

“启明” TensorFlow 手册

编写时间：2025 年 1 月

一、 介绍	3
1.1. 简介	3
二、 安装软件	3
2.1. 配置anaconda	3
2.2. 加载南科大镜像源	3
2.3. 创建虚拟环境	4
2.4. 安装tensorflow	4
三、 作业样例	5
3.1. 提交脚本	5
3.2. 提交并查看作业	6
3.3. 作业测试结果	6
四、 注意事项	6
4.1. 关于测试样例	6
4.2. 联系方式	7

一、介绍

1.1. 简介

TensorFlow是由Google Brain团队开发的开源机器学习库，用于各种复杂的数学计算，特别是涉及深度学习的计算。它提供了大量工具和资源，用于构建和训练机器学习模型。TensorFlow因其强大的功能和灵活性，在机器学习和深度学习领域得到了广泛应用。

更多内容，请参考官方网站：<https://tensorflow.google.cn/>

二、安装软件

2.1. 配置anaconda

我们建议使用anaconda来管理环境，目前启明集群已安装好anaconda3/2022.10版本，直接加载使用即可，如有其它版本需求可以自行安装。

加载环境：

```
module load anaconda3
```

```
source activate
```

```
[ccse-yangyc@b04u17l ~]$ module load python/anaconda3/2022.10  
[ccse-yangyc@b04u17l ~]$ source activate  
(base) [ccse-yangyc@b04u17l ~]$
```

2.2. 加载南科大镜像源

把以下镜像地址写入`~/.condarc`文件

channels:

- defaults

show_channel_urls: true

default_channels:

- <https://mirrors.sustech.edu.cn/anaconda/pkg/main>
- <https://mirrors.sustech.edu.cn/anaconda/pkg/free>
- <https://mirrors.sustech.edu.cn/anaconda/pkg/r>
- <https://mirrors.sustech.edu.cn/anaconda/pkg/pro>
- <https://mirrors.sustech.edu.cn/anaconda/pkg/msys2>

custom_channels:

conda-forge: <https://mirrors.sustech.edu.cn/anaconda/cloud>

msys2: <https://mirrors.sustech.edu.cn/anaconda/cloud>

bioconda: <https://mirrors.sustech.edu.cn/anaconda/cloud>

menpo: <https://mirrors.sustech.edu.cn/anaconda/cloud>

pytorch: <https://mirrors.sustech.edu.cn/anaconda/cloud>

simpleitk: <https://mirrors.sustech.edu.cn/anaconda/cloud>

nvidia: <https://mirrors.sustech.edu.cn/anaconda-extra/cloud>

2.3. 创建虚拟环境

在登录节点创建环境安装python，虚拟环境可以帮助您隔离不同项目的依赖项，避免不同项目之间的冲突。以下是创建虚拟环境的步骤：

1. 创建虚拟环境：运行以下命令来创建一个新的虚拟环境。您可以将<env_name>替换为您喜欢的环境名称，例如“pytorch_env”。

例如，要创建一个名为tf-2.12.0的虚拟环境，其中Python版本为3.11，可以运行：

```
conda create -n tf-2.12.0 python=3.11 -y
```

2. 激活虚拟环境：运行以下命令来激活刚刚创建的虚拟环境

```
conda activate tf-2.12.0
```

```
(base) [ccse-yangyc@b04u17l ~]$ conda activate tf-2.12.0  
(tf-2.12.0) [ccse-yangyc@b04u17l ~]$
```

2.4. 安装tensorflow

在启明集群上，我们强烈建议安装支持cuda11.8以下包含cuda11.8的tensorflow来兼容集群GPU驱动的使用，tensorflow-gpu-2.12.0版本以上已经合并，安装即可使用GPU和CPU，这里安装tensorflow-gpu-2.11.0为例：

```
pip install tensorflow==2.12.0
```

```
Installing collected packages: libclang, flatbuffers, wrapt, urllib3, typing-extensions, termcolor, tensorflow-io-gcs-filesystem, tensorflow-estimator, tensorboard-data-server, six, pyasn1, protobuf, packaging, opt-einsum, oauthlib, numpy, MarkupSafe, markdown, keras, idna, grpcio, gast, charset-normalizer, certifi, cachetools, absl-py, werkzeug, scipy, rsa, requests, pyasn1-modules, ml_dtypes, h5py, google-pasta, astunparse, requests-oauthlib, jaxlib, google-auth, jax, google-auth-oauthlib, tensorboard, tensorflow  
Successfully installed MarkupSafe-3.0.2 absl-py-2.1.0 astunparse-1.6.3 cachetools-5.5.0 certifi-2024.12.14 charset-normalizer-3.4.1 flatbuffers-24.12.23 gast-0.4.0 google-auth-2.37.0 google-auth-oauthlib-1.0.0 google-pasta-0.2.0 grpcio-1.69.0 h5py-3.12.1 idna-3.10 jax-0.4.30 jaxlib-0.4.30 keras-2.12.0 libclang-18.1.1 markdown-3.7 ml_dtypes-0.5.1 numpy-1.23.5 oauthlib-3.2.2 opt-einsum-3.4.0 packaging-24.2 protobuf-4.25.5 pyasn1-0.6.1 pyasn1-modules-0.4.1 requests-2.32.3 requests-oauthlib-2.0.0 rsa-4.9 scipy-1.15.1 six-1.17.0 tensorboard-2.12.3 tensorboard-data-server-0.7.2 tensorflow-2.12.0 tensorflow-estimator-2.12.0 tensorflow-io-gcs-filesystem-0.37.1 termcolor-2.5.0 typing-extensions-4.12.2 urllib3-2.3.0 werkzeug-3.1.3 wrapt-1.14.1  
(tf-2.12.0) [ccse-yangyc@b04u17l ~]$
```

#注意tensorflow-2.11.0以下版本需要做对应调试，如有其它版本需求，请联系我们再次安装调试

GPU					
Version	Python version	Compiler	Build tools	cuDNN	CUDA
tensorflow-2.16.1	3.9-3.12	Clang 17.0.6	Bazel 6.5.0	8.9	12.3
tensorflow-2.15.0	3.9-3.11	Clang 16.0.0	Bazel 6.1.0	8.9	12.2
tensorflow-2.14.0	3.9-3.11	Clang 16.0.0	Bazel 6.1.0	8.7	11.8
tensorflow-2.13.0	3.8-3.11	Clang 16.0.0	Bazel 5.3.0	8.6	11.8
tensorflow-2.12.0	3.8-3.11	GCC 9.3.1	Bazel 5.3.0	8.6	11.8
tensorflow-2.11.0	3.7-3.10	GCC 9.3.1	Bazel 5.3.0	8.1	11.2
tensorflow-2.10.0	3.7-3.10	GCC 9.3.1	Bazel 5.1.1	8.1	11.2
tensorflow-2.9.0	3.7-3.10	GCC 9.3.1	Bazel 5.0.0	8.1	11.2
tensorflow-2.8.0	3.7-3.10	GCC 7.3.1	Bazel 4.2.1	8.1	11.2
tensorflow-2.7.0	3.7-3.9	GCC 7.3.1	Bazel 3.7.2	8.1	11.2
tensorflow-2.6.0	3.6-3.9	GCC 7.3.1	Bazel 3.7.2	8.1	11.2
tensorflow-2.5.0	3.6-3.9	GCC 7.3.1	Bazel 3.7.2	8.1	11.2
tensorflow-2.4.0	3.6-3.8	GCC 7.3.1	Bazel 3.1.0	8.0	11.0
tensorflow-2.3.0	3.5-3.8	GCC 7.3.1	Bazel 3.1.0	7.6	10.1
tensorflow-2.2.0	3.5-3.8	GCC 7.3.1	Bazel 2.0.0	7.6	10.1
tensorflow-2.1.0	2.7, 3.5-3.7	GCC 7.3.1	Bazel 0.27.1	7.6	10.1
tensorflow-2.0.0	2.7, 3.3-3.7	GCC 7.3.1	Bazel 0.26.1	7.4	10.0
tensorflow_gpu-1.15.0	2.7, 3.3-3.7	GCC 7.3.1	Bazel 0.26.1	7.4	10.0

三、 作业样例

3.1. 提交脚本

提交脚本文件为test.lsf，测试tensorflow是否可用gpu，test.py 脚本来测试作业分配的gpu卡是否能检测到

1. 测试脚本内容：

```
import tensorflow as tf
print((tf.config.list_physical_devices('GPU')))
print(tf.test.is_gpu_available())
```

2. 作业脚本内容：

```
#!/bin/bash
```

```
#BSUB -J tensorflow      #作业名称，可以自定义
#BSUB -q 4a100-40 #提交队列
#BSUB -n 1          #申请1个核，和申请的-gpu卡数一致
#BSUB -e %J.err      #定义错误输出文件
#BSUB -o %J.out      #定义正常输出日志文件
#BSUB -R "span[ptile=1]" #申明一个节点核数，这个跟-n 申请的核数一致
#BSUB -gpu "num=1"      #申请1张gpu卡

#加载安装tensorflow、cuda、cudnn环境
module load  cuda/11.8 cudnn/8.8.1.3_cuda11
module load  python/anaconda3/2022.10
source activate
conda activate tf-2.12.0

#运行python脚本
python test.py > $LSB_JOBID.log 2>&1
```

3.2. 提交并查看作业

```
[ccse-yangyc@b04u17l tensorflow]$ bsub < test.lsf
Job <312017> is submitted to queue <4a100-40>.
[ccse-yangyc@b04u17l tensorflow]$ bjobs
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
312017 ccse-ya RUN   4a100-40  b04u17l    be1u03g    torch     Jan 19 16:36
[ccse-yangyc@b04u17l tensorflow]$
```

3.3. 作业测试结果

测试结果文件为312017.log

```
[ccse-yangyc@b04u17l tensorflow]$ cat 312017.log
2025-01-19 16:36:59.193489: I tensorflow/core/util/port.cc:110] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2025-01-19 16:36:59.350962: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2025-01-19 16:37:01.180518: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
WARNING:tensorflow:From /work/ccse-yangyc/jobtest/tensorflow/test.py:3: is_gpu_available (from tensorflow.python.framework.test_util) is deprecated and will be removed in a future version.
Instructions for updating:
Use `tf.config.list_physical_devices('GPU')` instead.
2025-01-19 16:37:04.606093: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1635] Created device /device:GPU:0 with 38339 MB memory: -> device: 0, name: NVIDIA A100-SXM4-40GB, pci bus id: 0000:17:00.0, compute capability: 8.0
[PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
gpu status: True
[ccse-yangyc@b04u17l tensorflow]$
```

四、 注意事项

4.1. 关于测试样例

- 本文档测试样例，只是验证改软件能正常运行。安装tensorflow之后，还得根据自己使用的其它模块在登录节点安装进行调试使用。

- 欢迎使用该软件的用户，提供更加好的python脚本样例，以便提高该软件的更多测试。

4.2. 联系方式

- 本文档由科学与工程计算中心编写，联系邮箱：hpc@sustech.edu.cn, 联系人：谢作扬，电话：0755-88015831