加法的attenion简单的实现

原创 2017年12月08日 17:03:00 16

一般Attention可以用于seq2seq中，也可以就用于简单的rnn模型中，下面介绍用于简单的rnn模型+attenion机制实现,流程答题可以分为：

rnn部分，假如模型输入是一个这样tensor：
rnn_outputs=[batch,seq_len,hidden_size]

attention初始化3个tensor：

w=[hidden_size,atten_dim]

b=[atten_dim]

u=[atten_dim]

attenion逻辑实现:

v=tanh(rnn_outs*w+b)
vu=v*u
exps=exp(vu)

alphas=exps/sum(exps)

output=rnn_outputs*alphas #如果用于文本分类，则要对seq_len这个维度进行sum，如果用于其他如词性标注则不要

output=rnn_outputs *alphas *如果用了文本分类，则要对seq_len这个维度进行sum，如果用了其他如词性标注则不需要

代码实现：

```
import tensorflow as tf

batch_size=2
seq_len=7
hidd_size=12
attention_dim=10

label=tf.one_hot([0,2],10,1,0)

rnn_outputs=tf.get_variable(name='output',shape=[batch_size,seq_len,hidd_size],
dtype=tf.float32)
# Attention mechanism
sequence_length = rnn_outputs.shape[
1].value # the length of sequences processed in the antecedent RNN layer
hidden_size = rnn_outputs.shape[2].value # hidden size of the RNN laye
W = tf.Variable(
tf.truncated_normal([hidden_size, attention_dim],
stddev=0.1), name="W"
)
print("w shape is:",W.get_shape()) #(12, 10)
b = tf.Variable(tf.random_normal([attention_dim], stddev=0.1),
name="b")
print("b shape is:",b.get_shape()) #(10,)
u = tf.Variable(tf.random_normal([attention_dim], stddev=0.1),
name="u")
print("u shape is:",u.get_shape()) #(10,)
v = tf.tanh(tf.matmul(tf.reshape(rnn_outputs, [-1, hidden_size]), W) + tf.reshape(b, [1, -
print("v shape is:",v.get_shape()) #(14, 10)
vu = tf.matmul(v, tf.reshape(u, [-1, 1]))
print("vu shape is:",vu.get_shape()) #(14, 1)
exps = tf.reshape(tf.exp(vu), [-1, sequence_length])
print("exps shape is:",exps.get_shape()) #(2, 7)
alphas = exps / tf.reshape(tf.reduce_sum(exps, 1), [-1, 1])
print("alphas shape is:",alphas.get_shape())# (2, 7)

# Output of Bi-gru is reduced with attention vector
print("reshape:",tf.reshape(alphas, [-1, sequence_length, 1]).get_shape())#(2, 7, 1)
print((rnn_outputs * tf.reshape(alphas, [-1, sequence_length, 1])).get_shape()) #(2, 7, 12
output = tf.reduce_sum(rnn_outputs * tf.reshape(alphas, [-1, sequence_length, 1]), 1)
print("output shape is:",output.get_shape())#(2, 12)

logits=tf.layers.dense(output,10)
logits=tf.nn.softmax(logits)
print("logits：",logits.get_shape()) #(2, 10)
cross_entropy=tf.nn.softmax_cross_entropy_with_logits(logits=logits,labels=label)
cost=tf.reduce_mean(cross_entropy)
optim=tf.train.AdamOptimizer(0.03).minimize(cost)


with tf.Session() as sess:
sess.run(tf.global_variables_initializer())
print("logists:",sess.run(logits))
for i in range(1,20):
cossst=sess.run(cost)
sess.run(optim)
```

```
if i%2==0:
print("i is {0},loss is {1}".format(i,cossst))
```

w shape is: (12, 10)
b shape is: (10,)
u shape is: (10,)
v shape is: (14, 10)
vu shape is: (14, 1)
exps shape is: (2, 7)
alphas shape is: (2, 7)
reshape: (2, 7, 1)
(2, 7, 12)
output shape is: (2, 12)
logits： (2, 10)
2017-12-08 15:26:22.009274: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could speed up CPU computations.

2017-12-08 15:26:22.009306: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are available on your machine and could speed up CPU computations.
2017-12-08 15:26:22.009314: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 instructions, but these are available on your machine and could speed up CPU computations.
2017-12-08 15:26:22.009322: W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA instructions, but these are available on your machine and could speed up CPU computations.
logists: [[0.09686147 0.08603875 0.10925905 0.11028122 0.09818964 0.09983405
0.10350803 0.10191522 0.10624968 0.08786286]
[0.13021383 0.10083188 0.08510906 0.10502309 0.09848842 0.08085372
0.09499053 0.1028346 0.10391016 0.0977447]]
i is 2,loss is 2.2936596870422363
i is 4,loss is 2.245049476623535
i is 6,loss is 2.1709165573120117
i is 8,loss is 2.0626285076141357
i is 10,loss is 1.923459768295288
i is 12,loss is 1.776179313659668
i is 14,loss is 1.6438379287719727
i is 16,loss is 1.545175552368164
i is 18,loss is 1.4922502040863037

相关文章推荐

java实现加法 简单的小例子 (http://download.csdn.net/download/bananas…

汇编的简单键盘输入的加法 (http://download.csdn.net/download/hanyuch…

2012年03月11日 12:22 1KB 下载

Qt下写一个简单的加法计算界面 (http://blog.csdn.net/weiwen12138/article/details/70741238)

Qt下写一个简单的加法计算界面 只有整数

weiwen12138 (http://blog.csdn.net/weiwen12138) 2017年04月25日 16:01 1024

简单加法计算器 (http://download.csdn.net/download/akinalive/6278487)

2013年09月16日 23:46 161B 下载

计算器 vb编写的简单加法计算器 (http://download.csdn.net/download/u01…

2015年07月16日 12:47 20KB 下载

内容举报

返回顶部