

360doc 个人图书馆

首页 阅览室 馆友 我的图书馆 搜文章 找馆友

登录 注册

和你说过的话，就是力量
顺着你的声音，我会努力生活
你要幸福！
手写

黄金线路直播

dnn神经网络

手写平板电脑

python趣

霸业传奇

pyth on环

原油直播间

TensorFlow人工智能引擎入门教程之四 TensorBoard面板可视化管理

基陆伯 图书馆 11495 馆藏 33889

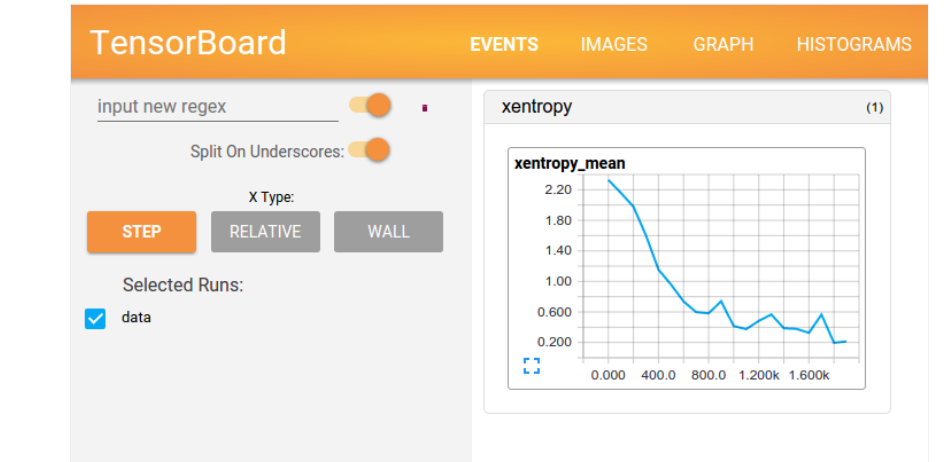
2016-05-15 基陆伯 阅 5

分享： 微信 转藏到我的图书馆

在tensorflow中自带了tensorboard面板 可以 让tensorflow 训练过程可视化显示

首先我们先看看我们上面的alexnet模型我们怎么添加tensorboard相关的op操作 添加到s session会话中，让给该op收集训练过程数据 写入日志，收集汇总起来用来图表可视化显示

官方文档 https://www.tensorflow.org/versions/r0.8/how_tos/summaries_and_tensorboard/index.html



1.tensorflow 网络 可视化，操作，对于每一步op 形成的无向图，我们可以使用tensorfl ow可视化，因为tensorflow默认每一次会话有一个默认的graph对象 graph.ref

所以

```
summary_writer = tf.train.SummaryWriter('tmp/tensorflowlogs', graph_def=sess.graph_def)
```

如果我们需要记录 图标训练时候的参数

我们需要 添加相应的op操作，上面的sumary_writer操作是复杂 将来训练时候的op 过程记录到日志里面，便于后面显示graph

因为变量很多，所以可以定义scope范围来管理，节点树一样展开查看，文档上有 很清晰，这里就不解释了，下面介绍个方法 记录 op 以及 汇总op的方法

1.4D数据虚拟化 tensorflow 比如CNN每一步生成的特征，也就是用图像虚拟化ndarry的特征，比如我们可以看到每一步cnn得到的特征是什么。比如下图 车子 每一步得到的CNN特征 用图像方式虚拟化显示image_summary

```
images = np.random.randint(256, size=shape).astype(np.uint8) tf.image_summary("Visualize_image", images)
```

TA的推荐 TA的最新馆藏

永远成功的秘密，就是每天淘汰自己
我们将永生还是灭绝？人工智能很...
我们将永生还是灭绝？人工智能很...
[转] 赞美的大能
他们还不信我要到几时呢？
基督徒的委身 【】

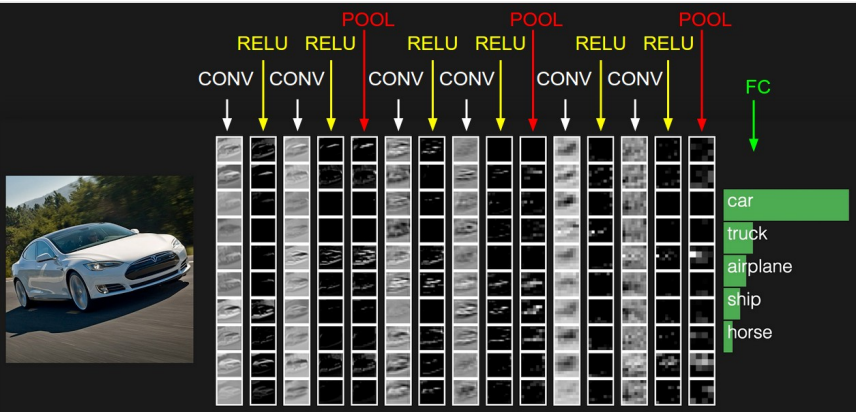


推荐阅读 更多

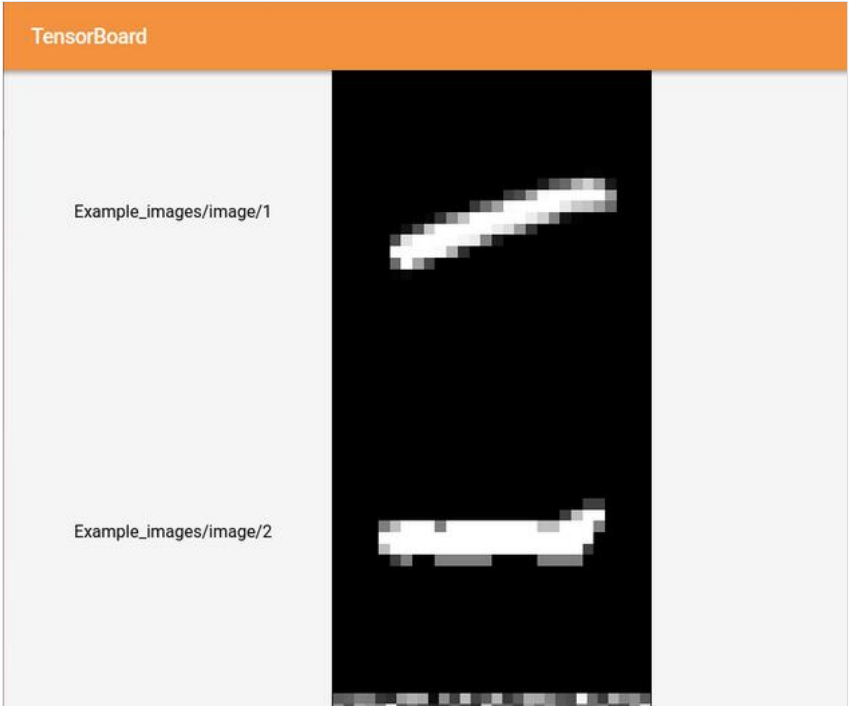
BetaCat 的前生后世
揪出bug！解析调试神经网络的技巧
深度学习计算模型中 “门函数（Ga...
简易的深度学习框架Keras代码解析...
国外公司开发新型移动无线网pCell...
enum的用法
再谈：义和团史实（转）
是还没有受洗，还没有正式参加某...
帧缓存



- | | |
|----------|------------|
| 1 美亚保险官网 | 7 北京口腔医院 |
| 2 美亚保险 | 8 企业邮箱 |
| 3 公司邮箱 | 9 用英语介绍美国 |
| 4 企业邮箱注册 | 10 led亮化照明 |
| 5 中老年妈妈装 | 11 企业邮箱申请 |
| 6 英语学习 | 12 钱爸爸理财 |



展示的样子大概

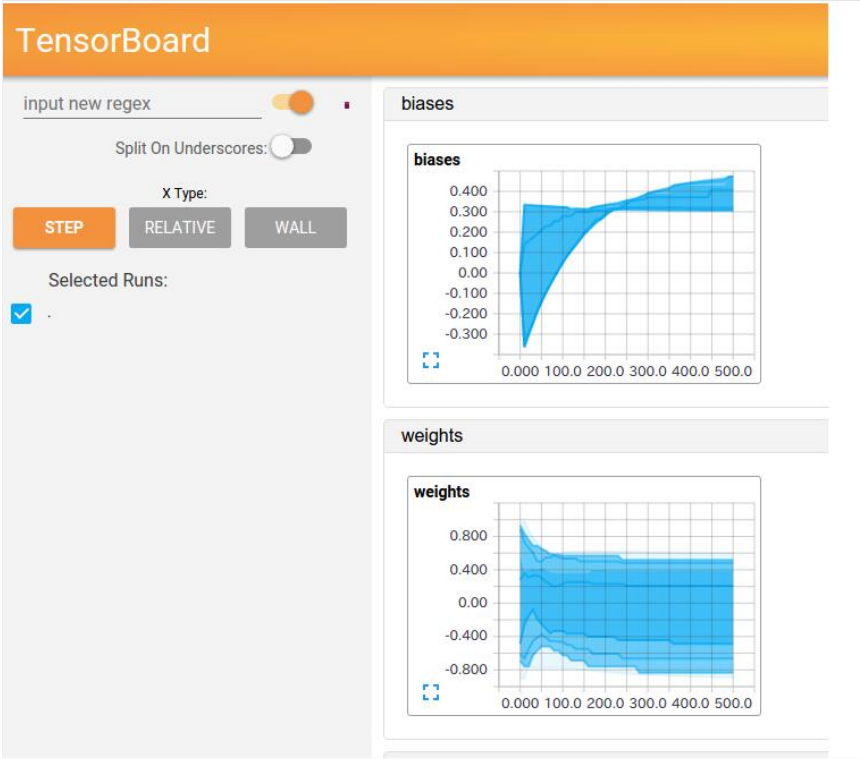


2.图标变化方式展示tensorflow数据特征

```
w_hist = tf.histogram_summary("weights", W) b_hist = tf.histogram_summary("biases", b) y_hist = tf.histogram_summary("y", y)
```

大概下面这个样子



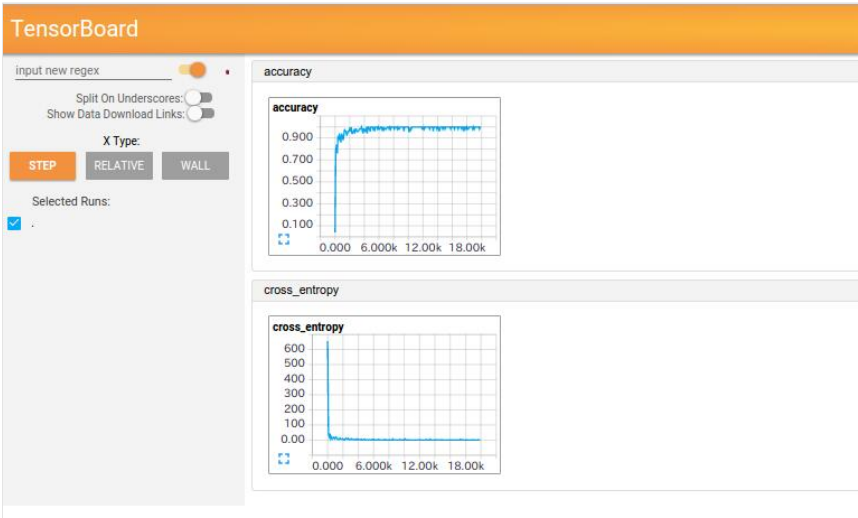


3.以变量批次事件变化的线路图显示表示tensorflow数据

比如下面记录精度的变化曲线图

```
accuracy_summary = tf.scalar_summary("accuracy", accuracy)
```

展示的样子大概这么个样子



因为我们上面这些op操作各种各样 多个 所以需要有个汇总的op操作

该操作会汇总上面的所有summary

```
merged = tf.merge_all_summaries()
```

最后在session中执行汇总summary操作op 并使用上面summarywriter 写入更新到graph 日志中

```
summary_all = sess.run(merged_summary_op, feed_dict={x: batch_xs, y: batch_ys}) summary_writer.add_summary(summary_all, i)
```

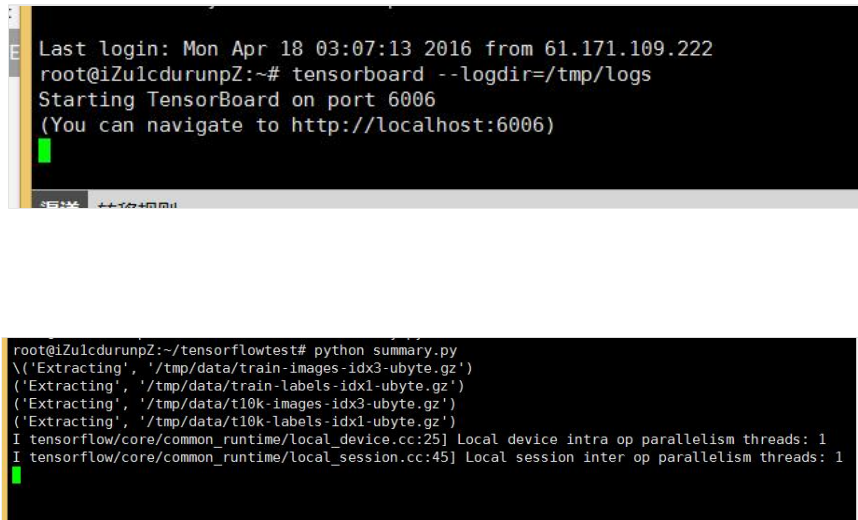
现在我们 回到 上面一篇文章的model alexnet中我们添加summary

下面看看我们分别加上了

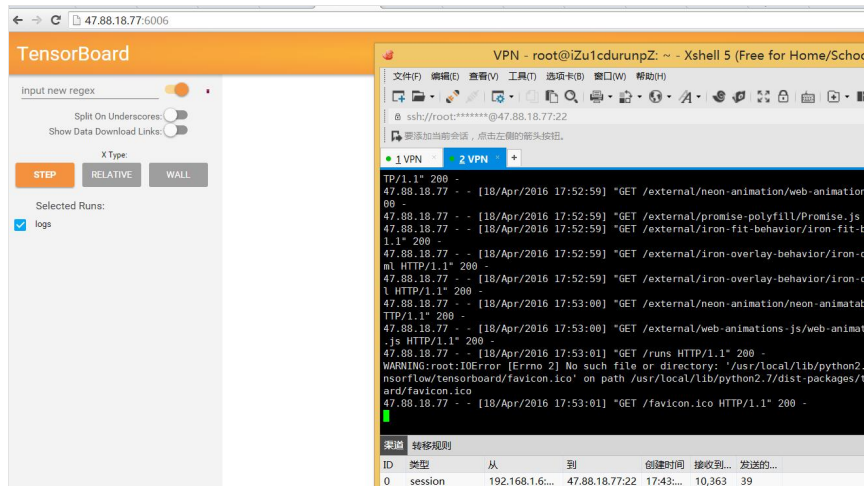
loss accuracy 的几个summary

```
import input_data mnist = input_data.read_data_sets("/tmp/data/", one_hot=True) import tensorflow as tf # Parameters learning_rate = 0.001 training_iters = 200000 batch_size = 64 display_step = 20 # Network Parameters n_input = 784 # MNIST data input (img shape: 28*28) n_classes = 10 # MNIST total classes (0-9 digits) dropout = 0.8 # Dropout, probability to keep units # tf Graph input x = tf.placeholder(tf.float32, [None, n_input]) y = tf.placeholder(tf.float32, [None, n_classes]) keep_prob = tf.placeholder(tf.float32) # dropout (keep probability) # Create custom model def conv2d(name, l_input, w, b): return tf.nn.conv2d(l_input, w, b, tf.nn.bias_add(tf.nn.conv2d(l_input, w, strides=[1, 1, 1, 1], padding='SAME'), b), name=name) def max_pool(name, l_input, k): return tf.nn.max_pool(l_input, [1, k, k, 1], stride=[1, k, k, 1], padding='SAME', name=name) def norm(name, l_input, lsize=4): return tf.nn.lrn(l_input, lsize, bias=1.0, alpha=0.001 / 9.0, beta=0.75, name=name) def customnet(X, _weights, _biases, dropout): # Reshape input picture _X = tf.reshape(X, shape=[-1, 28, 28, 1]) # Convolution Layer conv1 = conv2d('conv1', _X, _weights['wc1'], _biases['bc1']) # Max Pooling (down-sampling) pool1 = max_pool('pool1', conv1, k=2) # Apply Normalization norm1 = norm('norm1', pool1, lsize=4) # Apply Dropout norm1 = tf.nn.dropout(norm1, dropout) # conv1 image show tf.image_summary('conv1', conv1) # Convolution Layer conv2 = conv2d('conv2', norm1, _weights['wc2'], _biases['bc2']) # Max Pooling (down-sampling) pool2 = max_pool('pool2', conv2, k=2) # Apply Normalization norm2 = norm('norm2', pool2, lsize=4) # Apply Dropout norm2 = tf.nn.dropout(norm2, dropout) # Convolution Layer conv3 = conv2d('conv3', norm2, _weights['wc3'], _biases['bc3']) # Max Pooling (down-sampling) pool3 = max_pool('pool3', conv3, k=2) # Apply Normalization norm3 = norm('norm3', pool3, lsize=4) # Apply Dropout norm3 = tf.nn.dropout(norm3, dropout) # conv4 conv4 = conv2d('conv4', norm3, _weights['wc4'], _biases['bc4']) # Max Pooling (down-sampling) pool4 = max_pool('pool4', conv4, k=2) # Apply Normalization norm4 = norm('norm4', pool4, lsize=4) # Apply Dropout norm4 = tf.nn.dropout(norm4, dropout) # Fully connected layer dense1 = tf.reshape(norm4, [-1, _weights['wd1'].get_shape().as_list()[0]]) # Reshape conv3 output to fit dense layer input dense1 = tf.nn.relu(tf.matmul(dense1, _weights['wd1']) + _biases['bd1'], name='fc1') # Relu activation dense2 = tf.nn.relu(tf.matmul(dense1, _weights['wd2']) + _biases['bd2'], name='fc2') # Relu activation # Output, class prediction out = tf.matmul(dense2, _weights['out']) + _biases['out'] return out # Store layers weight & bias weights = {'wc1': tf.Variable(tf.random_normal([3, 3, 1, 64])), 'wc2': tf.Variable(tf.random_normal([3, 3, 128, 256])), 'wc3': tf.Variable(tf.random_normal([2, 2, 256, 512])), 'wc4': tf.Variable(tf.random_normal([2*2*512, 1024])), 'wd1': tf.Variable(tf.random_normal([1024, 1024])), 'wd2': tf.Variable(tf.random_normal([1024, 1024])), 'out': tf.Variable(tf.random_normal([1024, 10]))} biases = {'bc1': tf.Variable(tf.random_normal([64])), 'bc2': tf.Variable(tf.random_normal([128])), 'bc3': tf.Variable(tf.random_normal([256])), 'bc4': tf.Variable(tf.random_normal([512])), 'bd1': tf.Variable(tf.random_normal([1024])), 'bd2': tf.Variable(tf.random_normal([1024])), 'out': tf.Variable(tf.random_normal([n_classes]))} # Construct model pred = customnet(x, weights, biases, keep_prob) # Define loss and optimizer cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(pred, y)) optimizer = tf.train.AdamOptimizer(learning_rate=learning_rate).minimize(cost) # Evaluate model correct_pred = tf.equal(tf.argmax(pred, 1), tf.argmax(y, 1)) accuracy = tf.reduce_mean(tf.cast(correct_pred, tf.float32)) # Initializing the variables init = tf.initialize_all_variables() # tf.scalar_summary("loss", cost) tf.scalar_summary("accuracy", accuracy) # Merge all summaries to a single operator merged_summary_op = tf.merge_all_summaries() # Launch the graph with tf.Session() as sess: sess.run(init) summary_writer = tf.train.SummaryWriter('/tmp/logs', graph_def=sess.graph_def) step = 1 # Keep training until reach max iterations while step * batch_size < training_iters: batch_xs, batch_ys = mnist.train.next_batch(batch_size) # Fit training using batch data sess.run(optimizer, feed_dict={x: batch_xs, y: batch_ys, keep_prob: dropout}) if step % display_step == 0: # Calculate batch accuracy acc = sess.run(accuracy, feed_dict={x: batch_xs, y: batch_ys, keep_prob: 1.0}) # Calculate batch loss loss = sess.run(cost, feed_dict={x: batch_xs, y: batch_ys, keep_prob: 1.0}) print "Iter " + str(step*batch_size) + ", Minibatch Loss=" + "{:.6f}".format(loss) + ", Training Accuracy=" + "{:.5f}".format(acc) summary_str = sess.run(merged_summary_op, feed_dict={x: batch_xs, y: batch_ys, keep_prob: 1.0}) summary_writer.add_summary(summary_str, step) step += 1 print "Optimization Finished!" # Calculate accuracy for 256 mnist test images print "Testing Accuracy:", sess.run(accuracy, feed_dict={x: mnist.test.images[:256], y: mnist.test.labels[:256], keep_prob: 1.0})
```

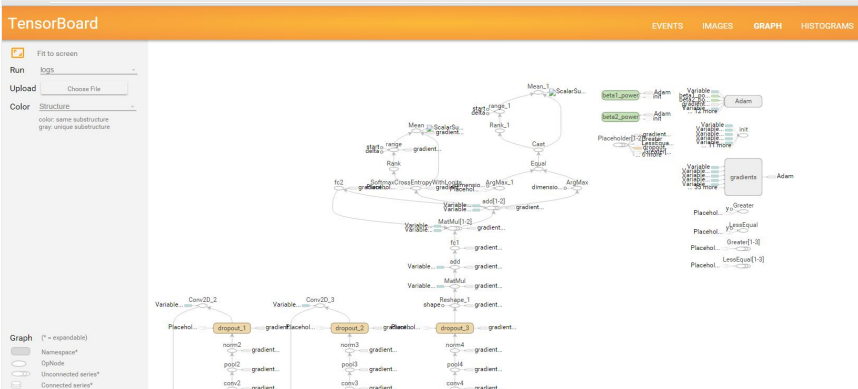
下面我们来启动运行tensorflow tensorborad面板



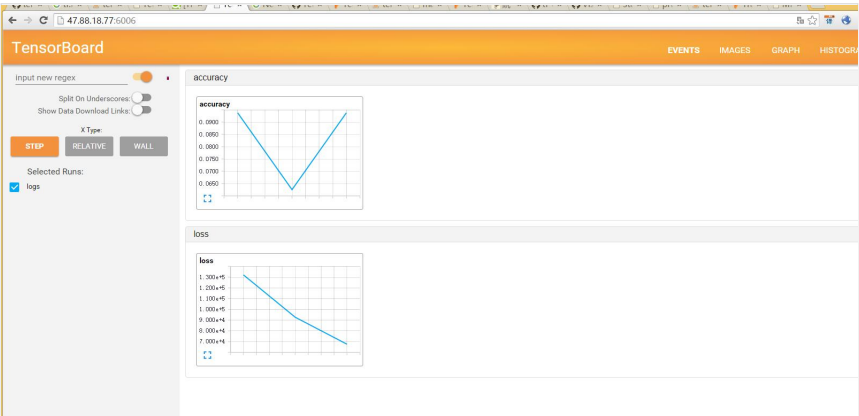
下面打开面板



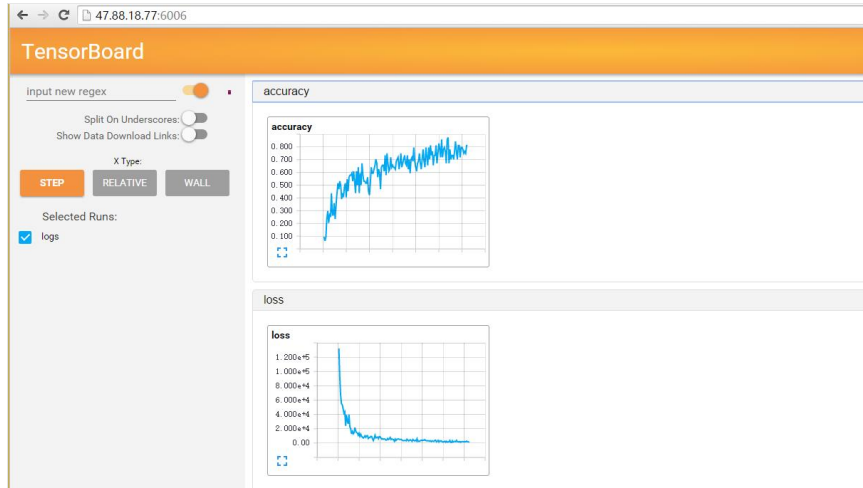
其实大家可能会发现 其实顶部的 四种类型 summary 就是我上面讲的4中类型，如果你添加了相应的summary 会显示相应的区域内



现在才开始 训练 几个的记录，后面曲线变化会很明显的 我就不等他运行完截图了。



最后 loss 慢慢 收敛 无变化



[转藏到我的图书馆](#)
[献花 \(0\)](#)
[分享：](#)
[微信](#)

来自：[基陆伯](#) > [《DeepMind》](#)
[以文找文](#) | [举报](#)

上一篇：[原 TensorFlow人工智能引擎入门教程之三 实现一个自创的CNN卷积神经网络](#)
 下一篇：[TensorFlow人工智能引擎入门教程之五 AlphaGo 的策略网络（CNN）简单的实现](#)

猜你喜欢

霸业传奇

拓展项目

原油分析师

角色扮演页游

现货白银

现货贵金属

原油模拟交易

python趣味编程

贵金属检测仪

传奇私

类似文章

[更多](#)

精选文章

- TensorFlow之CNN和GPU

【机器学习】Tensorflow学习笔记

当TensorFlow遇见CNTK

深入MNIST code测试

【TensorFlow教程】TensorFlow 实战之 K...

Deep learning：四十七(Stochastic Pool...

用Tensorflow基于Deep Q Learning DQN 玩...

TensorFlow CNN 测试CIFAR-10数据集
- 古人是怎样发誓与对待誓约的

月小，似眉弯

生活哲理精粹

爸爸该怎么成为孩子偶像

名医忠告：你知道疾病的克星是什么？

中国未来最值得投资的十大行业

老电影里的大美女

青春不能错过什么



自我护理能力



文件管理系统



《算命字典》举例



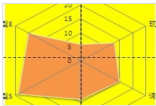
lol竞猜的首页



人有三个错误不能



调查问卷与量表的



霍兰德职业兴趣测



苏果lol的首页



中国居民膳食营养



lol职业联赛2f的首

- 1 生肖决定你是穷苦命,富贵命..
- 2 贵金属投资服务优选百利好..
- 3 北京特价二手房急售,不限购..

- | | |
|----------|----------|
| 1 美亚保险官网 | 4 英语学习 |
| 2 美亚保险 | 5 企业邮箱注册 |
| 3 公司邮箱 | 6 北京口腔医院 |

发表评论:

请 [登录](#) 或者 [注册](#) 后再进行评论

社交帐号登录: