

CSDN新首页上线啦，邀请你来立即体验！(<http://blog.csdn.net/>)

立即体

验

**CSDN**

博客 (<http://blog.csdn.net/>) 学院 (<http://edu.csdn.net?ref=toolbar>)

下载 (<http://download.csdn.net?ref=toolbar>) GitChat (<http://gitbook.cn/?ref=csdn>)

更多 

0

## 浅谈Attention-based Model【源码篇】

原创

2017年09月09日 22:54:19

696



转载请标明出处：<http://blog.csdn.net/wuzqchom/article/details/77918780>

(<http://blog.csdn.net/wuzqchom/article/details/77918780>)

源码不可能每一条都详尽解释，主要在一些关键步骤上加了一些注释和少许个人理解，如有不足之处，请予指正。

计划分为三个部分：

浅谈Attention-based Model【原理篇】(<http://blog.csdn.net/wuzqChom/article/details/75792501>)

浅谈Attention-based Model【源码篇】(你在这里)

浅谈Attention-based Model【实践篇】



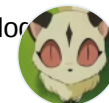
weixin\_3506...

(<http://my.csdn.net?ref=toolbar>)

(<http://write.blog.csdn.net/postedit?ref=toolbar>)

(<http://blog.csdn.net/wuzqchom>)

wuzqchom (<http://blog.cs...>)



+ 关注

(<http://blog.csdn.net/wuzqchom>)

码云

原创

粉丝

喜欢

未开通  
(<https://github.com/wuzqchom>)

56

6

5

utm\_sourc

### 他的最新文章

更多文章 (<http://blog.csdn.net/wuzqchom>)

使用BibTex格式时缩小参考文献的字体  
(<http://blog.csdn.net/wuzqChom/article/details/78592038>)

【转】计算机类学术论文 28个常见出版社一般写法（参考文献用）(<http://blog.csdn.net/wuzqChom/article/details/78196847>)

Batch Normalization (<http://blog.csdn.net/wuzqChom/article/details/78024653>)

在之前的博客：浅谈Attention-based Model【原理篇】

(<http://blog.csdn.net/wuzqchom/article/details/75792501>)中，介绍了Attention机制的原理，这篇主要通过tensorflow源码来看看注意力机制的工作过程。

## 0. 主要函数结构



0

先来看看主要函数的调用过程（接下来也是按照这样的结构解释源码）：



- 1.embedding\_attention\_seq2seq()
  - 2.embedding\_attention\_decoder()
- 3.attention\_decoder()
  - attention()



其中函数attention()是实现每一个时刻的attention机制的主要函数，即论文Grammar as a Foreign Language (<https://arxiv.org/abs/1412.7449>)中的Attention Mechanism基本上都是在这个函数里面实现的

为什么说基本？这是因为 $W_1' h_i$ 是在该函数之外使用卷积操作实现的，具体详见3.attention\_decoder()函数。

## 1. embedding\_attention\_seq2seq()

Python切片赋值操作 (<http://blog.csdn.net/wuzqChom/article/details/77917825>)

Sampled Softmax (<http://blog.csdn.net/wuzqChom/article/details/77073246>)

### 相关推荐

自然语言处理中的Attention Model：是什么及为什么 (<http://blog.csdn.net/malefactor/article/details/50550211>)

深度学习笔记——Attention Model（注意力模型）学习总结 ([http://blog.csdn.net/mpk\\_no1/article/details/72862348](http://blog.csdn.net/mpk_no1/article/details/72862348))

以Attention Model为例谈谈两种研究创新模式 (<http://blog.csdn.net/malefactor/article/details/50583474>)

浅谈Attention-based Model【原理篇】 (<http://blog.csdn.net/wuzqChom/article/details/75792501>)

embedding\_attention\_seq2seq函数参数说明如下：

```

1 def embedding_attention_seq2seq(encoder_inputs,# 编码器的输入
2     decoder_inputs,# 解码器的输入
3     cell,# 所用的cell
4     num_encoder_symbols,# 编码的符号总数
5     num_decoder_symbols,# 解码的符号总数
6     embedding_size,# 向量的维度
7     num_heads=1,# 这个也不知道啊，默认为1，基本上可以当它不存在了
8     output_projection=None,# 输出投影
9     feed_previous=False,# 当前输入是否要考虑前一个时刻的输出，这个一般在训练的时候为False，预测的时
10    dtype=None,
11    scope=None,
12    initial_state_attention=False):
13
14 返回值：
15 由（outputs，state）组成的元组
16 outputs的shape为[batch_size x num_decoder_symbols]
17 state为最后一个时刻decoder的状态，shape为[batch_size x cell.state_size]
```

这个函数首先为一个编码的过程：

使用EmbeddingWrapper将输入映射到embedding\_size大小的向量，然后通过调用static\_rnn得到了encoder的每一个时刻的输出，即为之后我们需要attention的向量。

然后再进行解码的过程：

首先使用OutputProjectionWrapper将解码器的输出映射成想要的维度



## 他的热门文章

TensorFlow中CNN的两种padding方式“SAME”和“VALID” (<http://blog.csdn.net/wuzqchom/article/details/74785643>)

📖 4769

tensorflow.nn.bidirectional\_dynamic\_rnn()函数的用法 (<http://blog.csdn.net/wuzqchom/article/details/75453327>)

接下来执行：

```
if isinstance(feed_previous, bool):
    return embedding_attention_decoder
```

## 2. embedding\_attention\_decoder()

```

1  def embedding_attention_decoder(decoder_inputs,
0  2      initial_state,
3      attention_states,
4      cell,
5      num_symbols,
6      embedding_size,
7      num_heads=1,
8      output_size=None,# 输出的大小，tensorflow中为num_decoder_symbols
9      output_projection=None,
10     feed_previous=False,
11     update_embedding_for_previous=True,
12     dtype=None,
13     scope=None,
14     initial_state_attention=False):# 初始attention向量的值，默认为0
15
16  返回值：
17  同上
```

这个函数主要是通过embedding\_ops.embedding\_lookup()函数把decoder\_inputs转换为向量的形式，之前decoder\_inputs用符号的index的表示。

## 3. attention\_decoder()

📖 3036

python中scrapy点击按钮 (<http://blog.csdn.net/wuzqchom/article/details/52677579>)

📖 2305

浅谈Attention-based Model【原理篇】 (<http://blog.csdn.net/wuzqchom/article/details/75792501>)

📖 1649

Seq2Seq模型 (<http://blog.csdn.net/wuzqchom/article/details/76651479>)

📖 1515

```

1 def attention_decoder(decoder_inputs,# 编码器的输入，向量形式
2     initial_state,# 初始状态，tensorflow照顾你为rnn编码器的最后一个时刻的state
3     attention_states,# 需要attention的向量，即为编码器的每一个时刻的输出
4     cell,
5     output_size=None,
6     num_heads=1,
7     loop_function=None,
8     dtype=None,
9     scope=None,
10    initial_state_attention=False):

```



终于看到核心的函数了，attention解码器参照论文

Grammar as a Foreign Language (<https://arxiv.org/abs/1412.7449>)实现

为了说明方便，我把论文当中的公式放上来了：



$$u_i^t = v^T \tanh(W_1' h_i + W_2' d_t)$$

$$a_i^t = \text{softmax}(u_i^t)$$

$$d_t' = \sum_{i=1}^{T_A} a_i^t h_i$$

<http://blog.csdn.net/wuzqChom>

我们依次把这三个公式标号为公式(1),(2),(3)。

首先来看看 $W_1' h_i$ 的实现:

```
1  for a in xrange(num_heads):
2      # filter的大小，输入的通道数为输入embedding_size的大小，输出的通道为attention_vec_size，但正如前面所说，tenso
3      k = variable_scope.get_variable("AttnW_%d" % a,
4                                      [1, 1, attn_size, attention_vec_size])
5      hidden_features.append(nn_ops.conv2d(hidden, k, [1, 1, 1, 1], "SAME"))
6      v.append(
7          variable_scope.get_variable("AttnV_%d" % a, [attention_vec_size]))
```

如上，使用卷积核来实现相乘的操作，attn\_size为attention的长度，即为编码器是输出的向量个数，attention\_vec\_size为attention向量的长度，tensorflow默认为embedding\_size的大小，而且并没有提供参数传递，但是这个假如手动改成其它的值也没有问题。

```
1 <font size=4>
2 #循环，依次将解码状态的每一个时刻的是state都做一次attention，然后和该时刻的decoder_inputs值共同决定该时刻的输
3 for i, inp in enumerate(decoder_inputs):
4     if i > 0:
5         variable_scope.get_variable_scope().reuse_variables()
6     # If loop_function is set, we use it instead of decoder_inputs.
7     if loop_function is not None and prev is not None:
8         with variable_scope.variable_scope("loop_function", reuse=True):
9             inp = loop_function(prev, i)
10    # Merge input and previous attentions into one vector of the right size.
11    input_size = inp.get_shape().with_rank(2)[1]
12    if input_size.value is None:
13        raise ValueError("Could not infer input size from input: %s" % inp.name)
14    # 和decoder_inputs该时刻的值共同决定该时刻的输入
15    # 这里的因为原文用的词为concat，但是这里应该是每一个维度的值相加
16    x = linear([inp] + attns, input_size, True)
17    # Run the RNN.
18    # 使用输入和上一个时刻的隐状态共同决定当前时刻的隐状态和解码的输出
19    cell_output, state = cell(x, state)
20    # Run the attention mechanism.
21    if i == 0 and initial_state_attention:
22        with variable_scope.variable_scope(
23            variable_scope.get_variable_scope(), reuse=True):
24            attns = attention(state)
25    else:
26        attns = attention(state)
27    with variable_scope.variable_scope("AttnOutputProjection"):
28        output = linear([cell_output] + attns, output_size, True)
29    if loop_function is not None:
30        prev = output
31    outputs.append(output)
```

最后的最后，还有最为核心的attention函数，上面是T个时刻的循环。但是每一个时刻的attention是由该函数完成的，即真正实现上面三个公式的地方，敲黑板：

```

1  def attention(query):
2      """Put attention masks on hidden using hidden_features and query."""
3      ds = [] # Results of attention reads will be stored here.
4      if nest.is_sequence(query): # If the query is a tuple, flatten it.
5          query_list = nest.flatten(query)
6          for q in query_list: # Check that ndims == 2 if specified.
7              ndims = q.get_shape().ndims
8              if ndims:
9                  assert ndims == 2
10             query = array_ops.concat(query_list, 1)
11         for a in xrange(num_heads):
12             with variable_scope.variable_scope("Attention_%d" % a):
13                 # 公式 ( 1 ) 中的  $W_2^{d_t}$ 
14                 y = linear(query, attention_vec_size, True)
15                 y = array_ops.reshape(y, [-1, 1, 1, attention_vec_size])
16                 # Attention mask is a softmax of  $v^T * \tanh(\dots)$ .
17                 # 公式 ( 1 ) 结果
18                 s = math_ops.reduce_sum(v[a] * math_ops.tanh(hidden_features[a] + y),
19                                         [2, 3])
20                 # 公式 ( 2 )
21                 a = nn_ops.softmax(s)
22                 # Now calculate the attention-weighted vector d.
23                 # 公式 ( 3 )
24                 d = math_ops.reduce_sum(
25                     array_ops.reshape(a, [-1, attn_length, 1, 1]) * hidden, [1, 2])
26                 ds.append(array_ops.reshape(d, [-1, attn_size]))
27         return ds

```



版权声明：本文为博主原创文章，未经博主允许不得转载。



发表你的评论

([http://my.csdn.net/weixin\\_35068028](http://my.csdn.net/weixin_35068028))



## 相关文章推荐



### 自然语言处理中的Attention Model：是什么及为什么 (<http://blog.csdn.net/malefactor/article...>)



要是关注深度学习在自然语言处理方面的研究进展，我相信你一定听说过Attention Model（后文有时会简称AM模型）这个词。AM模型应该说是过去一年来NLP领域中的重要进展之一，在很多场景被证明有...



malefactor (<http://blog.csdn.net/malefactor>) 2016年01月20日 18:26 58071

### 深度学习笔记——Attention Model（注意力模型）学习总结 ([http://blog.csdn.net/mpk\\_no1/a...](http://blog.csdn.net/mpk_no1/a...))

Attention Model（注意力模型）学习总结，包括soft Attention Model，Global Attention Model和Local Attention Model，静态AM，...



mpk\_no1 ([http://blog.csdn.net/mpk\\_no1](http://blog.csdn.net/mpk_no1)) 2017年08月06日 21:49 7780



广告

### 惊呆了！微博和阿里背后的数据库有多厉害？

想不到！数据库作为最关键的基础设施，渗透技术领域的方方面面，我阿里和微博的师哥们是这么分享的...

(http://www.baidu.com/cb.php?c=lgF\_pyfqHmknjTzrb0lZ0qnfK9ujYzP1nsrjD10Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1d9mHN-m1f3uWc4uhDLPHNB0AwY5HDdnHfznWRvPHn0lgF\_5y9YIZ0lQzq-uZR8mLPbUB48ugfEpZNGXy-jULNzTvRETVnzpyN1gVw-IA7GUatLPjqdIAdxTvqdThP-5yF\_UvTkn0KzujYk0AFV5H00TZcqn0KdpyfqHRLPjnvnfKEpyfqHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqPWRYrj0)

## 以Attention Model为例谈谈两种研究创新模式 (http://blog.csdn.net/malefactor/article/detail...

0

各位观众朋友好，也许此刻您刚打开电梯.....上了年纪的读者估计能看出上面一句是引用了韩乔生老先生的名言，我写东西就喜欢用名人名言开头，这好习惯这么多年怎么也改不了。您问韩乔生是谁？恭喜您，作为90后您以...



malefactor (http://blog.csdn.net/malefactor) 2016年01月26日 00:18 21394

## 浅谈Attention-based Model【原理篇】 (http://blog.csdn.net/wuzqChom/article/details/757...

转载请注明出处：http://blog.csdn.net/wuzqchom/article/details/75792501 计划分为三个部分：浅谈Attention-based Model【...



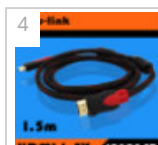
wuzqChom (http://blog.csdn.net/wuzqChom) 2017年07月22日 19:24 1662

## 深度学习方法（九）：自然语言处理中的Attention Model注意力模型 (http://blog.csdn.net/xb...

上一篇博文深度学习方法（八）：Encoder-Decoder模型，基本Sequence to Sequence模型描述了基本的Encoder-Decoder模型，在作为翻译模型的时候，这种基本的Enc...



xbinworld (http://blog.csdn.net/xbinworld) 2017年02月04日 00:27 6506



**2.60/条**  
高品质hdmi线 高清线  
1.5米 hdmi1.4版 hdmi




**560.00/台**  
工厂专业生产挖矿机电  
源800W-3300W，质



**6.00/卷**  
厂家直销 HDMI2.0版  
锌合金外壳高清线 支

## 自然语言处理中的Attention Model：是什么及为什么 (<http://blog.csdn.net/jdbc/article/detail...>)


要是关注深度学习在自然语言处理方面的研究进展，我相信你一定听说过Attention Model（后文有时会简称AM模型）这个词。AM模型应该说是过去一年来NLP领域中的重要进展之一，在很多场景被证明有...

 jdbc (<http://blog.csdn.net/jdbc>) 2016年01月26日 11:16 2336




## 【深度学习】聚焦机制DRAM(Deep Recurrent Attention Model)算法详解 (<http://blog.csdn.n...>)

Visual Attention基础，Multiple object recognition with visual attention算法解读。

 shenxiaolu1984 (<http://blog.csdn.net/shenxiaolu1984>) 2016年06月28日 22:14 6484


## attention 机制 ([http://blog.csdn.net/qq\\_26609915/article/details/52086772](http://blog.csdn.net/qq_26609915/article/details/52086772))

attention 机制什么是attentionattention机制是（非常）松散地基于人类的视觉注意机制。就是按照“高分辨率”聚焦在图片的某个特定区域并以“低分辨率”感知图像的周边区域的模式，然后...

 qq\_26609915 ([http://blog.csdn.net/qq\\_26609915](http://blog.csdn.net/qq_26609915)) 2016年08月01日 16:24 5285

## 深度学习中的Attention模型介绍及其进展 (<http://blog.csdn.net/jteng/article/details/52864401>)

近期对深度学习中的Attention模型进行了深入研究，该模型在图像识别、语音识别和自然语言处理三大深度学习的热门领域均有广泛的使用，是2014和2015年深度学习领域的重要进展。现对其原理、主要应用...


 jteng (<http://blog.csdn.net/jteng>) 2016年10月20日 15:29 7023


## 阅读理解任务中的Attention-over-Attention神经网络模型原理及实现 (<http://blog.csdn.net/liu...>)

本文是“Attention-over-Attention Neural Networks for Reading Comprehension”的阅读笔记。这篇论文所处理的任务是阅读理解里面的完形填空问...


 liuchonge (<http://blog.csdn.net/liuchonge>) 2017年06月06日 09:24  2569



## tensorflow seq2seq模型 代码阅读分析 ([http://blog.csdn.net/vincent\\_hbl/article/details/770...](http://blog.csdn.net/vincent_hbl/article/details/770...))

 如果刚开始入门该模型请阅读tf官方说明：Sequence-to-Sequence Models模型应用于机器翻译的示例代码：github如果还没有看懂tf的translate示例代码，请先理解透彻tr...

 vincent\_hbl ([http://blog.csdn.net/vincent\\_hbl](http://blog.csdn.net/vincent_hbl)) 2017年08月11日 15:08  971



## tensorflow学习笔记（十一）：seq2seq Model相关接口介绍 (<http://blog.csdn.net/u0124361...>)

 seq2seq Model源码地址调用外部的函数介绍tf.sampled\_softmax\_loss()tf.sampled\_softmax\_loss()中调用了\_compute\_sampled\_log...

 u012436149 (<http://blog.csdn.net/u012436149>) 2016年10月31日 00:04  14319


## basic\_rnn\_seq2seq与embedding\_attention\_seq2seq输入的差别 (<http://blog.csdn.net/u01...>)

在用embedding\_attention\_seq2seq 跑完数据之后，想用basic\_rnn\_seq2seq再跑一遍进行对比，发现报错了，错误是ValueError: Linear is expe...

 u013713117 (<http://blog.csdn.net/u013713117>) 2017年02月22日 16:34  1886


## 【增强学习】Recurrent Visual Attention源码解读 (<http://blog.csdn.net/shenxiaolu1984/art...>)

Mnih, Volodymyr, Nicolas Heess, and Alex Graves. "Recurrent models of visual attention." Advances in...

 shenxiaolu1984 (<http://blog.csdn.net/shenxiaolu1984>) 2016年06月17日 21:18  8261

## Seq2Seq源码解析（基于Theano框架）(<http://blog.csdn.net/yezhenxu1992/article/details/...>)


这是一篇关于seq2seq源码解析的文章。seq2seq是一个通用型 端到端（end-to-end）的学习框架，包括2个部分，一个是Encoder（编码器，负责编码源句子的输入），另外一个Decod...

 yezhenxu1992 (<http://blog.csdn.net/yezhenxu1992>) 2017年05月15日 15:57 2802

0


## cnn+rnn+attention (<http://blog.csdn.net/u014365862/article/details/78495870>)

下面是单层rnn+attention的代码，若考虑多层rnn请参考博主的：tf.contrib.rnn.static\_rnn与tf.nn.dynamic\_rnn区别 def attention

 u014365862 (<http://blog.csdn.net/u014365862>) 2017年11月10日 08:16 198


## 对Attention is all you need 的理解 (<http://blog.csdn.net/mijiaoxiaosan/article/details/7325...>)

对谷歌Attention is all you need 的理解。

 mijiaoxiaosan (<http://blog.csdn.net/mijiaoxiaosan>) 2017年06月14日 19:24 5994


## 【OpenCV】图像特征提取 (<http://blog.csdn.net/bleess2015/article/details/50681974>)

形状提取 #include using namespace cv; /\* 形状提取 \*/ int main() { IplImage\* pCanny...

 bleess2015 (<http://blog.csdn.net/bleess2015>) 2016年02月17日 18:06 1702

## 浅谈Attention-based Model【原理篇】 (<http://blog.csdn.net/wuzqChom/article/details/757...>)

转载请标明出处：http://blog.csdn.net/wuzqchom/article/details/75792501 计划分为三个部分：浅谈Attention-based Model  
【...】

 wuzqChom (http://blog.csdn.net/wuzqChom) 2017年07月22日 19:24 1662

## EKF based object detect and tracking for UAV by using visual-attention-...



(http://download.csdn.net/detail/wuzqchom/10444444)

0

2015年03月28日 15:40

999KB

下载

