

人工智能之NLP

Seq2Seq

主讲人: GerryLiu

课程要求

- 课上课下“九字”真言
 - 认真听，善摘录，勤思考
 - 多温故，乐实践，再发散
- 四不原则
 - 不懒散惰性，不迟到早退
 - 不请假旷课，不拖延作业
- 一点注意事项
 - 违反“四不原则”，不推荐就业

课程内容

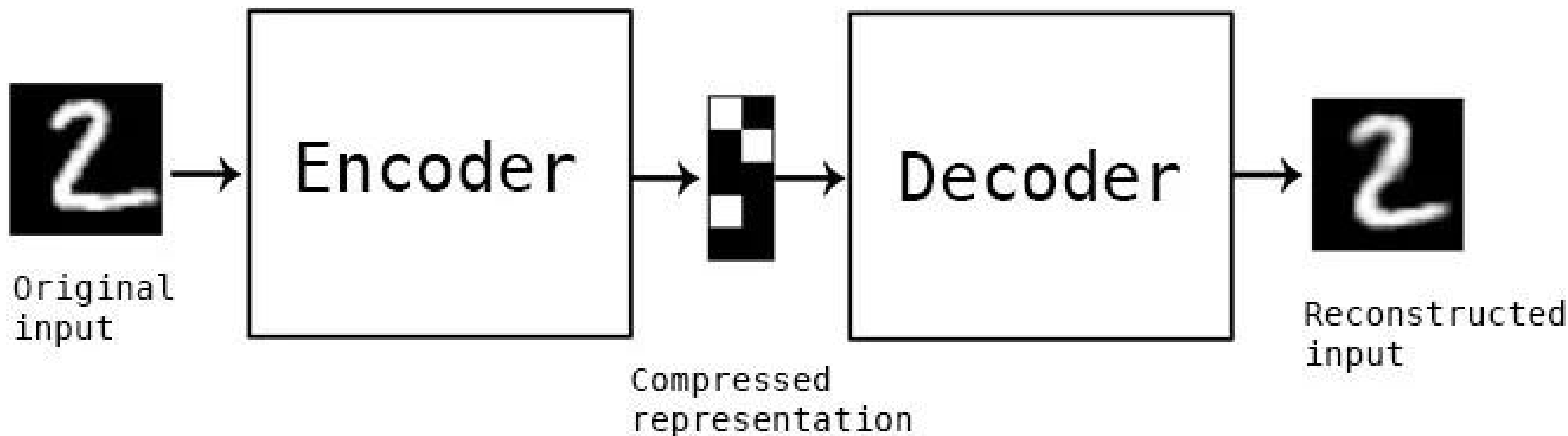
- 自编码神经网络回顾
- RNN、LSTM神经网络回顾
- Seq2Seq网络结构讲解
- Attention结构讲解
- Seq2Seq+Attention项目

自编码器回顾

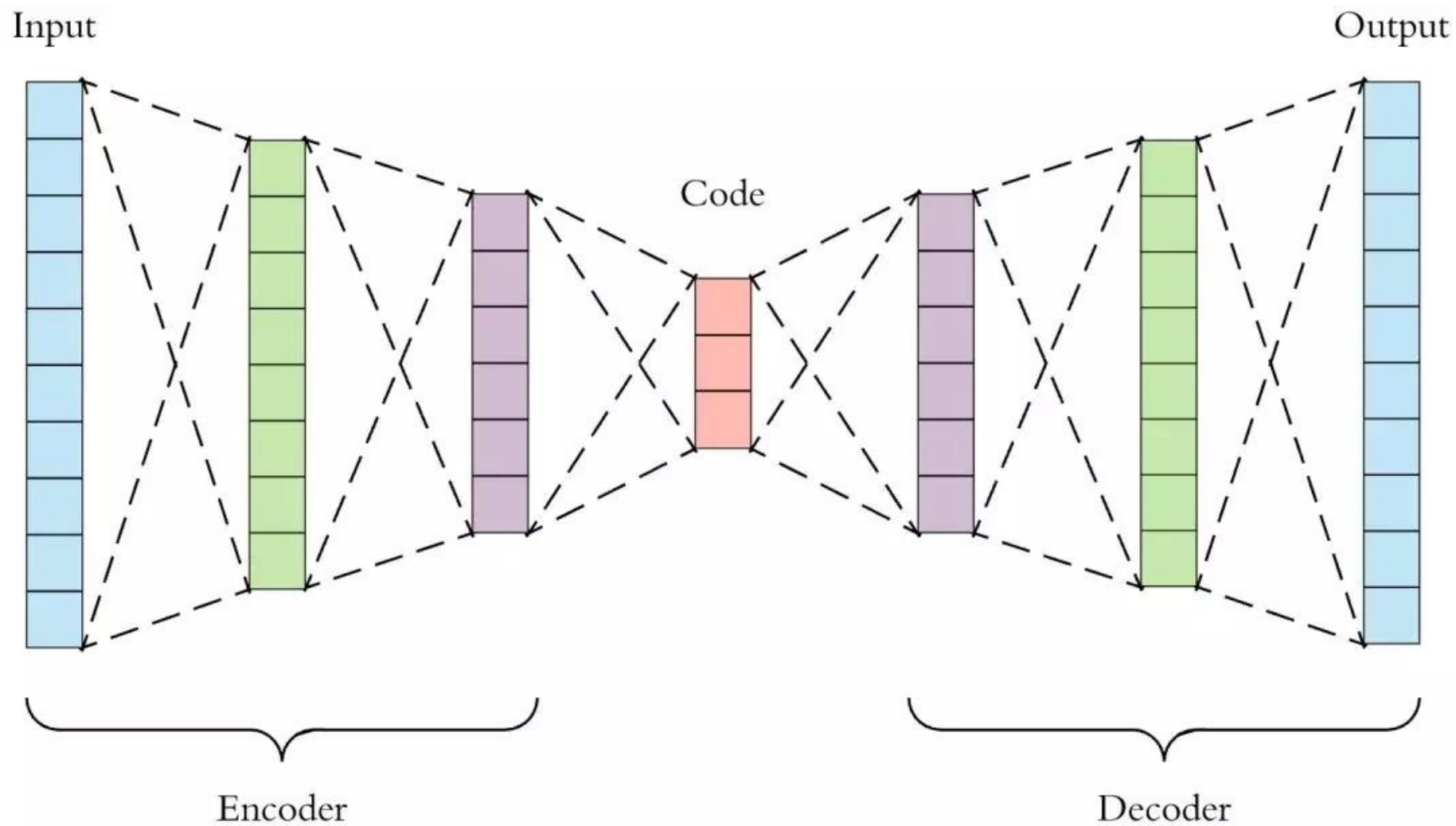
- 自编码器作为一种深度学习领域无监督的算法，本质上是一种数据压缩算法，和生成对抗网络一样，属于生成算法的一种。
- 自编码器(AutoEncoder, AE)就是一种利用反向传播使得输出值等于输入值的神经网络，它将输入压缩成潜在特征/高阶特征，然后将这种表征重构输出。主要包含以下三个特征：
 - 数据相关性。
 - 数据有损性。
 - 自动学习性。

自编码器回顾

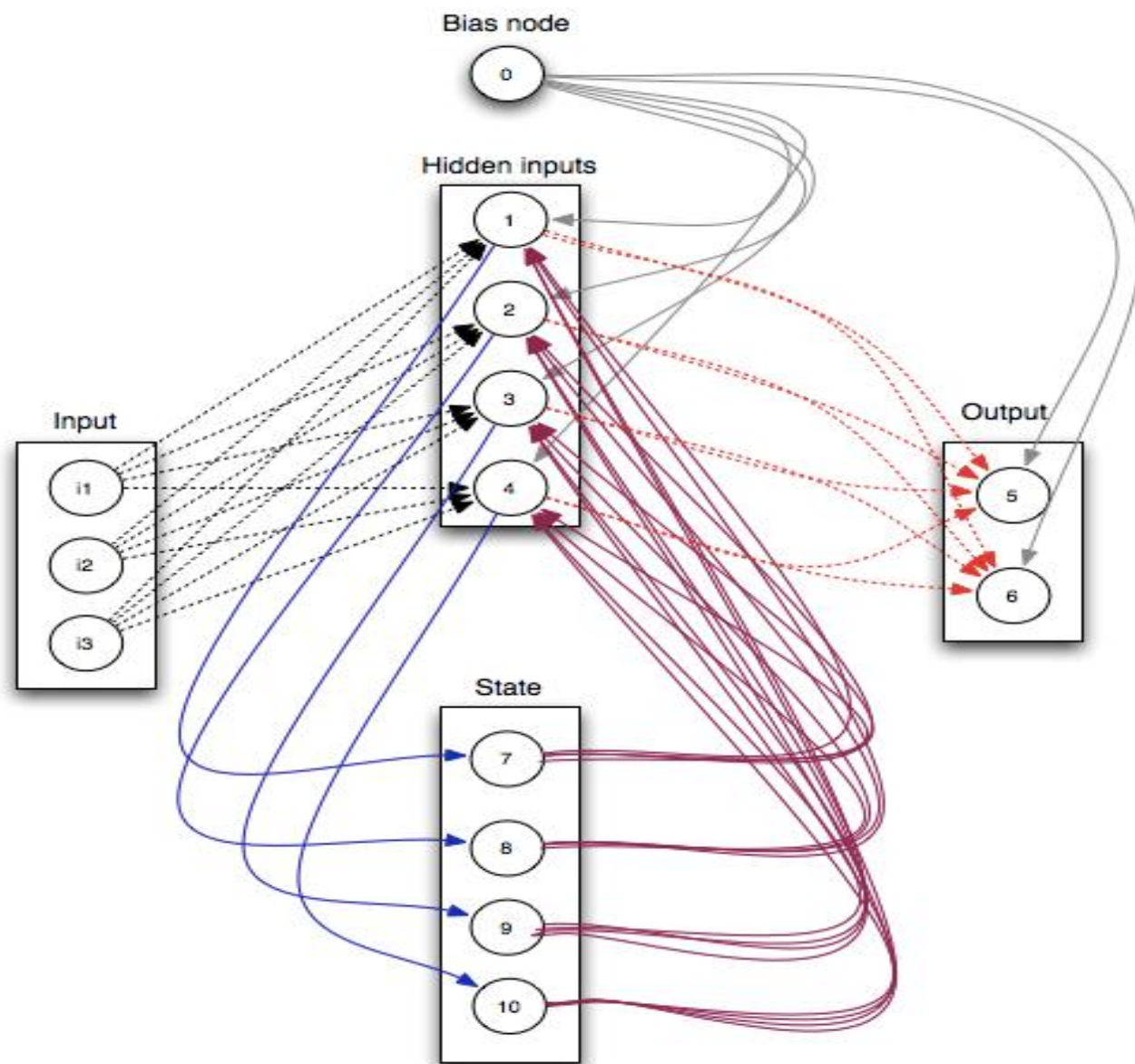
- 构建一个自编码器主要包括两部分：编码器(Encoder)和解码器(Decoder)。编码器将输入压缩为潜在空间特征，解码器将潜在空间特征重构输出。
- 自编码的核心价值是在于提取潜在的高阶空间特征信息。主要应用是两个方向：数据去噪以及进行可视化降维。



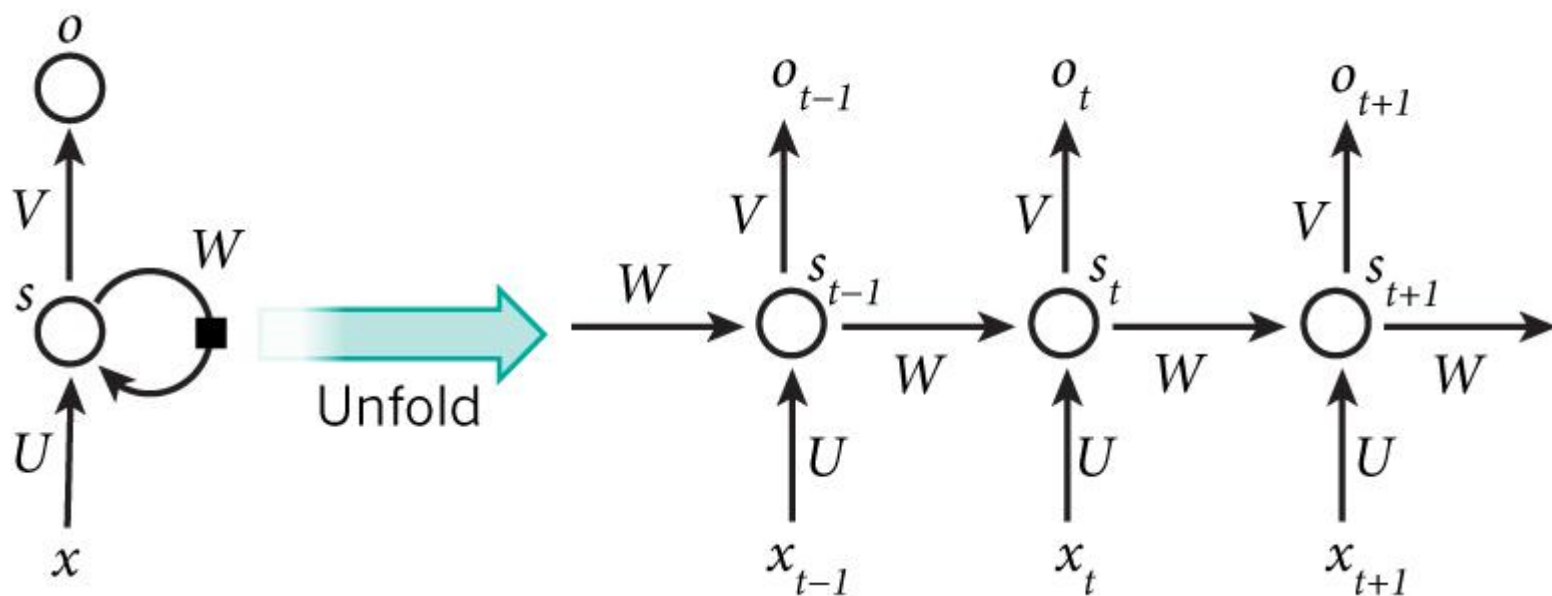
自编码器回顾



RNN回顾

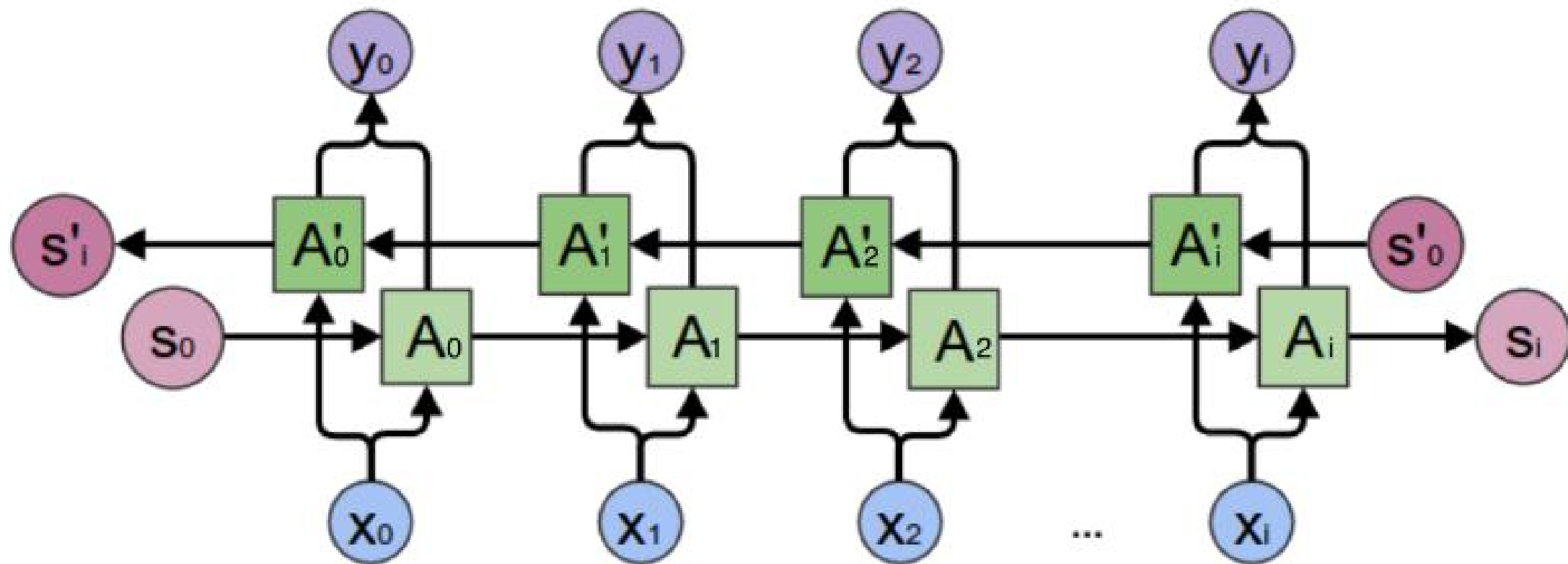
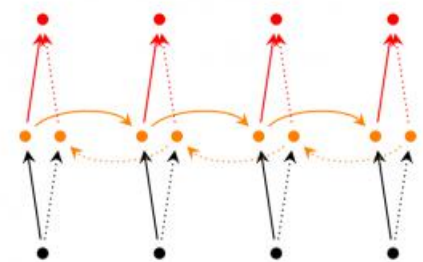


RNN回顾

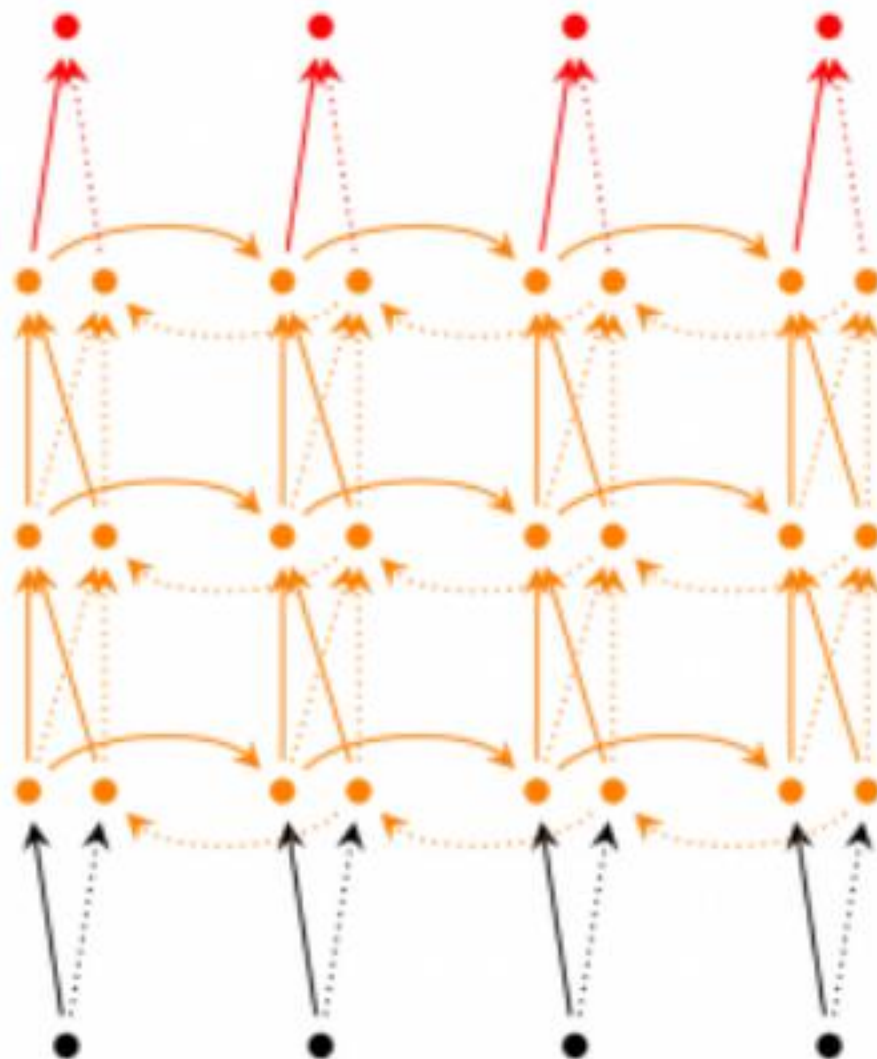


$$\begin{aligned} s_t &= Ux_t + Wh_{t-1} \\ h_t &= f(Ux_t + Wh_{t-1}) \\ o_t &= g(Vh_t) \end{aligned}$$

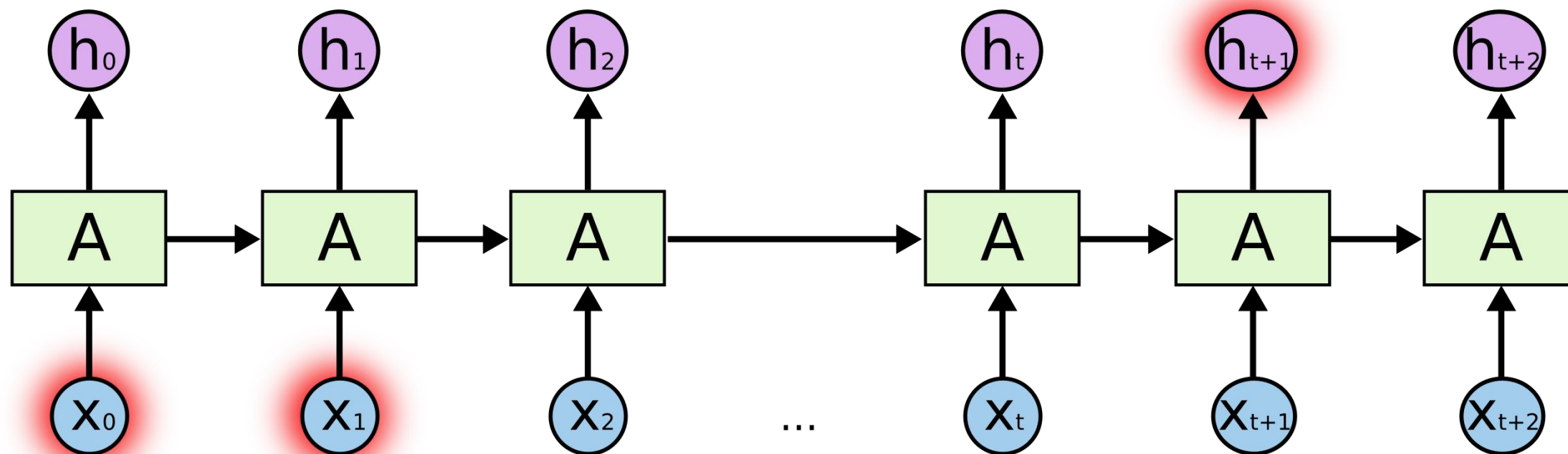
Bidirectional RNN回顾



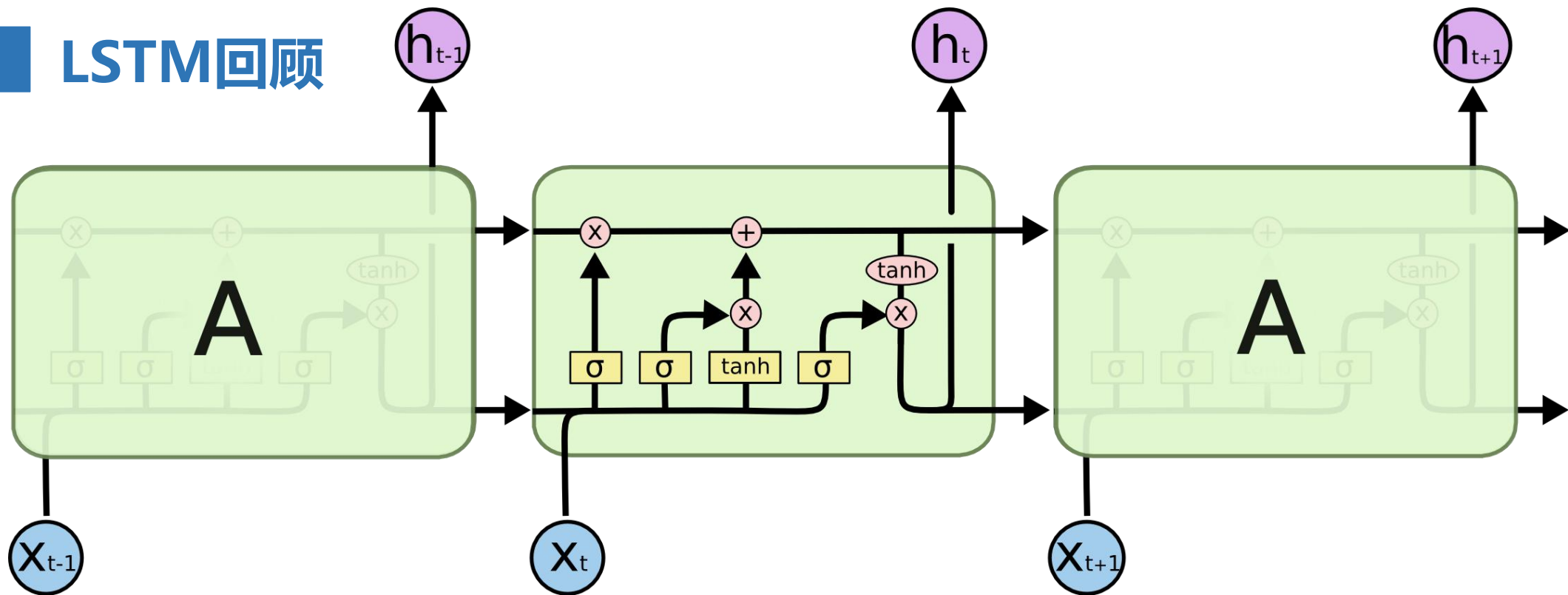
Deep Bidirectional RNN回顾



LSTM回顾



LSTM回顾



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

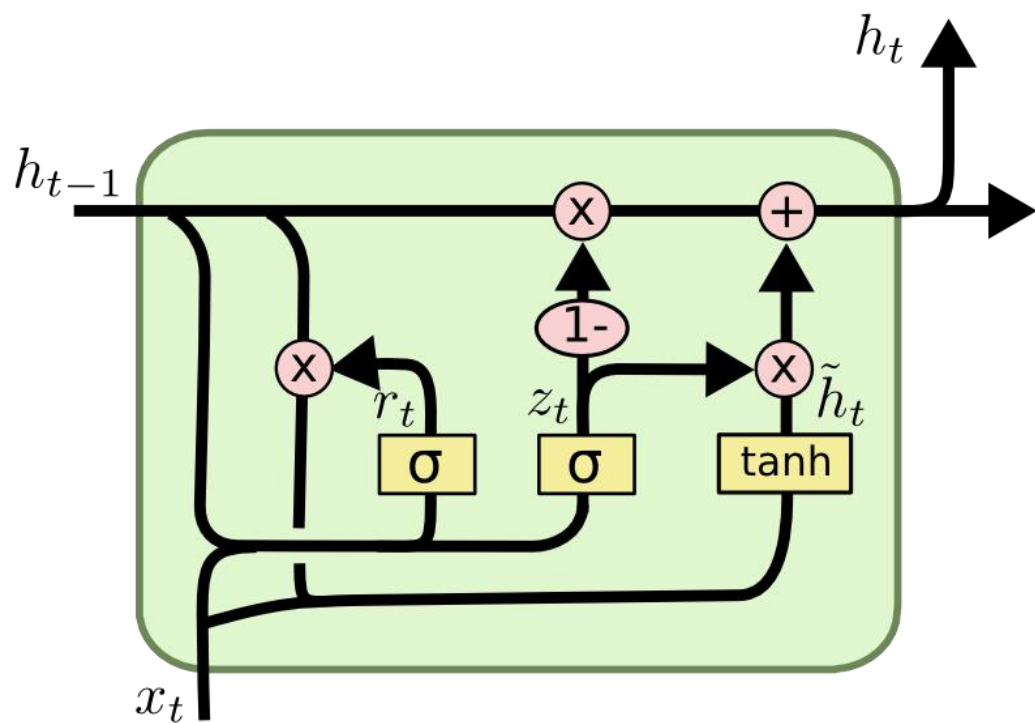
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

GRU回顾



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

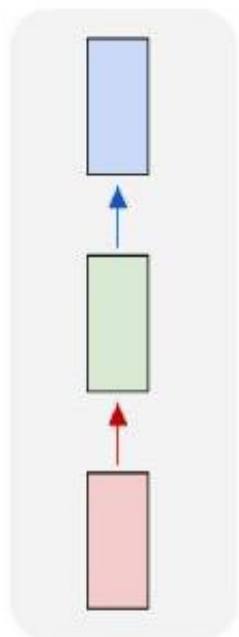
$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

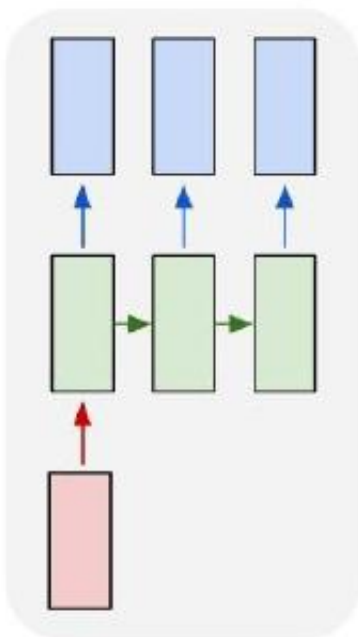
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

RNN结构回顾

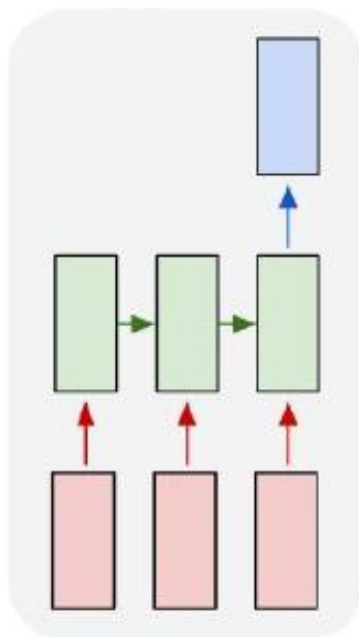
one to one



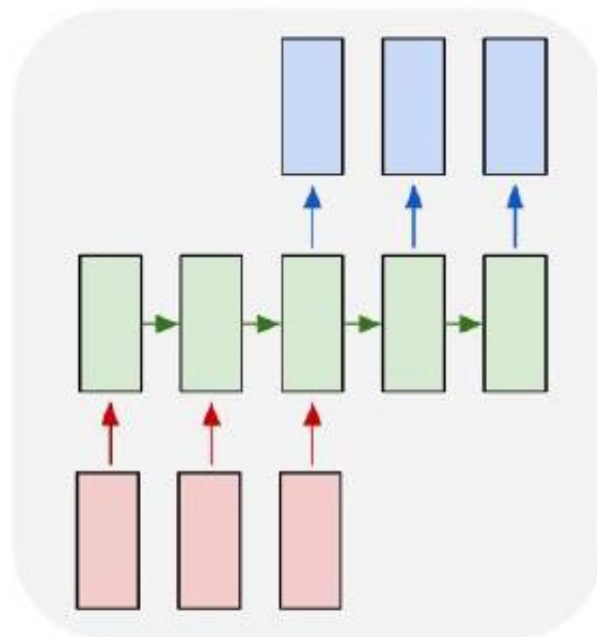
one to many



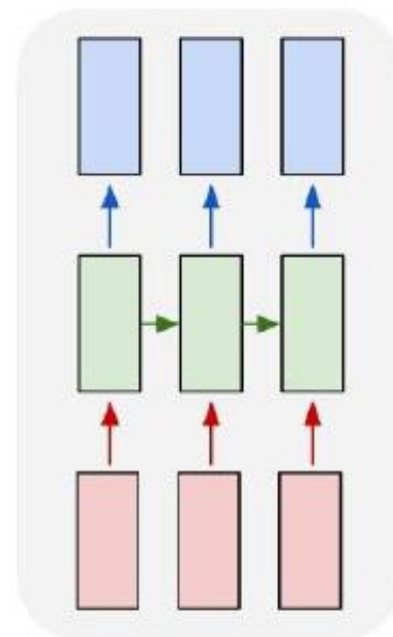
many to one



many to many

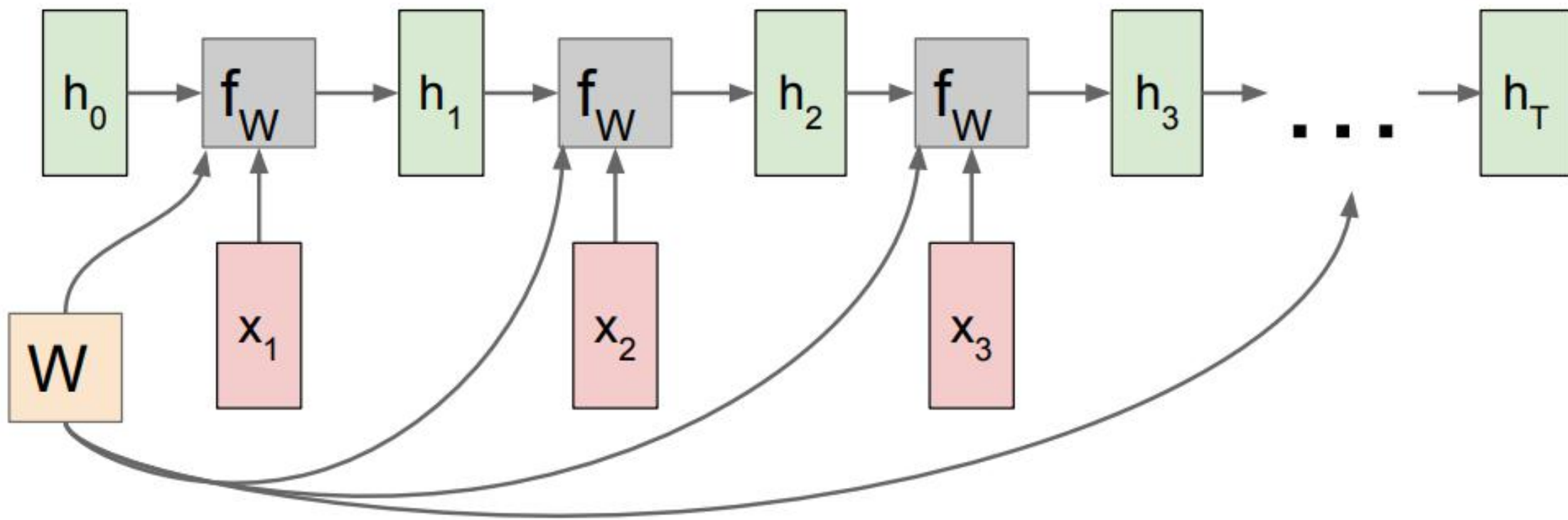


many to many



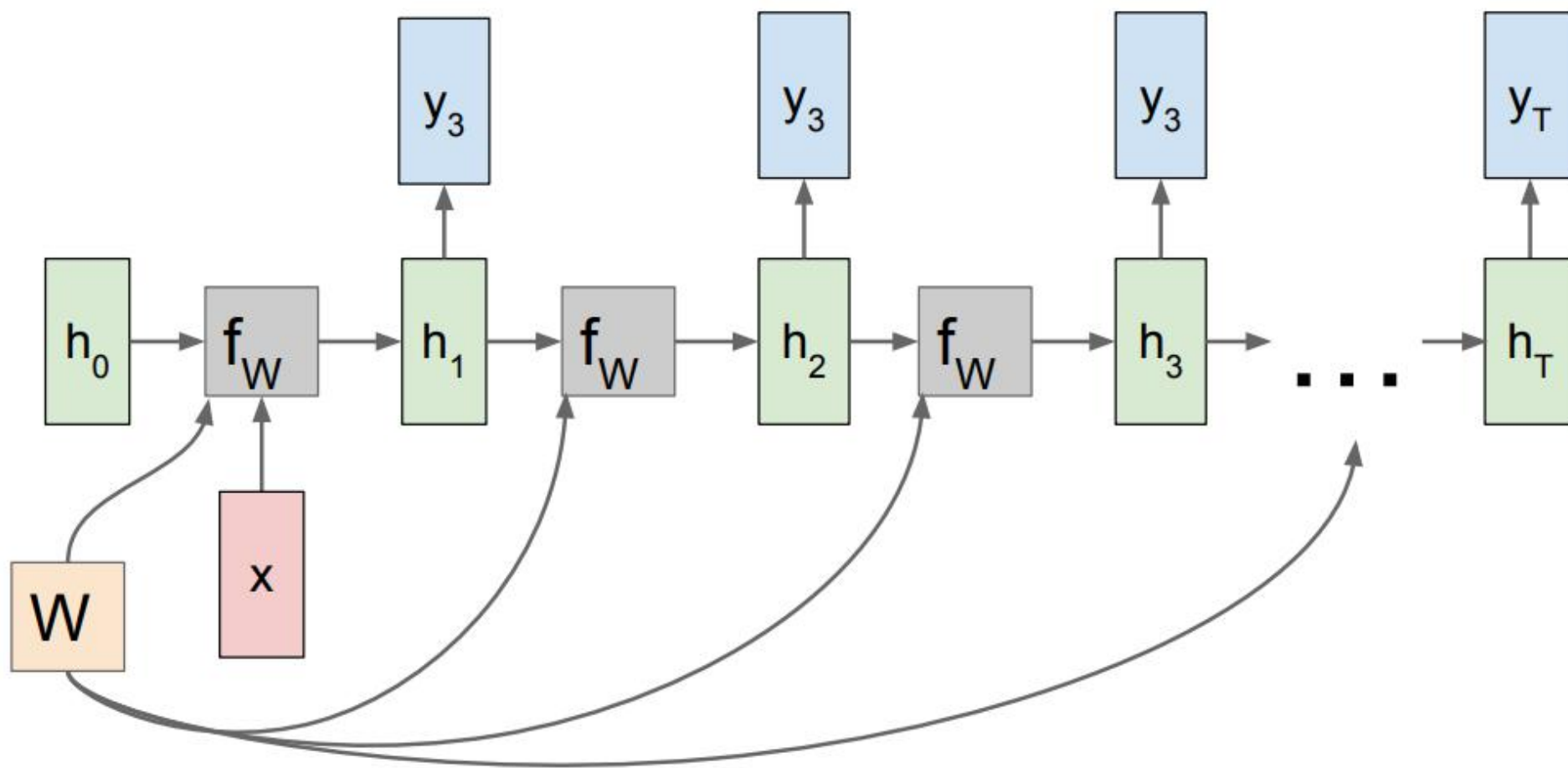
RNN结构回顾

Re-use the same weight matrix at every time-step



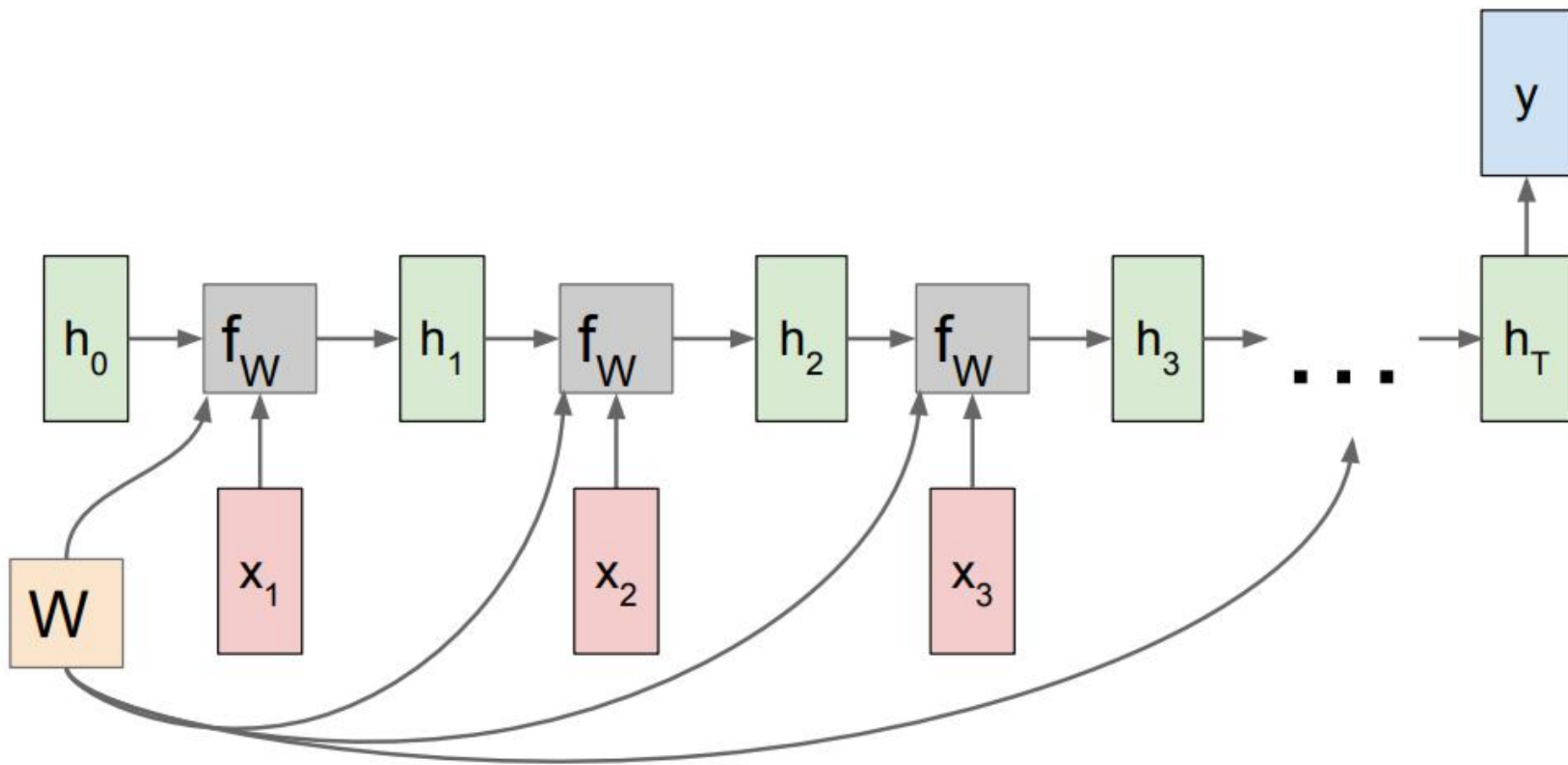
RNN结构回顾

RNN: Computational Graph: One to Many



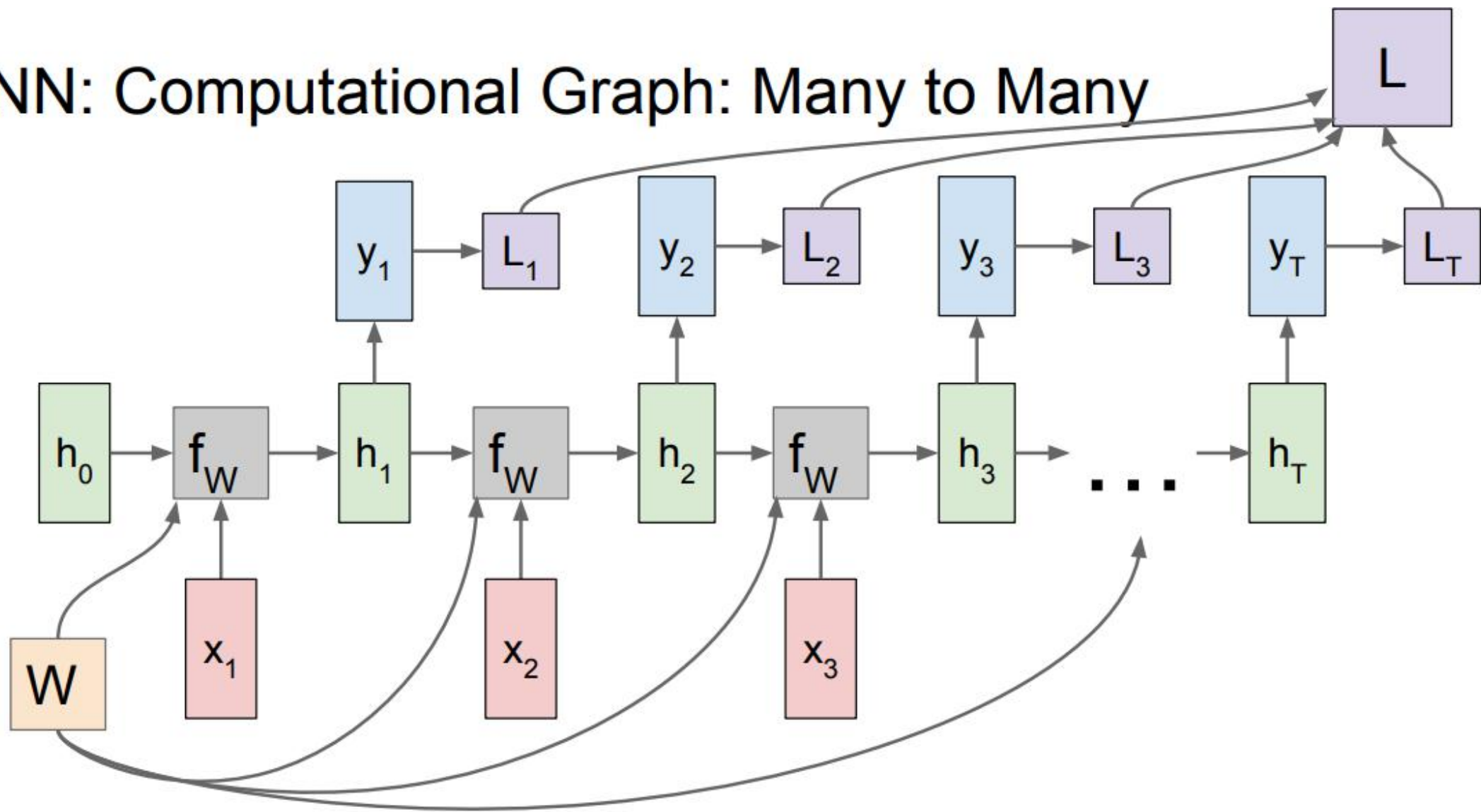
RNN结构回顾

RNN: Computational Graph: Many to One

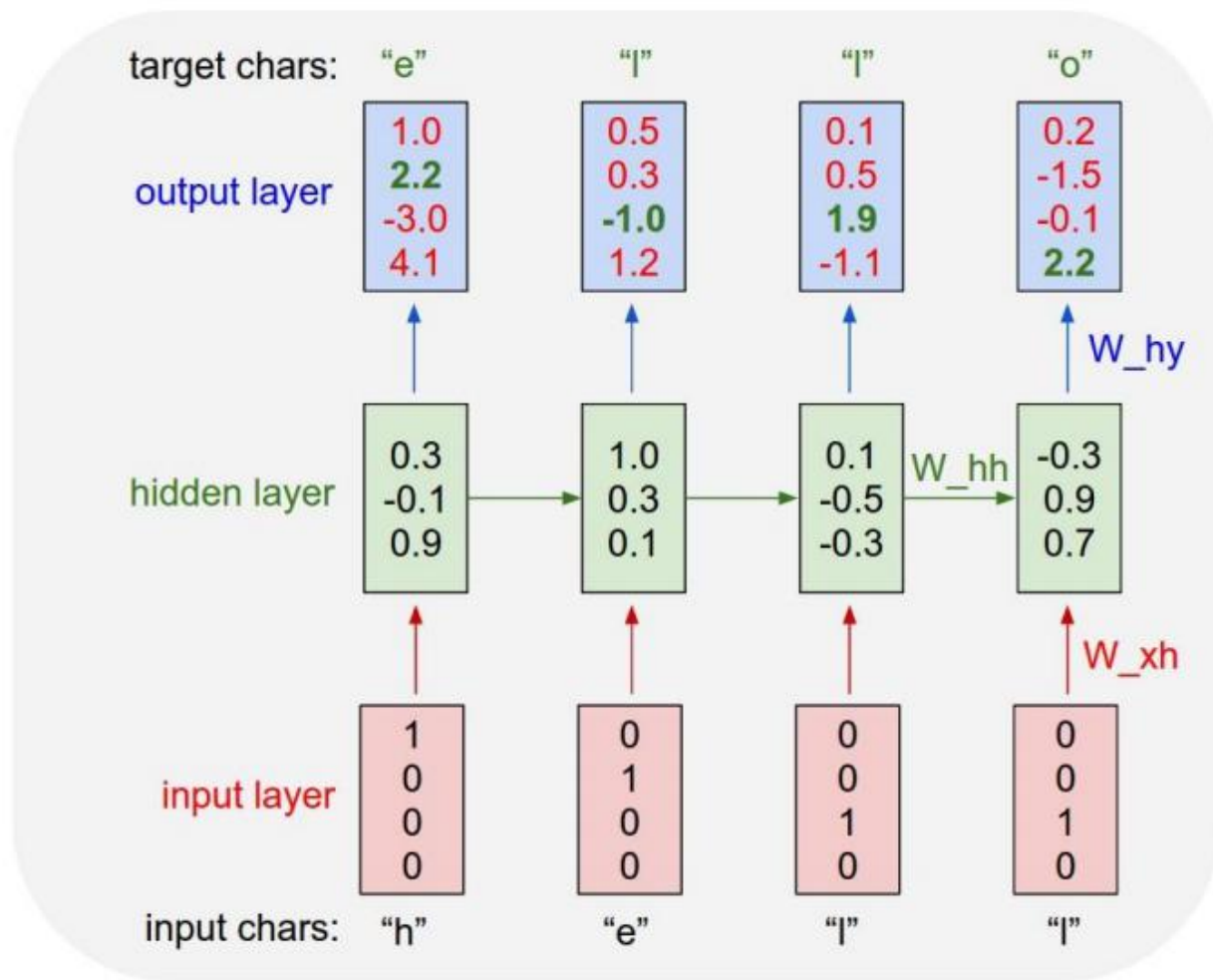


RNN结构回顾

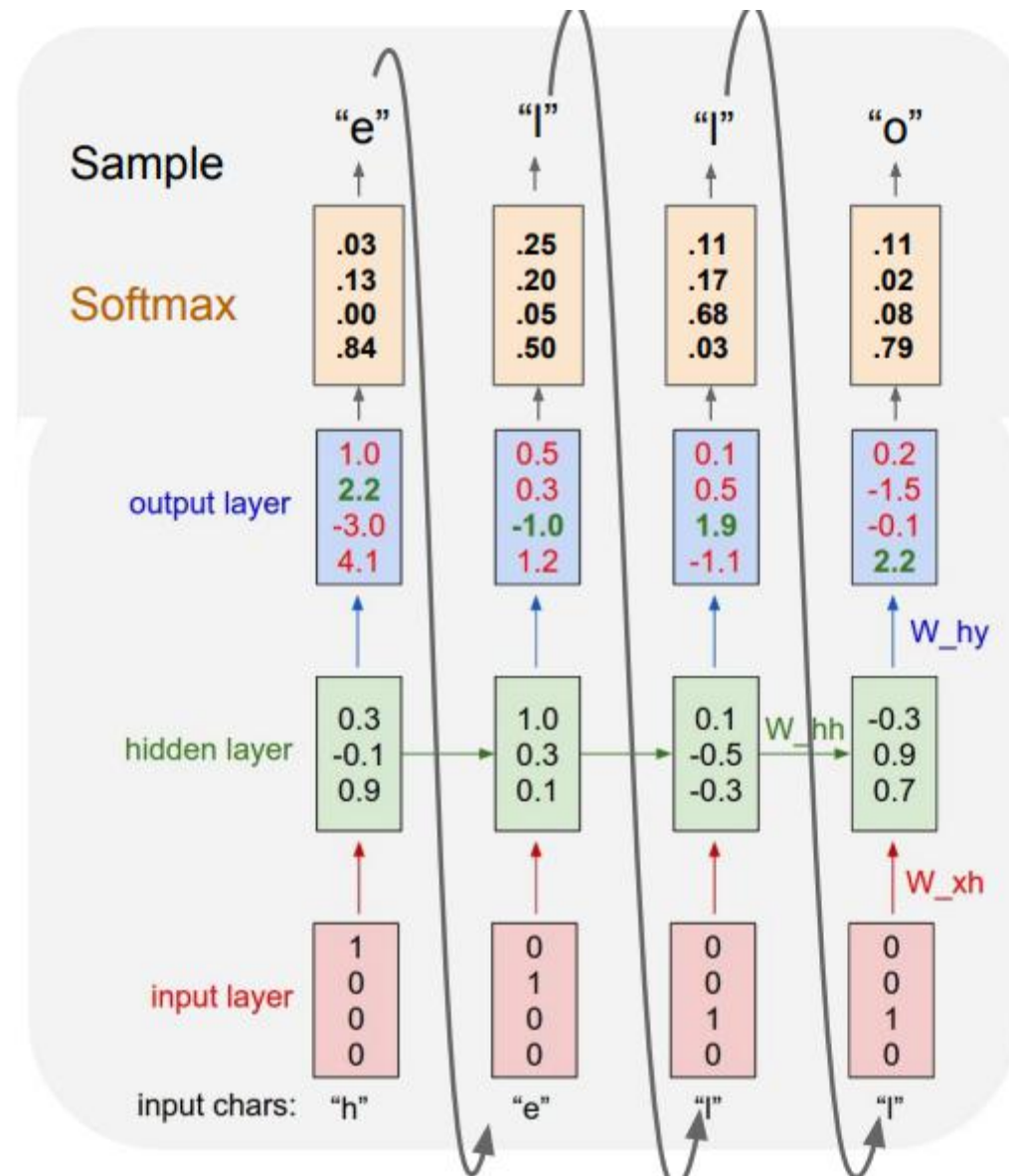
RNN: Computational Graph: Many to Many



RNN结构回顾



训练阶段



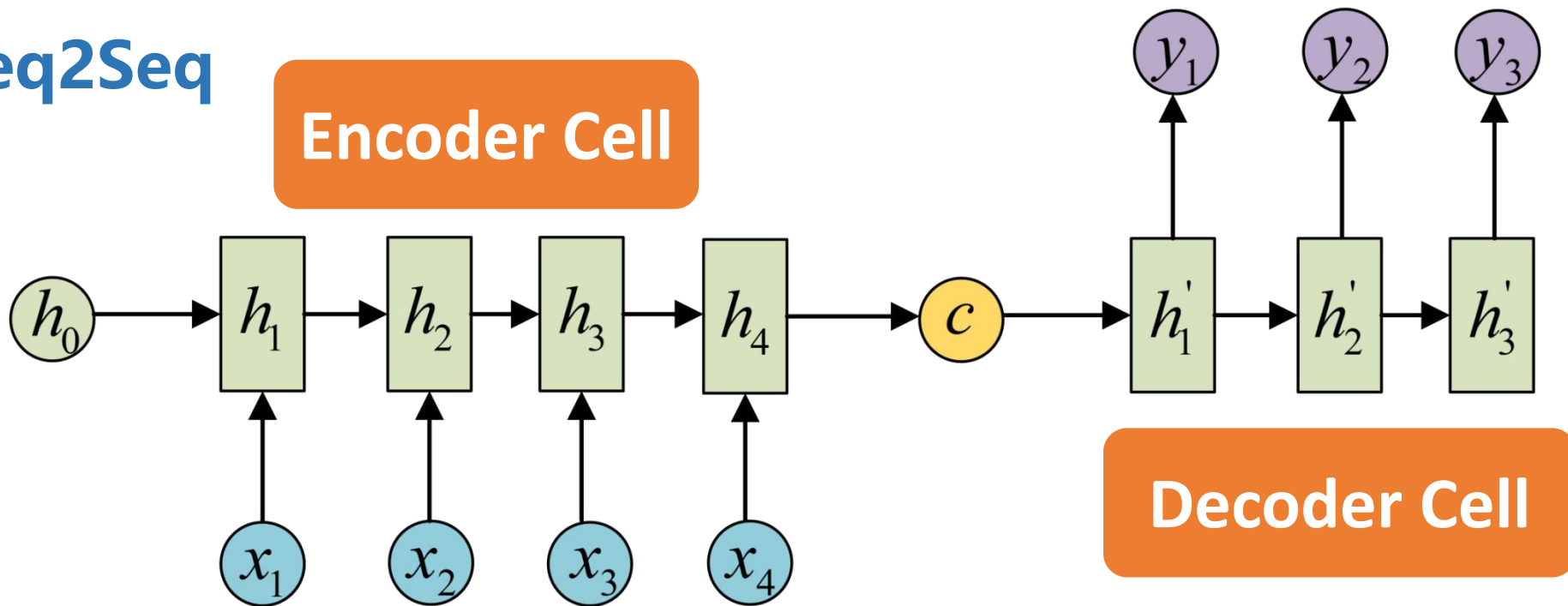
预测阶段

Seq2Seq

- Seq2Seq(Sequence to Sequence)，它被提出于2014年，最早由两篇文章独立地阐述了它主要思想，分别是Google Brain团队的《Sequence to Sequence Learning with Neural Networks》和Yoshua Bengio团队的《Learning Phrase Representation using RNN Encoder-Decoder for Statistical Machine Translation》。
- Seq2Seq属于一种Encoder-Decoder结构。



Seq2Seq



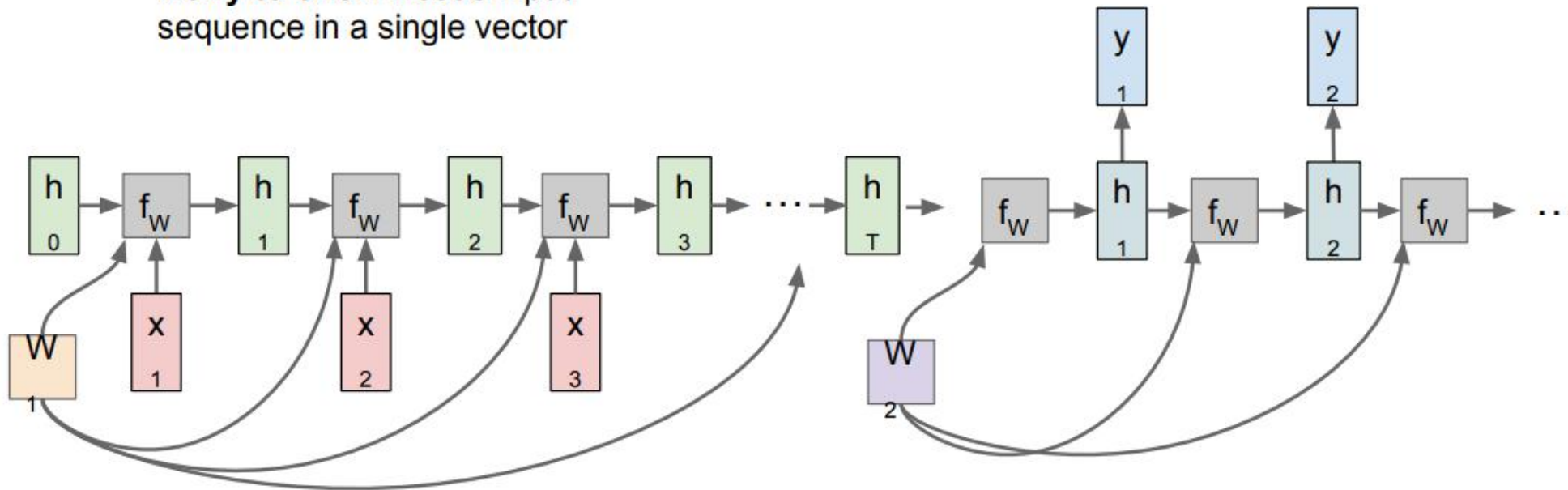
- Encoder-Decoder 这种结构的，其中 Encoder 是一个 RNNCell (RNN, GRU, LSTM 等) 结构。每个 timestep，我们向 Encoder 中输入一个字/词（一般是表示这个字/词的一个实数向量），直到我们输入这个句子的最后 一个字/词 x_T ，然后输出整个句子的语义向量 c （一般情况下， $c = h_T = F(W[x_T; h_{T-1}])$ ， x_T 是最后一个 timestep 输入）。因为 RNN 的特点就是把前面每一步的输入信息都考虑进来了，所以理论上这个 c 就能够把整个句子的信息都包含了，我们可以把 c 当成这个句子的一个语义表示，也就是一个句向量。在 Decoder 中，我们根据 Encoder 得到的句向量 c ，一步一步地把蕴含在其中的信息分析出来。

Seq2Seq

Sequence to Sequence: Many-to-one + one-to-many

Many to one: Encode input sequence in a single vector

One to many: Produce output sequence from single input vector



Seq2Seq

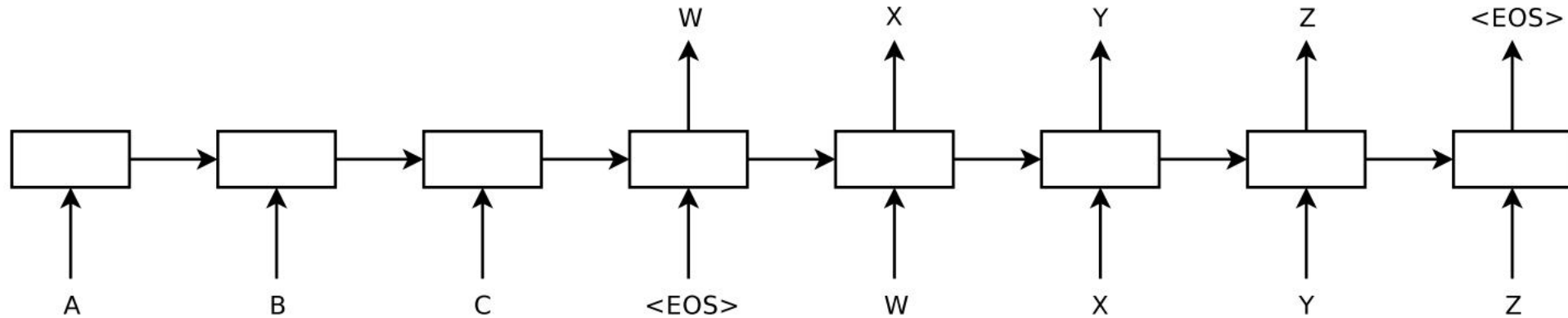


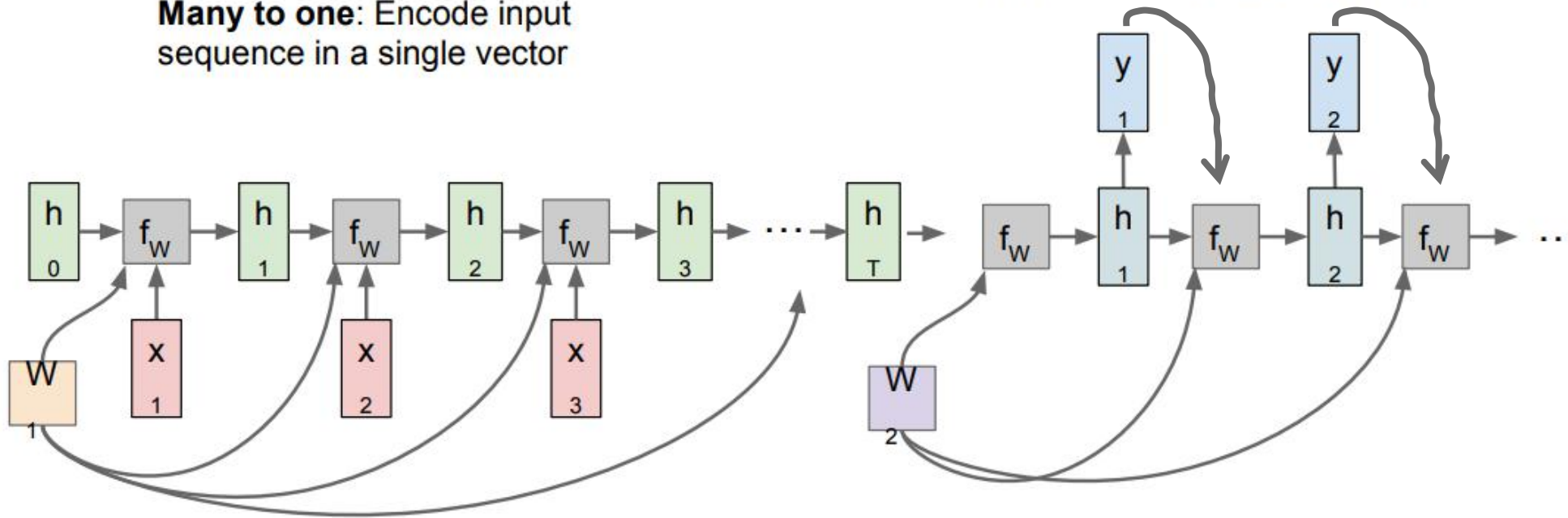
Figure 1: Our model reads an input sentence “ABC” and produces “WXYZ” as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

Seq2Seq

Sequence to Sequence: Many-to-one + one-to-many

Many to one: Encode input sequence in a single vector

One to many: Produce output sequence from single input vector

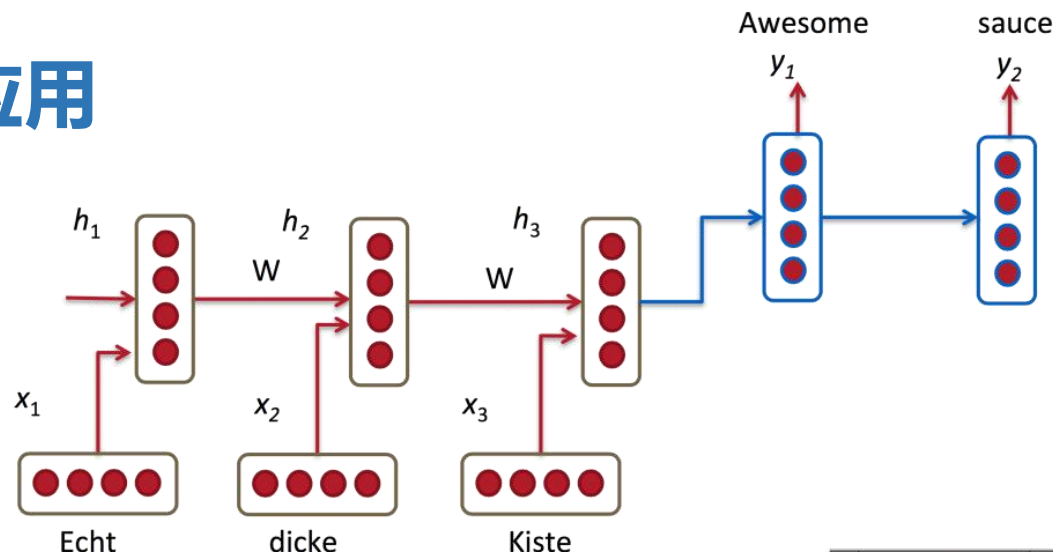


Seq2Seq理解

- 输入：小明 吃 苹果
- 希望输出： xiao ming eats apples

Seq2Seq应用

- 机器翻译



- 自动对话机器人

Post	Model	Response(Original)	Response(Translated)
情人节更庸俗的事就是，对情人节的接待规格待遇提出要求。 What is more vulgar than Valentine's day is to make specific demands on the reception of the Valentine's day.	Seq2Seq	这才是真正的情人节。	This is a real Valentine's day.
	ECM	Like 情人节快乐！	Happy Valentine's Day!
		Happiness 哈哈，这也太浪漫了吧！	Aha, this is too romantic!
		Sadness 我也想要这样的情人节。	I also want this kind of Valentine's day, sigh.
		Disgust 这就是所谓的情人节！	This is the so-called Valentine's day!
		Anger 这是赤裸裸的炫耀啊！	This is shameless show-off!
知道真相的我眼泪笑下来。 Knowing the truth, I laughed myself to tears.	Seq2Seq	知道真相的我眼泪掉下来。	After knowing the truth, I burst into tears.
	ECM	Like 太可爱了，太可爱了！	Too lovely, it's too lovely.
		Happiness 笑得我眼泪都出来了。	I can't help laugh myself to tears.
		Sadness 知道真相的我眼泪掉下来。	After knowing the truth, I burst into tears.
		Disgust 这是真的还是假的？	Is this true?
		Anger 什么意思？没看懂。	What do you mean? I don't understand.

ID	Post	NRM-glo	NRM-loc	NRM-hyb	Rtr-based
1	每年元旦都来高烧。 High fever attacks me every New Year's Day	多休息，增强免疫力！ Rest more to boost your immune system.	每年元旦都去医院看病 Go to see a doctor every New Year's Day	祝早日康复，身体健康！ Get well soon and stay healthy!	高烧... High fever...
	又老了一岁，感谢我们组，我很高兴 I gain one more year. Grateful to my group, so happy	生日快乐！永远支持你！ Happy birthday! Will stand by you forever!	生日快乐！永远年轻漂亮！ Happy birthday! Wish you always stay young and pretty!	老了老了，岁月不饶人啊！ Getting old now. Time has no mercy.	我只是老了一岁 I just gain only one more year.
3	吃素第一天，坚持住，崔朵拉。 First day of being a vegetarian. Hold on, CuiDuoLa	必须来支持！加油！ I come to offer my support. Keep it up!	坚持住，坚持下去。 Hold on, keep it up.	每天都是素食主义者 I have been being a vegetarian everyday	坚持住！ Hold on!
	这种天气真应该带个帅哥出去踏青赏花 We should go out with some cute guys to enjoy a great outing in such a nice weather.	这是哪儿啊？真美！ Where is it? It is so beautiful!	阳光明媚，心情舒畅。 Such a nice sunny day! I am in a great mood.	这种天气真不是盖的。 It is indeed a very nice weather.	文山啊 出去踏青寻找灵感 哈哈 WenShan, let's go out to get some inspiration. Ha! Ha!
4					

Seq2Seq应用

- 文档摘要自动生成

Good quality summary output

S: a man charged with the murder last year of a british backpacker confessed to the slaying on the night he was charged with her killing , according to police evidence presented at a court hearing tuesday . ian douglas previte , ## , is charged with murdering caroline stuttle , ## , of yorkshire , england

T: man charged with british backpacker 's death confessed to crime police officer claims

O: man charged with murdering british backpacker confessed to murder

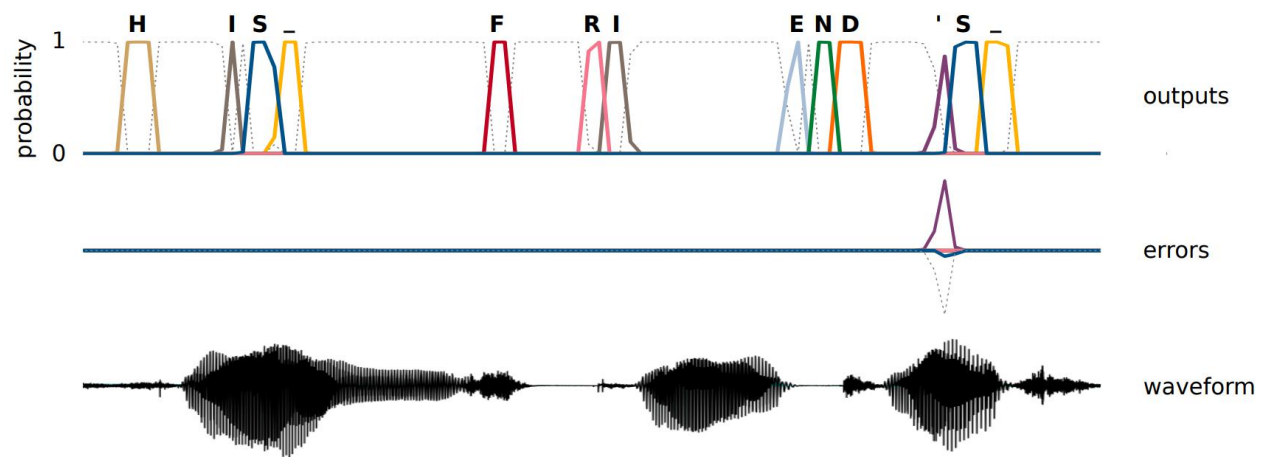
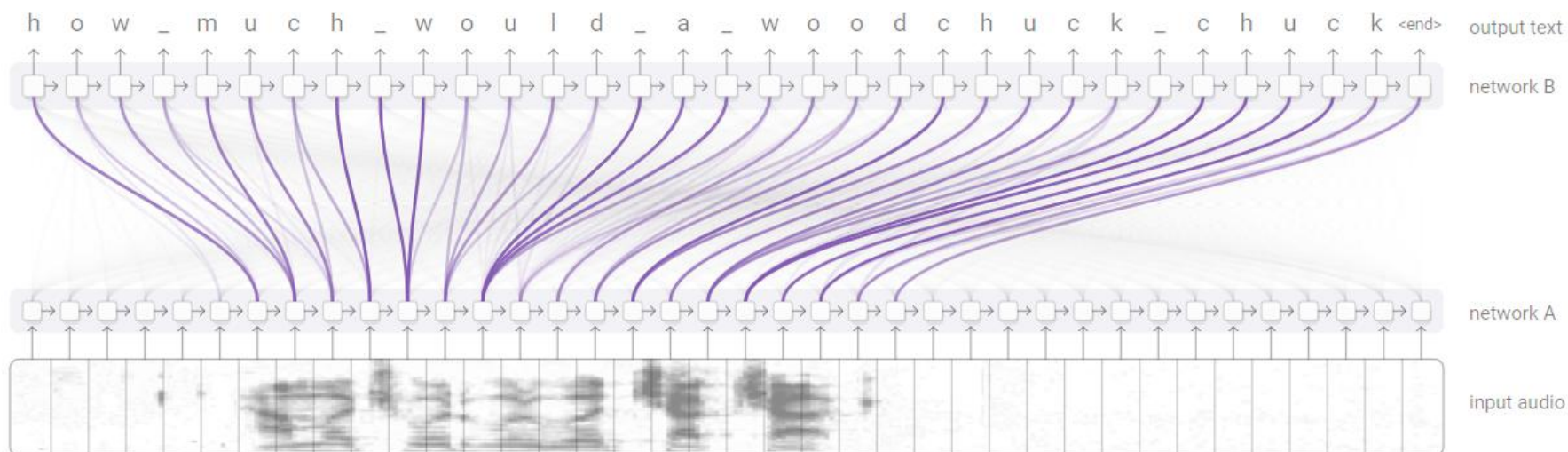
- 文本生成

秋夕湖上 By a Lake at Autumn Sunset 一夜秋凉雨湿衣， A cold autumn rain wetted my clothes last night, 西窗独坐对夕晖。 And I sit alone by the window and enjoy the sunset. 湖波荡漾千山色， With mountain scenery mirrored on the rippling lake, 山鸟徘徊万籁微。 A silence prevails over all except the hovering birds.	秋夕湖上 By a Lake at Autumn Sunset 荻花风里桂花浮， The wind blows reeds with osmanthus flying, 恨竹生云翠欲流。 And the bamboos under clouds are so green as if to flow down. 谁拂平湖新镜面， The misty rain ripples the smooth surface of lake, 飞来烟雨暮天愁。 And I feel blue at sunset .
---	---

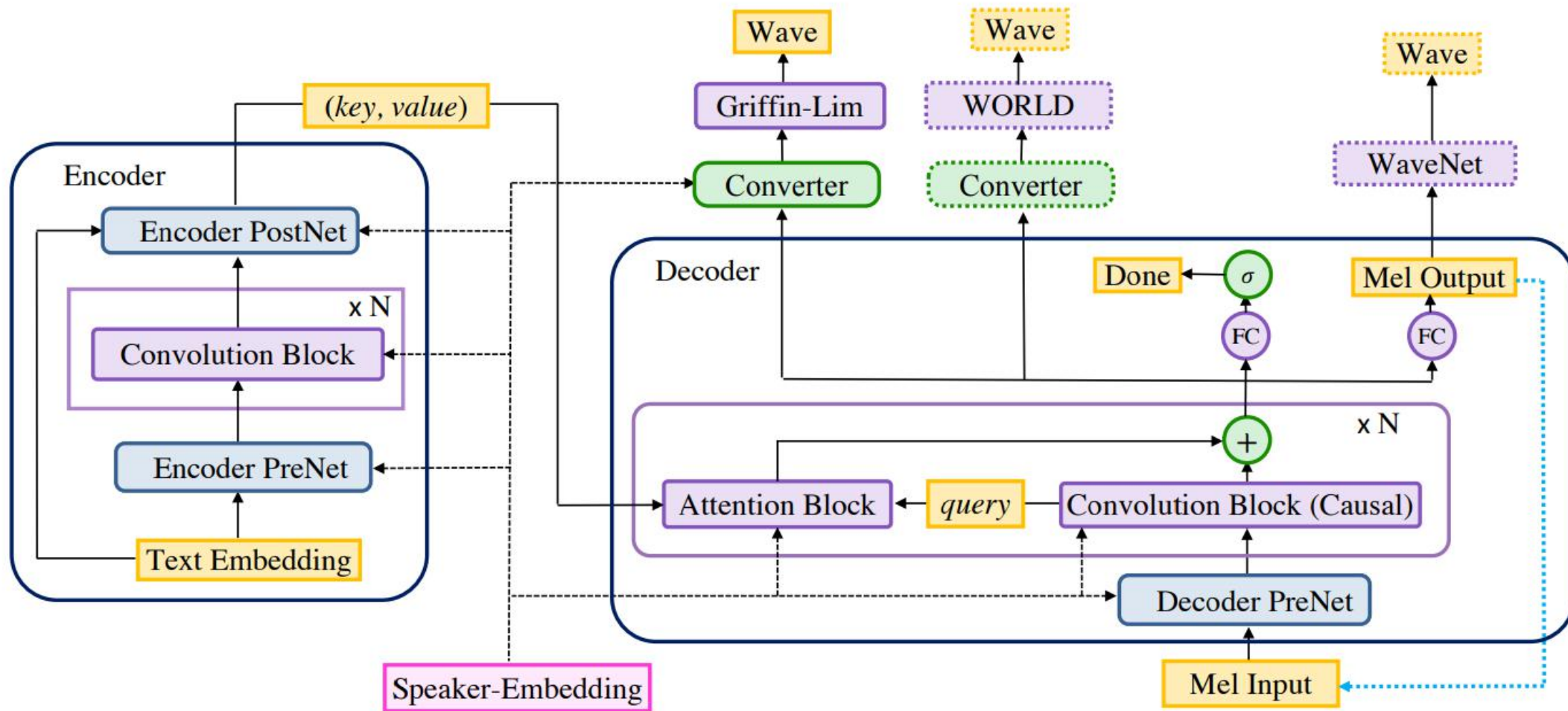
▲ 图5：左边是机器生成的诗词，右边是一首宋代诗词

Seq2Seq应用

- 语音识别/合成/语音-文本转换



Seq2Seq应用



百度Deep Voice v3

Seq2Seq应用

- 图片描述自动生成



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



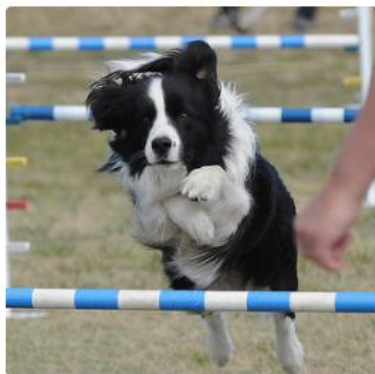
"two young girls are playing with lego toy."



"boy is doing backflip on wakeboard."



"girl in pink dress is jumping in air."



"black and white dog jumps over bar."

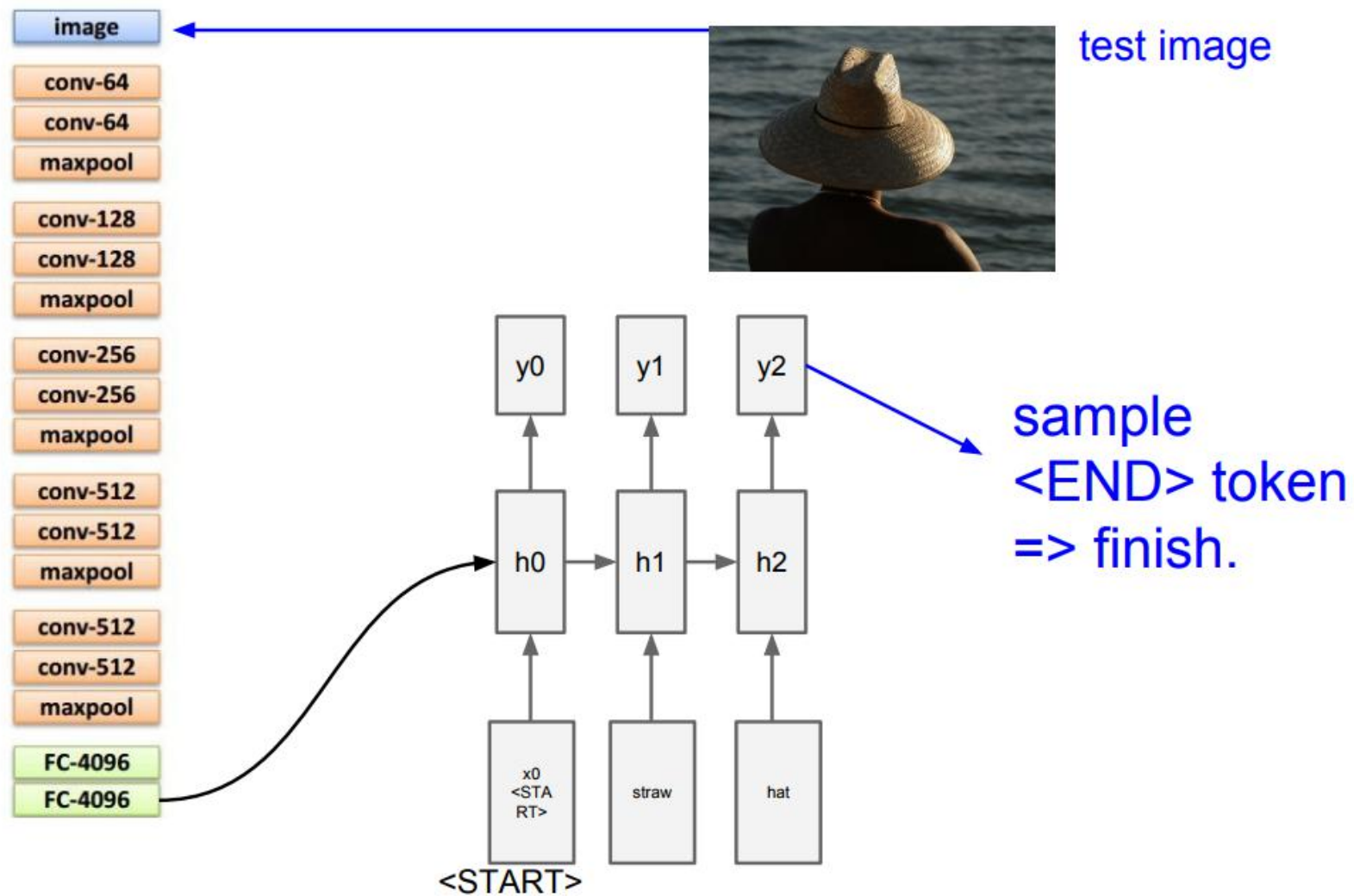


"young girl in pink shirt is swinging on swing."



"man in blue wetsuit is surfing on wave."


Seq2Seq应用



Seq2Seq应用

- Visual Question Answering(VQA, 视觉问答系统)
 - <https://visualqa.org/index.html>

Image



GT Question 戴帽子的男孩在干什么?
What is the boy in green cap doing?

GT Answer 他在玩滑板。
He is playing skateboard.



图片中有人么?
Is there any person in the image?

有。
Yes.

Who is wearing glasses?

man woman



Where is the child sitting?

fridge arms



GT Question 房间里的沙发是什么质地的?
What is the texture of the sofa in the room?

GT Answer 布艺。
Cloth.



这个人在挑菜么?
Is the man trying to buy vegetables?

是的。
Yes.

Is the umbrella upside down?

yes no

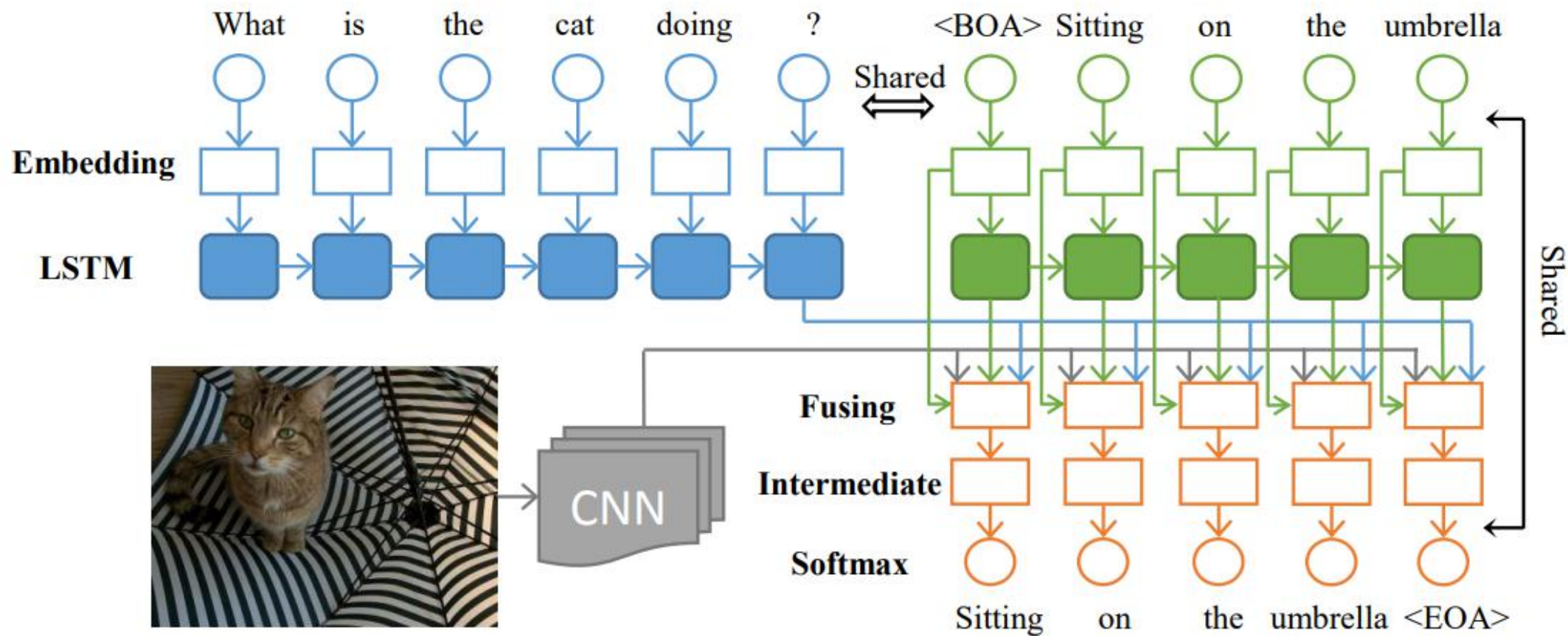


How many children are in the bed?

2 1



Seq2Seq应用

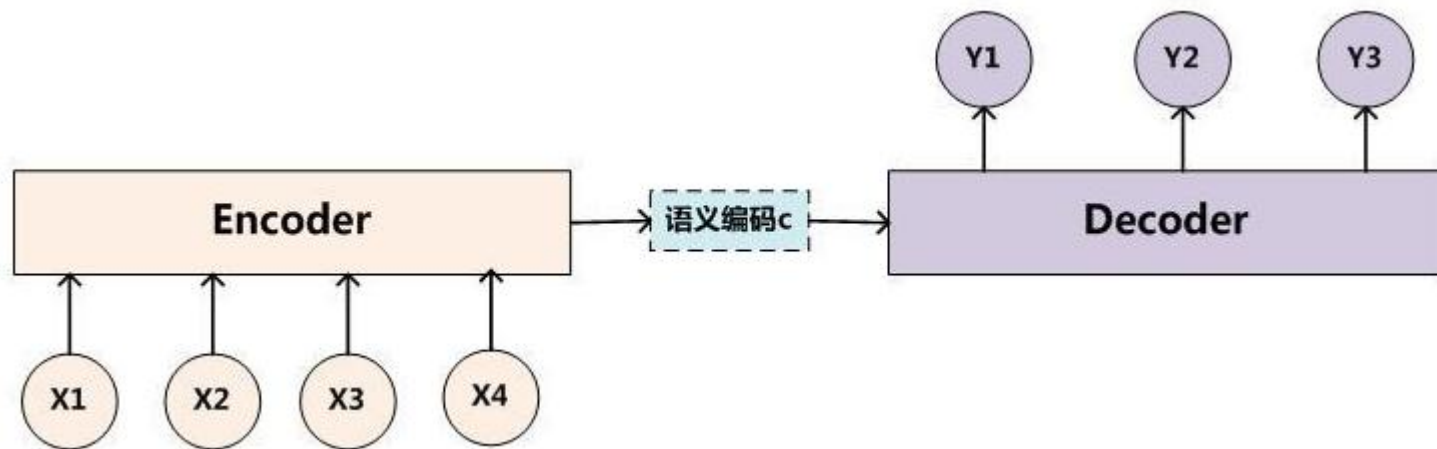


Seq2Seq应用总结

- 总而言之，Seq2Seq应用场景，包括了经典的机器翻译、文本摘要和对话生成等，也包括了一些非常有趣的应用，比如：根据公式图片生成 latex 代码，生成 commit message 等。自然语言生成（NLG）是一个非常有意思，也非常有前途的研究领域，简单地说，就是解决一个条件概率 $p(\text{output} | \text{context})$ 的建模问题，即根据 context 来生成 output，这里的 context 可以非常零活多样，大家都是利用深度学习模型对这个条件概率进行建模，同时加上大量的训练数据和丰富的想象力，可以实现很多有趣的工作。Seq2Seq 是一个简单易用的框架，开源的实现也非常多，但并不意味着直接生搬硬套就可以了，需要具体问题具体分析。此外，对于生成内容的控制，即 decoding 部分的研究也是一个非常有意思的方向，比如：如何控制生成文本的长度，控制生成文本的多样性，控制生成文本的信息量大小，控制生成文本的情感等等。

Seq2Seq原理

- 最基础的Seq2Seq模型包含了三个部分，即Encoder、Decoder以及连接两者的中间状态向量，Encoder通过学习输入，将其编码成一个固定大小的状态向量 c ，继而将 c 传给Decoder，Decoder再通过对状态向量 c 的学习来进行输出。下图中，图中每一个box代表了一个**RNN Cell**单元，通常是**LSTM**或者**GRU**。



Seq2Seq

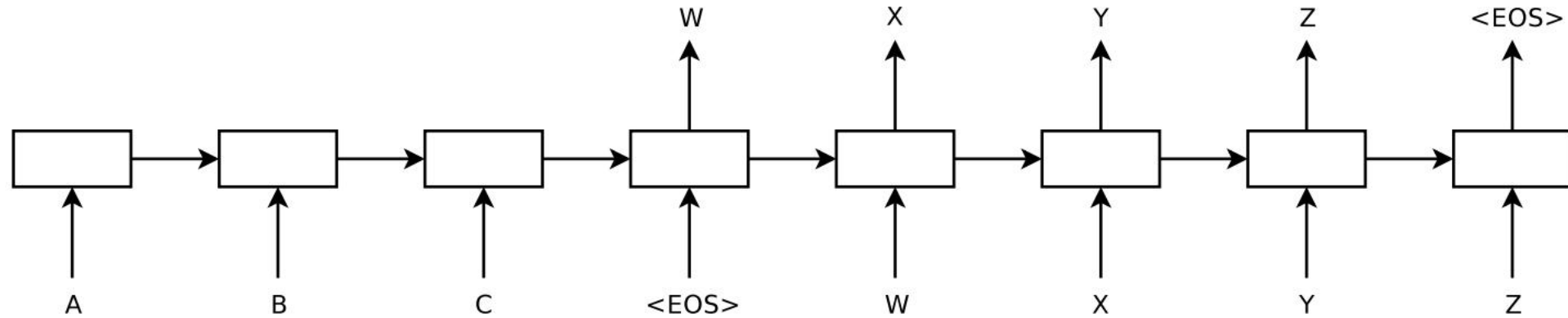


Figure 1: Our model reads an input sentence “ABC” and produces “WXYZ” as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

Seq2Seq

- Encoder-Decoder框架可以这么直观地去理解：可以把它看作适合处理由一个句子（或篇章）生成另外一个句子（或篇章）的通用处理模型。对于句子对<X,Y>，我们的目标是给定输入句子X，期待通过Encoder-Decoder框架来生成目标句子Y。X和Y可以是同一种语言，也可以是两种不同的语言。而X和Y分别由各自的单词序列构成：

$$X = (x_1, x_2, \dots, x_m)$$

$$Y = (y_1, y_2, \dots, y_n)$$

Seq2Seq

- Encoder顾名思义就是对输入句子X进行编码，将输入句子通过非线性变换转化为中间语义表示C:

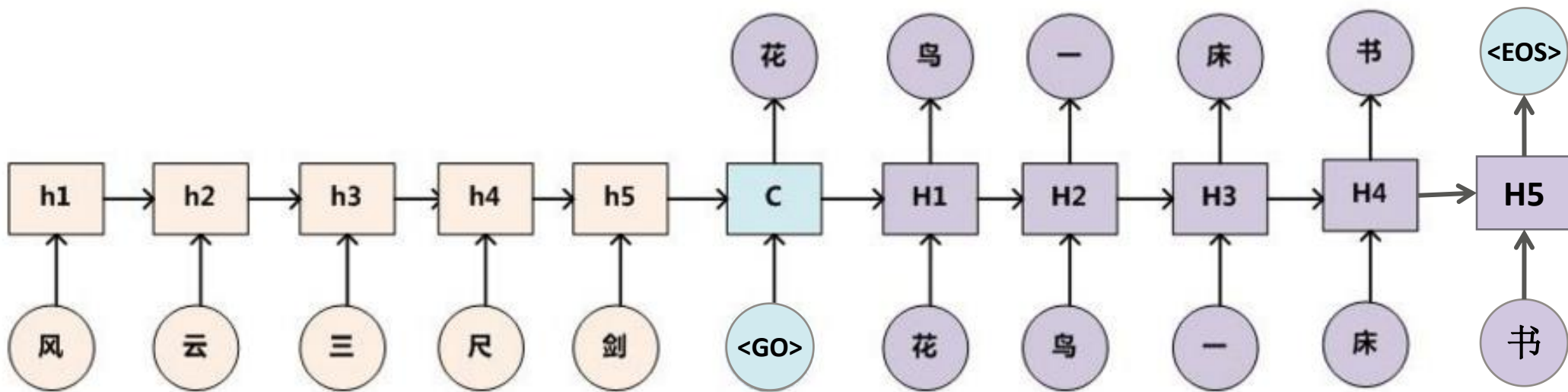
$$C = F(x_1, x_2, \dots, x_m)$$

- 对于解码器Decoder来说，其任务是根据句子X的中间语义表示C和之前已经生成的历史信息 y_1, y_2, \dots, y_{i-1} 来生成i时刻要生成的单词 y_i :每个 y_i 都依次这么产生，那么看起来就是整个系统根据输入句子X生成了目标句子Y。

$$y_i = G(C, y_1, y_2, \dots, y_{i-1})$$

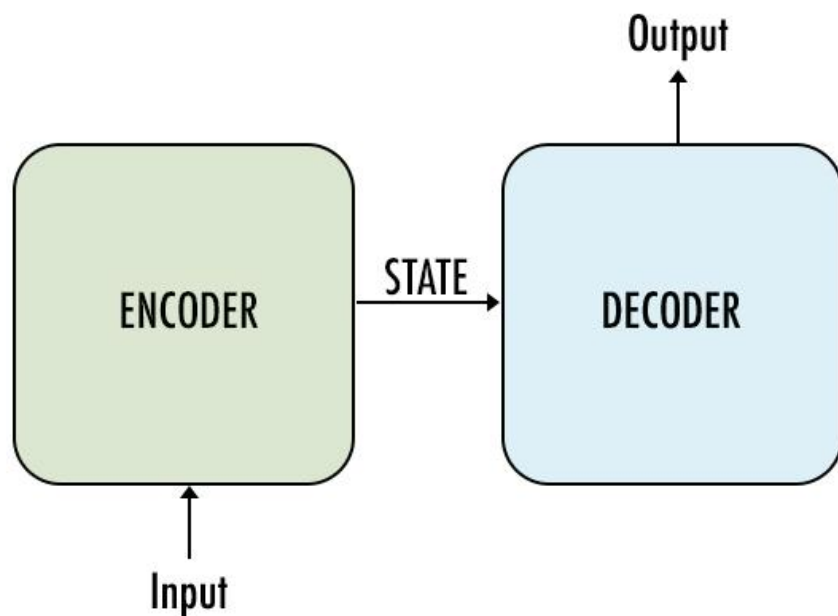
Seq2Seq案例

- 只需要找到大量的对联数据对这个模型进行训练，那么即可利用这个模型，输入上联，机器自动产生下联了。



Seq2Seq案例

Inputs	Target
How are you?	I am good
Can you fly that thing?	Not yet



编码器: [tf.nn.dynamic_rnn](#)

解码器: [tf.contrib.seq2seq.dynamic_rnn_decoder](#)

举例: `tf.nn.dynamic_rnn(cell, inputs, sequence_length=None, initial_state=None, dtype=None, parallel_iterations=None, swap_memory=False, time_major=False, scope=None)`

Cell为前面构建的RNN

`cell(tf.contrib.rnn.BasicLSTMCell);`

Inputs,为输入的文本数据, 通常是嵌入层的输出。
以及initial_state

Seq2Seq

- <PAD> 在训练中，我们将数据按批次输入。但同一批次中必须有相同的Sequence Length(序列长度/time_steps)。所以我们会用<PAD>填充较短的输入。
- <EOS> 它能告诉解码器句子在哪里结束，并且它允许解码器在其输出中表明句子结束的位置
- <UNK> 忽视词汇表中出现频率不够高而不足以考虑在内的文字,将这些单词替换为 <UNK>
- <GO> 解码器的第一个时间步骤的输入，以使解码器知道何时开始产生输出

Seq2Seq案例

0	<PAD>	11	can
1	<EOS>	12	you
2	<UNK>	13	fly
3	<GO>	14	that
4	how	15	thing
5	are	16	not
6	you	17	yet
7	?		
8	i		
9	am		
10	good		

Seq2Seq案例

Inputs

How are you?
Can you fly that thing?

Target

I am good
Not yet



填充

how
can

are
you

you
fly

?
that

<PAD>
thing

<PAD>
?

Seq2Seq案例

Inputs

How are you?

Can you fly that thing?



填充

how
can

are
you

you
fly

?
that



Word2id

4
11

5
12

6
13

7
14

0
15

0
7

Target

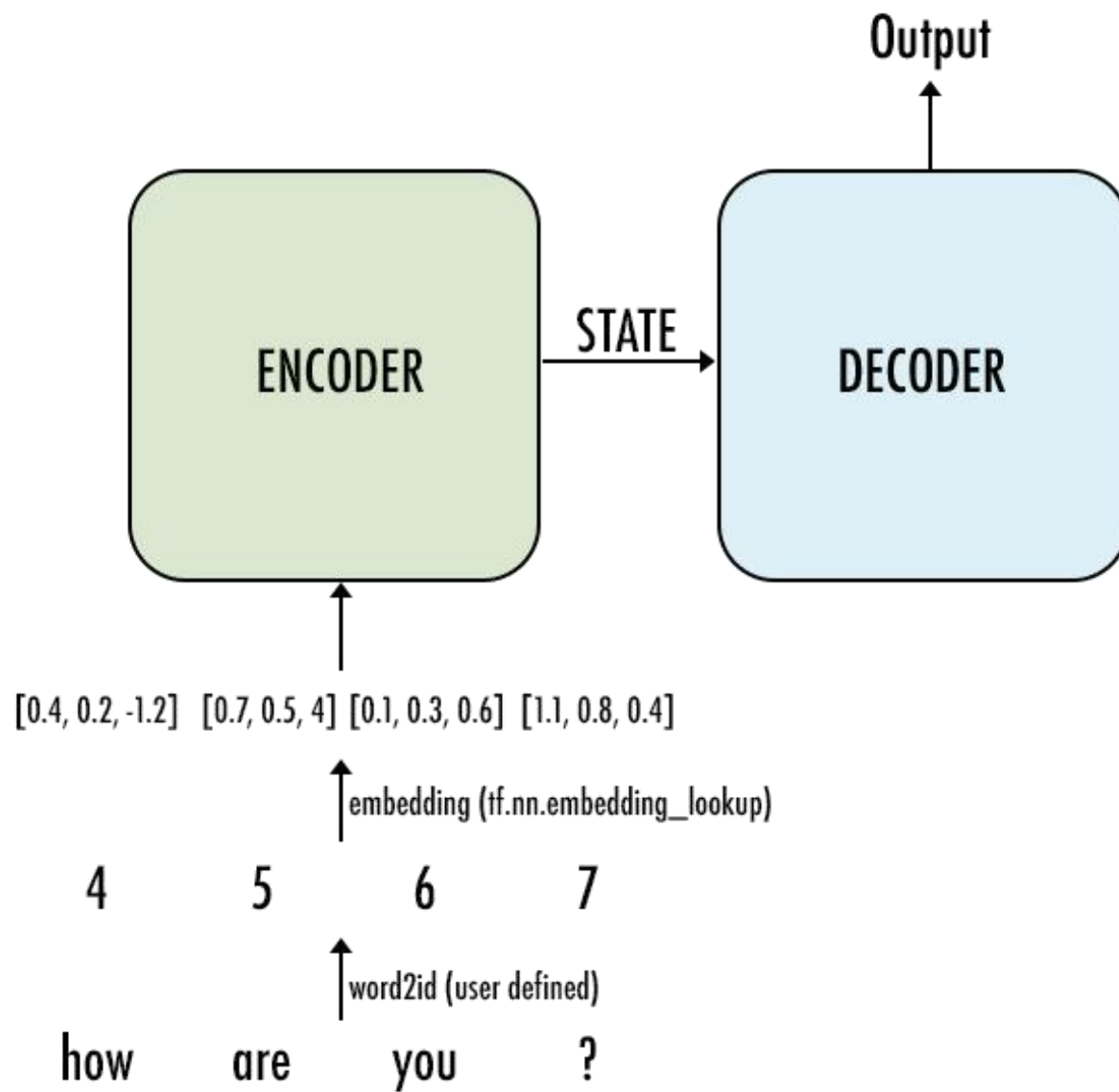
I am good

Not yet

<PAD>
thing

<PAD>
?

Seq2Seq案例



Seq2Seq案例

Inputs

How are you?
Can you fly that thing?

Target

I am good
Not yet

填充

<GO>
<GO>

i
not

am
yet

good
<EOS>

<EOS>
<PAD>

Seq2Seq案例

Inputs

How are you?
Can you fly that thing?

<GO>
<GO>

i
not

am
yet

3
3

8
16

9
17

Target

I am good
Not yet

填充

good
<EOS>

<EOS>
<PAD>

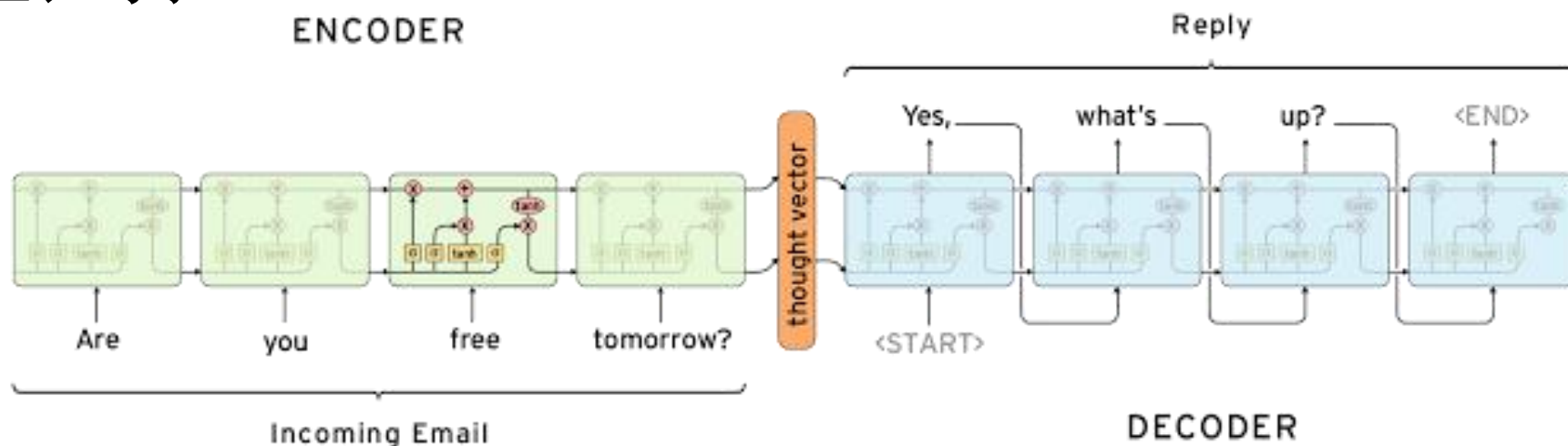
Word2id

10
1

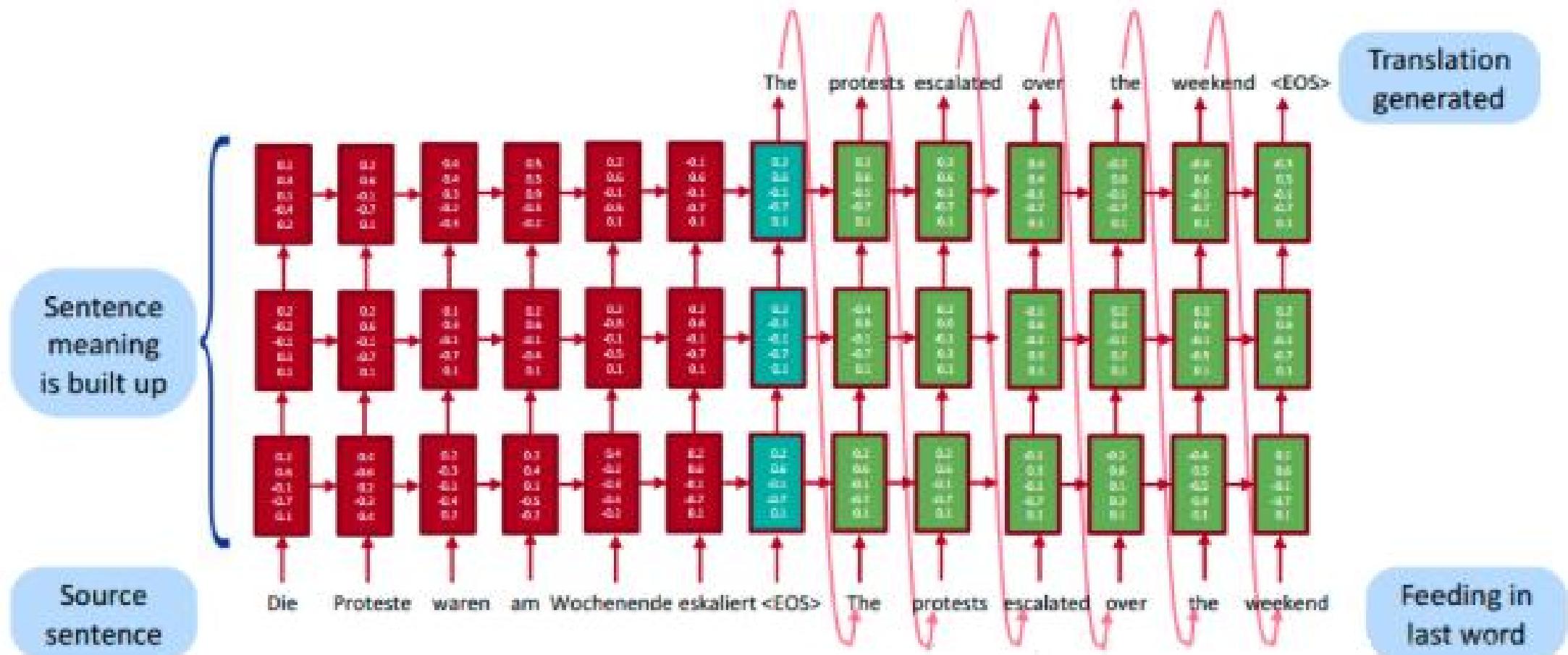
1
0

Seq2Seq

- 将RNN模块换成LSTM，则效果如下图。Encoder 和 Decoder 都是 4 个时间步长的 LSTM(但是只有两个RNN Cell)。小技巧：将源句子顺序颠倒后再输入 Encoder 中，比如源句子为“A B C”，那么输入 Encoder 的顺序为“C B A”，经过这样的处理后，取得了很大的提升，而且这样的处理使得模型能够很好地处理长句子。

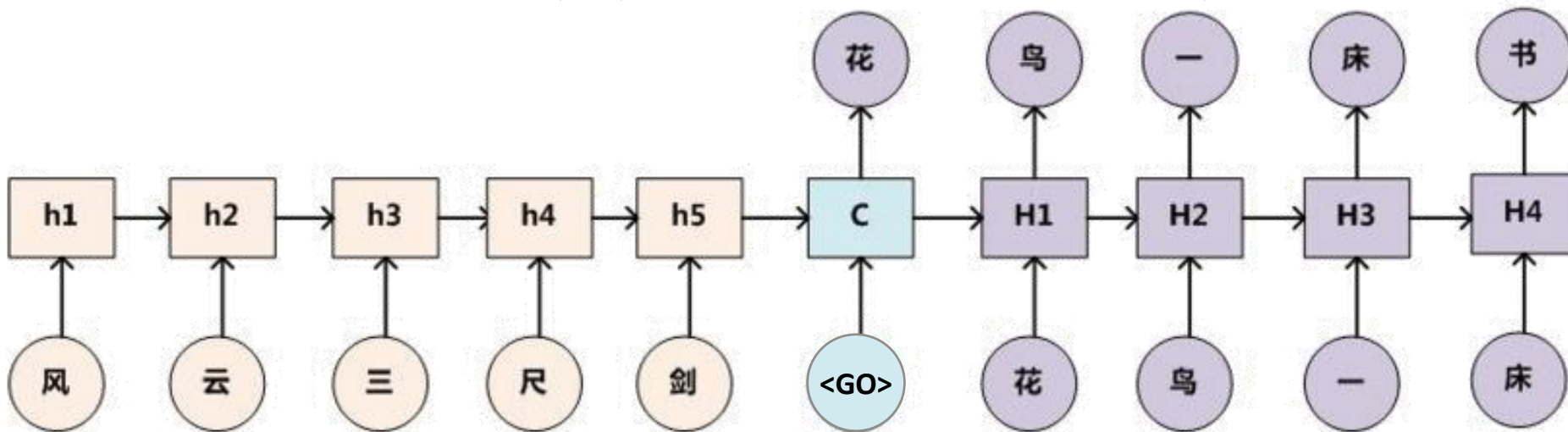


Seq2Seq



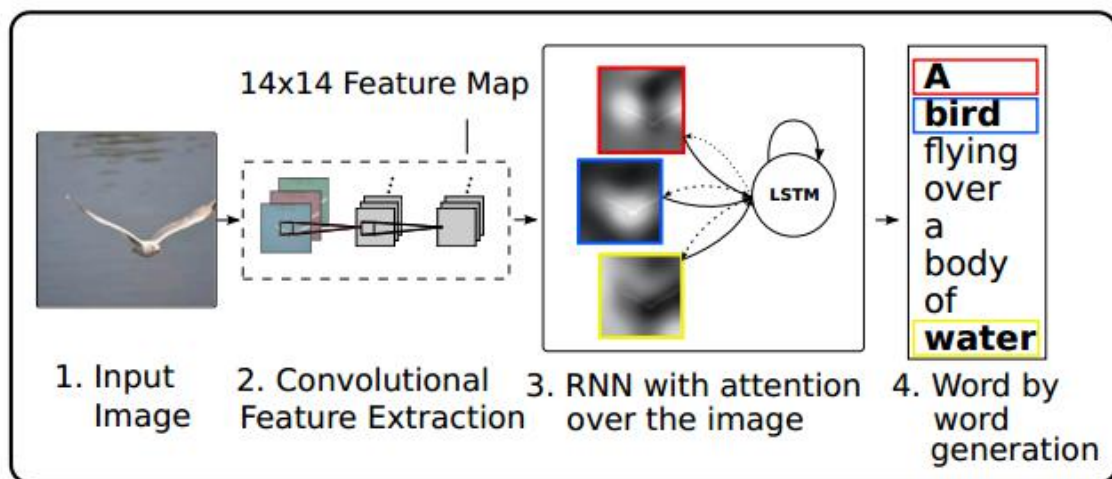
Seq2Seq问题

- 问题描述：“风”对应的特征对于下联的影响是最弱的。



1. 字句对等;
2. 词性对品;
3. 结构对应;
4. 节律对拍;
5. 平仄对立;
6. 形对意联;

Attention



Easy on the sensitive skin.

...soft, extra thick, gel-free protection ... baby's sensitive skin. The chlorine-free materials and ... is non-toxic and non-irritating. Clinically ... recommended for babies with allergies and sensitive skin.

<http://www.baby.com>

TM

If you are not satisfied with the baby leakage protection, you will get your money back. Read more about our leakfree guarantee at www.baby.com

Seq2Seq Attention

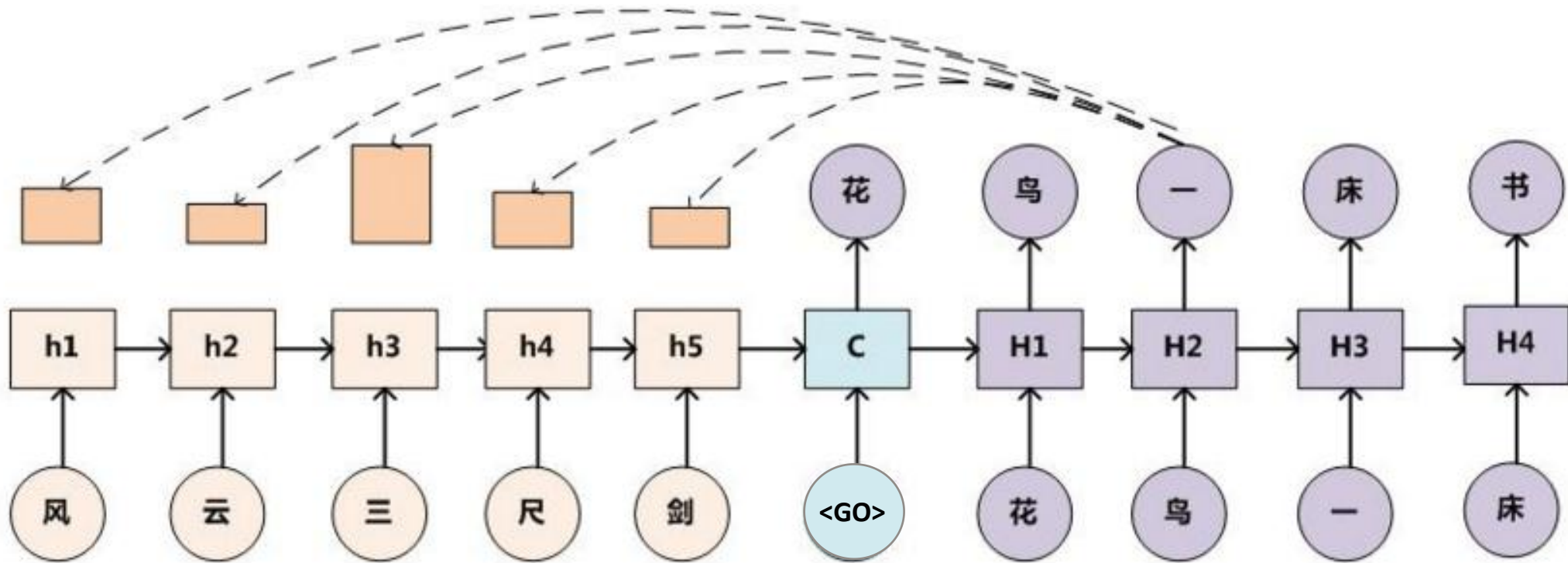


图3. Attention模型

Seq2Seq Attention

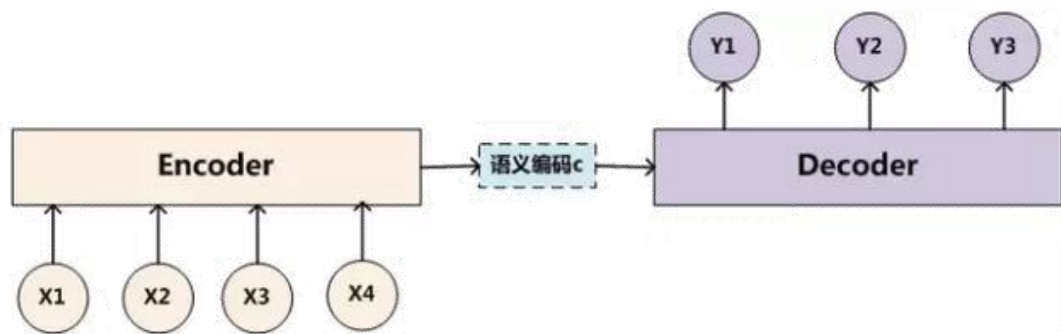


图2 抽象的文本处理领域的Encoder-Decoder框架

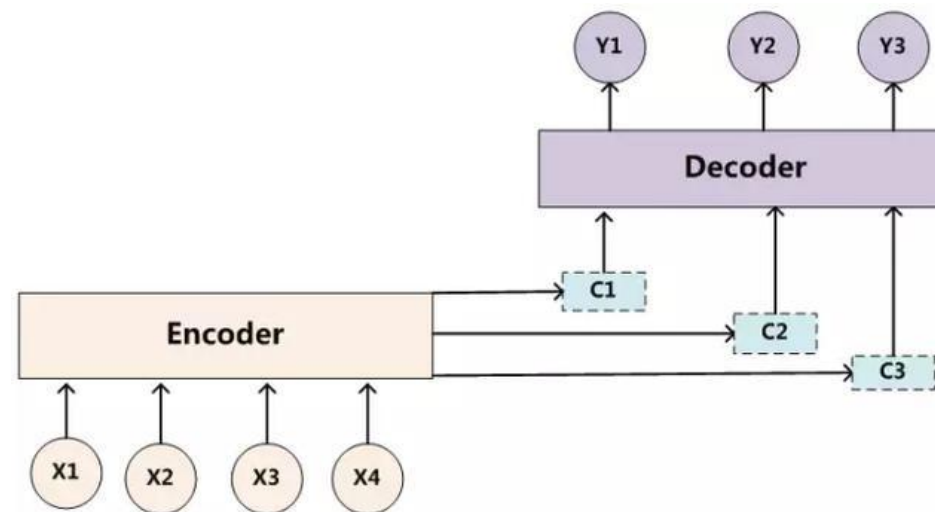
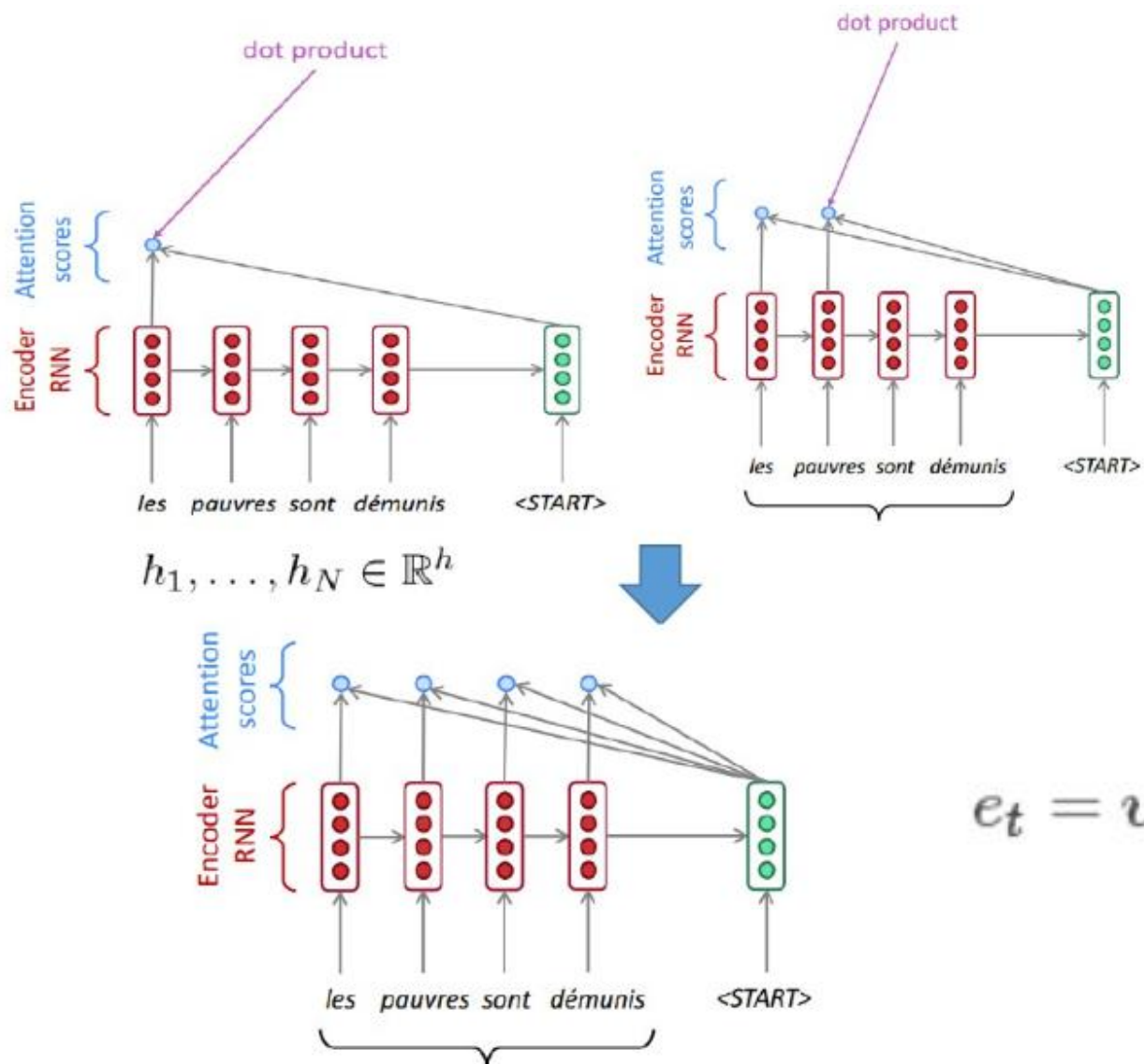


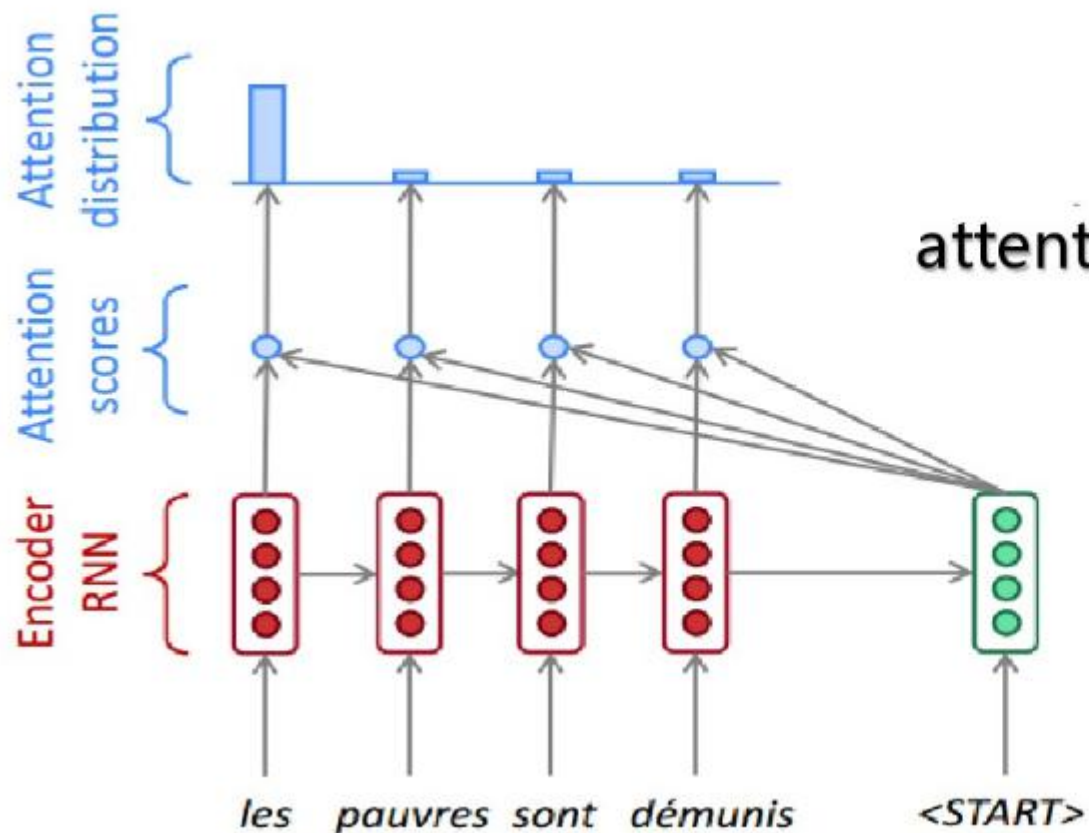
图3 引入注意力模型的Encoder-Decoder框架

Seq2Seq Attention计算过程



$$e_t = v_a^T \tanh(W_a s_{i-1} + U_a h_t)$$

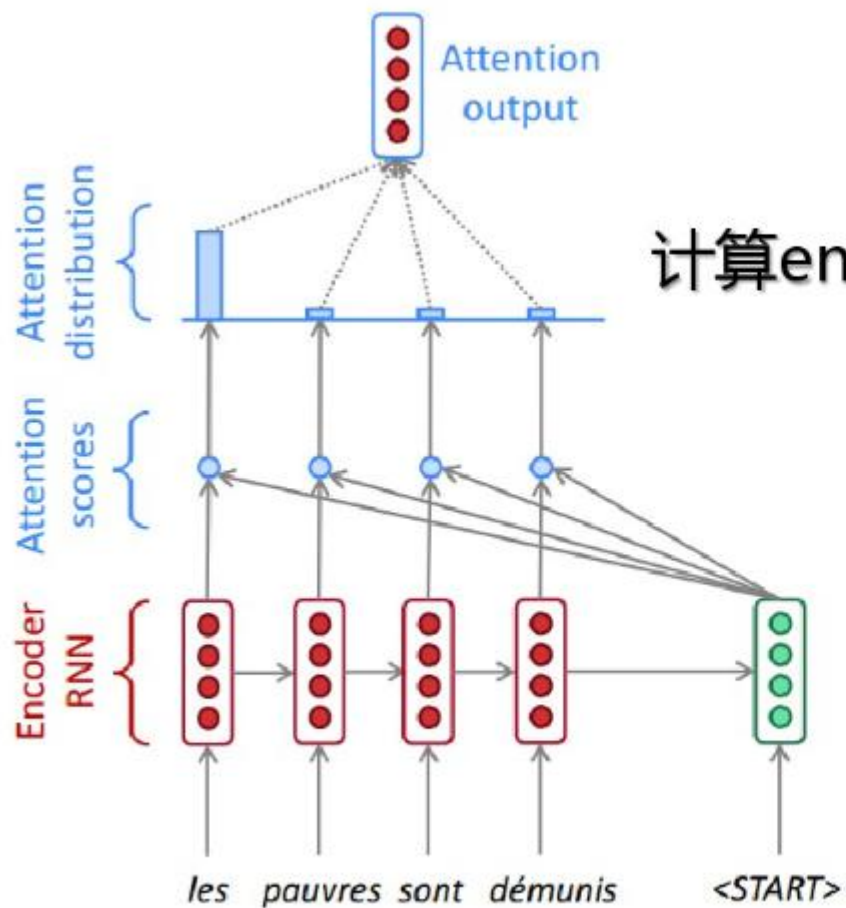
Seq2Seq Attention计算过程



利用softmax函数：
attention scores转化为概率分布

$$\alpha^t = \text{softmax}(\mathbf{e}^t) \in \mathbb{R}^N$$

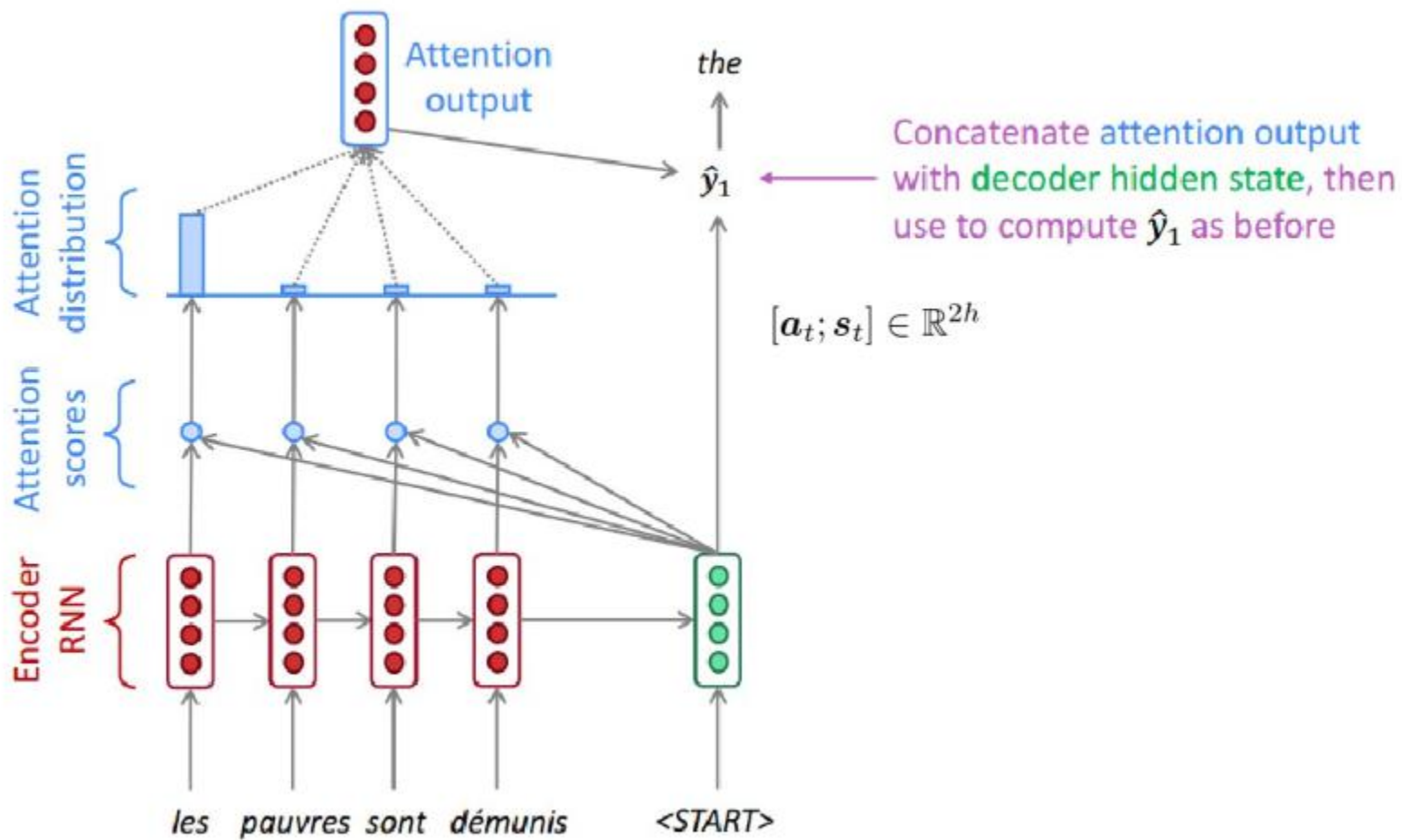
Seq2Seq Attention计算过程



按照上一步概率分布：
计算encoder的hidden states的加权求和

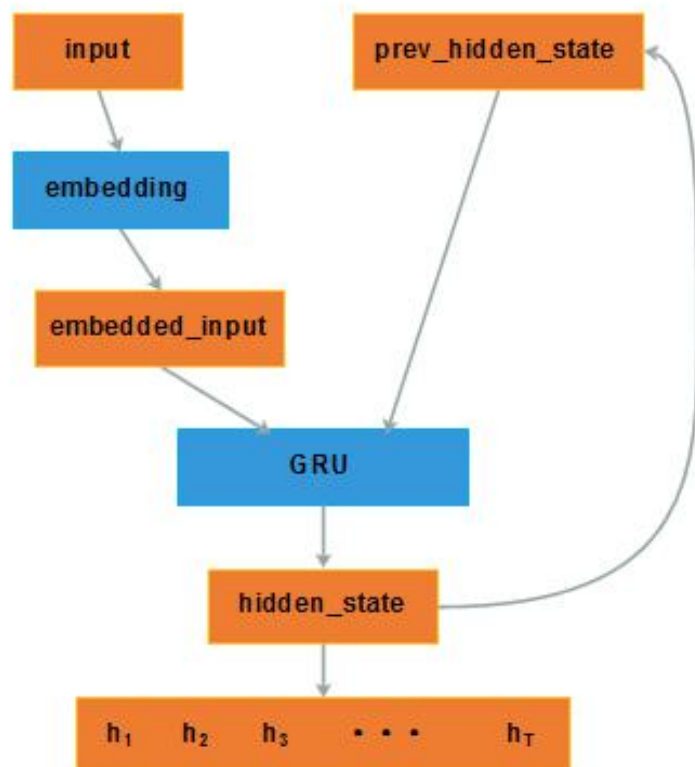
$$\mathbf{a}_t = \sum_{i=1}^N \alpha_i^t \mathbf{h}_i \in \mathbb{R}^h$$

Seq2Seq Attention计算过程



Seq2Seq Attention计算过程

Bi-RNN Encoder



Encoder的流程如上图所示，最终的输出结果是每个时刻的hidden_state $h_1, h_2, h_3, \dots, h_T$ 。

Seq2Seq Attention计算过程

其中的GRU使用的双向的，正向部分的公式如下

$$\vec{h}_0 = 0$$

$$\vec{z}_i = \sigma(\vec{W}_z \vec{E}x_i + \vec{U}_z \vec{h}_{i-1}) \quad (1)$$

$$\vec{r}_i = \sigma(\vec{W}_r \vec{E}x_i + \vec{U}_r \vec{h}_{i-1}) \quad (2)$$

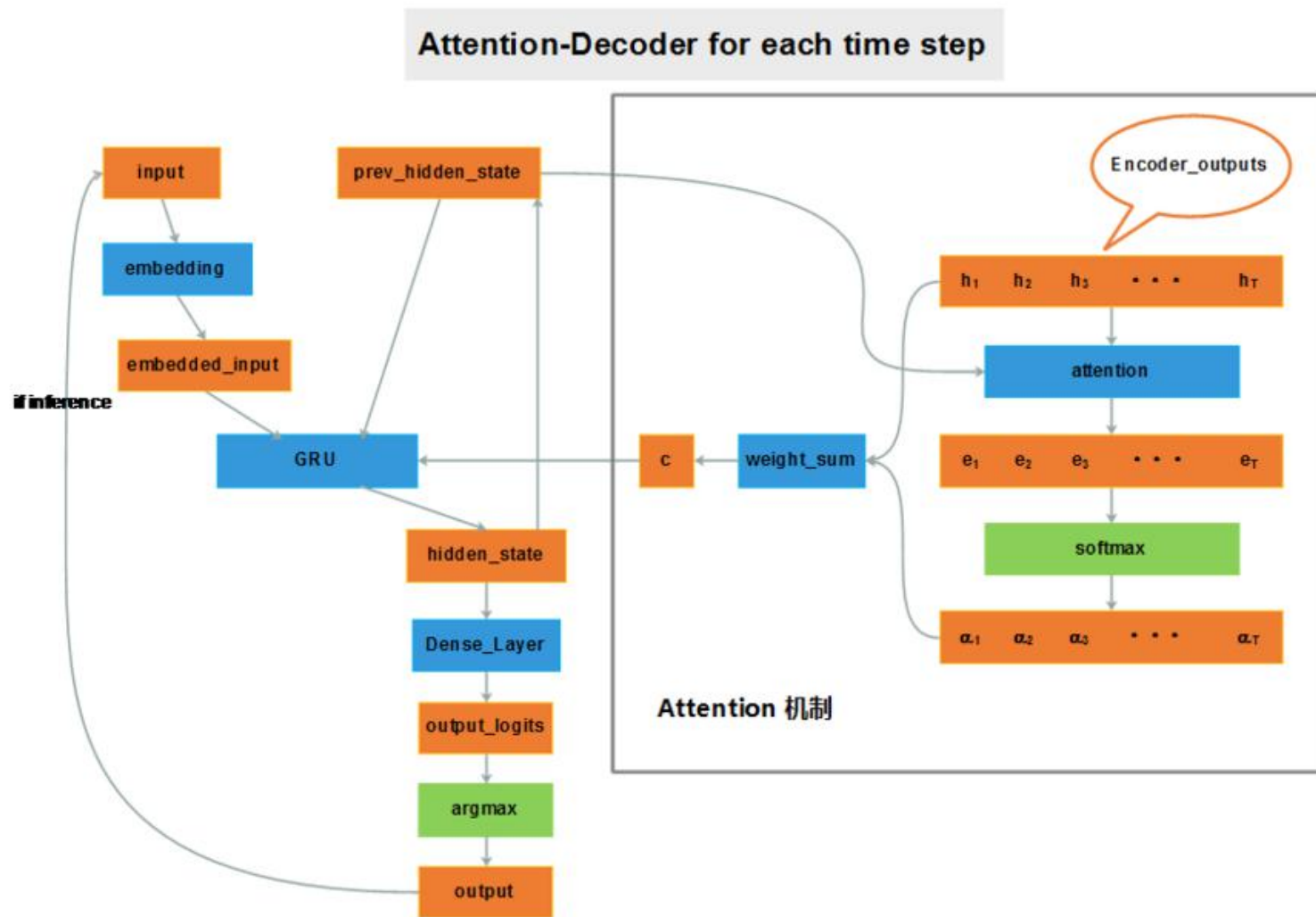
$$\hat{\vec{h}}_i = \tanh(\vec{W} \vec{E}x_i + \vec{U}[\vec{r}_i * \vec{h}_{i-1}]) \quad (3)$$

$$\vec{h}_i = (1 - \vec{z}_i) * \vec{h}_{i-1} + \vec{z}_i * \hat{\vec{h}}_i, i > 0 \quad (4)$$

反向的同上，最终的 h_t 为将正向与反向的结果concat得到的向量。

Seq2Seq Attention计算过程

Attention-Decoder



Seq2Seq Attention计算过程

整个decoding的过程可以拆分为以下几个部分

一、 离散的词ID转换为词向量

与Encoder 中的这个步骤是一样的，只不过embedding矩阵与Encoder的可能不一样，比如翻译源语言与目标语言需要使用不同的embedding矩阵，但是如文本摘要或是文本风格改写这种就可以使用同一个embedding矩阵。

二、 由encoder的输出结合decoder的prev_hidden_state生成energy

$$[(h_1, h_2, h_3, \dots, h_T), prev_hidden_state] \Rightarrow (e_1, e_2, e_3, \dots, e_T)$$

prev_hidden_state为 s_{i-1} ，由encoder所有时刻的输出 h_t 以及decoder的 **hidden_state** s_{i-1} 产生能量 e_1, e_2, \dots, e_T 的过程是Attention的关键步骤。能量 e_t 的含义也就是对应的源语言输入的词 x_t 对即将生成的目标语言的词 y_i 的影响力。既然我们需要一个对齐模型，根据此时每个输入词能量的大小，就可以知道应该使用哪个词与当前的 y_i 进行对齐。这样的对齐方式又称为soft alignment，也就是可以求得梯度，所以可以与整个模型一起优化。

$$e_t = v_a^T \tanh(W_a s_{i-1} + U_a h_t)$$

其中 $U_a h_t$ 的计算，与decoder的时刻无关，因此可以预先计算好，在每一个时刻将其代入即可。

Seq2Seq Attention计算过程

三、由energy 到概率

$$(e_1, e_2, e_3, \dots, e_T) \Rightarrow (\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_T)$$

使用attention的目的，是希望得到一个context向量，因此需要将 h_1, h_2, \dots, h_T 融合在一起，融合若干向量最容易想到的就是加权平均，但是如果直接把能量 e 作为权值，可能会将context向量缩放若干倍，所以需要将 e_t 转换为概率值 α_t ，使得它们的和为1，同时可以用来表示输入序列中每个词与当前待生成的词的匹配程度。即

$$\sum_{t=1}^T e_t = 1$$

使用softmax求得概率，公式如下

$$\alpha_t = \frac{\exp(e_t)}{\sum_{t=1}^T \exp(e_t)}$$

四、context 向量合成

$$[(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_T), (h_1, h_2, h_3, \dots, h_T)] \Rightarrow c$$

得到对输入序列每一时刻的权值 α_t 后，将其与encoder各时刻的输出 h_t 加权求和，即得到decoder在当前时刻的context向量 c_i ，公式如下

$$c_i = \sum_{t_1}^T \alpha_t h_t$$

Seq2Seq Attention计算过程

五、prev_hidden_state, 词向量, context向量通过GRU单元生成下一时刻 hidden_state

$$(embedded_input, prev_hidden_state, c) \Rightarrow hidden_state$$

decoder本质是一个GRU单元，与GRU不同的在于融合了由attention机制产生的context向量。在decoding的场景，只能用单向的RNN，因为后续时刻的结果在当前时刻是未知的。

将 decoder 在时刻 i 的 **hidden_state** 表示为 s_i ，**prev_hidden_state** 为 s_{i-1} ，context 向量 **c** 表示为 c_i ，**embedded_input**表示为 Ey_{i-1} ，则可由如下公式表示该过程：

$$s_i = (1 - z_i) * s_{i-1} + z_i * \tilde{s}_i$$

其中

$$\tilde{s}_i = \tanh(W Ey_{i-1} + U[r_i * s_{i-1}] + Cc_i)$$

$$z_i = \sigma(W_z Ey_{i-1} + U_z s_{i-1} + C_z c_i)$$

$$r_i = \sigma(W_r Ey_{i-1} + U_r s_{i-1} + C_r c_i)$$

Seq2Seq Attention计算过程

六、使用全连接层将hidden_state映射为vocabulary size的向量

$(hidden_state) \Rightarrow output_logist$

生成的hidden_state已经包含了待生成的词的信息了，但是要生成具体的词，我们还需要知道目标语言中每个词的条件概率 $p(y_i | s_i)$ ，如果 s_i 的维度就是目标语言的词典大小，那么使用softmax就可以算出每个词的概率，但是 s_i 的维度也属于模型的一个参数，通常是不会等于目标语言词典的大小的，因此再增加一个全连接层，将 s_i 映射为维度等于词典大小 L 的向量 v_i ，每个维度代表一个词，使用softmax计算出每个词的概率。

$$v_i = W s_i + b$$

$$prob_{ij} = \frac{\exp(v_{ij})}{\sum_{j=1}^L v_{ij}}$$

$$y_i = \arg \max prob_i$$

Seq2Seq Attention计算过程

- 此时给定Target中的某个元素Query，通过计算Query和各个Key的相似性或者相关性，得到每个Key对应Value的权重系数，然后对Value进行加权求和，即得到了最终的Attention数值。所以本质上Attention机制是对Source中元素的Value值进行加权求和，而Query和Key用来计算对应Value的权重系数。

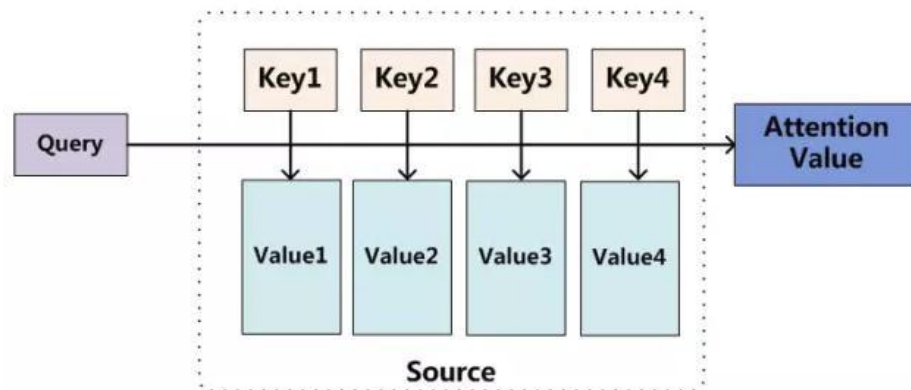
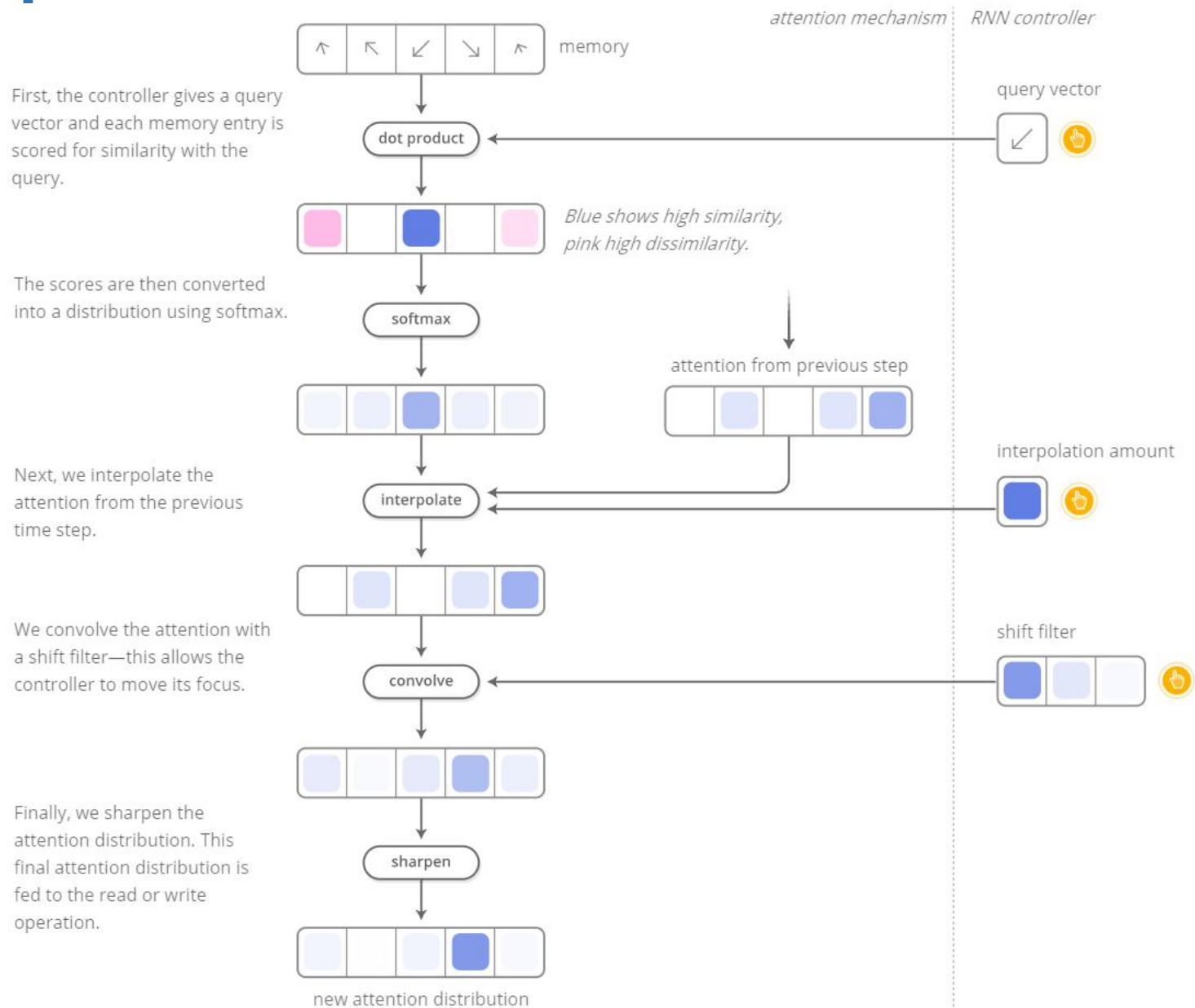
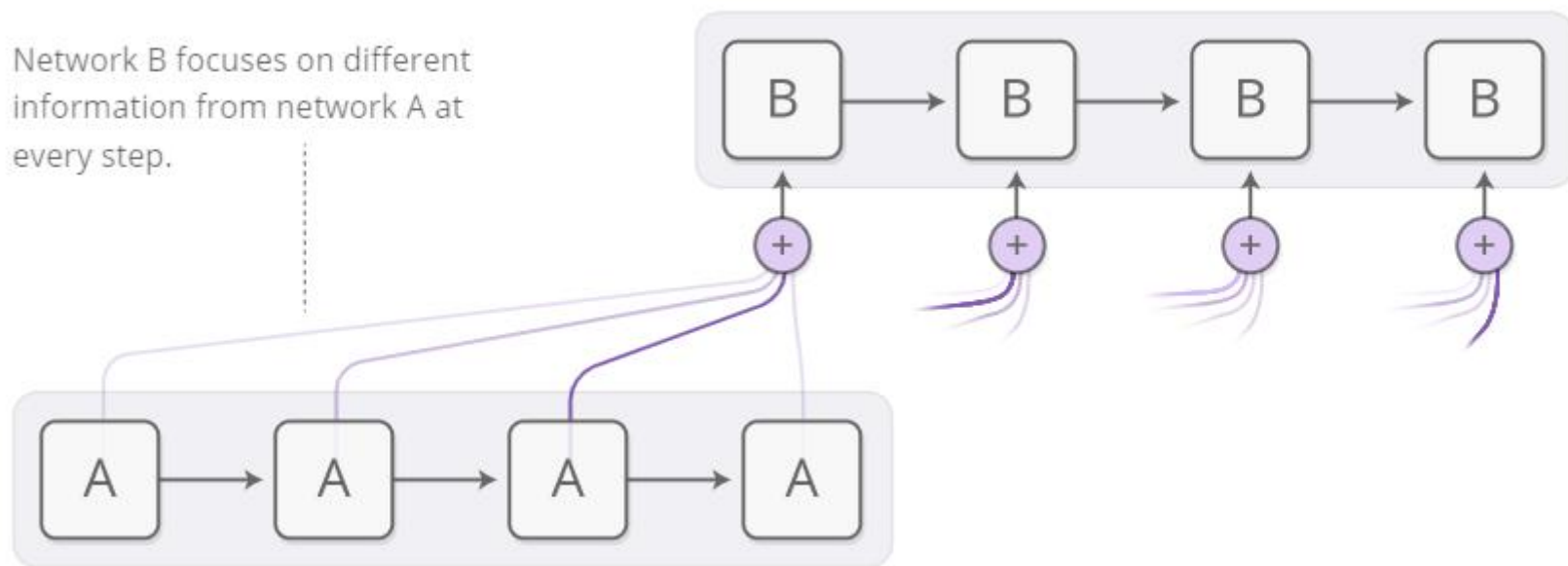


图9 Attention机制的本质思想

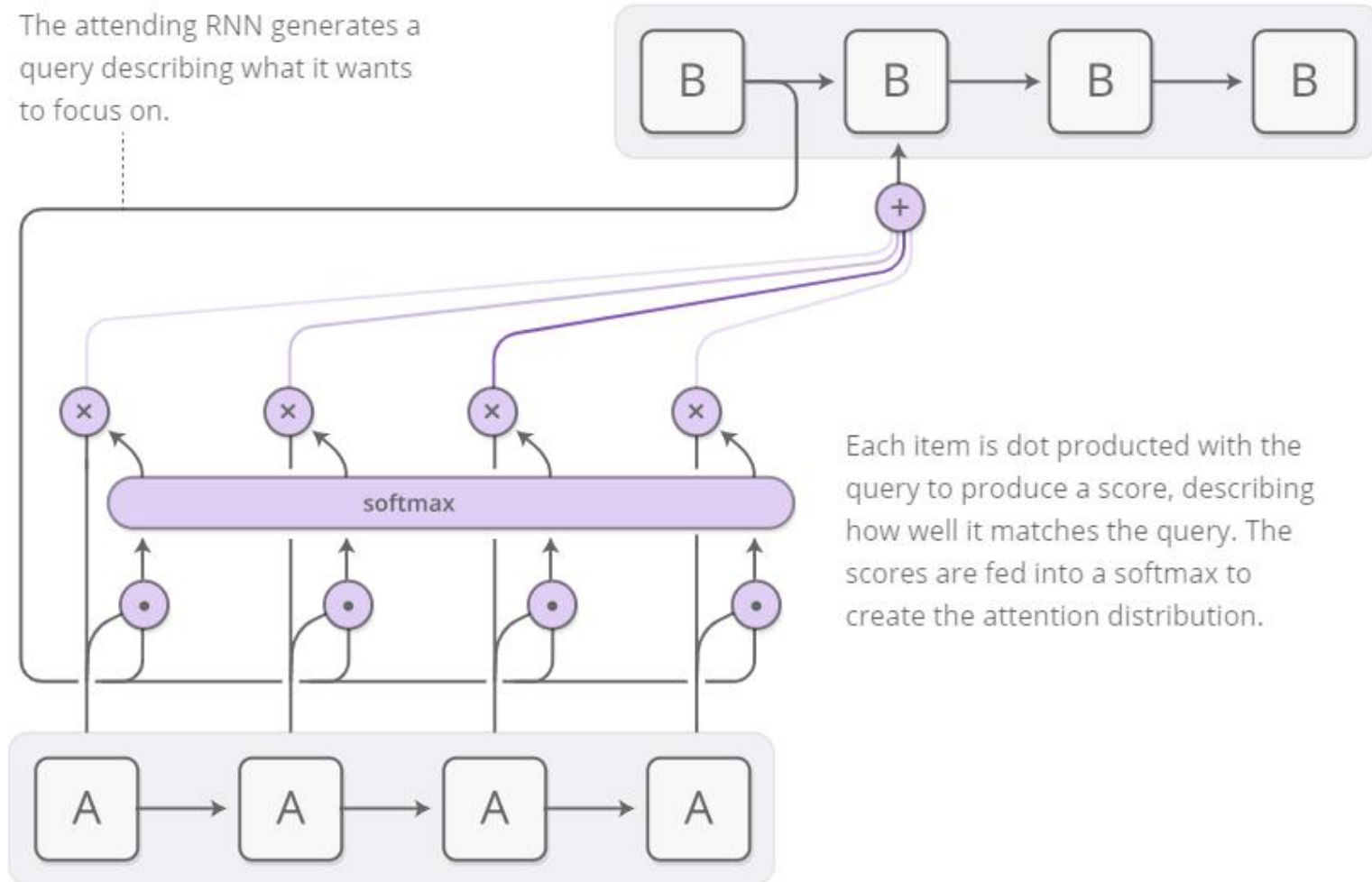
Seq2Seq Attention 计算过程



Seq2Seq Attention 计算过程

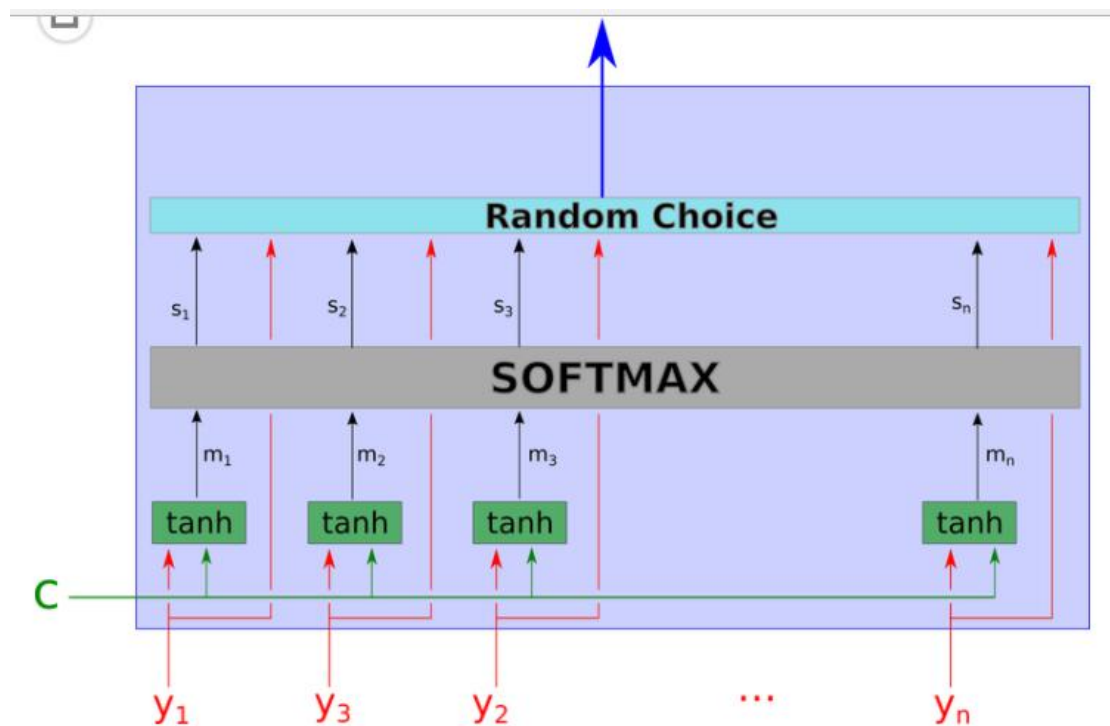
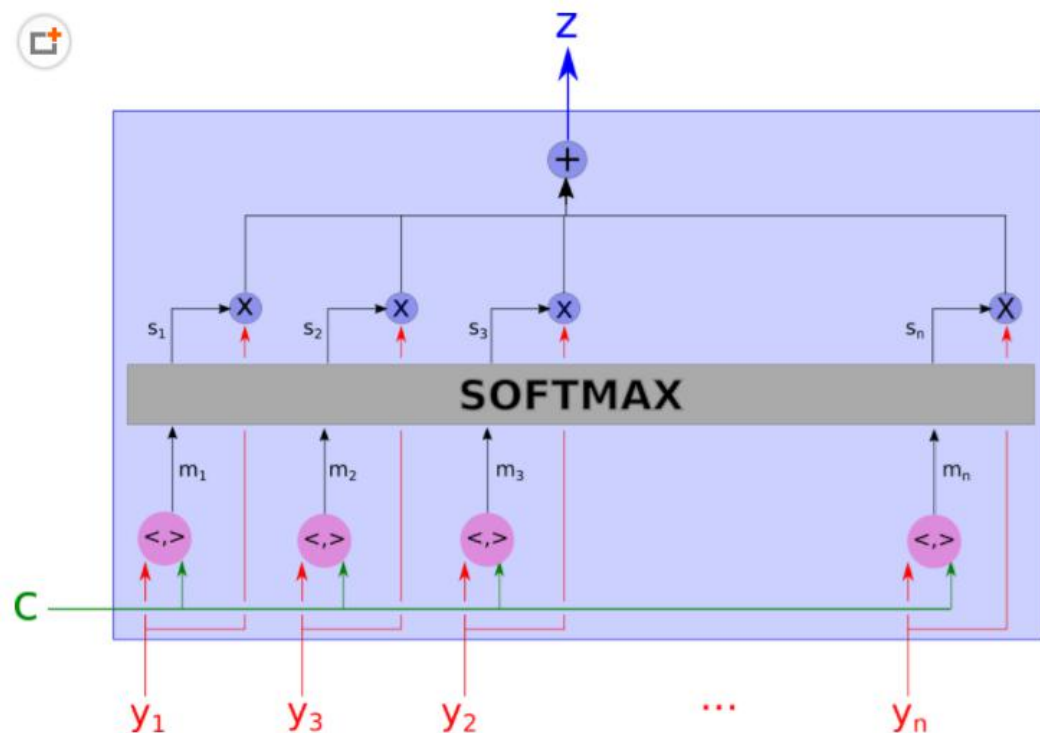


Seq2Seq Attention 计算过程



Soft Attention和Hard Attention区别

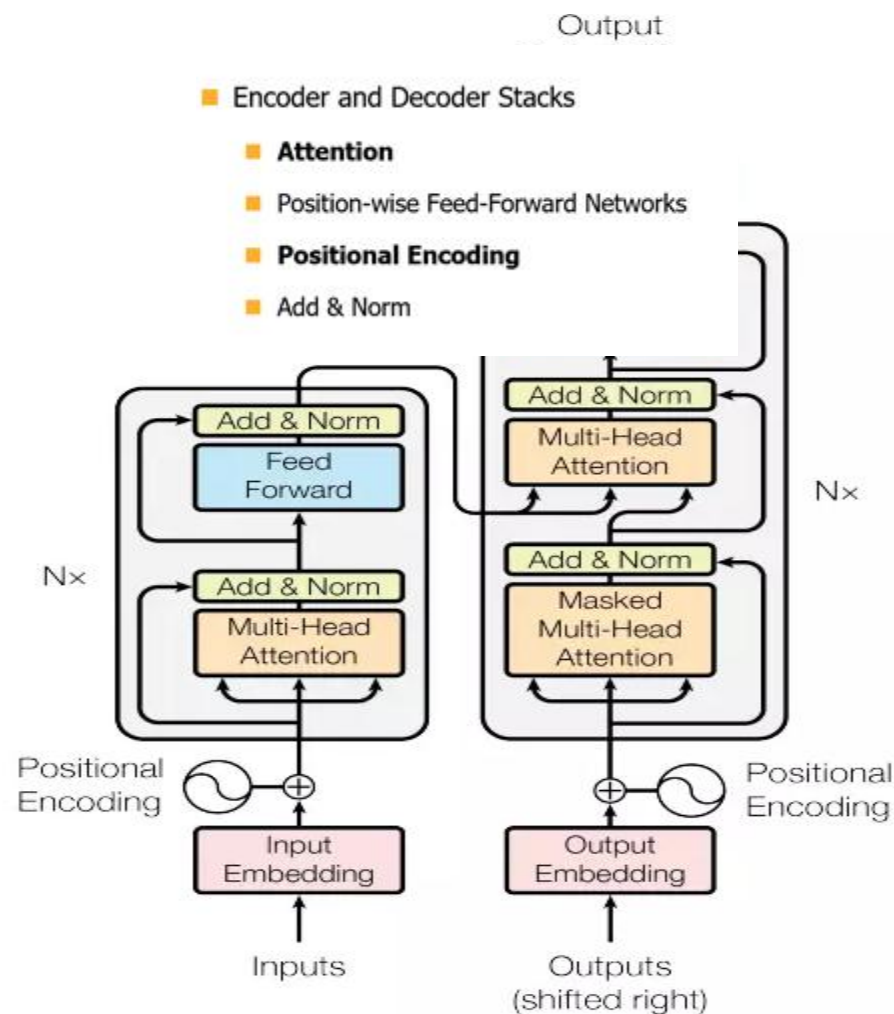
Soft Attention通常是指以上我们描述的这种全连接(如MLP计算Attention 权重), 对每一层都可以计算梯度和后向传播的模型; 不同于Soft attention那样每一步都对输入序列的所有隐藏层 $h_j(j=1\dots T_x)$ 计算权重再加权平均的方法, Hard Attention是一种随机过程, 每次以一定概率抽样, 以一定概率选择某一个隐藏层 h_j^* , 在估计梯度时也采用蒙特卡罗抽样Monte Carlo sampling的方法。左图为Soft Attention 模型, 右图为Hard Attention 模型:



注意力机制

多头注意力（Multi-headed attention）机制

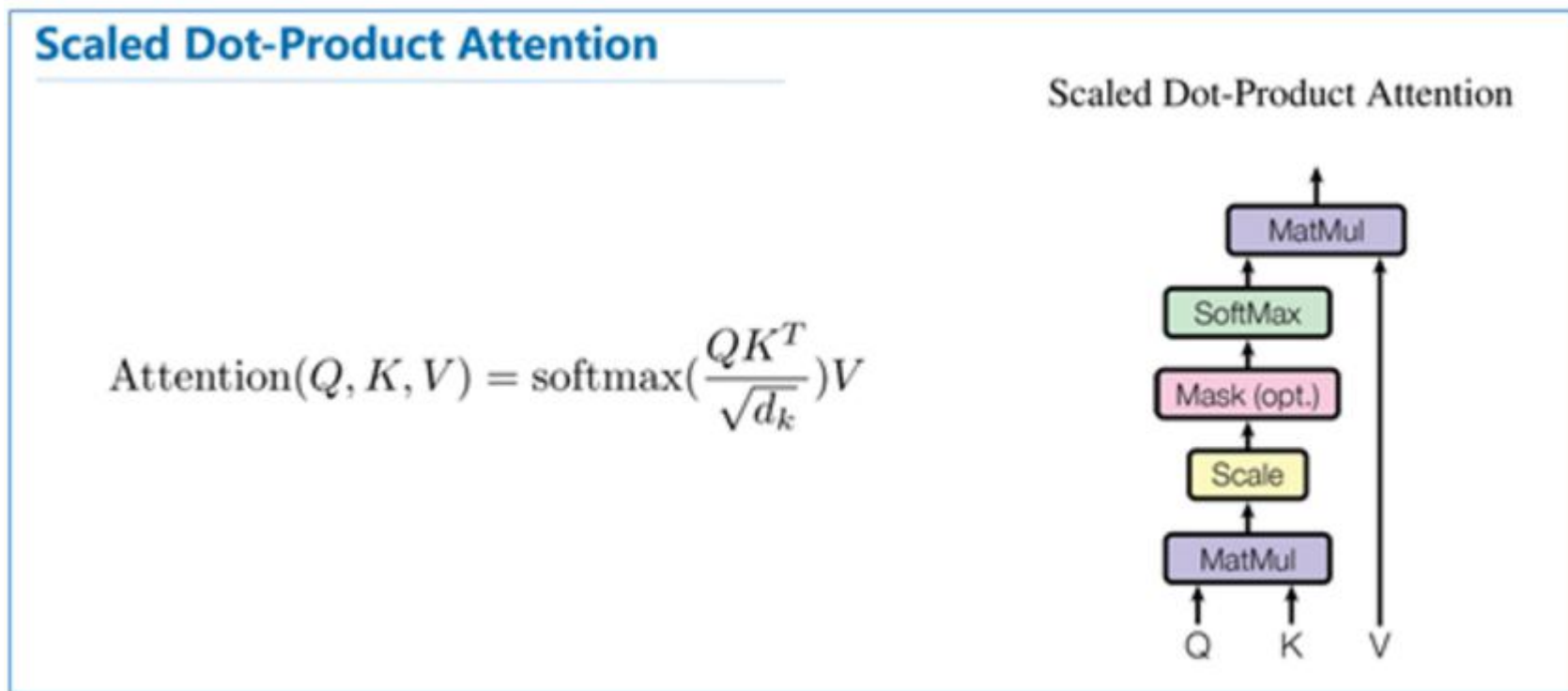
1、由编码器和解码器组成，在编码器的一个网络块中，由一个多头attention子层和一个前馈神经网络子层组成，整个编码器栈式搭建了N个块。类似于编码器，只是解码器的一个网络块中多了一个多头attention层。为了更好的优化深度网络整个网络使用了残差连接和对层进行了规范化（Add&Norm）。



注意力机制

多头注意力（Multi-headed attention）机制

2、放缩点积attention（scaled dot-Product attention）。对比我在前面背景知识里提到的attention的一般形式，其实scaled dot-Product attention就是我们常用的使用点积进行相似度计算的attention，只是多除了一个（为 K 的维度）起到调节作用，使得内积不至于太大。



注意力机制

3、多头attention的Query, Key, Value首先进过一个线性变换, 然后输入到放缩点积attention, 注意这里要做h次, 其实也就是所谓的多头, 每一次算一个头。而且每次Q, K, V进行线性变换的参数W是不一样的。然后将h次的放缩点积attention结果进行拼接, 再进行一次线性变换得到的值作为多头attention的结果。

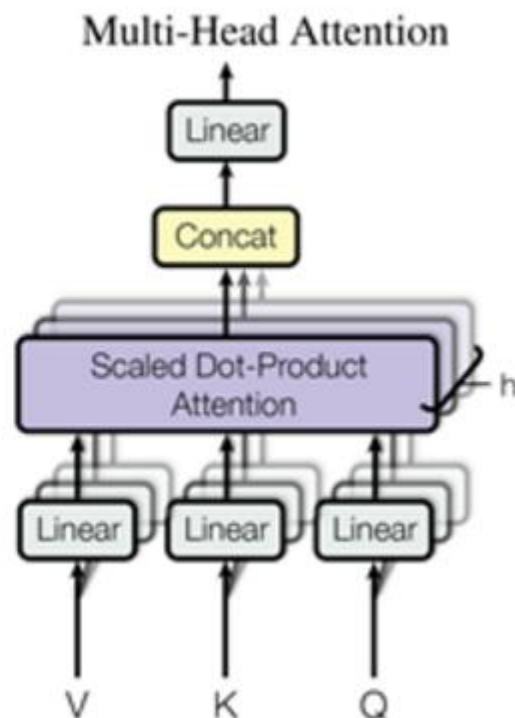
<https://zhuanlan.zhihu.com/p/31547842>

Multi-Head Attention

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

- Multi-head attention allows the model to jointly attend to **information from different representation subspaces** at different positions.





Seq2Seq Attention



Seq2Seq Attention



Seq2Seq Attention

