



鼠标手写输入



python 环境



python 项目



霸业传奇

TensorFlow 人工智能引擎 入门教程之一 基本概念以及理解

2016-05-15 基陆伯 阅 21 分享： 微信 转藏到我的图书馆

首先大家看2 张图片 在0.8版本时候已经支持分布式了，所以未来的发展 方面 人工智能的趋势 不可挡，而且个人也是非常喜欢caffe 以及 tensorflow 的我创业 用的就用到了它。



基陆伯 图书馆

11495 馆藏 33889

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 【】

[北京育才美容培训]

美容美发
化妆培训

<师资力量雄厚>

报名电话：
010-63969632



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



有点贵 但却很有型

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

中国电信
仅限紧急呼叫

4G 36%

09:10

×

36氪

⋮

Google 升级开源深度学习算法 TensorFlow，已支持分布式计算

不到半年前 Google 开源了其第二代深度学习技术 TensorFlow，如今它迎来了重大更新——分布式版本，使得其搜索、图像识别以及邮箱的深度学习框架更有意义。此前 TensorFlow 只开源了“单机版”，而非能够识别猫的“分布式版本”。由于目前的计算机还达不到模拟人脑数量庞大的神经元（千亿级），因此便有了用到成千上万大型计算机（计算节点集群）来吸收数据并对其进行训练的“分布式版本”。

Facebook 成立新硬件实验室，挖来谷歌高管 Dugan 负责

Facebook 成立了一个名为“Building 8”、专门开发硬件产品的实验室，并挖来了 Google 高管 Dugan。扎克伯格称该实验室将打造“推动我们连接世界使命”的产品（被认为与 AR 相关）、“在未来数年里向这一领域投资数亿美元以及数百名人力”。据悉，Building 8 的产品主要强调“虚拟与现实”的结合。

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

和你走过的日
没有你的未来
你要幸福！

手写

关闭

CBD 迷你仓

智能迷你仓

微信扫描 可量价

可存放 企业文档 搬家

400-103-396

十里河·陶然亭·丰

关闭

http://www.360doc.com/content/16/0515/08/1317564_559242288.shtml

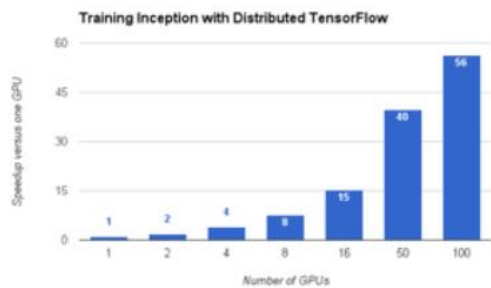
2/8

TensorFlow 0.8 发布，Google 机器学习系统

oschina 发布于 22分钟前，共有 1 条评论

TensorFlow 0.8 发布了，谷歌在众多的产品中使用机器学习。为了不断改进性能，使其尽可能快的运行，Google不断的测试优化。要做到这一点，方法之一就是使用尽可能多服务器，目前跨数百台机器运行，机器学习时间从数周缩短至几小时。TensorFlow自开源以来，很多用户都期待TensorFlow 支持分布式计算，现在Google已经实现该功能。

此次，发布的TensorFlow 0.8支持分布式计算，也提供了高性能GRPC分布式计算模块。它支持TensorFlow 并行运行在数百台服务器上。



下面我们来讲一下 学习tensorflow要了解的基础知识。

首先安装方面 很简单

```
root@izulcdurup2:~# pip install https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-0.5.0-cp27-none-linux_x86_64.whl
Downloading/unpacking https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-0.5.0-cp27-none-linux_x86_64.whl
  Downloading tensorflow-0.5.0-cp27-none-linux_x86_64.whl (10.9MB): 10.9MB downloaded
Requirement already satisfied (use --upgrade to upgrade): numpy>=1.9.2 in /usr/local/lib/python2.7/dist-packages (from tensorflow==0.5.0)
Requirement already satisfied (use --upgrade to upgrade): six>=1.10.0 in /usr/local/lib/python2.7/dist-packages (from tensorflow==0.5.0)
Installing collected packages: tensorflow
Successfully installed tensorflow
Cleaning up...
```

ubuntu 下 `pip install` 这里 知识安装普通的0.5版本，如果需要最新版 看github上最新的0.8版本

安装好了 就可以开始使用了

首先 1.tensorflow = tensor + flow = ndarray + dag图(网络)

也就是说 通过 每一个 有向图 dag 把每一步操作Op 连接起来 传递 输入 输出都是一个nd array 多维数组(numpy) 然后在tensorflow的回路中进行计算操作，

1.占位符

```
x = tf.placeholder(tf.float32, [None, 10]) y = tf.placeholder("float") z = tf.placeholder(tf.int32)
```

占位符表示当OP 操作进行时候传递进来的过程,,也就是 feed 喂养 给予操作，所以 需要外部传递过

来，比如训练图像时候

X 为图像数据 Y 为图像标签，这样的時候 X Y 都是来自训练测试数据，所以前面需要定义2个用来传递的占位符来传递X Y

feed 需要传递 python中的 tuple元祖 feed_dict={x: xdata, y: ydata} ,当执行op操作时候 feed传入替换，执行完后销毁

`tf.placeholder(dtype, shape=None, name=None)`

Inserts a placeholder for a tensor that will be always fed.

Important: This tensor will produce an error if evaluated. Its value must be fed using the feed_dict optional argument to `Session.run()`, `Tensor.eval()`, or `Operation.run()`.

For example:

```
x = tf.placeholder(tf.float32, shape=(1024, 1024))
y = tf.matmul(x, x)

with tf.Session() as sess:
    print(sess.run(y)) # ERROR: will fail because x was not fed.

    rand_array = np.random.rand(1024, 1024)
    print(sess.run(y, feed_dict={x: rand_array})) # Will succeed.
```

Args:

- dtype: The type of elements in the tensor to be fed.
- shape: The shape of the tensor to be fed (optional). If the shape is not specified, you can feed a tensor of any shape.
- name: A name for the operation (optional).

这里上面 第二个参数 shape 表示约束维度 比如 shape=(1024,1024) 就是约束为1024 行 1024列的二维数组

如果shape=(None,1024) None 是python中的空值，表示 可以任何维度行的 1024列的二维数组

2.变量

在tensorflow 一些随时用来计算 变化 操作 共享的量，我们知道OP 操作 在python传递的是ndarray多维数组 C++中就是tensorflow的tensor类，多维数组在tensorflow表示就是变量

```
w = tf.Variable(0.0, name="weights") b= tf.Variable (tf.zeros([100]),name="bias")
```

Variable第一个参数 表示 初始化的值，比如w 初始化为0 这样 w 在后面的过程中会一直变化，下次可能获取w时候是w=xxx 某个值，在tensorflow中这种 可以通过fetch 也就是当需要获得 传递 某个最新的w 时候 可以 session.run([w])，这时候 传递的w 就是每次最新变化的最新的w

如果有时候需要常量 使用

```
k = tf.constant(3.0) #表示定义一个值为3.0的常量
```

See the [Variables How To](#) for a high level overview.

A variable maintains state in the graph across calls to `run()`. You add a variable to the graph by constructing an instance of the class `Variable`.

The `Variable()` constructor requires an initial value for the variable, which can be a `Tensor` of any type and shape. The initial value defines the type and shape of the variable. After construction, the type and shape of the variable are fixed. The value can be changed using one of the assign methods.

If you want to change the shape of a variable later you have to use an `assign` Op with `validate_shape=False`.

Just like any `Tensor`, variables created with `Variable()` can be used as inputs for other Ops in the graph. Additionally, all the operators overloaded for the `Tensor` class are carried over to variables, so you can also add nodes to the graph by just doing arithmetic on variables.

```
import tensorflow as tf

# Create a variable.
w = tf.Variable(<initial-value>, name=<optional-name>)

# Use the variable in the graph like any Tensor.
y = tf.matmul(w, ...another variable or tensor...)

# The overloaded operators are available too.
z = tf.sigmoid(w + b)
```

3.Session 回话

TensorFlow 通过回话 进行 连接 操作 执行所有的op (操作) 形成一个有向图 , 进行 执行运行 图计算 graph compute

每一个回话 的 只要有变量存在的情况 网络图 第一个bottom都是 initop 也就是初始化 变量

```
init = tf.initialize_all_variables() with tf.Session() as session:    session.run(init)
#1. 第一个图的节点op    session.run(googlenet)#2. 第二个图的节点op 1-2-n    sess.close()#释放资源
```

类似于ipython一样 有交互式的

InteractiveSession交互式回话session

```
sess = tf.InteractiveSession()
```



```
tf.Session.run(fetches, feed_dict=None, options=None, run_metadata=None)
```

Runs the operations and evaluates the tensors in `fetches`.

This method runs one "step" of TensorFlow computation, by running the necessary graph fragment to execute every `Operation` and evaluate every `Tensor` in `fetches`, substituting the values in `feed_dict` for the corresponding input values.

The `fetches` argument may be a list of graph elements or a single graph element, and these determine the return value of this method. A graph element can be one of the following types:

- If the *i*th element of `fetches` is an `Operation`, the *i*th return value will be `None`.
- If the *i*th element of `fetches` is a `Tensor`, the *i*th return value will be a numpy ndarray containing the value of that tensor.
- If the *i*th element of `fetches` is a `SparseTensor`, the *i*th return value will be a `SparseTensorValue` containing the value of that sparse tensor.
- If the *i*th element of `fetches` is produced by a `get_tensor_handle` op, the *i*th return value will be a numpy ndarray containing the handle of that tensor.

The optional `feed_dict` argument allows the caller to override the value of tensors in the graph. Each key in `feed_dict` can be one of the following types:

- If the key is a `Tensor`, the value may be a Python scalar, string, list, or numpy ndarray that can be converted to the same dtype as that tensor. Additionally, if the key is a `placeholder`, the shape of the value will be checked for compatibility with the placeholder.
- If the key is a `SparseTensor`, the value should be a `SparseTensorValue`.

The optional `options` argument expects a `[RunOptions]` proto. The options allow controlling the behavior of this particular step (e.g. turning tracing on).

TensorFlow = tensor (多维数组) + flow (graph 图 op) session回话上下文管理 variable == tensor(多维数据变量)

placeholder == 外部传入的参数变量

下面看看官方文档上的一个例子

这里numpy 其实 tensor 二维数组就是numpy中的ndarray

tf.random_uniform 可以见api手册 就是随机的生成均匀分布的 一些-1 1 之间的shape 为1 2的二维数组

`GradientDescentOptimizer` 表示训练时候使用梯度下降来优化 , 还有很多优化策略。

```
import tensorflow as tf
import numpy as np

# 使用 NumPy 生成假数据(phony data), 总共 100 个点.
x_data = np.float32(np.random.rand(2, 100)) # 随机输入
y_data = np.dot([0.100, 0.200], x_data) + 0.300

# 构造一个线性模型
#
b = tf.Variable(tf.zeros([1]))
W = tf.Variable(tf.random_uniform([1, 2], -1.0, 1.0))
y = tf.matmul(W, x_data) + b

# 最小化方差
loss = tf.reduce_mean(tf.square(y - y_data))
optimizer = tf.train.GradientDescentOptimizer(0.5)
train = optimizer.minimize(loss)

# 初始化变量
init = tf.initialize_all_variables()

# 启动图 (graph)
sess = tf.Session()
sess.run(init)

# 拟合平面
for step in xrange(0, 201):
    sess.run(train)
    if step % 20 == 0:
        print step, sess.run(W), sess.run(b)

# 得到最佳拟合结果 W: [[0.100 0.200]], b: [0.300]
```

这一章都是基础概念。后面 可能 要去工作了，时间不是那么有空，周末 可能 有空才会更新，后面章节我们来使用 tensorflow 来定一个 深度学习网络DNN 以及 卷积神经网络CNN .我创业 就到了TensorFlow 所以放心 tensorflow 我还是比较了解的至少可以熟练使用吧，也许很多地方有些不足，请见谅 下面我们使用运行 测试 看看。

这里 提一下 就是因为上面有中文，所以在前面加上coding 让他utf-8 这是python基础知识

```
# -*- coding: utf-8 -*-
import tensorflow as tf
import numpy as np

# 使用 NumPy 生成假数据(phony data), 总共 100 个点.
x_data = np.float32(np.random.rand(2, 100)) # 随机输入
y_data = np.dot([0.100, 0.200], x_data) + 0.300

# 构造一个线性模型
#
```

运行结果

```
root@iZulcdurupZ:~/ai# vi chap1.py
root@iZulcdurupZ:~/ai# python chap1.py
I tensorflow/core/common_runtime/local_device.cc:25] Local device intra op parallelism threads: 1
I tensorflow/core/common_runtime/local_session.cc:45] Local session inter op parallelism threads: 1
0 [[[-0.02269062 0.30589515]] [ 0.6866346]]
20 [[ [ 0.02262159 0.15802984]] [ 0.36292556]]
40 [[ [ 0.07153023 0.17590532]] [ 0.3278482]]
60 [[ [ 0.08839625 0.18830992]] [ 0.3123652]]
80 [[ [ 0.09502661 0.19462496]] [ 0.30549768]]
100 [[ [ 0.0978208 0.19757718]] [ 0.30244562]]
120 [[ [ 0.09903632 0.19891632]] [ 0.30108815]]
140 [[ [ 0.09957226 0.19951679]] [ 0.30048421]]
160 [[ [ 0.09980982 0.19978477]] [ 0.30021548]]
180 [[ [ 0.09991541 0.19990419]] [ 0.30009592]]
200 [[ [ 0.09996235 0.19995736]] [ 0.30004269]]
```

上一篇：如果你想赚大钱，就切记不要太用力！

下一篇：TensorFlow人工智能引擎入门教程之二 CNN卷积神经网络的基本定义理解。

猜你喜欢



霸业传奇



文件管理系统



小生意项目



角色扮演页游



拓展项目



自我护理能力



现货白银



原油分析师



原油交易点差



原油模拟交易

类似文章

更多

精选文章

- TensorFlow之尝试

【机器学习】Tensorflow基本使用

TensorFlow在图像识别中的应用

【机器学习】Tensorflow学习笔记

TensorFlow的安装

深度学习与 Spark 和 TensorFlow

Deep Learning with Spark and TensorFl...

当TensorFlow遇见CNTK
- 650个电影 叫你一次看个够

范冰冰自曝起生理反应的十部激情裸戏

美兰盛开庆佳节

做个有品位的男人

我的父亲是流氓——于建嵘

让自己变得更积极

有时少是很美

世界上最美丽的小镇



汉密尔顿手表



python趣味编程



人有三个错误不能



《算命字典》举例



调查问卷与量表的



lol竞猜的首页



霍兰德职业兴趣测



中国居民膳食营养



lol职业联赛2f的首



苏果lol的首页

1 生肖决定你是穷苦命,富贵命..

2 夫妻俩开店年收100万

3 看老股民如何从15万变300万

1 美亚保险官网

2 美亚保险

3 公司邮箱

4 北京口腔医院

5 企业邮箱注册

6 英语学习

发表评论：

请 [登录](#) 或者 [注册](#) 后再进行评论

社交帐号登录：