





莫烦PYTHON 教程 ▼ 关于我 赞助 大家说

Tensorflow

 Python基础 ▼

 机器学习 ▼

 数据处理 ▼

 其他 ▼

莫烦PYTHON 子2教程 ▼ 络 关于我 赞助 大家说



切换到 优酷 视频 (如优酷播放出现问题, 请 [点击这里](#))

莫烦PYTHON 教程 ▼ 关于我 赞助 大家说

TensorFlow 转到 [tensorflow.org](https://www.tensorflow.org)



« 上一个

下一个 »

例子2

作者: **Morvan** 编辑: **Morvan**

- 学习资料:
 - [相关代码](#)

Tensorflow 是非常重视结构的, 我们得建立好了神经网络的结构, 才能将数字放进去, 运行这个结构.

这个例子简单的阐述了 tensorflow 当中如何用代码来运行我们搭建的结构.

首先, 我们这次需要加载 tensorflow 和 numpy 两个模块, 并且使用 numpy 来创建我们的数据.

```
import tensorflow as tf
```

莫烦PYTHON 教程 ▼ 关于我 赞助 大家说

```
import numpy as np

# create data
x_data = np.random.rand(100).astype(np.float32)
y_data = x_data*0.1 + 0.3
```

接着, 我们用 `tf.Variable` 来创建描述 `y` 的参数. 我们可以把 `y_data = x_data*0.1 + 0.3` 想象成 `y=Weights * x + biases`, 然后神经网络也就是学着把 `Weights` 变成 0.1, `biases` 变成 0.3.

```
Weights = tf.Variable(tf.random_uniform([1], -1.0, 1.0))
biases = tf.Variable(tf.zeros([1]))

y = Weights*x_data + biases
```

接着就是计算 `y` 和 `y_data` 的误差:

```
loss = tf.reduce_mean(tf.square(y-y_data))
```

反向传递误差的工作就教给 `optimizer` 了, 我们使用的误差传递方法是梯度下降法: `Gradient Descent` 让我们使用 `optimizer` 来进行参数的更新.

```
optimizer = tf.train.GradientDescentOptimizer(0.5)
train = optimizer.minimize(loss)
```

莫烦PYTHON 教程 ▼ 关于我 赞助 大家说

```
# init = tf.initialize_all_variables() # tf 马上就要废弃这种写法
init = tf.global_variables_initializer() # 替换成这样就好
```

接着,我们再创建会话 **Session**. 我们会在下一节中详细讲解 **Session**. 我们用 **Session** 来执行 **init** 初始化步骤. 并且, 用 **Session** 来 **run** 每一次 training 的数据. 逐步提升神经网络的预测准确性.

```
sess = tf.Session()
sess.run(init)          # Very important

for step in range(201):
    sess.run(train)
    if step % 20 == 0:
        print(step, sess.run(Weights), sess.run(biases))
```

如果你觉得这篇文章或视频对你的学习很有帮助, 请你也分享它, 让它能再次帮助到更多的需要学习的人.

莫烦没有正式的经济来源, 如果你也想支持 莫烦Python 并看到更好的教学内容, 请拉倒屏幕最下方, 赞助他一点点, 作为鼓励他继续开源的动力.

莫烦PYTHON 教程 ▼ 关于我 赞助 大家说

使用社交网站账户登录

或使用来必力便捷评论 ?

邮件

写评论

总评论数 25

按时间正序



Yuri_xie 2017年2月27日

The TensorFlow library wasn't compiled to use SSE instructions, but these are available on your machine and could speed up CPU computations.

The TensorFlow library wasn't compiled to use SSE2 instructions, but these are available on your machine and could speed up CPU computations.

The TensorFlow library wasn't compiled to use SSE3 instructions, but these are available on your machine and could speed up CPU computations.

The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could speed up CPU computations.

The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could speed up CPU computations.

翻看评论

莫烦PYTHON 教程 ▼ 关于我 赞助 大家说

支持 让教学变得更优秀

点我 赞助 莫烦

关注我的动向:

[Youtube频道](#) [优酷频道](#) [Github](#) [微博](#)

Email: morvanzhou@hotmail.com

© 2016 morvanzhou.github.io. All Rights Reserved