



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



博客 (//blog.csdn.net/54093161) 学院 (//edu.csdn.net?ref=toolbar)

下载 (//download.csdn.net?ref=toolbar) GitChat (//gitbook.cn/?ref=csdn)

更多 ▾

## tensorflow笔记:使用tf来实现word2vec

原创 2017年01月05日 14:50:21

标签：python (http://so.csdn.net/so/search/s.do?q=python&t=blog) /

tensorflow (http://so.csdn.net/so/search/s.do?q=tensorflow&t=blog) /

word2vec (http://so.csdn.net/so/search/s.do?q=word2vec&t=blog)

10500

- （一）tensorflow笔记：流程，概念和简单代码注释 (http://blog.csdn.net/u014595019/article/details/52677412)
- （二）tensorflow笔记：多层CNN代码分析 (http://blog.csdn.net/u014595019/article/details/52728886)
- （三）tensorflow笔记：多层LSTM代码分析 (http://blog.csdn.net/u014595019/article/details/52759104)
- （四）tensorflow笔记：常用函数说明 (http://blog.csdn.net/u014595019/article/details/52805444)
- （五）tensorflow笔记：模型的保存与训练过程可视化 (http://blog.csdn.net/u014595019/article/details/53912710)
- （六）tensorflow笔记：使用tf来实现word2vec (http://blog.csdn.net/u014595019/article/details/54093161)

时隔若干个月，又绕到了word2vec。关于word2vec的原理我就不叙述了，具体可见word2vec中的数学 (http://download.csdn.net/detail/mzg12345678/7988741)，写的非常好。我后来自己用Python实现了一遍word2vec,过程写在自己动手写word2vec (一):主要概念和流程 (http://blog.csdn.net/u014595019/article/details/51884529)以及后续的若干文章中

我当时使用的是Hierarchical Softmax+CBOW的模型。给我的感觉是比较累，既要费力去写huffman树，还要自己写计算梯度的代码，完了按层softmax速度还慢。这次我决定用tensorflow来写，除了极大的精简了代码以外，可以使用gpu对运算进行加速。此外，这次使用了负采样(negative sampling)+skip-gram模型，从而避免了使用Huffman树导致训练速度变慢的情况，适合大规模的文本。

一些相关的资料：

word2vec 中的数学原理详解-基于 Negative Sampling 的模型

(http://blog.csdn.net/u014595019/article/details/51970582)

自己动手写word2vec (四):CBOW和skip-gram模型 (http://blog.csdn.net/u014595019/article/details/51943428)

tensorflow笔记：流程，概念和简单代码注释 (http://blog.csdn.net/u014595019/article/details/52677412)

tensorflow笔记：常用函数说明 (http://blog.csdn.net/u014595019/article/details/52805444)

其实google已经实现过一遍word2vec了（点这里 (https://www.tensorflow.org/tutorials/word2vec/)），我看了一下代码,感觉核心代码非常简介干练，我自己写的许多运算和函数调用也是参照它来的，但是关于外围的代码，包括数据集的生成等方面，我不是很喜欢，而且也与我的要求不符，所以我重新写了一下，并且进行了封装，增加了模型的存/取，训练过程可视化等功能，并且简化了流程。

我的模型主要分成两部分：由输入单词生成训练集的外围代码，以及用于描述模型，训练的核心代码。在训练的时候，外围代码收到一个分好词的句子，例如['我','中午','吃饭'],然后根据skip-gram模型，将其转化



电脑租赁

达内的

气垫bb怎么样 达内证书有用吗  
199元智能手机 排名第一的手机  
世界十大邮轮 泰国佛牌不要碰  
带摄像头的眼镜 带投影的手机



+ 关注

(http://blog.csdn.net/u014595019)

码云  
未开通  
(https://gite  
utm\_sourc

### 他的最新文章

更多文章 (http://blog.csdn.net/u014595019)

Jacobian矩阵，Hessian矩阵和牛顿法  
(http://blog.csdn.net/u014595019/articl  
e/details/58588383)

tensorflow笔记：模型的保存与训练过  
程可视化 (http://blog.csdn.net/u014595  
019/article/details/53912710)

ubuntu16.04下安装CUDA，cuDNN及t  
ensorflow-gpu版本过程 (http://blog.csd  
n.net/u014595019/article/details/53732  
015)

electron开发记录(五)：调用jquery.boot  
strap并对其一些用法的总结 (http://blo  
g.csdn.net/u014595019/article/details/  
53610731)

electron开发记录(四)：electron中组件  
的一些用法 (http://blog.csdn.net/u0145  
95019/article/details/53516584)

### 相关标签



电脑租赁

达内的

气垫bb怎么样 达内证书有用吗  
199元智能手机 排名第一的手机  
世界十大邮轮 泰国佛牌不要碰  
带摄像头的眼镜 带投影的手机



+ 关注

p://blog.csdn.net/u012436149/article/deta  
ils/52848013)



立即体

验

内容举报

返回顶部

成输入集和标签集。例如

[我，中午，中午，吃饭]

[中午，我，吃饭，中午]

当然了，实际过程中输入集和标签集都是用id来表示的。生成输入集和标签集以后，将其输入核心代码进行训练。那就先从核心代码讲起吧。这篇文章中的代码是不完全的，想看完整版的可以移步<https://github.com/multiangle/tfword2vec> (<https://github.com/multiangle/tfword2vec>)

## 核心代码

核心代码主要就是描述模型，计算loss,根据loss优化参数等步骤。这里计算loss直接使用了tf封装好的tf.nn.nce\_loss方法，比较方便。优化方法这里也是选的最简单的梯度下降法。具体的描述就放在代码里说好了

```
1 self.graph = tf.Graph()
2 self.graph = tf.Graph()
3 with self.graph.as_default():
4     # 首先定义两个用作输入的占位符，分别输入输入集(train_inputs)和标签集(train_labels)
5     self.train_inputs = tf.placeholder(tf.int32, shape=[self.batch_size])
6     self.train_labels = tf.placeholder(tf.int32, shape=[self.batch_size, 1])
7
8     # 词向量矩阵，初始时为均匀随机正态分布
9     self.embedding_dict = tf.Variable(
10         tf.random_uniform([self.vocab_size, self.embedding_size], -1.0, 1.0)
11     )
12
13     # 模型内部参数矩阵，初始为截断正态分布
14     self.nce_weight = tf.Variable(tf.truncated_normal([self.vocab_size, self.embedding_size],
15         stddev=1.0/math.sqrt(self.embedding_size)))
16     self.nce_biases = tf.Variable(tf.zeros([self.vocab_size]))
17
18     # 将输入序列向量化，具体可见我的【常用函数说明】那一篇
19     embed = tf.nn.embedding_lookup(self.embedding_dict, self.train_inputs) # batch_size
20
21     # 得到NCE损失(负采样得到的损失)
22     self.loss = tf.reduce_mean(
23         tf.nn.nce_loss(
24             weights = self.nce_weight, # 权重
25             biases = self.nce_biases, # 偏差
26             labels = self.train_labels, # 输入的标签
27             inputs = embed, # 输入向量
28             num_sampled = self.num_sampled, # 负采样的个数
29             num_classes = self.vocab_size # 类别数目
30         )
31     )
32
33     # tensorboard 相关
34     tf.scalar_summary('loss',self.loss) # 让tensorflow记录参数
35
36     # 根据 nce_loss 来更新梯度和embedding，使用梯度下降法(gradient descent)来实现
37     self.train_op = tf.train.GradientDescentOptimizer(learning_rate=0.1).minimize(self.loss) # 训练操作
38
```



### 博主专栏



深度学习  
&tensorflow笔记  
13  
(<http://blog.csdn.net/column/details/13414.html>)




自然语言处理  
(<http://blog.csdn.net/column/details/13461.html>)  
118343  
(<http://blog.csdn.net/column/details/13461.html>)


内容举报

返回顶部


### 达人课




电脑租赁




达内的




气垫bb怎么样




达内证书有用吗




199元智能手机




排名第一的手机




世界十大邮轮



泰国佛牌不要碰



带摄像头的眼镜



带投影的手机



tensorflow笔记：流程，概念和简单代码



达内证书有用吗



199元智能手机



排名第一的手机



世界十大邮轮



泰国佛牌不要碰



带摄像头的眼镜



带投影的手机

内容举报

返回顶部

```
30
31
32
33
34
35
36
37
38
39 # 计算与指定若干单词的相似度
40 self.test_word_id = tf.placeholder(tf.int32,shape=(None))
41 vec_l2_model = tf.sqrt( # 求各词向量的L2模
42
43     tf.reduce_sum(tf.square(self.embedding_dict),1,keep_dims=True)
44 )
45
46
47
48 avg_l2_model = tf.reduce_mean(vec_l2_model)
49 tf.scalar_summary('avg_vec_model',avg_l2_model)
50
51
52
53 self.normed_embedding = self.embedding_dict / vec_l2_model
54 # self.embedding_dict = norm_vec # 对embedding向量正则化
55 test_embed = tf.nn.embedding_lookup(self.normed_embedding, self.test_word_id)
56 self.similarity = tf.matmul(test_embed, self.normed_embedding, transpose_b=True)
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

tensorflow笔记:使用tf来实现word2vec  
(http://blog.csdn.net/u014595019/article/details/52562159)  
26059



电脑租赁 达内的  
气垫bb怎么样 达内证书有用吗  
199元智能手机 排名第一的手机  
世界十大邮轮 泰国佛牌不要碰  
带摄像头的眼镜 带投影的手机  
广告

# 外围代码

外围代码其实有很多，例如训练过程中变量的记录，模型的保存与读取等等，不过这与训练本身没什么关系，这里还是贴如何将句子转化成输入集和标签集的代码。对其他方面感兴趣的看官可以到github (https://github.com/multiangle/tfword2vec)上看完整的代码。

```
1 def train_by_sentence(self, input_sentence=[]):
2     # input_sentence: [sub_sent1, sub_sent2, ...]
3     # 每个sub_sent是一个单词序列，例如['这次','大选','让']
4     sent_num = input_sentence.__len__()
5     batch_inputs = []
6     batch_labels = []
7     for sent in input_sentence: # 输入有可能是多个句子，这里每个循环处理一个句子
8         for i in range(sent.__len__()): # 处理单个句子中的每个单词
9             start = max(0,i-self.win_len) # 窗口为 [-win_len,win_len],总计长2*win_len+1
10            end = min(sent.__len__(),i+self.win_len+1)
11            # 将某个单词对应窗口中的其他单词转化为id计入label，该单词本身计入input
12            for index in range(start,end):
13                if index == i:
14                    continue
15            label =
```




电脑租赁 达内的  
气垫bb怎么样 达内证书有用吗  
199元智能手机 排名第一的手机  
世界十大邮轮 泰国佛牌不要碰  
带摄像头的眼镜 带投影的手机  
广告

```
15         continue
16         input_id = self.word2id.get(sent[i])
17         label_id = self.word2id.get(sent[index])
18         if not (input_id and label_id): # 如果单词不在词典中，则跳过
19             continue
20         batch_inputs.append(input_id)
21         batch_labels.append(label_id)
22     if len(batch_inputs)==0: # 如果标签集为空，则跳过
23         return
24     batch_inputs = np.array(batch_inputs,dtype=np.int32)
25     batch_labels = np.array(batch_labels,dtype=np.int32)
26     batch_labels = np.reshape(batch_labels,[batch_labels.__len__(),1])
27
28     # 生成供tensorflow训练用的数据
29     feed_dict = {
30         self.train_inputs: batch_inputs,
31         self.train_labels: batch_labels
32     }
33     # 这句操控tf进行各项操作。数组中的选项，train_op等，是让tf运行的操作，feed_dict选项用来输入数据
34     _, loss_val, summary_str = self.sess.run([self.train_op,self.loss,self.merged_summary_op], feed_dict=feed_dict)
35
36     # train loss，记录这次训练的loss值
37     self.train_loss_records.append(loss_val)
38     # self.train_loss_k10 = sum(self.train_loss_records)/self.train_loss_records.__len__()
39     self.train_loss_k10 = np.mean(self.train_loss_records) # 求loss均值
40     if self.train_sents_num % 1000 == 0:
41         self.summary_writer.add_summary(summary_str,self.train_sents_num)
42
43     print("(a) sentences dealed, loss: (b)"
44           .format(a=self.train_sents_num,b=self.train_loss_k10))
45
46     # train times
47     self.train_words_num += batch_inputs.__len__()
48     self.train_sents_num += input_sentence.__len__()
49     self.train_times_num += 1
```


版权声明：本文为博主原创文章，未经博主允许不得转载。

本文已收录于以下专栏：深度学习&tensorflow笔记 (http://blog.csdn.net/column/details/13414.html)

自然语言处理 (http://blog.csdn.net/column/details/13461.html)

 发表你的评论

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

-  qq\_35725871 (/qq\_35725871)

qq\_35725871 (/qq\_35725871) 2017-09-18 15:24 9楼
-  shevchenkoNIIT (/shevchenkoNIIT)

(/shevchenkoNIIT) 2017-08-24 14:51 8楼
-  weixin\_39599665 (/weixin\_39599665)

(/weixin\_39599665) 2017-08-22 19:24 7楼




气垫bb怎么样 达内证书有用吗


199元智能手机 排名第一的手机

世界十大邮轮 泰国佛牌不要碰

带摄像头的眼镜 带投影的手机

广告

 内容举报

 返回顶部



相关文章推荐

TensorFlow实现word2vec 详细代码解释 (http://blog.csdn.net/nanjifengzi/article/details/71...

参考1 : http://blog.csdn.net/mylove0414/article/details/69789203 参考2 : 《TensorFlow实战》 参考3 : http://www.jians...


 nanjifengzi (http://blog.csdn.net/nanjifengzi)

2017年05月09日 18:58


 2258

从零学习word2vect到Tensorflow (一) (http://blog.csdn.net/chinwuforwork/article/detail...

最近实习领导给安排了任务，由于业务需要，要求我学习word2vect和Tensorflow。基于此，将看到的好的博客，和学习心得分享给大家。 一、word2vect基础 原博地址：ht...

 chinwuforwork (http://blog.csdn.net/chinwuforwork)

2016年09月08日 11:26

 5025



2017年前端报告：程序员薪酬上涨70%！

前端程序员的薪酬曝光，2017年，平均上涨70%，月薪20的人最为常见！以下为详细数据.....

(http://www.baidu.com/cb.php?c=lgF\_pyfqnhmknj0dP1f0IZ0qnfK9ujYzP1nYPH0k0Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1dbnhDkuHf3uHIBrj0znvc40AwY5HDdnHndPj6vn1f0lgF\_5y9YIZ0lQzq-uZR8mLPbUB48ugfEIAqspynElvNBnHqdlAdxTvqdThP-5yF\_UvTkn0KzujYk0AFV5H00TZcqn0KdpyfqnhRLPjrnvnfKEpyfqhHc4rj6kP0KWpyfqP1cvtHnz0AqLUWYs0ZK45HcsP6KWThnqrH0Lrf)

TensorFlow word2vec\_basic 代码分析 (http://blog.csdn.net/luozirong/article/details/7327...

将字母向量化（即 word embedding）是NLP的基础，在TensorFlow中有一个简单的实现即word2vec（这篇文章主要关注他的基本实现,demo在这个目录下：tensorflow/e...


 luozirong (http://blog.csdn.net/luozirong)

2017年06月15日 11:35


 1078

tensorflow 0.10 word2vec 源码解析 (http://blog.csdn.net/u012436149/article/details/52848...

关于word2vec 的解释见word2vec的数学原理。 本代码主要是实现了skip-gram模型，通过神经网络，对概率进行建模（概率模型中的最大似然，其实就是神经网络中的梯度下降）# Copyr...


 u012436149 (http://blog.csdn.net/u012436149)

2016年10月18日 15:20


 5009

Python Tensorflow下的Word2Vec代码解释 (http://blog.csdn.net/mydear\_11000/article/det...

作为一个深度学习的重度狂热者，在学习了各项理论后一直想通过项目练手来学习深度学习的框架以及结构用在实战中的知识。心愿是好的，但机会却不好找。最近刚好有个项目，借此机会练手的过程中，我发现其实各大机器学习...

 mydear\_11000 (http://blog.csdn.net/mydear\_11000)

2016年10月10日 11:35

 3021



区块链与程序员：赚钱还是创业

比特币和区块链是信息技术发展的产物，只有程序猿能够深入到代码层。与其他人相比，程序员们对区块链理解更深刻，投资更可能成功。你将获得巨大的认知快感：还能这么玩啊，为啥我没有想到！

(http://www.baidu.com/cb.php?c=lgF\_pyfqnhmknjTYPHR0IZ0qnfK9ujYzP1f4Pjn10Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1YvuAN9nhc4njwbPH-WmW640AwY5HDdnHndPj6vn1R0lgF\_5y9YIZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq\_lyd8Q1N9nyuWrAcYnWf5HDznHNWmhkEusKzujYk0AFV5H00TZcqn0KdpyfqnhRLPjrnvnfKEpyfqhHnsnj0YnsKWpyfqP1cvtHnz0AqLUWYs0ZK



- 电脑租赁
- 达内的
- 气垫bb怎么样
- 达内证书有用吗
- 199元智能手机
- 排名第一的手机
- 世界十大邮轮
- 泰国佛牌不要碰
- 带摄像头的眼镜
- 带投影的手机

1

内容举报



返回顶部



内容举报




返回顶部



- 电脑租赁
- 达内的
- 气垫bb怎么样
- 达内证书有用吗
- 199元智能手机
- 排名第一的手机
- 世界十大邮轮
- 泰国佛牌不要碰


tensorflow笔记:使用tf来实现word2vec (http://blog.csdn.net/u013378306/article/details/62...

时隔若干个月，又绕到了word2vec。关于word2vec的原理我就不叙述了，具体可见word2vec中的数学，写的非常好。 我后来自己用Python实现了一遍word2vec,过程写在自己动手...

 u013378306 (http://blog.csdn.net/u013378306) 2017年03月16日 16:56 1354


TensorFlow实战11：实现Word2Vec (http://blog.csdn.net/Felaim/article/details/70160667)

1.Word2Vec简介Word2Vec也称Word Embeddings，中文比较常见的叫法是“词向量”或者是“词嵌入”。通俗的来说就是把单词进行编码，变成数字的形式让计算机知道那个单词的代号。哈哈...

 Felaim (http://blog.csdn.net/Felaim) 2017年04月13日 18:15 1810


tensorflow学习笔记之word2vec (http://blog.csdn.net/zhangjcsd/article/details/52945540)

Intrduction word2vec是训练词向量的方法，tensorflow自带的word2vec代码实现的是Skip-gram方法，即使用周边词预测中心词。 代码比较简单，BP的一个前向后向过程...

 zhangjcsd (http://blog.csdn.net/zhangjcsd) 2016年10月27日 14:45 3543


Word2Vec (Part 2): NLP With Deep Learning with Tensorflow (CBOW) (http://blog.csdn.n...

Tensorflow上其实本来已经有word2vec的代码了，但是我第一次看的时候也是看得云里雾里，还是看得不太明白。并且官方文档中只有word2vec的skip-gram实现，所以google了一下...

 linuxheng (http://blog.csdn.net/linuxheng) 2017年04月14日 12:01 541


自己动手写word2vec (四):CBOW和skip-gram模型 (http://blog.csdn.net/u014595019/article...

CBOW和skip-gram应该可以说算是word2vec的核心概念之一了。这一节我们就来仔细的阐述这两个模型。其实这两个模型有很多的相通之处，所以这里就以阐述CBOW模型为主，然后再阐述skip-g...

 u014595019 (http://blog.csdn.net/u014595019) 2016年07月18日 16:54 34265

TensorFlow-9-词的向量表示 (http://blog.csdn.net/aliceyangxi1987/article/details/72566754)

今日资料： https://www.tensorflow.org/tutorials/word2vec 中文版： http://wiki.jikexueyuan.com/project/tens...

 aliceyangxi1987 (http://blog.csdn.net/aliceyangxi1987) 2017年05月19日 23:02 7197


Word2Vec (Part 2): NLP With Deep Learning with Tensorflow (CBOW) (http://blog.csdn.n...

Tensorflow上其实本来已经有word2vec的代码了，但是我第一次看的时候也是看得云里雾里，还是看得不太明白。并且官方文档中只有word2vec的skip-gram实现，所以google了一下...

 xiaocong1990 (http://blog.csdn.net/xiaocong1990) 2017年06月15日 17:04 177


经典的把一篇英文文章转成word2id形式的dict的一段python程序 (http://blog.csdn.net/guoto...

import collectionsimport tensorflow as tf def \_read\_words(filename): with tf.gfile.GFile(filename,...

 guotong1988 (http://blog.csdn.net/guotong1988) 2017年03月02日 14:10 849

统计SQuAD的词汇得到word2id 并把词都转成id的python代码 (http://blog.csdn.net/guotong...

import json import collectionsjson\_file = open("train-v1.1.json") data = json.load(json\_file)all\_wor...

 guotong1988 (http://blog.csdn.net/guotong1988) 2017年07月03日 20:11 260

带摄像头的眼镜 带投影的手机



内容举报

返回顶部



电脑租赁 达内的

气垫bb怎么样 达内证书有用吗  
199元智能手机 排名第一的手机  
世界十大邮轮 泰国佛牌不要碰  
带摄像头的眼镜 带投影的手机





再谈word2vec (http://blog.csdn.net/jj12345jj198999/article/details/27352659)

之前写过一篇博文介绍如何使用word2vec，最近老板让我讲一讲word2vec，显然光讲word2vec的使用是不够的，更重要的是介绍原理。 背景介绍 Word2vec是google在2013年...

jj12345jj198999 (http://blog.csdn.net/jj12345jj198999) 2014年05月28日 17:17 73

【Spark Mlib】TF-IDF&Word2Vec——文本相似度 (http://blog.csdn.net/u011239443/article...)

http://blog.csdn.net/u011239443/article/details/517286591 从数据中抽取合适的特征1.1 TF-IDF短语加权表示 TF-IDF公式的含义是...

u011239443 (http://blog.csdn.net/u011239443) 2016年06月21日 16:59 7303

自己动手写word2vec (一):主要概念和流程 (http://blog.csdn.net/u014595019/article/details/...)

word2vec 是 Google 于 2013 年开源推出的一个用于获取词向量（word vector）的工具包，它简单、高效，因此引起了很多人的关注。我在看了@peghoty所写的《word2ve...

u014595019 (http://blog.csdn.net/u014595019) 2016年07月11日 22:35 17005

word2vec原理及实现 (http://blog.csdn.net/FANGPINLEI/article/details/52200832)

word2vec，字词的向量空间模型依靠将语意相近的词语聚在一起来提高自然语言处理的表现。...

FANGPINLEI (http://blog.csdn.net/FANGPINLEI) 2016年08月13日 18:24 15021

word2vec使用指导 (http://blog.csdn.net/zhoubi668/article/details/24314769)

word2vec是一个将单词转换成向量形式的工具。可以把对文本内容的处理简化为向量空间中的向量运算，计算出向量空间上的相似度，来表示文本语义上的相似度。 一、理论概述（主要来源于http://l...

zhoubi668 (http://blog.csdn.net/zhoubi668) 2014年04月22日 16:34 131511

机器学习算法实现解析——word2vec源码解析 (http://blog.csdn.net/google19890102/article...)

在阅读本文之前，建议首先阅读“简单易学的机器学习算法——word2vec的算法原理”，掌握如下的几个概念：什么是统计语言模型 神经概率语言模型的网络结构 CBOW模型和Skip-gram模型的网络结...

google19890102 (http://blog.csdn.net/google19890102) 2017年03月16日 16:32 8994



内容举报



返回顶部



电脑租赁 达内的  
气垫bb怎么样 达内证书有用吗  
199元智能手机 排名第一的手机  
世界十大邮轮 泰国佛牌不要碰  
带摄像头的眼镜 带投影的手机

广告



内容举报



返回顶部