

[CSDN首页 \(http://www.csdn.net?ref=toolbar\)](http://www.csdn.net?ref=toolbar)[学院 \(http://edu.csdn.net?ref=toolbar\)](http://edu.csdn.net?ref=toolbar)[下载 \(http://download.csdn.net?ref=toolbar\)](http://download.csdn.net?ref=toolbar)

更多 ▾

[下载 CSDN APP \(http://www.csdn.net/app?ref=toolbar\)](http://www.csdn.net/app?ref=toolbar)[写博客 \(http://write.blog.csdn.net/postedit?ref=toolbar\)](http://write.blog.csdn.net/postedit?ref=toolbar)[登录 \(https://passport.csdn.net/account/login?ref=toolbar\)](http://blog.csdn.net/) | [注册 \(http://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobileRegister\)](http://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobileRegister)[首页 \(http://blog.csdn.net/\)](http://blog.csdn.net/)[移动开发 \(http://blog.csdn.net/mobile/newarticle.html\)](http://blog.csdn.net/mobile/newarticle.html)

全部 ▮

CSDN (<http://www.csdn.net>)

喜欢

tensorflow之训练数据制作



收藏

原创

2017年05月11日 22:16:10

2261

0

0



评论



分享

[TensorFlow高效读取数据的方法 \(http://blog.csdn.net/u012759136/article/details/52232266\)](http://blog.csdn.net/u012759136/article/details/52232266)

TensorFlow 学习（二）制作自己的TFRecord数据集，读取，显示及代码详解

 [\(http://blog.csdn.net/miaomiaoyuan/article/details/56865361\)](http://blog.csdn.net/miaomiaoyuan/article/details/56865361)

convert_to_records.py

[python]

```
01. """Converts MNIST data to TFRecords file format with Example protos."""
02. from __future__ import absolute_import
03. from __future__ import division
04. from __future__ import print_function
05.
06. import argparse
07. import os
08. import sys
```

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

+ 关注

 [\(http://blog.csdn.net/u014568921\)](http://blog.csdn.net/u014568921)

码云

未开通

原创

粉丝

喜欢

[\(https://github.com/\)](https://github.com/)

452

202

0

utm_source

他的最新文章

[更多文章 \(http://blog.csdn.net/u014568921\)](http://blog.csdn.net/u014568921)

机器学习中非平衡数据的处理

 [\(/u014568921/article/details/71440743\)](/u014568921/article/details/71440743)

模型融合

 [\(/u014568921/article/details/77414799\)](/u014568921/article/details/77414799)

caffe提供服务

 [\(/u014568921/article/details/77689591\)](/u014568921/article/details/77689591)

image caption

 [\(/u014568921/article/details/53188964\)](/u014568921/article/details/53188964)

返回顶部

编辑推荐

最热专栏

```

09.
10. import tensorflow as tf
11.
12. from tensorflow.contrib.learn.python.learn.datasets import mnist
13.
14. FLAGS = None
15.
16.
17. def _int64_feature(value):
18.     return tf.train.Feature(int64_list=tf.train.Int64List(value=[value]))
19.
20.
21. def _bytes_feature(value):
22.     return tf.train.Feature(bytes_list=tf.train.BytesList(value=[value]))
23.
24.
25. def convert_to(data_set, name):
26.     """Converts a dataset to tfrecords."""
27.     images = data_set.images
28.     labels = data_set.labels
29.     num_examples = data_set.num_examples
30.
31.     if images.shape[0] != num_examples:
32.         raise ValueError('Images size %d does not match label size %d.' %
33.                            (images.shape[0], num_examples))
34.     rows = images.shape[1]
35.     cols = images.shape[2]
36.     depth = images.shape[3]
37.
38.     filename = os.path.join(FLAGS.directory, name + '.tfrecords')
39.     print('Writing', filename)
40.     writer = tf.python_io.TFRecordWriter(filename)
41.     for index in range(num_examples):
42.         image_raw = images[index].tostring()
43.         example = tf.train.Example(features=tf.train.Features(feature={
44.             'height': _int64_feature(rows),
45.             'width': _int64_feature(cols),
46.             'depth': _int64_feature(depth),
47.             'label': _int64_feature(int(labels[index])),
48.             'image_raw': _bytes_feature(image_raw)}))
49.         writer.write(example.SerializeToString())

```



喜欢



收藏



评论



分享

tensorflow数据集制作/文件队列读取方...

Ubuntu启动问题以及Grub Rescue修复...

Tensorflow 训练自己的数据集（一）（...

在线课程



【免费】深入理解Docker

内部原理及网络配置

(http://edu.csdn.net/huiyi

Course/detail/563?

utm_source=blog9)



SDCC 2017之区块链技术

应用

(http://edu.csdn.net/huiyi

Course/series_detail/66?

utm_source=blog9)

```
50.     writer.close()
51.
52.
53.     def main(unused_argv):
54.         # Get the data.
55.         data_sets = mnist.read_data_sets(FLAGS.directory,
56.                                           dtype=tf.uint8,
57.                                           reshape=False,
58.                                           validation_size=FLAGS.validation_size)
59.
60.         # Convert to Examples and write the result to TFRecords.
61.         convert_to(data_sets.train, 'train')
62.         convert_to(data_sets.validation, 'validation')
63.         convert_to(data_sets.test, 'test')
64.
65.
66.     if __name__ == '__main__':
67.         parser = argparse.ArgumentParser()
68.         parser.add_argument(
69.             '--directory',
70.             type=str,
71.             default='/tmp/data',
72.             help='Directory to download data files and write the converted result'
73.         )
74.         parser.add_argument(
75.             '--validation_size',
76.             type=int,
77.             default=5000,
78.             help="""\
79.             Number of examples to separate from the training data for the validation
80.             set.\
81.             """
82.         )
83.         FLAGS, unparsed = parser.parse_known_args()
84.         tf.app.run(main=main, argv=[sys.argv[0]] + unparsed)
```



喜欢



收藏



评论



分享

fully_connected_reader.py



返回顶部



喜欢



收藏



评论



分享

[python]

```
01. #coding:utf-8
02.
03. """Train and Eval the MNIST network.
04. This version is like fully_connected_feed.py but uses data converted
05. to a TFRecords file containing tf.train.Example protocol buffers.
06. See:
07. https://www.tensorflow.org/programmers_guide/reading_data#reading_from_files
08. for context.
09. YOU MUST run convert_to_records before running this (but you only need to
10. run it once).
11. """
12. from __future__ import absolute_import
13. from __future__ import division
14. from __future__ import print_function
15.
16. import argparse
17. import os.path
18. import sys
19. import time
20.
21. import tensorflow as tf
22.
23. from tensorflow.examples.tutorials.mnist import mnist
24.
25. # Basic model parameters as external flags.
26. FLAGS = None
27.
28. # Constants used for dealing with the files, matches convert_to_records.
29. TRAIN_FILE = 'train.tfrecords'
30. VALIDATION_FILE = 'validation.tfrecords'
31.
32.
33. def read_and_decode(filename_queue):
34.     reader = tf.TFRecordReader()
35.     _, serialized_example = reader.read(filename_queue)
36.     features = tf.parse_single_example(
37.         serialized_example,
38.         # Defaults are not specified since both keys are required.
39.         features={
40.             'image_raw': tf.FixedLenFeature([], tf.string),
```



返回顶部

```

41.         'label': tf.FixedLenFeature([], tf.int64),
42.     })
43.
44.     # Convert from a scalar string tensor (whose single string has
45.     # length mnist.IMAGE_PIXELS) to a uint8 tensor with shape
46.     # [mnist.IMAGE_PIXELS].
47.     image = tf.decode_raw(features['image_raw'], tf.uint8)
48.     image.set_shape([mnist.IMAGE_PIXELS])
49.
50.     # OPTIONAL: Could reshape into a 28x28 image and apply distortions
51.     # here. Since we are not applying any distortions in this
52.     # example, and the next step expects the image to be flattened
53.     # into a vector, we don't bother.
54.
55.     # Convert from [0, 255] -> [-0.5, 0.5] floats.
56.     image = tf.cast(image, tf.float32) * (1. / 255) - 0.5
57.
58.     # Convert label from a scalar uint8 tensor to an int32 scalar.
59.     label = tf.cast(features['label'], tf.int32)
60.
61.     return image, label
62.
63.
64. def inputs(train, batch_size, num_epochs):
65.     """Reads input data num_epochs times.
66.     Args:
67.         train: Selects between the training (True) and validation (False) data.
68.         batch_size: Number of examples per returned batch.
69.         num_epochs: Number of times to read the input data, or 0/None to
70.             train forever.
71.     Returns:
72.         A tuple (images, labels), where:
73.         * images is a float tensor with shape [batch_size, mnist.IMAGE_PIXELS]
74.           in the range [-0.5, 0.5].
75.         * labels is an int32 tensor with shape [batch_size] with the true label,
76.           a number in the range [0, mnist.NUM_CLASSES).
77.         Note that an tf.train.QueueRunner is added to the graph, which
78.         must be run using e.g. tf.train.start_queue_runners().
79.     """
80.     if not num_epochs: num_epochs = None
81.     filename = os.path.join(FLAGS.train_dir,

```



喜欢



收藏



评论



分享



返回顶部



喜欢



收藏



评论



分享

```
82.         TRAIN_FILE if train else VALIDATION_FILE)
83.
84.     with tf.name_scope('input'):
85.         filename_queue = tf.train.string_input_producer(
86.             [filename], num_epochs=num_epochs)
87.
88.         # Even when reading in multiple threads, share the filename
89.         # queue.
90.         image, label = read_and_decode(filename_queue)
91.
92.         # Shuffle the examples and collect them into batch_size batches.
93.         # (Internally uses a RandomShuffleQueue.)
94.         # We run this in two threads to avoid being a bottleneck.
95.         images, sparse_labels = tf.train.shuffle_batch(
96.             [image, label], batch_size=batch_size, num_threads=2,
97.             capacity=1000 + 3 * batch_size,
98.             # Ensures a minimum amount of shuffling of examples.
99.             min_after_dequeue=1000)
100.
101.     return images, sparse_labels
102.
103.
104. def run_training():
105.     """Train MNIST for a number of steps."""
106.
107.     # Tell TensorFlow that the model will be built into the default Graph.
108.     with tf.Graph().as_default():
109.         # Input images and labels.
110.         images, labels = inputs(train=True, batch_size=FLAGS.batch_size,
111.                                 num_epochs=FLAGS.num_epochs)
112.
113.         # Build a Graph that computes predictions from the inference model.
114.         logits = mnist.inference(images,
115.                                 FLAGS.hidden1,
116.                                 FLAGS.hidden2)
117.
118.         # Add to the Graph the loss calculation.
119.         loss = mnist.loss(logits, labels)
120.
121.         # Add to the Graph operations that train the model.
122.         train_op = mnist.training(loss, FLAGS.learning_rate)
```



返回顶部



喜欢



收藏



评论



分享

```
123.
124.     # The op for initializing the variables.
125.     init_op = tf.group(tf.global_variables_initializer(),
126.                        tf.local_variables_initializer())
127.
128.     # Create a session for running operations in the Graph.
129.     sess = tf.Session()
130.
131.     # Initialize the variables (the trained variables and the
132.     # epoch counter).
133.     sess.run(init_op)
134.
135.     # Start input enqueue threads.
136.     coord = tf.train.Coordinator()
137.     threads = tf.train.start_queue_runners(sess=sess, coord=coord)
138.
139.     try:
140.         step = 0
141.         while not coord.should_stop():
142.             start_time = time.time()
143.
144.             # Run one step of the model. The return values are
145.             # the activations from the `train_op` (which is
146.             # discarded) and the `loss` op. To inspect the values
147.             # of your ops or variables, you may include them in
148.             # the list passed to sess.run() and the value tensors
149.             # will be returned in the tuple from the call.
150.             _, loss_value = sess.run([train_op, loss])
151.
152.             duration = time.time() - start_time
153.
154.             # Print an overview fairly often.
155.             if step % 100 == 0:
156.                 print('Step %d: loss = %.2f (%.3f sec)' % (step, loss_value,
157.                                                            duration))
158.                 step += 1
159.             except tf.errors.OutOfRangeError:
160.                 print('Done training for %d epochs, %d steps.' % (FLAGS.num_epochs, step))
161.             finally:
162.                 # When done, ask the threads to stop.
163.                 coord.request_stop()
```



返回顶部

```
164.
165.     # Wait for threads to finish.
166.     coord.join(threads)
167.     sess.close()
168.
169.
170. def main(_):
171.     run_training()
172.
173.
174. if __name__ == '__main__':
175.     parser = argparse.ArgumentParser()
176.     parser.add_argument(
177.         '--learning_rate',
178.         type=float,
179.         default=0.01,
180.         help='Initial learning rate.'
181.     )
182.     parser.add_argument(
183.         '--num_epochs',
184.         type=int,
185.         default=2,
186.         help='Number of epochs to run trainer.'
187.     )
188.     parser.add_argument(
189.         '--hidden1',
190.         type=int,
191.         default=128,
192.         help='Number of units in hidden layer 1.'
193.     )
194.     parser.add_argument(
195.         '--hidden2',
196.         type=int,
197.         default=32,
198.         help='Number of units in hidden layer 2.'
199.     )
200.     parser.add_argument(
201.         '--batch_size',
202.         type=int,
203.         default=100,
204.         help='Batch size.'
```



喜欢



收藏



评论



分享



返回顶部


```

205.     )
206.     parser.add_argument(
207.         '--train_dir',
208.         type=str,
209.         default='/tmp/data',
210.         help='Directory with the training data.'
211.     )
212.     FLAGS, unparsed = parser.parse_known_args()
213.     tf.app.run(main=main, argv=[sys.argv[0]] + unparsed)

```



喜欢



收藏



评论



分享

TensorFlow支持从csv文件和TFRecords文件读取数据，如果从二进制的TFRecords文件读取，可以采用QueueRunner和Coordinator的方式进行多线程读取，通过设置epoch参数控制训练数据文件迭代训练的次数，通过设置batch_size的大小来控制一次训练中从训练数据中取得的样本数量，还可以设置随机选取，有利于加快训练速度。

[python]

```

01. def read_and_decode(filename_queue):#从TFRecords中读取数据
02.     reader = tf.TFRecordReader()
03.     _, serialized_example = reader.read(filename_queue)
04.     features = tf.parse_single_example(serialized_example,
05.         features={
06.             "label": tf.FixedLenFeature([], tf.float32),
07.             "features": tf.FixedLenFeature([FEATURE_SIZE], tf.float32),
08.         })
09.     label = features["label"]
10.     features = features["features"]
11.     return label, features
12.
13.
14. filename_queue = tf.train.string_input_producer(tf.train.match_filenames_once(trainFile), num_e
15. label, features = read_and_decode(filename_queue)

```



返回顶部

```
16. batch_labels, batch_features = tf.train.shuffle_batch([label, features], batch_size=batch_size,
```

这里的trainFile可以是一个文件名的列表：



喜欢

```
trainFile = ['./data/train_1.tfrecords', './data/train_2.tfrecords']
```

还可以是一个正则表达式：



收藏

```
trainFile = './data/*.tfrecords'
```



评论

使用Coordinator来管理队列：



分享

```
[python]
01. coord = tf.train.Coordinator()
02. threads = tf.train.start_queue_runners(coord=coord, sess=sess)
03. try:
04.     while not coord.should_stop():
05.         _, loss_value, step = sess.run([train_op, loss, global_step])
06.         saver.save(sess, "./checkpoint/checkpoint.ckpt", global_step=step)
07. except tf.errors.OutOfRangeError:
08.     print("Done training after reading all data")
09. finally:
10.     coord.request_stop()
```

这里经常会碰到的一个问题是在没有训练之前队列就关闭了，类似“get 'OutOfRange', the queue will be closed”的问题，这是因为epoch设置过小，在开始训练前就把数据读完退出了，可以把epoch设置的大一



返回顶部

些，如果设置成Nnoe，程序会无限制地一直跑下去，当然你可以在结果足够好的时候手动中断程序的运行。这里就是我的问题啦，有没有什么好的方法来设置epoch参数？

Notes on tensorflow（七）将数据集转换为TFRecord

(http://blog.csdn.net/weixin_35653315/article/details/71015845)

Notes on tensorflow（八）read tfrecords with slim

(http://blog.csdn.net/weixin_35653315/article/details/71023596)



tensorflow学习笔记（四十二）：输入流水线 (<http://blog.csdn.net/u012436149/article/details/72353313>)

喜欢 TensorFlow数据读取方法 (<http://honggang.io/2016/08/19/tensorflow-data-reading/>)



Tensorflow中使用tfrecord方式读取数据 (<http://blog.csdn.net/u010358677/article/details/70544241>)

tensorflow中next_batch (<http://blog.csdn.net/appleml/article/details/57413615>)

收藏

Tensorflow之构建自己的图片数据集TFrecords (<http://blog.csdn.net/csuzhaoqinghui/article/details/51377941>)



【TensorFlow动手玩】数据导入2 (<http://blog.csdn.net/shenxiaolu1984/article/details/52857437>)

评论 自己用到的TensorFlow的几种输入 (<http://blog.csdn.net/jyshee/article/details/52566152>)



深度学习之图片转成pkl格式的训练集 (<http://blog.csdn.net/csuzhaoqinghui/article/details/54346637>)

学习TensorFlow，生成tensorflow输入输出的图像格式 (<http://blog.csdn.net/helei001/article/details/51354404>)

分享

Tenforflow之构建自己的cpkt训练模型 (<http://blog.csdn.net/csuzhaoqinghui/article/details/51839169>)

tensorflow载入数据的三种方式 (<http://blog.csdn.net/lujiandong1/article/details/53376802>)

<https://github.com/kevin28520/My-TensorFlow-tutorials/tree/master/03%20TFRecord>

tf.train.shuffle_batch (https://www.tensorflow.org/api_docs/python/tf/train/shuffle_batch)

版权声明：

举报



返回顶部

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



喜欢



收藏



评论





分享

相关文章推荐



tensorflow数据集制作/文件队列读取方式 (/continueoo/article/details/72236510)

TensorFlow程序读取数据3种方法: tensorflow数据集制作

 continueOo (<http://blog.csdn.net/continueOo>) 2017-05-16 10:45  259

Ubuntu启动问题以及Grub Rescue修复方法 (/xixinyan/article/details/6969469)

Ubuntu启动问题以及Grub Rescue修复方法 问题：之前系统是Windows7 64bit（C盘）+ D，E盘（都是NTFS）+ Ubuntu。今天，在...

 xixinyan (<http://blog.csdn.net/xixinyan>) 2011-11-14 17:22  0



精选：深入理解 Docker 内部原理及网络配置 (http://edu.csdn.net/huiyiCourse/detail/563?utm_source=blog10)

网络绝对是任何系统的核心，对于容器而言也是如此。Docker 作为目前最火的轻量级容器技术，有很多令人称道的功能，如 Docker 的镜像管理。然而，Docker的网络一直以来都比较薄弱，所以我们有必要深入了解Docker的



返回顶部

网络知识，以满足更高的网络需求。

Tensorflow 训练自己的数据集（一）（数据直接导入到内存） (/best_coder/article/details/70141075)

制作自己的训练集下图是我们数据的存放格式，在data目录下有验证集与测试集分别对应iris_test，iris_train 为了向伟大的MNIST致敬，我们采用的数据名称格式和MNIST类似 ...



喜欢

Best_Coder (http://blog.csdn.net/Best_Coder) 2017-04-12 11:32 2377



收藏

SqueezeNet (/u014568921/article/details/72565976)



评论

深度学习方法（七）：最新SqueezeNet 模型详解，CNN模型参数降低50倍，压缩461倍！



u014568921 (<http://blog.csdn.net/u014568921>) 2017-05-19 21:00 239

分享

TensorFlow——训练自己的数据（四）模型测试 (/xinyu3307/article/details/75008458)

测试一张图片获取一张图片函数：def get_one_image(train): 输入参数：train,训练图片的路径 返回参数：image，从训练图片中随机抽取一张图片 n = len(train)...

xinyu3307 (<http://blog.csdn.net/xinyu3307>) 2017-07-12 12:25 539



TensorFlow——训练自己的数据（三）模型训练 (/xinyu3307/article/details/74979842)

参考：Tensorflow教程-猫狗大战数据集 文件training.py 导入文件 import os import numpy as np import tensorflow as tf im...

xinyu3307 (<http://blog.csdn.net/xinyu3307>) 2017-07-11 16:45 1113

TensorFlow——训练自己的数据——CIFAR10（一）数据准备 (/xinyu3307/article/details/77072789)

参考教程:Tensorflow教程：深度学习 图像分类 CIFAR10数据集Reading Data 所用函数def read_cifar10(data_dir, is_train, batch_s...

 xinyu3307 (<http://blog.csdn.net/xinyu3307>) 2017-08-10 20:06  207



喜欢

白平衡 (/ken_yjj/article/details/5593779)



白平衡是摄像领域一个非常重要的概念，通过它可以解决色彩还原和色调处理的一系列问题。白平衡是随着电子影像再现色彩真实而产生的，在专业摄像领域白平衡应用的较早，现在家用电子产品（家用摄像机、数码照相机）中...



ken_yjj (http://blog.csdn.net/ken_yjj) 2010-05-14 23:15  933



评论



tensorflow之训练数据制作 (/u014568921/article/details/71566454)


分享

TensorFlow高效读取数据的方法 TensorFlow 学习（二）制作自己的TFRecord数据集，读取，显示及代码详解

 u014568921 (<http://blog.csdn.net/u014568921>) 2017-05-11 22:16  2262



TensorFlow 训练 MNIST 数据（二） (/willduan1/article/details/52036831)

上一篇博客讲了一个简单的基于 SoftMax 回归的学习模型，准确率大概在91%左右，这篇构建一个深度卷积神经网络。主要的教程还是来自于极客学院，但是讲的很琐碎，我把自己整理的思路和最后写的完整的代码...

 willduan1 (<http://blog.csdn.net/willduan1>) 2016-07-26 19:46  9098

TensorFlow入门之训练mnist数据集 (/rongrongyaofeiqi/article/details/65445830)

```
import sys,os
import numpy as np
import tensorflow as tf
from tensorflow.examples.tutorials.mnist import im...
```



 rongrongyaofeiqi (<http://blog.csdn.net/rongrongyaofeiqi>) 2017-03-23 17:06  246

TensorFlow 入门之训练 MNIST 数据 (/willduan1/article/details/52024254)

学习深度神经网络一段时间了，把当初跑的代码贴上来，这篇代码是跟着极客学院的文档做的，我把指导重新整理和汇总了一下下面贴出来。主要有以下几步：1. 新建module，我命名为my_tensorflo...



喜欢

 willduan1 (<http://blog.csdn.net/willduan1>) 2016-07-25 15:51  13322



收藏

TensorFlow个人学习（训练 MNIST 数据 ）(/q17686617156/article/details/73024381)





评论

（看了TensorFlow中文社区和极客学院的教程之后，来总结下TensorFlow的例子mnist）新建input_data.py，用于下载 MNIS数据集并且当做输入，具体见下面代码:...



分享

 q17686617156 (<http://blog.csdn.net/q17686617156>) 2017-06-11 12:16  73



TensorFlow 组合训练数据（batching）(/chaipp0607/article/details/73016068)

在之前的文章中我们提到了TensorFlow TensorFlow 队列与多线程的应用以及TensorFlow TFRecord数据集的生成与显示，通过这些操作我们可以得到自己的TFRecord文件，...

 chaipp0607 (<http://blog.csdn.net/chaipp0607>) 2017-06-11 14:35  586

Tensorflow训练mnist数据（完整版）(/u013841196/article/details/76403302)

重构之后的代码会被拆分成3个程序，第一个是mnist_inference.py,它定义了前向传播的过程以及神经网络中的参数。第二个程序是mnist_train.py，它定义了神经网络的训练过程。第三个...

 u013841196 (<http://blog.csdn.net/u013841196>) 2017-07-30 19:51  41



返回顶部

TensorFlow Object Detection API 训练VOC2012数据集 (/weixin_35654926/article/details/75578324)

安装好Tensorflow 和 Tensorflow Object Detection API 下载好VOC2012数据集1.准备训练集Tensorflow对象检测API使用TFRecord文件格式...

 weixin_35654926 (http://blog.csdn.net/weixin_35654926) 2017-07-20 21:31  754



喜欢

tensorflow实现AlexNet训练mnist数据 (/qq_23926575/article/details/71310366)



一、AlexNet网络介绍 AlexNet 可以说是具有历史意义的一个网络结构，可以说在AlexNet之前，深度学习已经沉寂了很久。历史的转折在2012年到来，Alex Net 在当年...



qq_23926575 (http://blog.csdn.net/qq_23926575) 2017-05-07 11:59  613

评论



TensorFlow 研究实践二 (/forest_world/article/details/51321197)

分享

学习文献资料：《TensorFlow 官方文档中文版 - v1.2》摘录训练 TensorFlow 神经网络模型:learning@learning-virtual-machine:~/tensorflow...

 forest_world (http://blog.csdn.net/forest_world) 2016-05-05 11:15  4246