GPU 支持 | TensorFlow

硬件要求

支持以下带有 GPU 的设备：

- CUDA® 架构为 3.5、5.0、6.0、7.0、7.5、8.0 或更高的 NVIDIA® GPU 卡。请参阅支持 CUDA® 的 GPU 卡列表。
- 如果 GPU 采用的 CUDA® 架构不受支持，或为了避免从 PTX 进行 JIT 编译，亦或是为了使用不同版本的 NVIDIA® 库，请参阅在 Linux 下从源代码编译指南。
- 软件包不包含 PTX 代码，但最新支持的 CUDA® 架构除外；因此，如果设置了 `CUDA_FORCE_PTX_JIT=1`，TensorFlow 将无法在旧款 GPU 上加载。（有关详细信息，请参阅应用兼容性。）

★ **注意：**"状态：设备内核映像无效"错误消息表示 TensorFlow 软件包不包含适用于您的架构的 PTX。您可以通过从源代码构建 TensorFlow 启用计算功能。

https://developer.nvidia.com/cuda-gpus

Tesla Workstation Products

GPU	Compute Capability
Tesla K80	3.7
Tesla K40	3.5
Tesla K20	3.5
Tesla C2075	2.0
Tesla C2050/C2070	2.0

NVIDIA Data Center Products

GPU	Compute Capability
NVIDIA A100	8.0
NVIDIA A40	8.6
NVIDIA A30	8.0
NVIDIA A10	8.6
NVIDIA A16	8.6
NVIDIA A2	8.6
NVIDIA T4	7.5
NVIDIA V100	7.0
Tesla P100	6.0
Tesla P40	6.1
Tesla P4	6.1
Tesla M60	5.2
Tesla M40	5.2
Tesla K80	3.7
Tesla K40	3.5
Tesla K20	3.5
Tesla K10	3.0

Tesla P100的CUDA**架构**为6.0

软件要求

必须在系统中安装以下 NVIDIA® 软件：

- NVIDIA® GPU 驱动程序 - CUDA® 11.2 要求 450.80.02 或更高版本。
- CUDA® 工具包：TensorFlow 支持 CUDA® 11.2（TensorFlow 2.5.0 及更高版本）
- CUDA® 工具包附带的 CUPTI。
- cuDNN SDK 8.1.0 cuDNN 版本。
- （可选）TensorRT 6.0，可缩短用某些模型进行推断的延迟时间并提高吞吐量。

cuda工具包对应TensorFlow版本：

[从源代码构建 | TensorFlow]

GPU

版本	Python 版本	编译器	构建工具	cuDNN	CUDA
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
tensorflow_gpu-1.14.0	2.7、3.3-3.7	GCC 4.8	Bazel 0.24.1	7.4	10.0

打算下载

tensorflow-2.6.0	3.6-3.9	GCC 7.3.1	Bazel 3.7.2	8.1	11.2

python -V 查看python版本

```
Singularity [~/deepxde] > python -V
Python 3.9.12
```

满足

命令：nvidia-smi

显示显卡为Tesla P100　(能显示说明驱动已安装)

驱动程序版本为：　支持的cuda版本为：

Terminal 1

```
Singularity [~] > ls
deepxde
Singularity [~] > cd deepxde/
Singularity [~/deepxde] > ls
Singularity [~/deepxde] > nvidia-smi
Sat May 28 04:12:55 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.129.06   Driver Version: 470.129.06   CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla P100-PCIE...  Off  | 00000000:00:07.0 Off |                    0 |
| N/A   27C    P0    27W / 250W |      0MiB / 12198MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

驱动程序版本>450.xxx **满足**

支持的CUDA为11.4（向下兼容）>11.2 **满足**

conda源配置：

进这个网址就知道了 https://mirrors.tuna.tsinghua.edu.cn/help/anaconda/

下载安装CUDA和cuDNN:

————————————————————

官方例子：（不使用官方例子）

Ubuntu 18.04 (CUDA 11.0)

```
# Add NVIDIA package repositories
wget
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-ubuntu1804.pin
sudo mv cuda-ubuntu1804.pin /etc/apt/preferences.d/cuda-repository-pin-600
sudo apt-key adv --fetch-keys
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2af80.pub
sudo add-apt-repository "deb
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/ /"
sudo apt-get update

wget http://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64/nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb

sudo apt install ./nvidia-machine-learning-repo-ubuntu1804_1.0.0-1_amd64.deb
sudo apt-get update

wget https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64/libnvinfer7_7.1.3-1+cuda11.0_amd64.deb
sudo apt install ./libnvinfer7_7.1.3-1+cuda11.0_amd64.deb
sudo apt-get update

# Install development and runtime libraries (~4GB)
sudo apt-get install --no-install-recommends \
    cuda-11-0 \
    libcudnn8=8.0.4.30-1+cuda11.0  \
    libcudnn8-dev=8.0.4.30-1+cuda11.0

# Reboot. Check that GPUs are visible using the command: nvidia-smi

# Install TensorRT. Requires that libcudnn8 is installed above.
sudo apt-get install -y --no-install-recommends libnvinfer7=7.1.3-1+cuda11.0 \
    libnvinfer-dev=7.1.3-1+cuda11.0 \
    libnvinfer-plugin7=7.1.3-1+cuda11.0
```

————————————————————

使用以下方式更简洁，版本号也对应TensorFlow2.6.0

tensorflow-2.6.0	3.6-3.9	GCC 7.3.1	Bazel 3.7.2	8.1	11.2

创建新环境conda create -n tensorflow-gpu python=3.7

激活环境activate tensorflow-gpu

安装：在具体使用中，其实真正需要的并不是整个CUDA，而是cudatoolkit，所以直接安装cudatoolkit，不需要再下载安装CUDA。依次执行命令：
conda install cudatoolkit=11.2
conda install cudnn=8.1
pip install tensorflow-gpu==2.6.0

测试代码：
import tensorflow as tf
print(tf.**version**)
print(tf.test.is_gpu_available())

————————————————————
https://blog.csdn.net/qq_41298034/article/details/120796756

1 查看 Nvidia 显卡利用率：显存占用和算力情况。

```
# 0.5 秒更新一次显卡利用情况，并查看 NVIDIA 驱动版本
watch -n 0.5 nvidia-smi
```

2. 查看 Cuda 版本：

```
cat  /usr/local/cuda/version.txt
1
```

3.查看 Cudnn 版本：

```
cat /usr/local/cuda/include/cudnn.h | grep CUDNN_MAJOR -A 2
```

TensorFlow

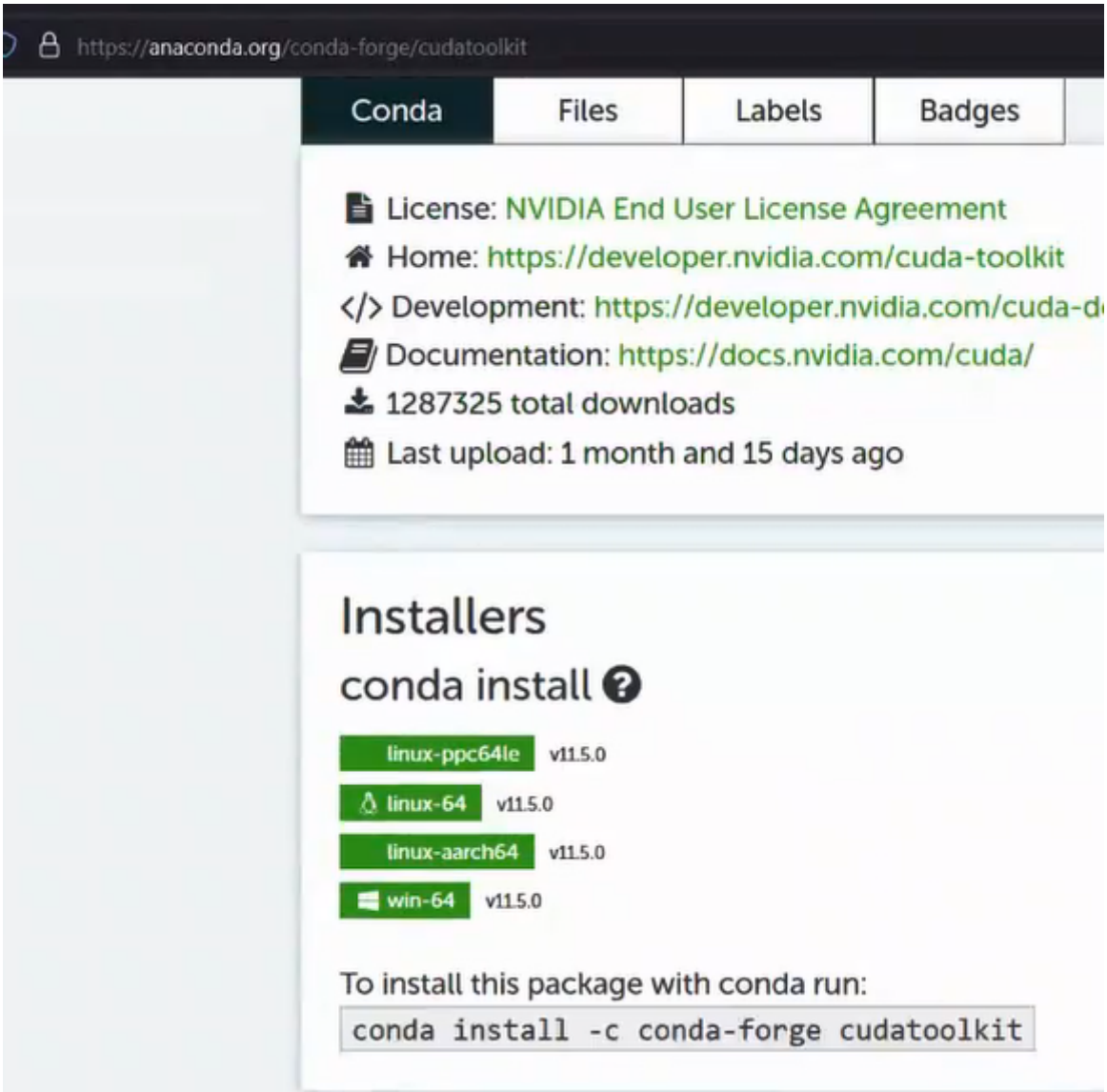
- https://www.tensorflow.org

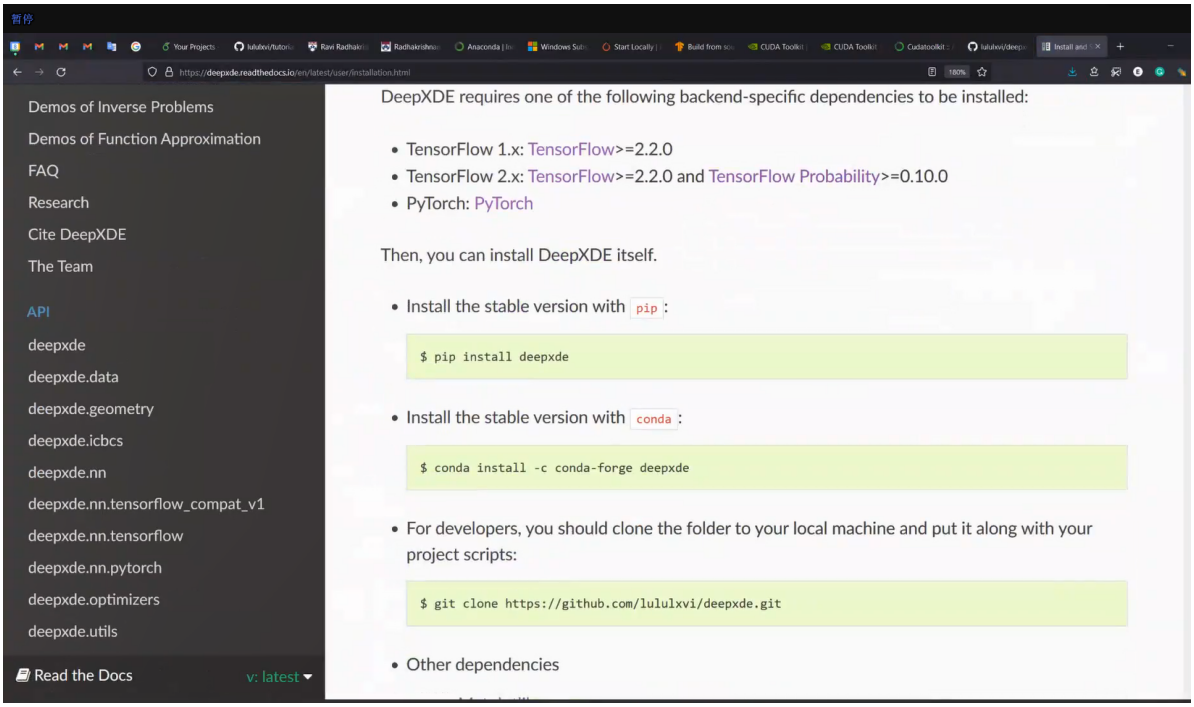
```
# Current stable release for CPU and GPU
$ pip install tensorflow
```

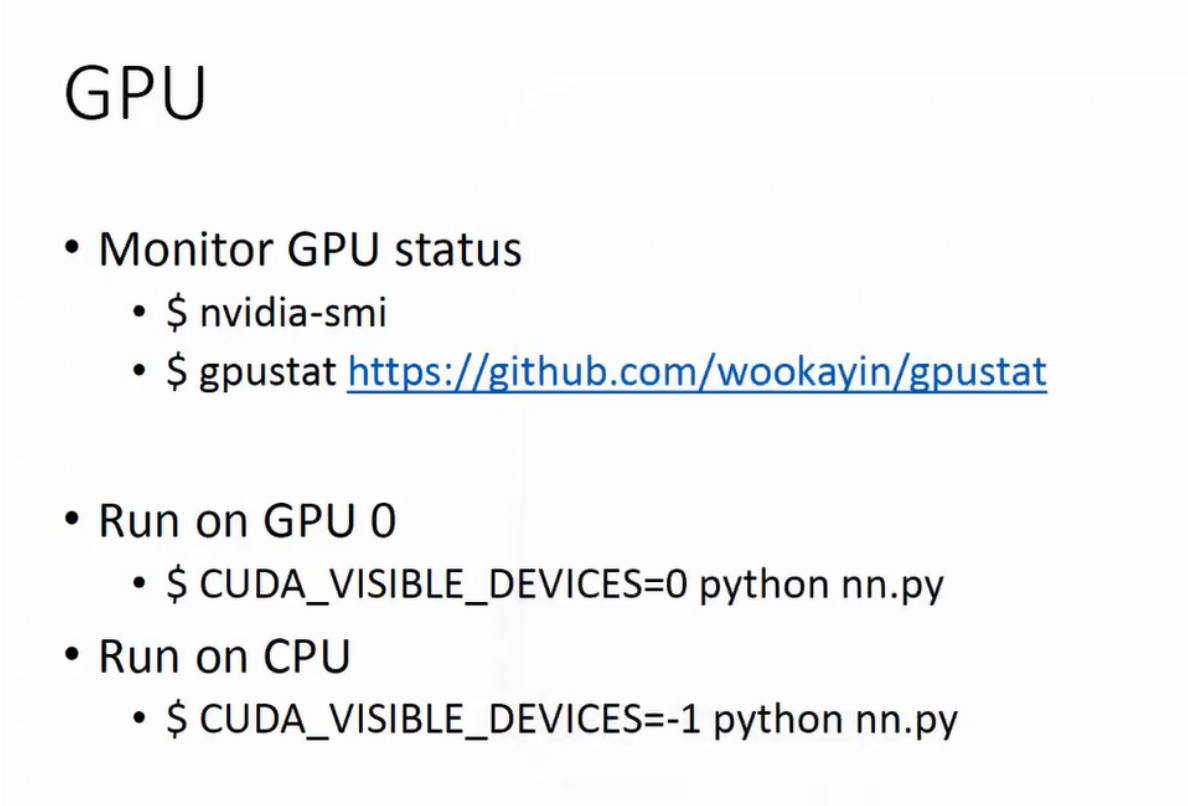
- GPU: compatible versions of NVIDIA driver, CUDA, cuDNN
  - Can be directly installed on OS
  - Install CUDA and cuDNN in Anaconda

Package	Version	Source
cudatoolkit	11.2.2	conda-forge
cudnn	8.1.0.77	conda-forge
TensorFlow	2.6.2	pip
tensorflow-probability	0.14.1	pip
tensorflow-addons	0.14.0	pip

Updated on 11/17/2021







3. 在官方文档中查看安装TensorFlow版本对应的CUDNN、CUDA的版本号

GPU

版本	Python 版本	编译器	构建工具	cuDNN	CUDA
tensorflow_gpu-2.6.0	3.6-3.9	MSVC 2019	Bazel 3.7.2	8.1	11.2

4. 创建新环境conda create -n tensorflow-gpu python=3.7

5. 激活环境activate tensorflow- gpu

最终输入命令：

conda create - n tensorflow-gpu python=3.9 （之后一直y）

```
# To activate this environment, use
#
#     $ conda activate tensorflow-gpu
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

conda activate tensorflow-gpu

conda install -c conda-forge deepxde

conda install -c conda-forge cudatoolkit=11.2
conda install -c conda-forge cudnn=8.1
pip install tensorflow==2.6.0

pip install tensorflow-probability==0.14.1

pip install tensorflow-addons==0.14.0

测试代码：
python

```
import tensorflow as tf
print(tf.__version__)
print(tf.test.is_gpu_available())
```

第一句import tf如果报错：

TypeError: Descriptors cannot not be created directly.
If this call came from a _pb2.py file, your generated code is out of date and must be regenerated with protoc >= 3.19.0.
If you cannot immediately regenerate your protos, some other possible workarounds are:

1. Downgrade the protobuf package to 3.20.x or lower.
2. Set PROTOCOL_BUFFERS_PYTHON_IMPLEMENTATION=python (but this will use pure-Python parsing and will be much slower).

解决：

pip uninstall protobuf

pip install protobuf==3.20.1

deepxde的后端版本选TensorFlow2，具体做法：

Deepxde backend not selected or invalid. Assuming tensorflow.compat.v1 for now.
Setting the default backend to "tensorflow.compat.v1". You can change it in the ~/.deepxde/config.json file or export the DDEBACKEND environment variable. Valid options are: tensorflow.compat.v1, **tensorflow**, pytorch, jax, paddle (all lowercase)

错误：
ImportError: cannot import name 'dtensor' from 'tensorflow.compat.v2.experimental'
解决：
我降级到 Keras v2.6.0，解决了这个问题。这似乎是由 v2.9 中的错误引起的。

要降级到 v2.6.0，在Anacondad的tensorflow虚拟环境中运行：

pip install keras==2.6.* -i https://pypi.douban.com/simple/

错误：找不到ptxas
2022-06-02 03:18:16.841335: I tensorflow/compiler/xla/service/service.cc:171] XLA service 0x55fa1dda3c60 initialized for platform CUDA (this does not guarantee that XLA will be used). Devices:
2022-06-02 03:18:16.841373: I tensorflow/compiler/xla/service/service.cc:179] StreamExecutor device (0): Tesla P100-PCIE-12GB, Compute Capability 6.0
2022-06-02 03:18:16.854943: I tensorflow/compiler/mlir/tensorflow/utils/dump_mlir_util.cc:210] disabling MLIR crash reproducer, set env var `MLIR_CRASH_REPRODUCER_DIRECTORY` to enable.
2022-06-02 03:18:17.263143: I tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child process: No such file or directory
2022-06-02 03:18:17.265299: I tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child process: No such file or directory
2022-06-02 03:18:17.265331: W tensorflow/stream_executor/gpu/asm_compiler.cc:77] Couldn't get ptxas version string: Internal: Couldn't invoke ptxas --version
2022-06-02 03:18:17.265982: I tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child process: No such file or directory
2022-06-02 03:18:17.266049: W tensorflow/stream_executor/gpu/redzone_allocator.cc:314] Internal: Failed to launch ptxas
Relying on driver to perform ptx compilation.
Modify $PATH to customize ptxas location.
This message will be only logged once.
2022-06-02 03:18:17.275523: I tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child process: No such file or directory
2022-06-02 03:18:17.275560: W tensorflow/stream_executor/gpu/asm_compiler.cc:77] Couldn't get ptxas version string: Internal: Couldn't invoke ptxas --version
2022-06-02 03:18:17.276316: I tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child process: No such file or directory
2022-06-02 03:18:17.276376: W tensorflow/compiler/xla/service/gpu/buffer_comparator.cc:512] Internal: Failed to launch ptxas
Relying on driver to perform ptx compilation.
Setting XLA_FLAGS=--xla_gpu_cuda_data_dir=/path/to/cuda or modifying $PATH can be used to set the location of ptxas
This message will only be logged once.
2022-06-02 03:18:17.435514: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:75] Can't find libdevice directory ${CUDA_DIR}/nvvm/libdevice. This may result in compilation or runtime failures, if the program we try to run uses routines from libdevice.
2022-06-02 03:18:17.435539: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:76] Searched for CUDA in the following directories:
2022-06-02 03:18:17.435551: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:79] ./cuda_sdk_lib
2022-06-02 03:18:17.435559: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:79] /usr/local/cuda-11.2
2022-06-02 03:18:17.435567: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:79] /usr/local/cuda

2022-06-02 03:18:17.435575: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:79] .
2022-06-02 03:18:17.435584: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:81] You can choose the search directory by setting xla_gpu_cuda_data_dir in HloModule's DebugOptions. For most apps, setting the environment variable XLA_FLAGS=--xla_gpu_cuda_data_dir=/path/to/cuda will work.
2022-06-02 03:18:17.490077: I tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child process: No such file or directory
2022-06-02 03:18:17.490113: W tensorflow/stream_executor/gpu/asm_compiler.cc:77] Couldn't get ptxas version string: Internal: Couldn't invoke ptxas --version
2022-06-02 03:18:17.490680: I tensorflow/core/platform/default/subprocess.cc:304] Start cannot spawn child process: No such file or directory
2022-06-02 03:18:17.490745: F tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:480] ptxas returned an error during compilation of ptx to sass: 'Internal: Failed to launch ptxas' If the error message indicates that a file could not be written, please verify that sufficient filesystem space is provided.
解决：
安装nvcc就自动把位置写好了
conda install -c nvidia cuda-nvcc
参考：https://github.com/google/jax/discussions/6843

错误：tensorflow.python.framework.errors_impl.InternalError: **libdevice not found** at ./libdevice.10.bc

(40条消息) 【conda虚拟环境安装CUDA路径】一*苇以航*aw的博客-CSDN博客_conda安装的cuda位置

Installation issue with libdevice · Discussion #6479 · google/jax (github.com)

https://blog.csdn.net/hedongya/article/details/79671469

复制文件：(40条消息) Ubuntu中复制文件或目录的命令*YY.Jiang的博客-CSDN博客*ubuntu复制文件

编译安装tensorflow GPU版本时报错：Cannot find libdevice.10.bc under /usr/local/cuda-8.0

解决（没用）：

没使用conda安装：

将/usr/local/cuda-8.0/nvvm/libdevice/libdevice.compute_50.10.bc改为libdevice.10.bc，并复制一份至/usr/local/cuda-8.0/

用conda安装的cuda：（这才是我的情况）

需要的文件在/anaconda3/envs/tensorflow-gpu/nvvm/libdevice的libdevice.10.bc

解决：

报错中有：

Can't find libdevice directory ${CUDA_DIR}/nvvm/libdevice. This may result in compilation or runtime failures, if the program we try to run uses routines from libdevice.

所以只要给我的anaconda的环境(名字叫tensorflow-gpu)新建一个环境变量CUDA_DIR就行了。

(40条消息) conda虚拟环境中设置环境变量*robot8me的博客-CSDN博客*conda 环境变量

conda虚拟环境中可以单独设置当前环境的环境变量，只有当前环境被激活(conda activate)时，自定义设置的环境变量才起作用，当conda deactivate后自定义的环境变量会自动清除。

可以使用conda env config vars set my_var=value设置当前虚拟环境中的自定义环境变量。

（CUDA_DIR=/anaconda3/envs/tensorflow-gpu）

但是设置完环境变量后必须重新激活环境conda activate env_name。
如果要查看自定义的环境变量是否设置生效可以用echo %my_var%（在Windows命令行使用%%这种形式）或者conda env config vars list，conda evn config vars list会列出当前虚拟环境中所有自定义的环境变量。
还可以通过-n指定要给那个虚拟环境设置自定义环境变量，例如：在虚拟环境conda env config vars set my_test_var=123 -n env_test_var

若要去除设置的环境变量

使用`conda env config vars unset my_var -n test-env`。`-n`同样是指定去除那个虚拟环境中设置的自定义环境变量。

但是，指定了正确的环境变量，它依然找不到。

只能把这个文件复制到以下路径之一了：

Can't find libdevice directory ${CUDA_DIR}/nvvm/libdevice. This may result in compilation or runtime failures, if the program we try to run uses routines from libdevice.
2022-06-02 04:25:35.090014: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:76] Searched for CUDA in the following directories:
2022-06-02 04:25:35.090023: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:79] **./cuda_sdk_lib**
2022-06-02 04:25:35.090032: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:79] **/usr/local/cuda-11.2**
2022-06-02 04:25:35.090037: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:79] **/usr/local/cuda**
2022-06-02 04:25:35.090042: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:79]

.
2022-06-02 04:25:35.090069: W tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:81] You can choose the search directory by setting xla_gpu_cuda_data_dir in HloModule's DebugOptions. For most apps, setting the environment variable XLA_FLAGS=--xla_gpu_cuda_data_dir=/path/to/cuda will work.

经过试验，只有第四个路径.可以用。

故在要运行的python文件路径下

cp -i /anaconda3/envs/tensorflow-gpu/nvvm/libdevice/libdevice.10.bc ./

即可

即使我在conda的tensorflow-gpu虚拟环境里，.../tensorflow-gpu/并不是我的根目录。

测试deepxde的ode例子，可使用gpu正常运行！

```
(tensorflow-gpu) Singularity [~/examples/pinn_forward] > python ode_system.py
Using backend: tensorflow
```

```
/anaconda3/envs/tensorflow-gpu/lib/python3.9/site-
packages/skopt/sampler/sobol.py:246: UserWarning: The balance properties of
Sobol' points require n to be a power of 2. 0 points have been previously
generated, then: n=0+37=37.
  warnings.warn("The balance properties of Sobol' points require "
2022-06-02 04:47:40.699589: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-06-02 04:47:40.713549: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-06-02 04:47:40.714893: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-06-02 04:47:40.716639: I tensorflow/core/platform/cpu_feature_guard.cc:142]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
2022-06-02 04:47:40.717406: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-06-02 04:47:40.718737: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-06-02 04:47:40.720030: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-06-02 04:47:41.522867: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-06-02 04:47:41.523724: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-06-02 04:47:41.524465: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:937] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-06-02 04:47:41.525135: W
tensorflow/core/common_runtime/gpu/gpu_bfc_allocator.cc:39] Overriding
allow_growth setting because the TF_FORCE_GPU_ALLOW_GROWTH environment variable
is set. Original config value was 0.
2022-06-02 04:47:41.525197: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device
/job:localhost/replica:0/task:0/device:GPU:0 with 11323 MB memory:  -> device: 0,
name: Tesla P100-PCIE-12GB, pci bus id: 0000:00:07.0, compute capability: 6.0
Compiling model...
'compile' took 0.000485 s
```

```
Training model...

WARNING:tensorflow:AutoGraph could not transform <function <lambda> at
0x7f651aef55e0> and will run it as-is.
Cause: could not parse the source code of <function <lambda> at 0x7f651aef55e0>:
no matching AST found
To silence this warning, decorate the function with
@tf.autograph.experimental.do_not_convert
WARNING:tensorflow:AutoGraph could not transform <function <lambda> at
0x7f651aef5820> and will run it as-is.
Cause: could not parse the source code of <function <lambda> at 0x7f651aef5820>:
no matching AST found
To silence this warning, decorate the function with
@tf.autograph.experimental.do_not_convert
2022-06-02 04:47:43.055607: I tensorflow/compiler/xla/service/service.cc:171] XLA
service 0x5609df2a3220 initialized for platform CUDA (this does not guarantee
that XLA will be used). Devices:
2022-06-02 04:47:43.055653: I tensorflow/compiler/xla/service/service.cc:179]
StreamExecutor device (0): Tesla P100-PCIE-12GB, Compute Capability 6.0
2022-06-02 04:47:43.068472: I
tensorflow/compiler/mlir/tensorflow/utils/dump_mlir_util.cc:210] disabling MLIR
crash reproducer, set env var `MLIR_CRASH_REPRODUCER_DIRECTORY` to enable.
2022-06-02 04:47:43.727578: W
tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:75] Can't find libdevice
directory ${CUDA_DIR}/nvvm/libdevice. This may result in compilation or runtime
failures, if the program we try to run uses routines from libdevice.
2022-06-02 04:47:43.727650: W
tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:76] Searched for CUDA in
the following directories:
2022-06-02 04:47:43.727670: W
tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:79]   ./cuda_sdk_lib
2022-06-02 04:47:43.727681: W
tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:79]   /usr/local/cuda-11.2
2022-06-02 04:47:43.727692: W
tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:79]   /usr/local/cuda
2022-06-02 04:47:43.727702: W
tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:79]   .
2022-06-02 04:47:43.727714: W
tensorflow/compiler/xla/service/gpu/nvptx_compiler.cc:81] You can choose the
search directory by setting xla_gpu_cuda_data_dir in HloModule's DebugOptions.
For most apps, setting the environment variable XLA_FLAGS=--
xla_gpu_cuda_data_dir=/path/to/cuda will work.
2022-06-02 04:47:43.889281: I
tensorflow/compiler/jit/xla_compilation_cache.cc:363] Compiled cluster using XLA!
This line is logged at most once for the lifetime of the process.
Step      Train loss                                  Test loss
                Test metric
0         [7.49e-02, 2.61e-01, 0.00e+00, 1.00e+00]    [7.51e-02, 2.67e-01,
0.00e+00, 1.00e+00]    [1.18e+00]
1000      [8.60e-03, 8.25e-03, 3.03e-08, 3.62e-04]    [9.61e-03, 7.60e-03, 3.03e-
08, 3.62e-04]    [7.91e-01]
2000      [5.21e-03, 3.93e-03, 4.56e-07, 9.13e-05]    [5.45e-03, 3.82e-03, 4.56e-
07, 9.13e-05]    [5.68e-01]
3000      [2.06e-03, 2.52e-03, 1.10e-06, 3.93e-05]    [2.23e-03, 2.30e-03, 1.10e-
06, 3.93e-05]    [3.64e-01]
4000      [6.93e-04, 1.15e-03, 7.31e-07, 1.09e-05]    [7.55e-04, 9.73e-04, 7.31e-
07, 1.09e-05]    [2.07e-01]
```

```
5000      [1.88e-04, 4.07e-04, 2.52e-07, 2.50e-06]    [2.07e-04, 3.10e-04, 2.52e-
07, 2.50e-06]    [9.86e-02]
6000      [1.11e-04, 1.32e-04, 6.06e-06, 1.94e-05]    [1.10e-04, 8.18e-05, 6.06e-
06, 1.94e-05]    [4.38e-02]
7000      [1.14e-04, 8.30e-05, 4.66e-06, 1.28e-06]    [1.06e-04, 5.97e-05, 4.66e-
06, 1.28e-06]    [2.49e-02]
8000      [2.76e-05, 2.10e-05, 3.28e-08, 5.92e-09]    [2.65e-05, 1.33e-05, 3.28e-
08, 5.92e-09]    [1.47e-02]
9000      [2.37e-05, 1.49e-05, 6.72e-08, 5.36e-08]    [2.31e-05, 9.86e-06, 6.72e-
08, 5.36e-08]    [1.14e-02]
10000     [2.08e-05, 1.40e-05, 2.10e-11, 9.23e-07]    [1.95e-05, 1.06e-05, 2.10e-
11, 9.23e-07]    [1.05e-02]
11000     [2.33e-05, 1.25e-05, 6.19e-07, 1.58e-08]    [2.58e-05, 1.11e-05, 6.19e-
07, 1.58e-08]    [1.02e-02]
12000     [1.69e-05, 7.76e-06, 9.64e-08, 1.03e-07]    [1.73e-05, 6.02e-06, 9.64e-
08, 1.03e-07]    [7.96e-03]
13000     [1.50e-05, 6.75e-06, 5.95e-09, 3.12e-09]    [1.49e-05, 5.52e-06, 5.95e-
09, 3.12e-09]    [8.03e-03]
14000     [1.36e-05, 7.01e-06, 1.09e-11, 5.41e-07]    [1.31e-05, 6.08e-06, 1.09e-
11, 5.41e-07]    [7.69e-03]
15000     [1.22e-05, 5.01e-06, 1.07e-08, 3.71e-10]    [1.22e-05, 4.70e-06, 1.07e-
08, 3.71e-10]    [6.68e-03]
16000     [2.51e-04, 1.18e-04, 4.01e-05, 1.39e-05]    [2.45e-04, 1.07e-04, 4.01e-
05, 1.39e-05]    [1.77e-02]
17000     [1.84e-05, 9.80e-06, 1.24e-06, 5.35e-08]    [1.66e-05, 9.28e-06, 1.24e-
06, 5.35e-08]    [7.04e-03]
18000     [3.83e-04, 2.23e-04, 7.88e-05, 2.78e-04]    [4.03e-04, 2.35e-04, 7.88e-
05, 2.78e-04]    [1.92e-02]
19000     [8.57e-06, 3.15e-06, 7.63e-09, 8.96e-09]    [8.74e-06, 3.67e-06, 7.63e-
09, 8.96e-09]    [4.69e-03]
20000     [8.60e-06, 4.42e-06, 6.04e-08, 1.22e-06]    [7.81e-06, 4.49e-06, 6.04e-
08, 1.22e-06]    [4.95e-03]

Best model at step 19000:
  train loss: 1.17e-05
  test loss: 1.24e-05
  test metric: [4.69e-03]

'train' took 25.933613 s

Saving loss history to /root/examples/pinn_forward/loss.dat ...
Saving training data to /root/examples/pinn_forward/train.dat ...
Saving test data to /root/examples/pinn_forward/test.dat ...
```

公式 → 公式+点

点 → 公式+点 （论文）