

# Server Configuration

## 一、配置pip用户环境

pip安装到用户目录(不需要管理员权限), 执行以下操作:

```
mkdir .pip
cd .pip
vi pip.conf
# 以下为写入pip.conf文件中的内容
[install]
install-option=--prefix=~/.local
```

## 二、安装miniconda

安装miniconda管理python安装包环境, 执行以下操作:

```
cd
wget https://repo.anaconda.com/miniconda/Miniconda2-latest-Linux-x86_64.sh
sh Miniconda2-latest-Linux-x86_64.sh
```

## 三、配置python3.5环境

利用conda配置python3.5环境, 执行以下操作:

```
conda create -n py35 python=3.5
conda activate py35 //If ERROR, try command: activate py35
conda install tensorflow-gpu
```

以下为TensorFlow-GPU版的测试代码:

```
import tensorflow as tf

with tf.device('/cpu:0'):
    a = tf.constant([1.0,2.0,3.0],shape=[3],name='a')
    b = tf.constant([1.0,2.0,3.0],shape=[3],name='b')
with tf.device('/gpu:1'):
    c = a+b
```

```
#注意: allow_soft_placement=True表明: 计算设备可自行选择, 如果没有这个参数, 会报错。
#因为不是所有的操作都可以被放在GPU上, 如果强行将无法放在GPU上的操作指定到GPU上, 将会报错。
sess = tf.Session(config=tf.ConfigProto(allow_soft_placement=True,log_device_placement=True))
#sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))
sess.run(tf.global_variables_initializer())
print(sess.run(c))
```

然后, 执行测试代码。如果出现类似以下数据, 说明成功:

```
add: (Add): /job:localhost/replica:0/task:0/device:GPU:1
```

安装Keras, 执行以下代码:

```
conda install keras
```