

法律声明

□ 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

□ 课程详情请咨询

■ 微信公众号：小象

■ 新浪微博：ChinaHadoop



神经序列模型 VI

主讲人： 史兴

07/28/2017

提纲

□ Beam Search 实现细节(续)

□ Seq2Seq 模型可视化

- Hidden States 可视化
- Word Embedding 可视化
- Attention 可视化

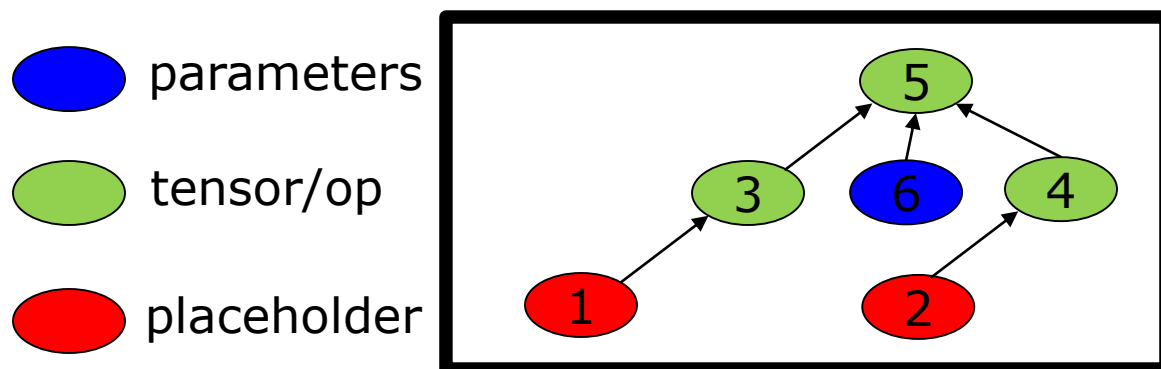
Beam Search 实现细节(续)

□ Tensorflow 一般的工作流程

■ 1) “画图”：构造计算图(computation graph)

□ 入口：placeholder

□ 出口：任何一个可以计算的操作(operation)或者中间变量(tensor)



Beam Search 实现细节(续)

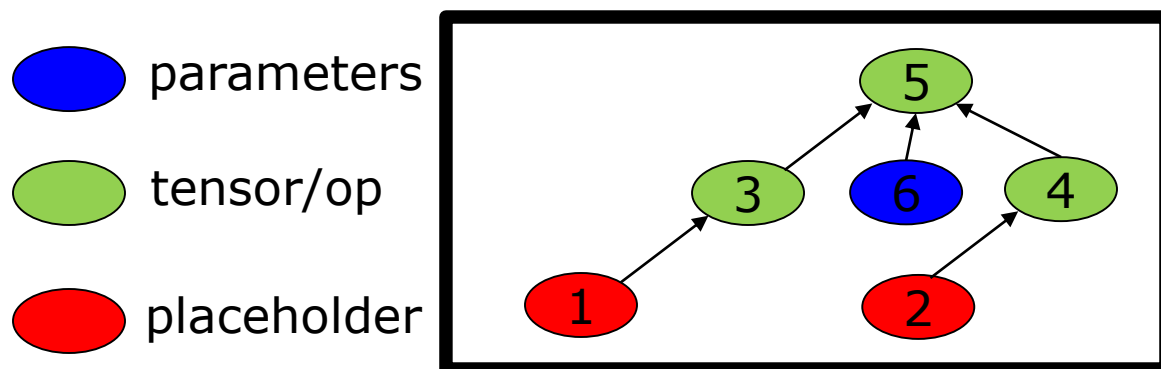
□ Tensorflow 一般的工作流程

■ 1) “画图”：构造计算图(computation graph)

■ 2) 对图进行“编译”：必要的优化

□ 构建图和编译图都是比较耗时的

□ 一旦图画好，后续的操作最好不要对图进行更改



Beam Search 实现细节(续)

□ Tensorflow 一般的工作流程

■ 3) 进行图计算 `session.run(output_feed, input_feed)`

□ “图”和“外界”交流的唯一途径

□ 仅仅进行必要的计算

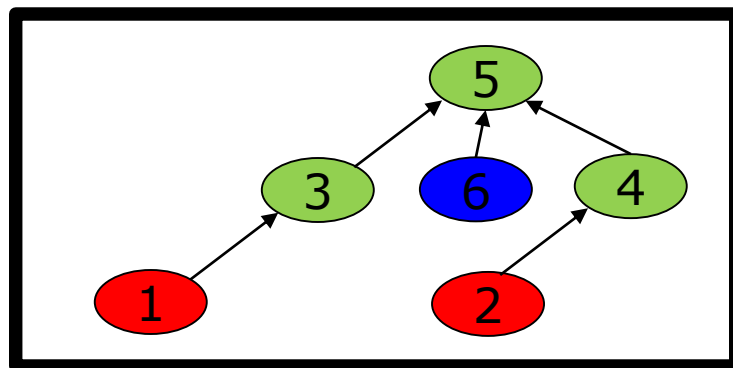
■ `output_feed = [5], input_feed = ?`

■ `output_feed = [3], input_feed = ?`

● parameters

● tensor/op

● placeholder



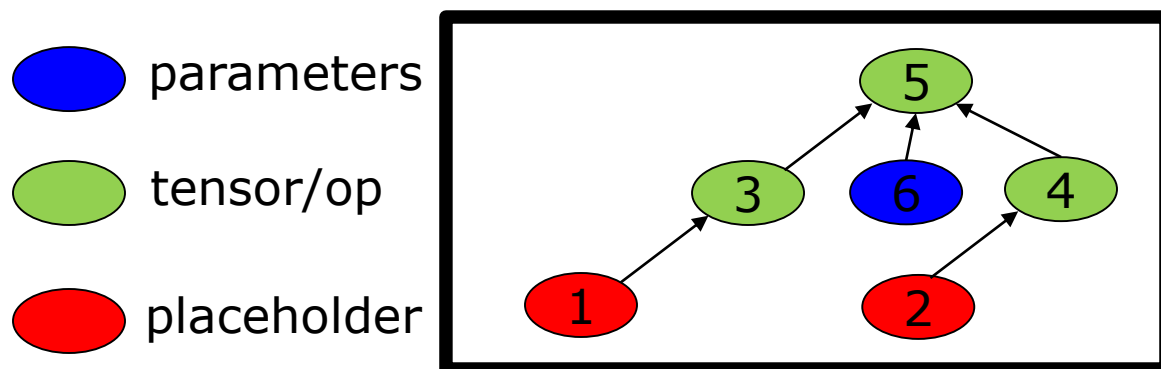
Beam Search 实现细节(续)

□ Tensorflow 一般的工作流程

■ 3) 进行图计算 `session.run(output_feed, input_feed)`

□ 首先调用: `session.run([3], [1])`

□ 然后再要计算5, `output_feed = [5], input_feed = ?`



Beam Search 实现细节(续)

□ Tensorflow 一般的工作流程

■ 3) 进行图计算 session.run(output_feed, input_feed)

□ 首先调用: session.run([3], [1])

□ 然后再要计算5, output_feed = [5], input_feed = [1,2]

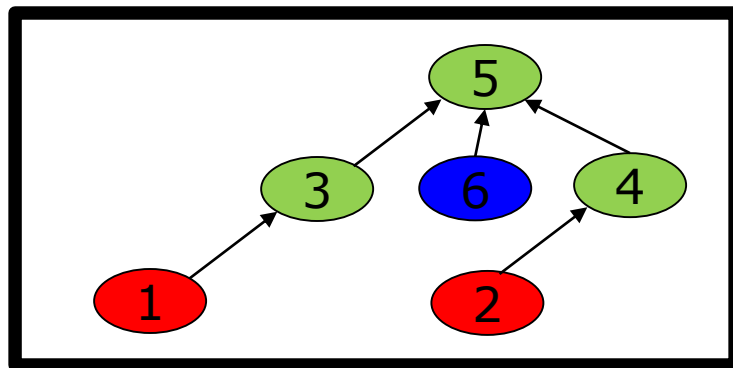
■ 每次调用session.run, **中间变量**的都需要重新计算, 但是参数parameter的值不会重新计算

■ 如何改变节省计算量?

blue oval parameters

green oval tensor/op

red oval placeholder



Beam Search 实现细节(续)

□ Tensorflow 一般的工作流程

■ 方案一:

□ 将计算结果从“图”中取出来, 保存在“外部”

■ `temp = session.run([3],[1])`

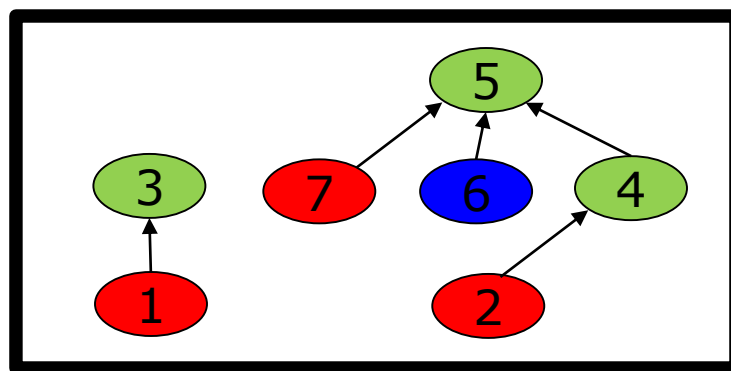
□ 将外部结果再重新输入“图中”

■ `session.run([5],[7=temp,2])`

● parameters

● tensor/op

● placeholder



Beam Search 实现细节(续)

□ Tensorflow 一般的工作流程

■ 方案一:

□ 如果3是一个比较大的矩阵?

■ 数据来回复制耗费时间

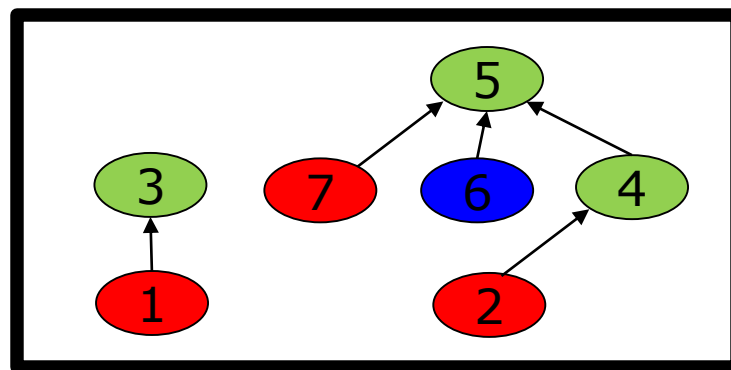
□ 如果“图”在GPU上面呢?

■ 更加耗费时间!

● parameters

● tensor/op

● placeholder



Beam Search 实现细节(续)

□ Tensorflow 一般的工作流程

■ 方案二：将temp作为一个参数放在“图”中

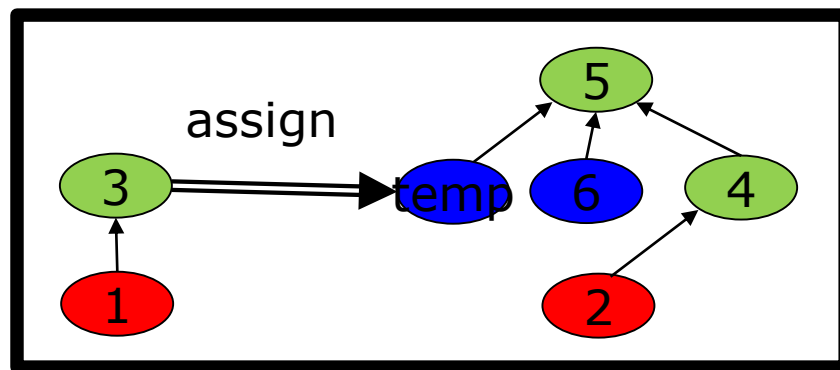
画图：1 = tf.placeholder(); 3 = f(1);
temp = tf.get_variable(); 6=tf.get_variable()
2= tf.placeholder(); 4=f(2); 5=h(temp,6,4)
copy = temp.assign(3)
调用： session.run([3,copy],[1])
session.run([5],[2])

⇒ operation

● parameters

● tensor/op

● placeholder



Beam Search 实现细节(续)

□ Tensorflow 一般的工作流程

■ 方案二：将temp作为一个参数放在“图”中

□ 可不可以把temp变成中间变量？

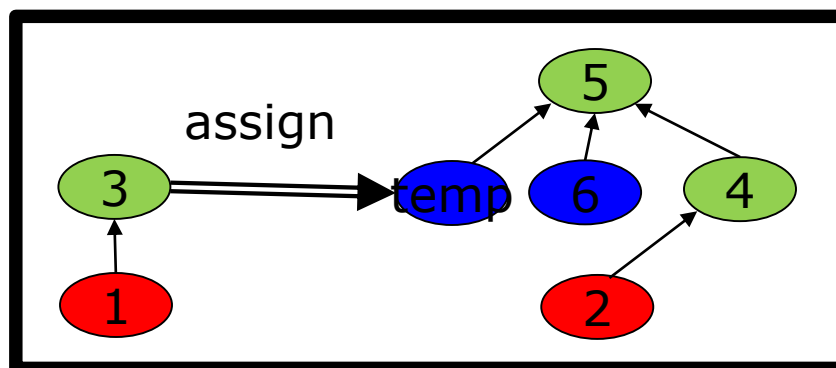
画图：1 = tf.placeholder; 3 = f(1);
temp = ?

==> operation

● parameters

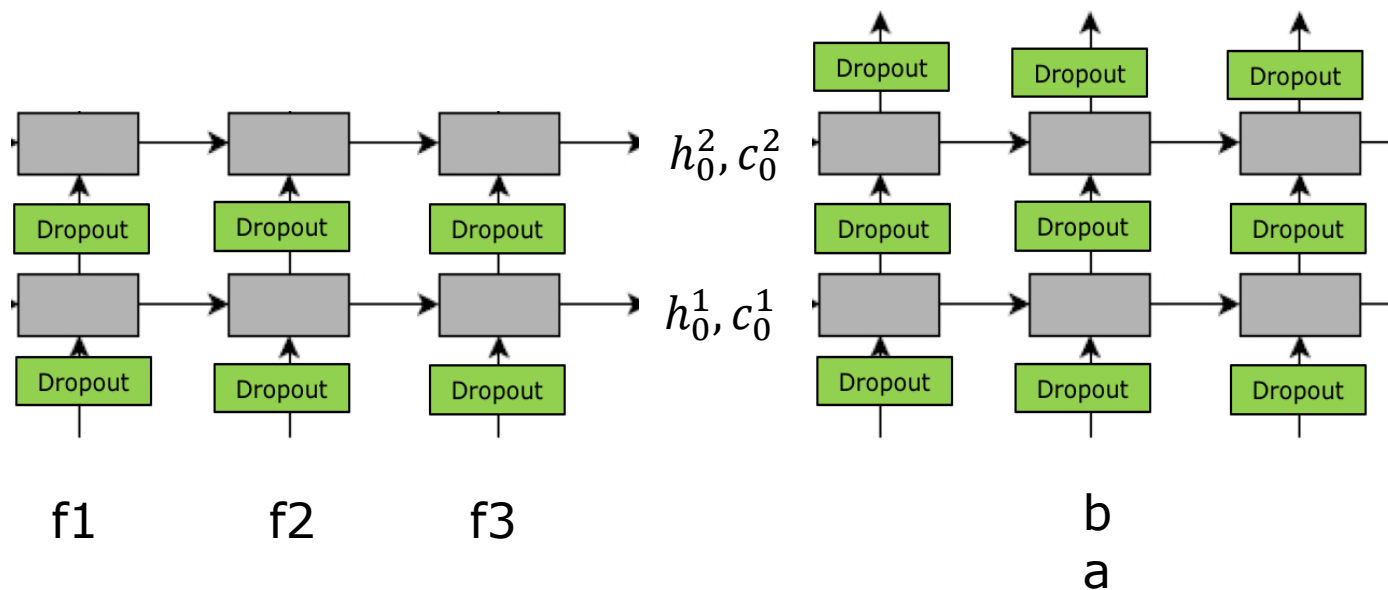
● tensor/op

● placeholder



Beam Search 实现细节(续)

output_feed = [a:0.5 b:0.5
a:0.8 b:0.2]



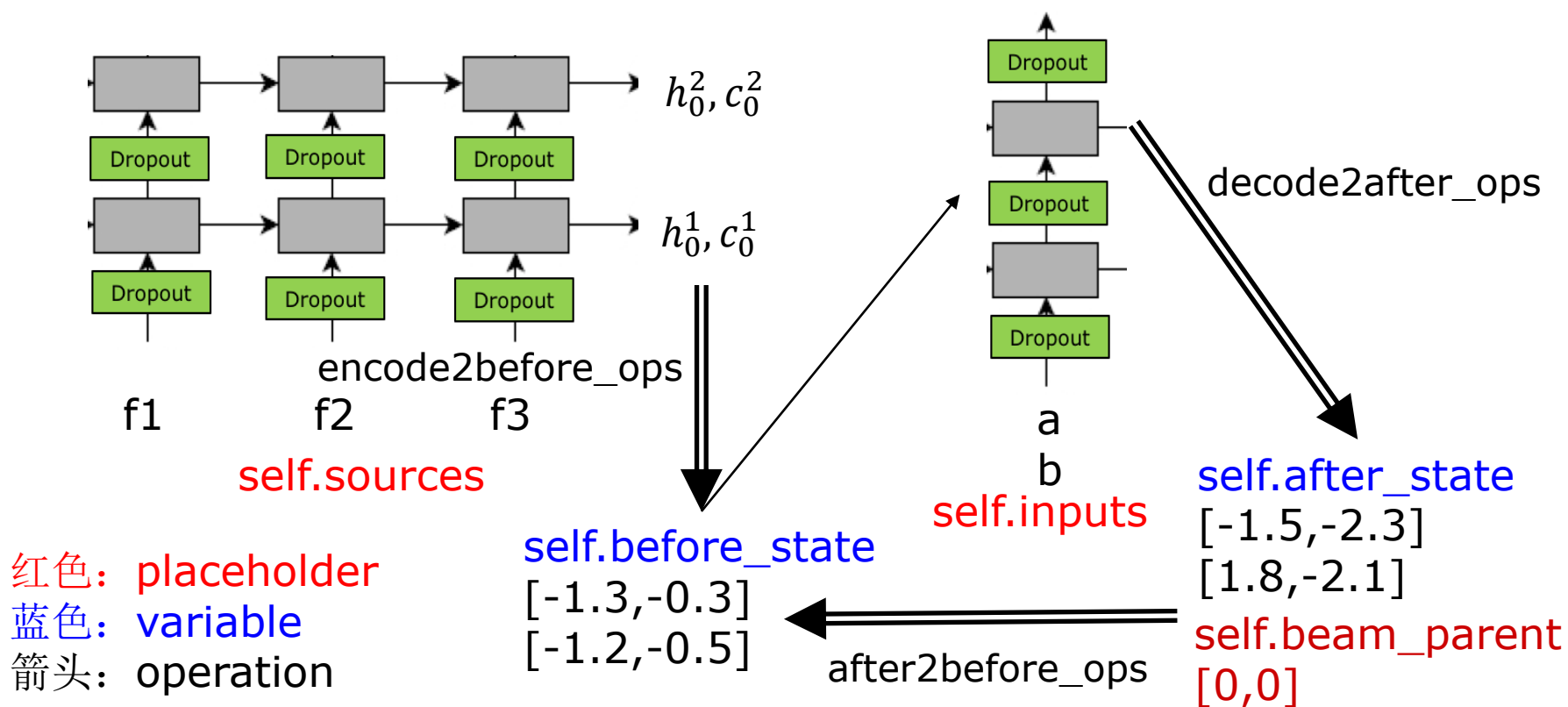
input_feed = [[f1,f2,f3],[_GO,_GO],[b,a]] ?

Beam Search 代码实现

Single-step Decoder

self.top_index
self.top_value
self.eos_value

a:0.3 b:0.7
a:0.5 b:0.5



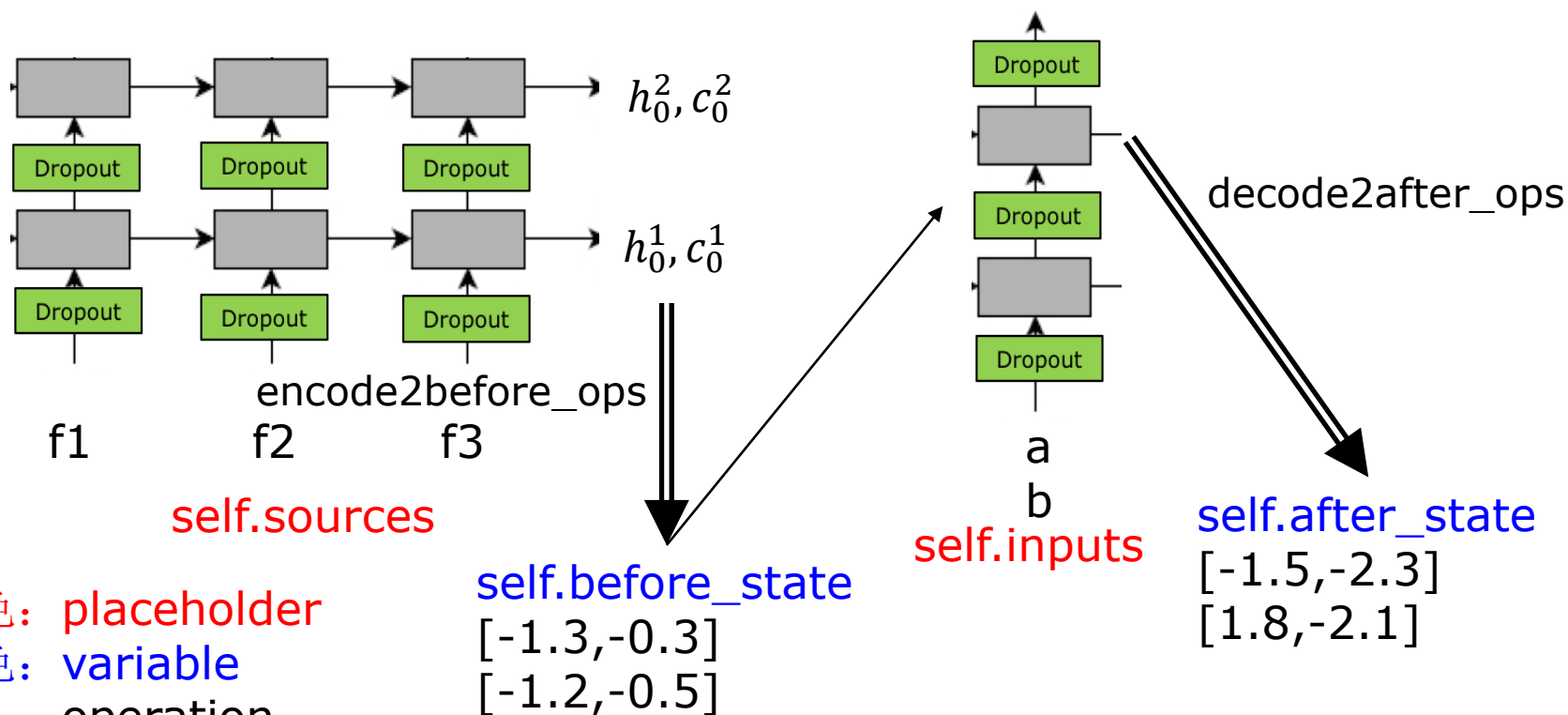
Beam Search 代码实现

beam_step(index=0)

```
session.run([encoder2before_ops],[[f1,f2,f3]])
```

```
session.run([[self.top_index,...],decode2after_ops],[self.inputs])
```

a:0.3 b:0.7
a:0.5 b:0.5
self.top_index
self.top_value
self.eos_value



红色: placeholder
蓝色: variable
箭头: operation

Beam Search 代码实现

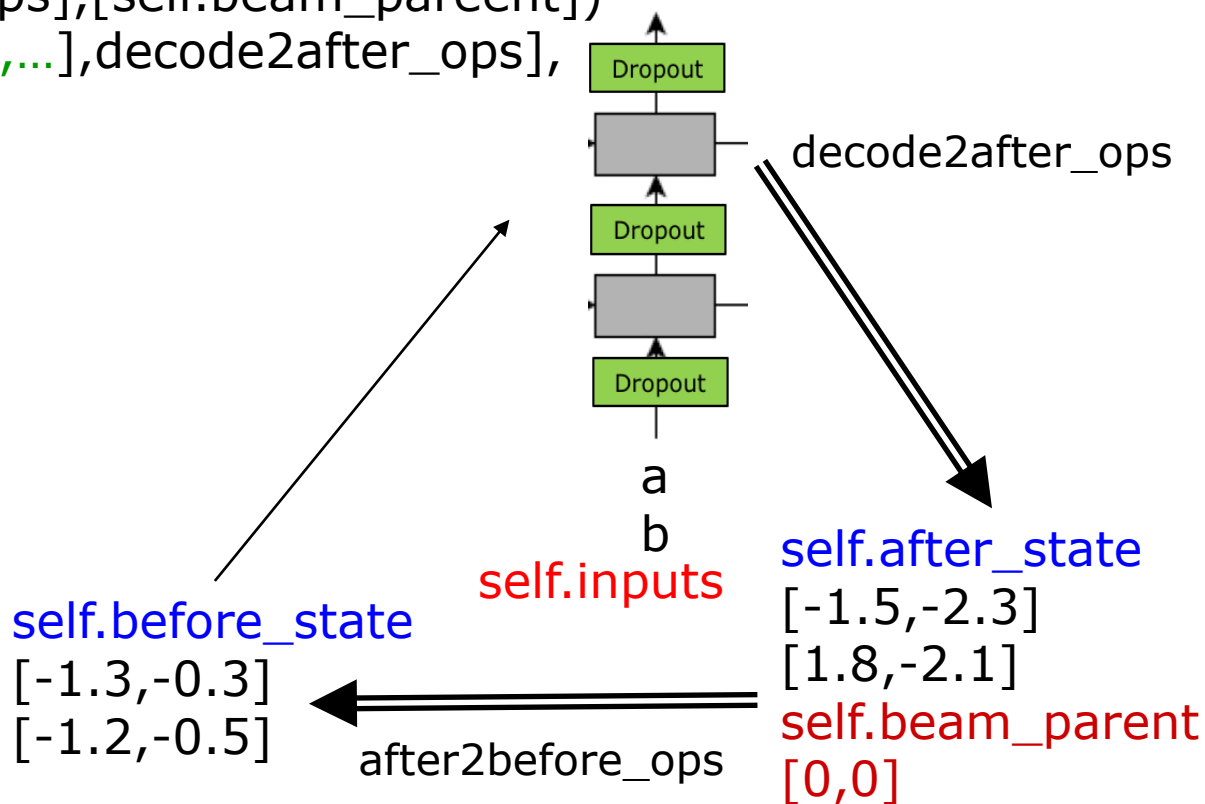
beam_step(index>0)

a:0.3 b:0.7
a:0.5 b:0.5

self.top_index
self.top_value
self.eos_value

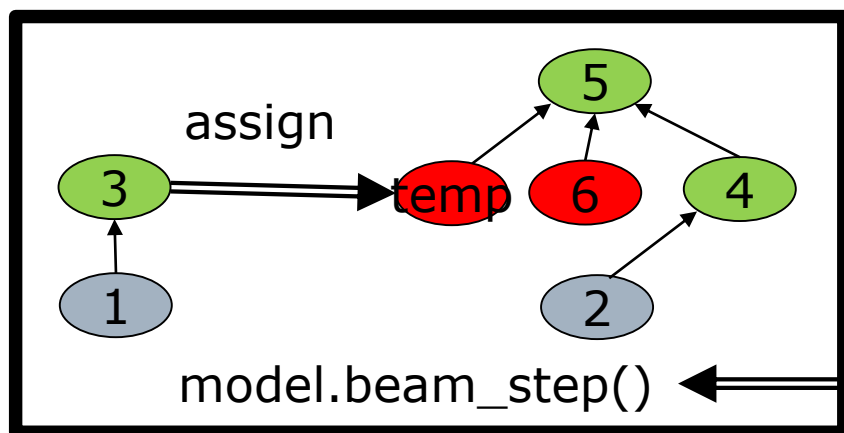
```
session.run([after2before_ops],[self.beam_parent])  
session.run([[self.top_index,...],decode2after_ops],  
            [self.inputs])
```

红色: placeholder
蓝色: variable
箭头: operation



Beam Search 实现细节(续)

SeqModel.py



run.py

```
def beam_decode():
```

source_inputs
target_inputs
beam_parent

top_value
top_index
eos_value

Beam Search 实现细节(续)

run.py:beam_decode()

变量:

source_inputs = [f_1, f_2, f_3]

	scores	sentences	beam parent	target inputs	top index	top value	eos value
0							
1							

results = [] 存放“熟了”的句子 ($w_1, \dots, w_i, \dots, w_n = \text{EOS}$)

sentences: 存放“半熟”的句子: 部分生成的句子(w_1, w_2, \dots, w_{i-1})

scores: $\log P(w_1, w_2, \dots, w_{i-1})$

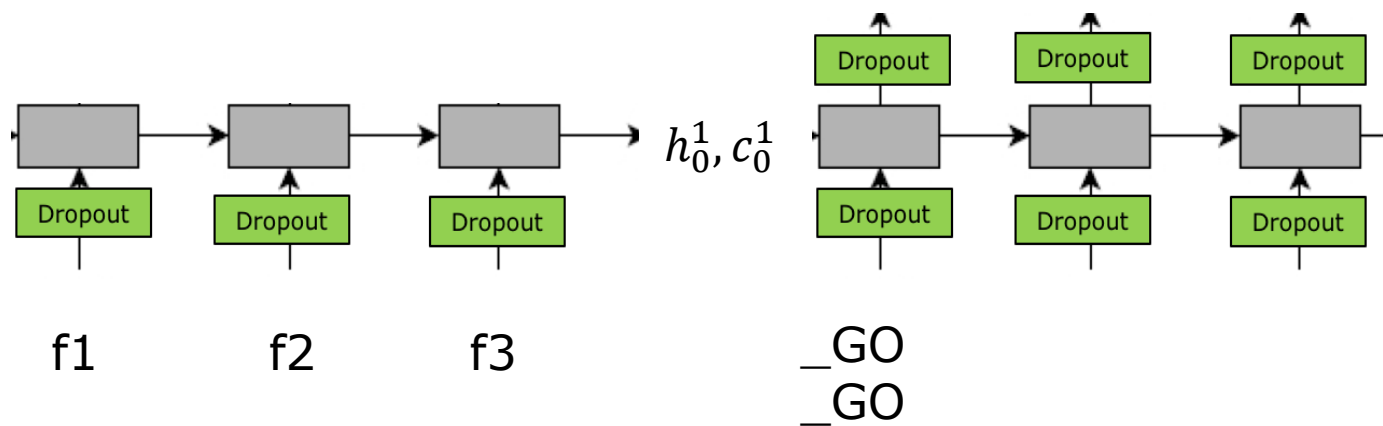
top_index: $\operatorname{argmax}_{\{w_i\}} P(w_i)$

top_value: $\max P(w_i)$

eos_value: $P(\text{EOS})$

Beam Search 代码实现 index = 0

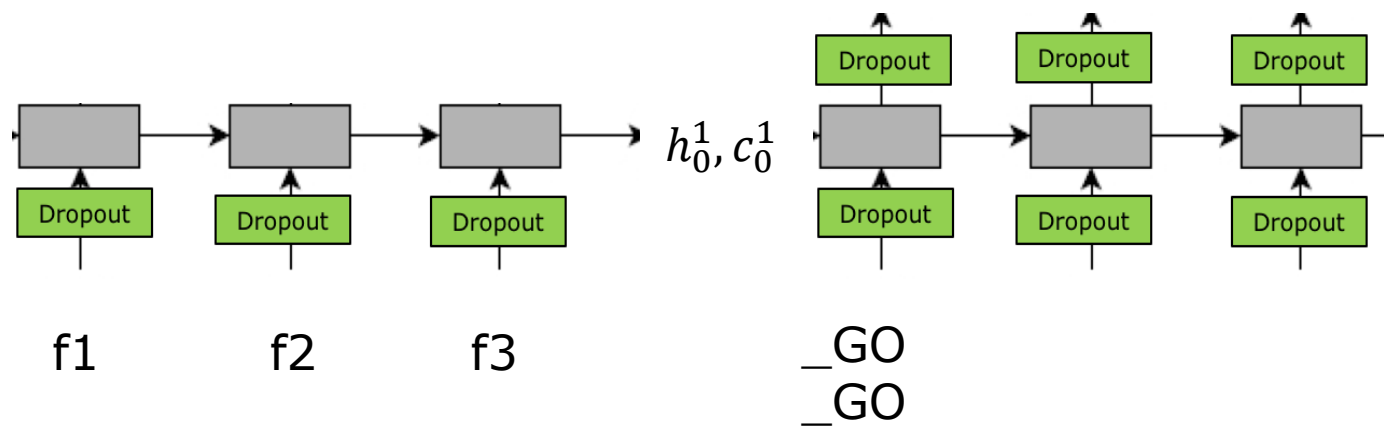
	scores	sentences	beam parent	target inputs	top index	top value	eos value
0	0.0	[]	0	_GO			
1	0.0	[]	1	_GO			



$$h_0^1 : \begin{bmatrix} 0.2, -0.3 \\ 0.2, -0.3 \end{bmatrix} = \gg h_1^1 :$$

Beam Search 代码实现 index = 0

	scores	sentences	beam parent	target inputs	top index	top value	eos value
0	0.0	[]	0	_GO	b, a	0.7,0.3	0.0001
1	0.0	[]	1	_GO	b, a	0.7,0.3	0.0001



$$h_0^1 : \begin{bmatrix} 0.2, -0.3 \\ 0.2, -0.3 \end{bmatrix} = \gg h_1^1 : \begin{bmatrix} 1.0, -3.3 \\ 1.0, -3.3 \end{bmatrix}$$

Beam Search 代码实现 index = 0

	scores	sentences	beam parent	target inputs	top index	top value	eos value
0	0.0	[]	0	_GO	b, a	0.7,0.3	0.0001
1	0.0	[]	1	_GO	b, a	0.7,0.3	0.0001

index = 0

```
global_queue = [(score, beam_index, word_index)]  
               = [(0+0.7,0,          b),  
                  (0+0.3,0,          a)]
```

```
target_inputs = [b, a]
```

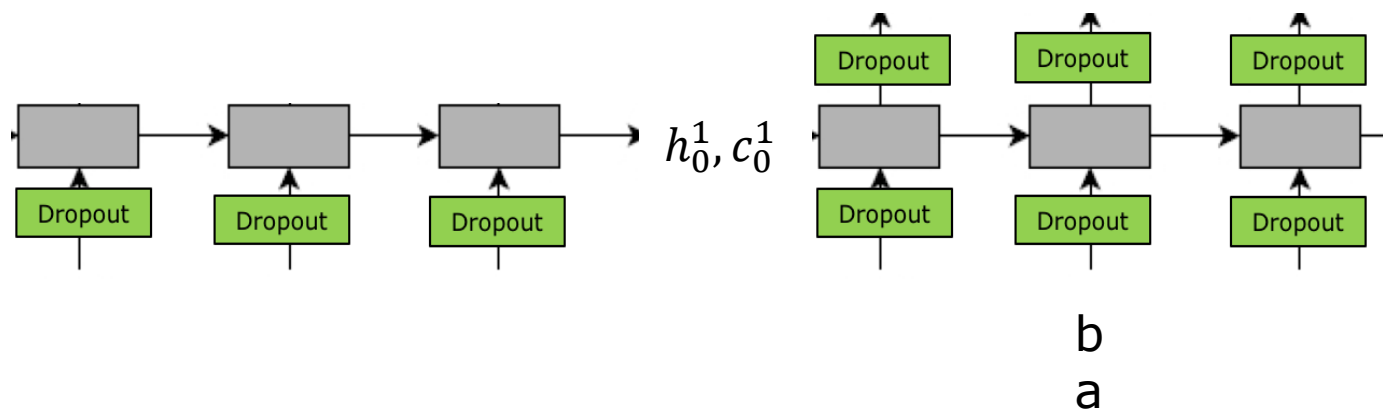
```
scores = [0.7,0.3]
```

```
beam_parent = [0,0]
```

```
sentences = [[b], [a]]
```

Beam Search 代码实现 index = 1

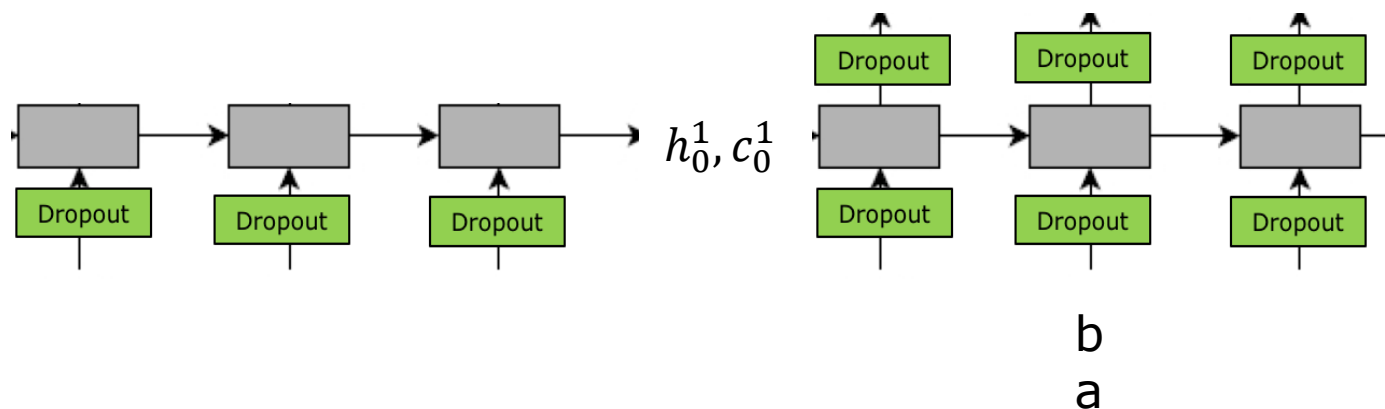
	scores	sentences	beam parent	target inputs	top index	top value	eos value
0	0.7	[b]	0	b			
1	0.3	[a]	0	a			



$$h_1^1 : \begin{bmatrix} 1.0, -3.3 \\ 1.0, -3.3 \end{bmatrix} \begin{matrix} \nearrow \\ \searrow \end{matrix} \begin{bmatrix} 1.0, -3.3 \\ 1.0, -3.3 \end{bmatrix} = \gg$$

Beam Search 代码实现 index = 1

	scores	sentences	beam parent	target inputs	top index	top value	eos value
0	0.7	[b]	0	b	a, b	0.5,0.5	0.001
1	0.3	[a]	0	a	a, b	0.8,0.2	0.001



$$h_1^1 : \begin{bmatrix} 1.0, -3.3 \\ 1.0, -3.3 \end{bmatrix} \begin{matrix} \nearrow \\ \searrow \end{matrix} \begin{bmatrix} 1.0, -3.3 \\ 1.0, -3.3 \end{bmatrix} = \gg h_2^1 : \begin{bmatrix} -1.3, -0.3 \\ 1.1, -1.5 \end{bmatrix}$$

Beam Search 代码实现 index = 1

	scores	sentences	beam parent	target inputs	top index	top value	eos value
0	0.7	[b]	0	b	a, b	0.5,0.5	0.001
1	0.3	[a]	0	a	a, b	0.8,0.2	0.001

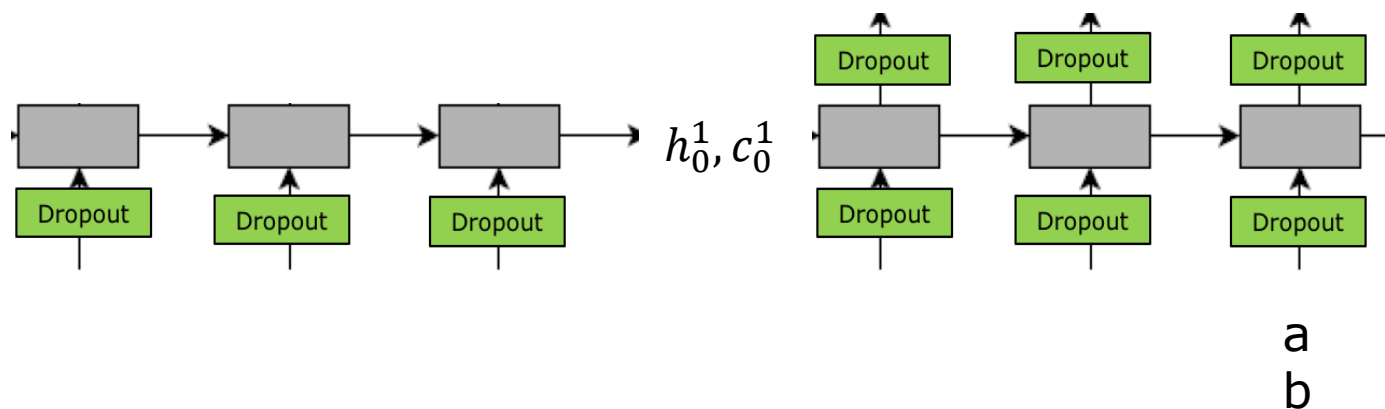
index = 1

```
global_queue = [(score, beam_index, word_index)]  
               = [(0.7*0.5,0, a),  
                  (0.7*0.5,0, b),  
                  (0.3*0.8,1, a),  
                  (0.3*0.2,1, b)]
```

```
target_inputs = [a, b]  
scores = [0.35,0.35]  
beam_parent = [0,0]  
sentences = [[b,a], [b,b]]
```


Beam Search 代码实现 index = 2

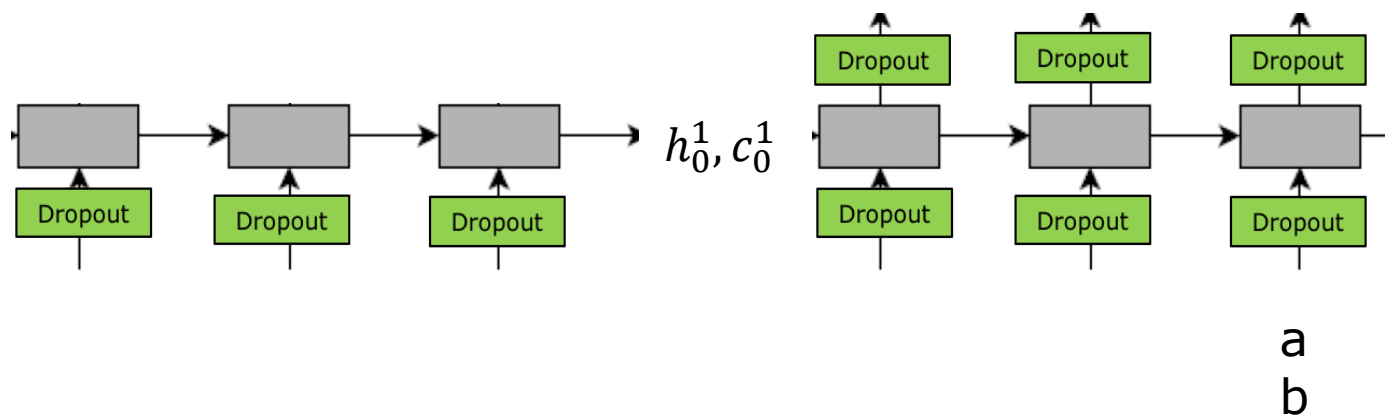
	scores	sentences	beam parent	target inputs	top index	top value	eos value
0	0.35	[b,a]	0	a			
1	0.35	[b,b]	0	b			



$$h_2^1 : \begin{bmatrix} -1.3, -0.3 \\ 1.1, -1.5 \end{bmatrix} \begin{matrix} \nearrow \\ \searrow \end{matrix} \begin{bmatrix} -1.3, -0.3 \\ 1.1, -1.5 \end{bmatrix} = \gg$$

Beam Search 代码实现 index = 2

	scores	sentences	beam parent	target inputs	top index	top value	eos value
0	0.35	[b,a]	0	a	b, a	0.7,0.3	0.0001
1	0.35	[b,b]	0	b	a, b	0.5,0.5	0.0001



$$h_2^1 : \begin{bmatrix} -1.3, -0.3 \\ 1.1, -1.5 \end{bmatrix} \begin{matrix} \nearrow \\ \searrow \end{matrix} \begin{bmatrix} -1.3, -0.3 \\ 1.1, -1.5 \end{bmatrix} = \gg h_3^1 : \begin{bmatrix} -1.5, -2.3 \\ 1.8, -2.1 \end{bmatrix}$$

Beam Search 代码实现 index = 2

	scores	sentences	beam parent	target inputs	top index	top value	eos value
0	0.35	[b,a]	0	a	b, a	0.7,0.3	0.0001
1	0.35	[b,b]	0	b	a, b	0.5,0.5	0.0001

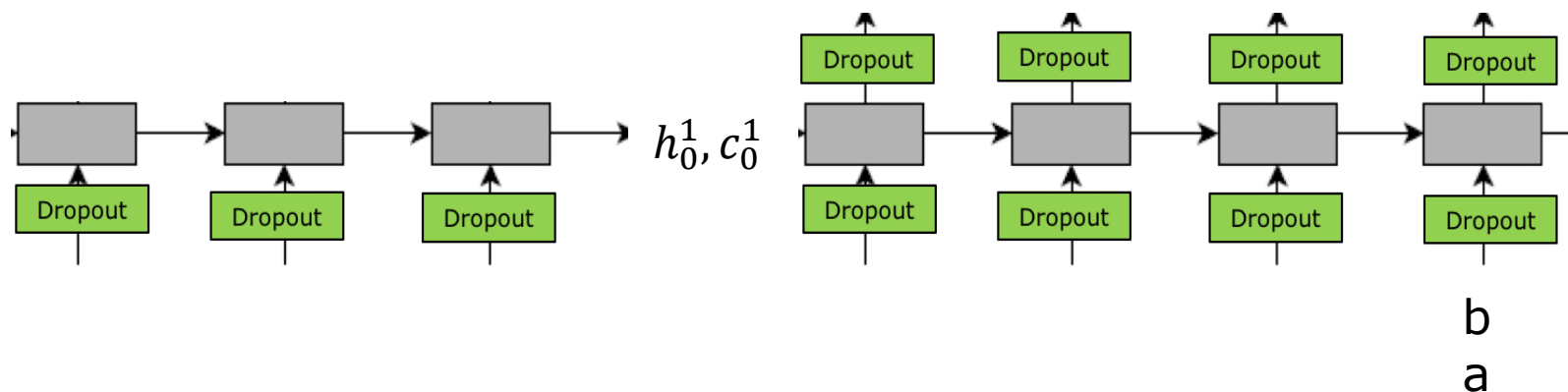
index = 2

```
global_queue = [(score, beam_index, word_index)]  
               = [(0.35*0.7,0, b),  
                  (0.35*0.3,0, b),  
                  (0.35*0.5,1, a),  
                  (0.35*0.5,1, b)]
```

```
target_inputs = [b, a]  
scores = [0.245,0.175]  
beam_parent = [0,1]  
sentences = [[b,a,b], [b,b,a]]
```

Beam Search 代码实现 index = 3

	scores	sentences	beam parent	target inputs	top index	top value	eos value
0	0.245	[b,a,b]	0	b	b, a	0.7,0.2	0.1
1	0.175	[b,b,a]	1	a	a, b	0.5,0.3	0.2



$$h_2^1 : \begin{bmatrix} -1.5, -2.3 \\ 1.8, -2.1 \end{bmatrix} \rightarrow \begin{bmatrix} -1.5, -2.3 \\ 1.8, -2.1 \end{bmatrix} \Rightarrow h_3^1 : \begin{bmatrix} 0.4, 2.2 \\ 1.2, -1.1 \end{bmatrix}$$

Beam Search 代码实现 index = 3

	scores	sentences	beam parent	target inputs	top index	top value	eos value
0	0.245	[b,a,b]	0	b	b, a	0.7,0.2	0.1
1	0.175	[b,b,a]	1	a	a, b	0.5,0.3	0.2

index = 3 == (max-length-1) 最后一步

```
global_queue = [(score, beam_index, word_index)]  
              = [(0.245*0.1, 0, EOS),  
                (0.175*0.2, 1, EOS)]
```

```
target_inputs = [EOS, EOS]
```

```
scores = [0.035, 0.0245]
```

```
beam_parent = [1, 0]
```

```
sentences = [[b,b,a,EOS], [b,a,b,EOS]]
```

```
results = [([b,b,a,EOS], 0.035),  
           ([b,a,b,EOS], 0.0245)]
```

Beam Search 代码实现 回到index=2

如果top_index是中包含EOS怎么办？

	scores	sentences	beam parent	target inputs	top index	top value	eos value
0	0.35	[b,a]	0	a	EOS, a	0.7,0.2	0.1
1	0.35	[b,b]	0	b	a, b	0.5,0.5	0.0001

index = 2

```
global_queue = [(score, beam_index, word_index)]  
               = [(0.35*0.7,0, EOS),  
                  (0.35*0.2,0, a),  
                  (0.35*0.5,1, a),  
                  (0.35*0.5,1, b)]
```

```
results = [[b,a,EOS], 0.245]
```

```
target_inputs = [a, b]
```

```
scores = [0.175,0.175]
```

```
beam_parent = [1,1]
```

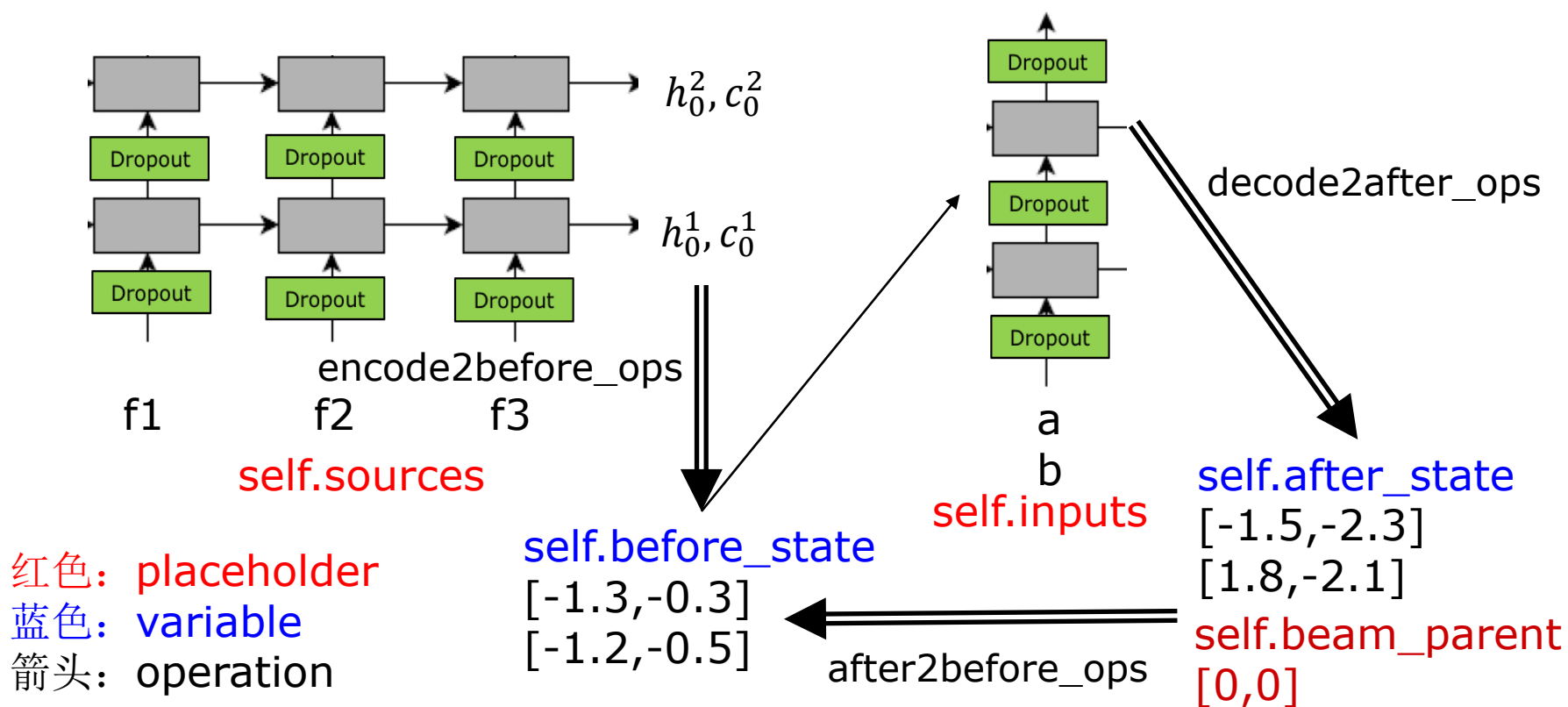
```
sentences = [[b,b,a], [b,b,b]]
```

Beam Search 代码实现

Single-step Decoder

`self.top_index`
`self.top_value`
`self.eos_value`

a:0.3 b:0.7
a:0.5 b:0.5



Beam Search 代码实现

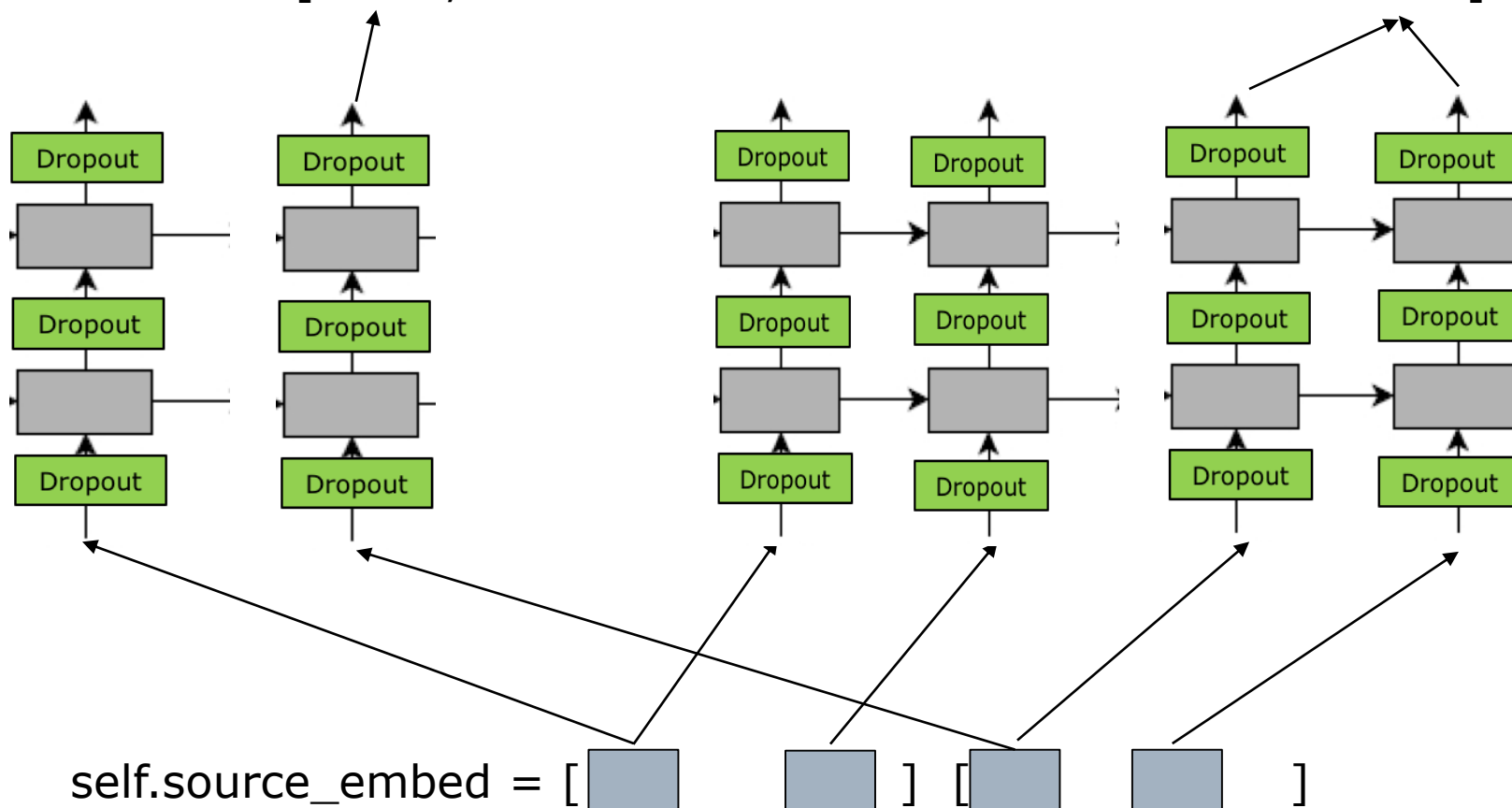
□ SeqModel.py

- `init_beam_decoder()` # “画”额外的“图”
- `beam_with_buckets()` #
- `beam_basic_seq2seq()` # Single-step decoder
- `beam_attention_seq2seq()` # **TODO**
- `show_before_state()` # 调试用
- `show_after_state()` # 调试用
- `beam_step()`
- `get_batch_test()`

Seq2Seq 代码实现

```
self.losses = [ loss1,
```

loss2]



```
self.source_embed = [ ]
```

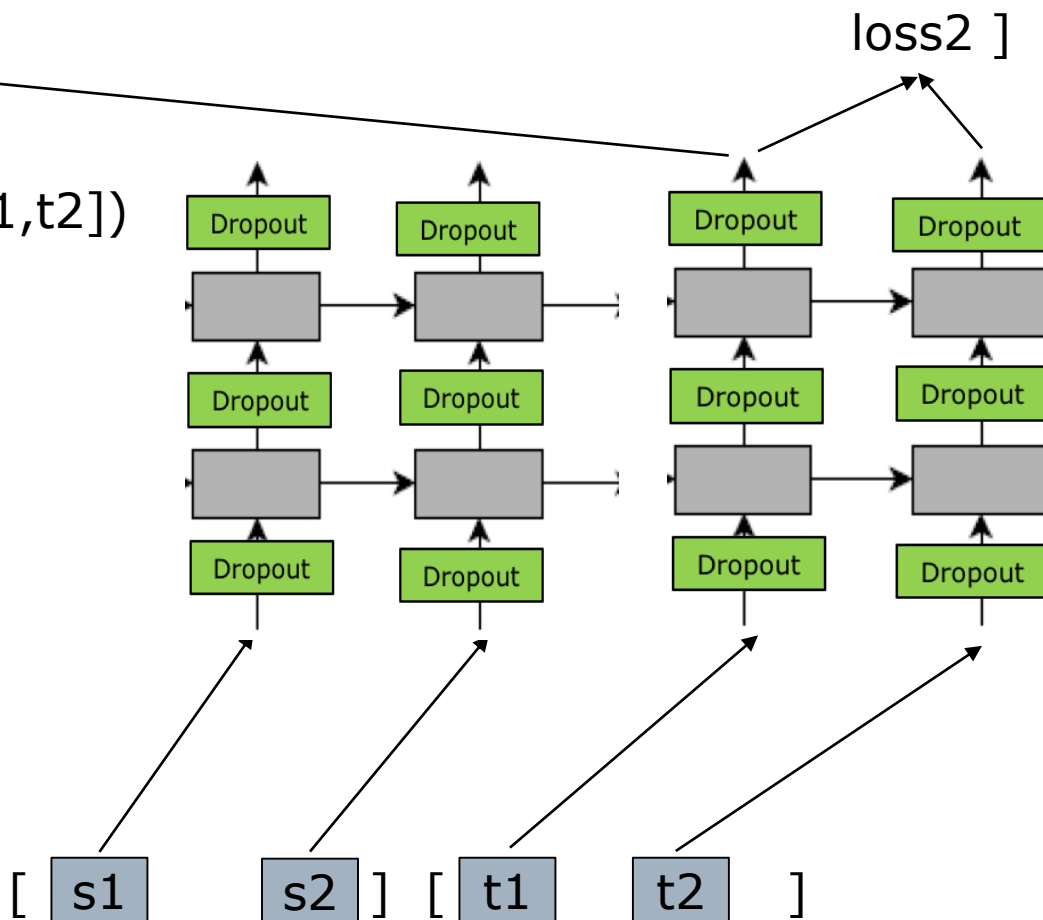
Seq2Seq 代码实现

为什么不“画”成这样？

```
self.losses = [ loss1,
```

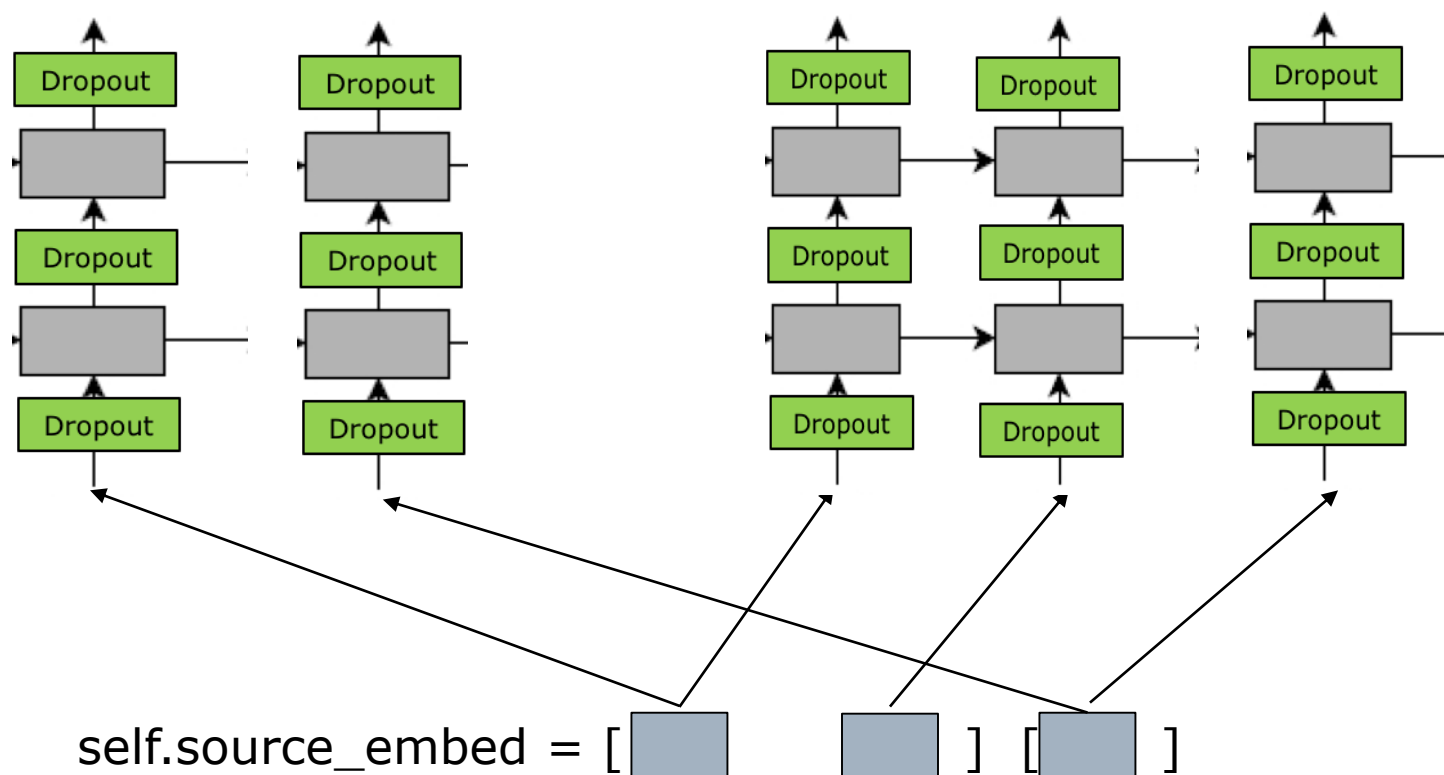
```
sess.run([loss2],[s1,s2,t1,t2])
```

```
sess.run([loss1],[?]
```



Beam Search 代码实现

beam_with_buckets()



Beam Search 代码实现

□ BLEU score

■ 评价机器翻译的标准

$$\text{BLEU} = \min \left(1, \frac{\text{output-length}}{\text{reference-length}} \right) \left(\prod_{i=1}^4 \text{precision}_i \right)^{\frac{1}{4}}$$

Beam Search 代码实现

□ BLEU score

SYSTEM A: Israeli officials responsibility of airport safety
2-GRAM MATCH 1-GRAM MATCH

REFERENCE: Israeli officials are responsible for airport security

SYSTEM B: airport security Israeli officials are responsible
2-GRAM MATCH 4-GRAM MATCH

Metric	System A	System B
precision (1gram)	3/6	6/6
precision (2gram)	1/5	4/5
precision (3gram)	0/4	2/4
precision (4gram)	0/3	1/3
brevity penalty	6/7	6/7
BLEU	0%	52%

Beam Search 代码实现

□ BLEU score

- `bash beam_decode_small.sh`

- `bash bleu_small.sh`

Beam Search 代码实现

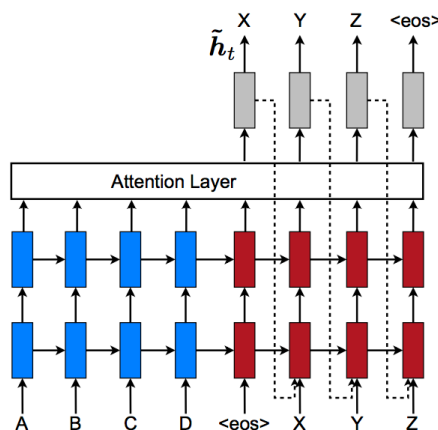
□ 高难度，高价值的作业

■ 实现attention model的beam search

□ beam_attention_seq2seq()

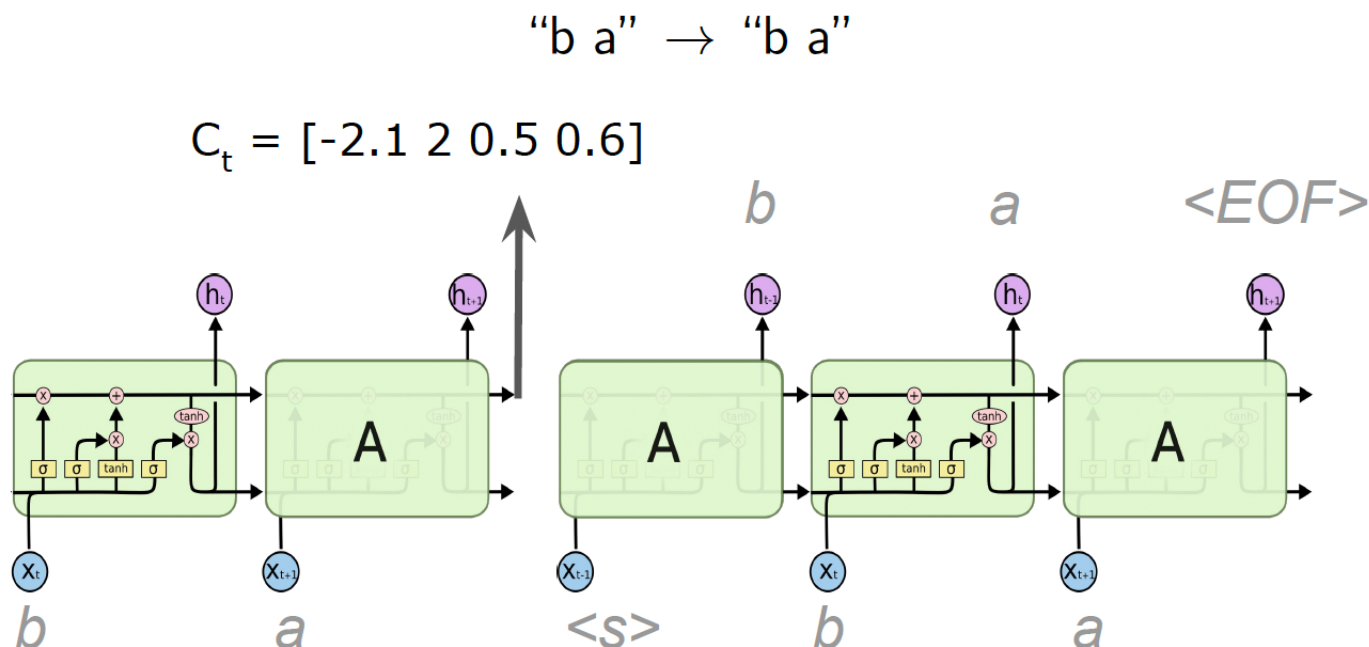
□ feed_input是否需要加before_ht_att和after_ht_att

□ 一周时间实现，实现成功的有神秘礼物



Seq2Seq 模型可视化

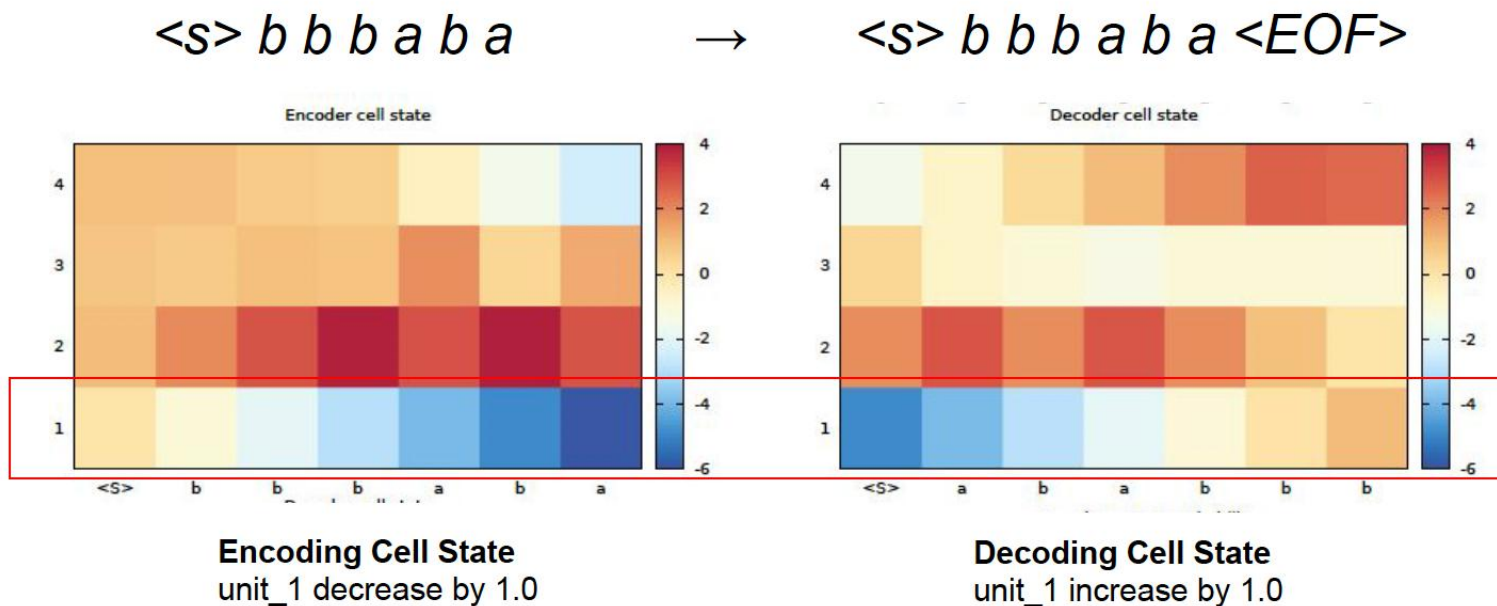
□ Hidden State 可视化



C_t only involves **element-wise** “+” and “X”

Seq2Seq 模型可视化

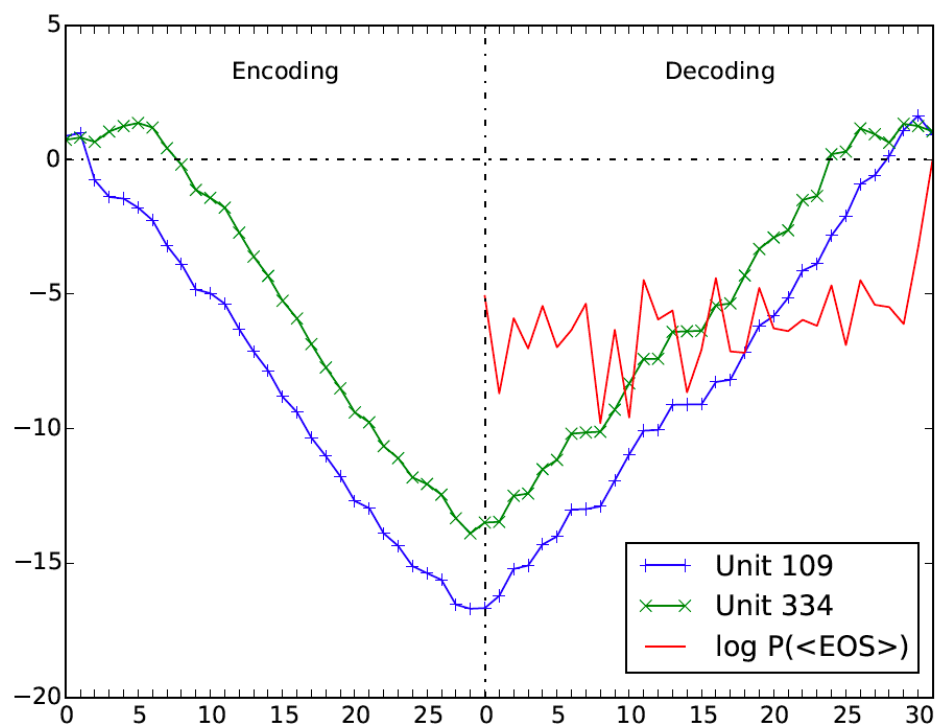
□ Hidden State 可视化



Seq2Seq 模型可视化

□ Hidden State 可视化

■ French-English 机器翻译模型



联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：小象
- 新浪微博：ChinaHadoop

