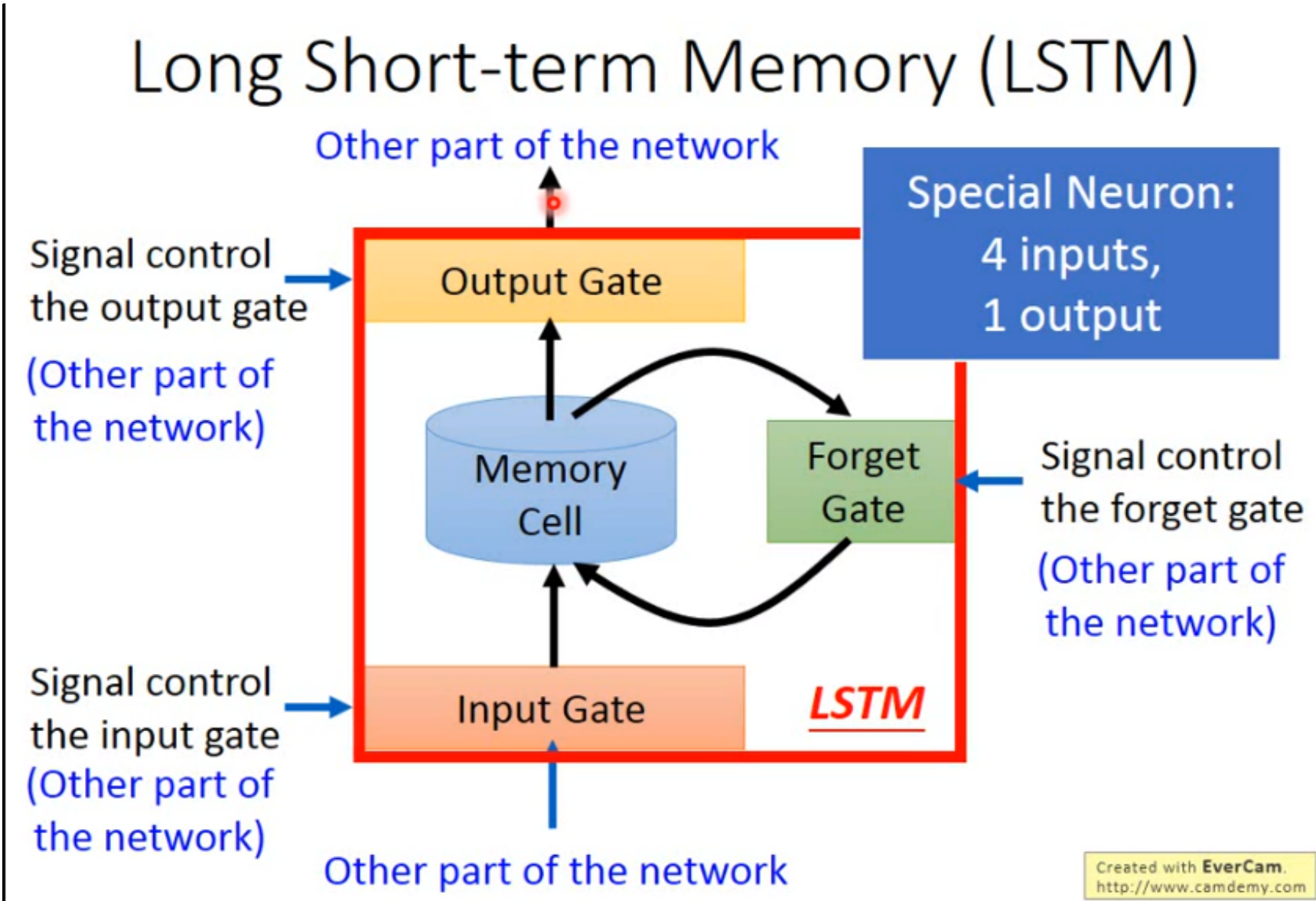


Long Short-term Memory (LSTM)

Long Short-term可以理解为你比较长时间的 短期记忆

LSTM结构

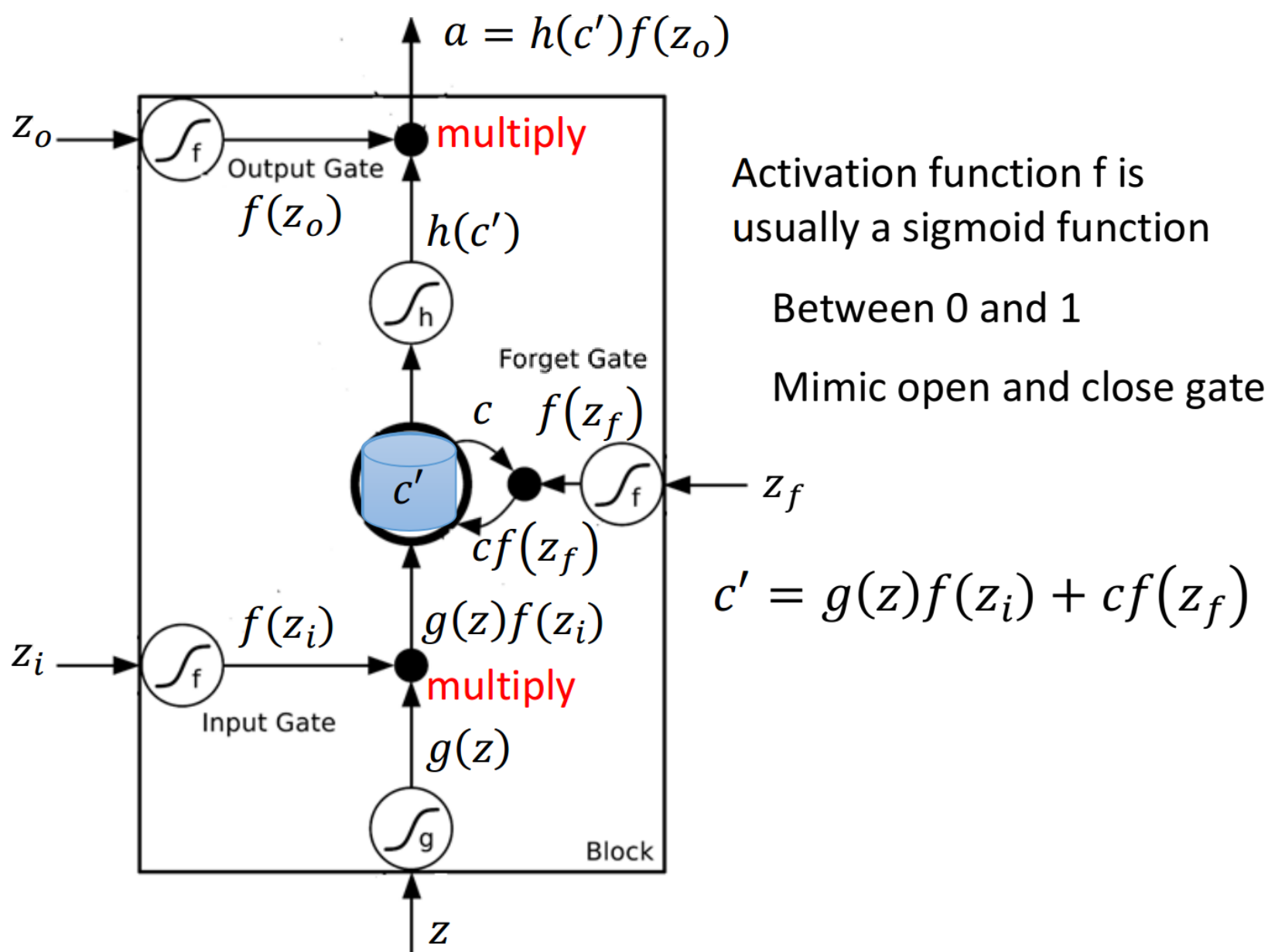
- Input Gate 决定外界是否可以把内容写入memory cell，Input Gate 的打开和关闭时机都是自己学习的
- Output Gate 外界是否可以把memory cell 中的内容读出来，output gate关闭的时候，外界则不可以将其内容读出
- Forget Gate 决定什么时候memory cell 要把过去记得的东西忘掉，打开的时候代表记得，关闭代表遗忘



整个LSTM有4个input，一个output

Memory Cell

- z 是想要被存储到cell里的input值
- z_i 是操纵 input Gate 的信号
- z_0 是操纵 output Gate 的信号
- z_f 是操纵 Forget Gate 的信号
- a 是 output值



LSTM - Example

- x_1 为输入
- $x_2 = 1$ 的时候, 将 x_1 存入到memory(下图蓝色方框) 中
- $x_2 = -1$ 的时候, 将重置memory
- $x_3 = 1$ 的时候, 输入 memory中的值

	0	0	3	3	7	7	7	0	6
x_1	1	3	2	4	2	1	3	6	1
x_2	0	1	0	1	0	0	-1	1	0
x_3	0	0	0	0	0	1	0	0	1
y	0	0	0	0	0	7	0	0	6

When $x_2 = 1$, add the numbers of x_1 into the memory

When $x_2 = -1$, reset the memory

When $x_3 = 1$, output the number in the memory.

