链滴

核显 DirectML 深度学习环境

作者：bingoct

原文链接：https://ld246.com/article/1664965187410

来源网站：链滴

参考文档：

tensorflow-directml 官方文档

Enable TensorFlow with DirectML in WSL

环境： win10 + wsl2-ubuntu20.04

安装显卡驱动（确保型号匹配，我是 amd 的 r5 核显）：

- amd：见 Radeon™ Software Adrenalin 2020 Edition for Microsoft® DirectX on Windows ubsystem for Linux Release Notes 支持的型号，只要大于 20.20.01.05 driver or newer 即可。
- intel: 6xx 系 HD 核显 以及更高的版本。 28.20.100.8322 driver or newer
- Nvida? cuda不香吗。

1. py36 环境安装

截至 2022 年 10 月 5 日， tensorflow-directml==1.15.8 最高支持 py37，但是其依赖 gast 0.2.2 不支持 pep517 的方式安装，推荐用 py36 版本。

若已有 py3.6 或者用 conda 安装可以跳过这一步。

```
cd $HOME
curl -Lo https://www.python.org/ftp/python/3.6.14/Python-3.6.14.tar.xz
tar xJvf Python-3.6.14.tar.xz
cd Python-3.6.14
install -dv $HOME/.local/lib/python3.6
./configure --prefix=$HOME/.local/lib/python3.6
make && make install

# 添加环境变量
install -dv $HOME/.local/bin/
ln -s $HOME/.local/lib/python3.6/bin/python3.6 $HOME/.local/bin/python36
# 检测是否成功
python36 -V
# Python 3.6.14

# 安装成功后移除安装包
rm -rf Python-3.6.14/ Python-3.6.14.tar.xz
```

2. pdm 配置

仍推荐直接用 conda，用 pdm 多半是为了节省硬盘。pdm 安装和使用可以参考 pdm 简易使用说明 pdm官方手册

自行添加需要的依赖。注意，因为使用 py3.6，所以 py 模块的编译使用的是 flit_core， 而非 pdm 认的 pdm.pep517.api。

```
install -dv PDM-Project/tensorflow-gpu
cd PDM-Project/tensorflow-gpu
# 生成PDM配置文件
cat <<'EOF' | tee pyproject.toml
[project]
name = ""
```

```
version = ""
description = ""
authors = [
    {name = "your_name", email = "your_email"},
]
dependencies = [
    "tensorflow-directml==1.15.8",
    "matplotlib==3.3.4",
    "jupyter==1.0.0",
    "scipy==1.5.4",
]
license = {text = "MIT"}
requires-python = ">=3.6.2"

[tool.pdm]

[tool.pdm.dev-dependencies]
lint = [
    "flake8",
    "black"
]

[[tool.pdm.source]]
url = "https://mirrors.aliyun.com/pypi/simple/"
verify_ssl = true
name = "pypi"

[build-system]
requires = ["flit_core >=3.2,<4"]
build-backend = "flit_core.buildapi"
'EOF'
```

然后选择 py 版本，并下载依赖包。

```
# 选择对应的python36版本
 pdm use
Please enter the Python interpreter to use
0. /home/bingo/.local/pipx/venvs/pdm/bin/python (3.8)
1. /usr/bin/python3.8 (3.8)
2. /home/bingo/.local/bin/python37 (3.7)
3. /home/bingo/.local/bin/python36 (3.6)
Please select (0): 3
Using Python interpreter: /home/bingo/.local/bin/python36 (3.6)
# 下载依赖
 pdm update --save-compatible
```

3. 结合 vscode python 工作区使用

vscode python插件从2022.10.0开始不再支持py36debug，要回滚至 v2022.8.1。

1. 创建 python 工作区，添加 python 相关开发的插件。vscode 配置、调试 python 的已经有很教程了，自行百度。

2. 将 PDM-Project 添加至工作区，方便后面复制 Path

3. 将需要写的项目添加至工作区

4. 在项目 .vscode 目录下添加 settings.json 文件（第一步的配置应该在工作区，这一步的配置只项目生效），添加 pylance 的 extra 路径配置。（如果不配置，pylance 插件会提示缺少模块文件，及没有对应的代码补全）

```
// settings.json
// 根据自己的PDM-Project/tensorflow-gpu库和py36路径调整
{
    "python.autoComplete.extraPaths": [
        "your_path/PDM-Project/tensorflow-gpu/__pypackages__/3.6/lib"
    ],
    "python.analysis.extraPaths": [
        "your_path/PDM-Project/tensorflow-gpu/__pypackages__/3.6/lib"
    ],
    "python.analysis.autoSearchPaths": true,
    "python.defaultInterpreterPath": "your_py36Path",
    "python.envFile": "${workspaceFolder}/.env",
}
```

5. 在项目 .vscode 目录下添加 .env 文件，这样子 launchF5 的时候可以关联 python.analysis.extraPaths 声明的目录到 PYTHONPATH 变量。根据实际路径调整

```
PYTHONPATH=your_path/PDM-Project/tensorflow-gpu/__pypackages__/3.6/lib
```

不推荐在 .vsocode/launch.json 中添加 env，因为对 vscode-jupyter 不生效。

6. 测试是否使用显卡（请务必用 vscode 的 luanch 运行，不然是无法链接到 python.analysis.extraPaths）

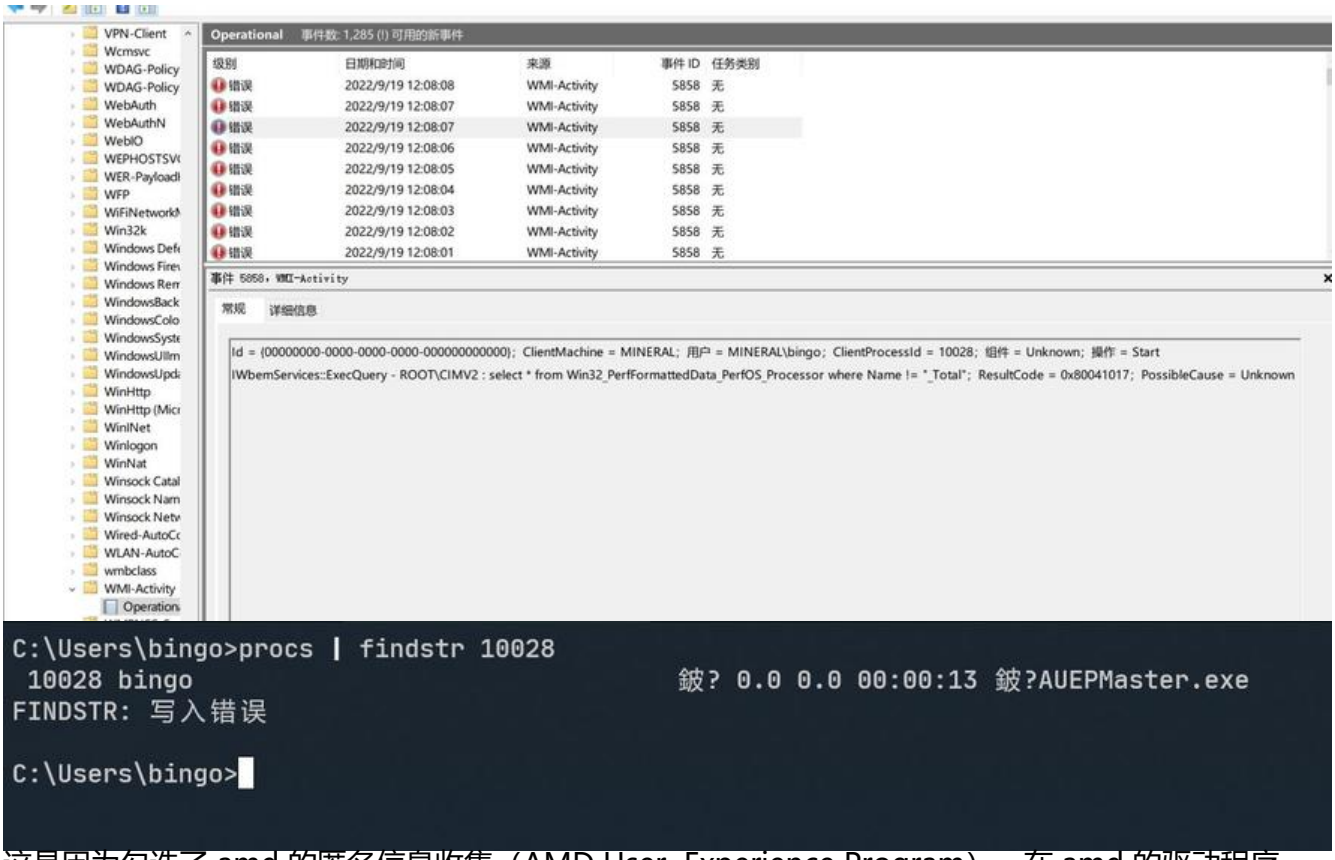
```
import tensorflow as tf

tf.test.is_gpu_available()
print(tf.test.is_gpu_available(cuda_only=False))

# output
DirectML device enumeration: found 1 compatible adapters.
DirectML: creating device on adapter 0 (AMD Radeon(TM) Graphics
True
```

性能就不测试了，只能说比用 cpu 快那么一丢丢。

注意事项

1. 安装 amd 显卡驱动后，待机时 cpu 负载异常

在WMI错误事件中锁定pid号



这是因为勾选了 amd 的匿名信息收集（AMD User Experience Program），在 amd 的驱动程序关闭收集就好了。

为什么使用 pdm?

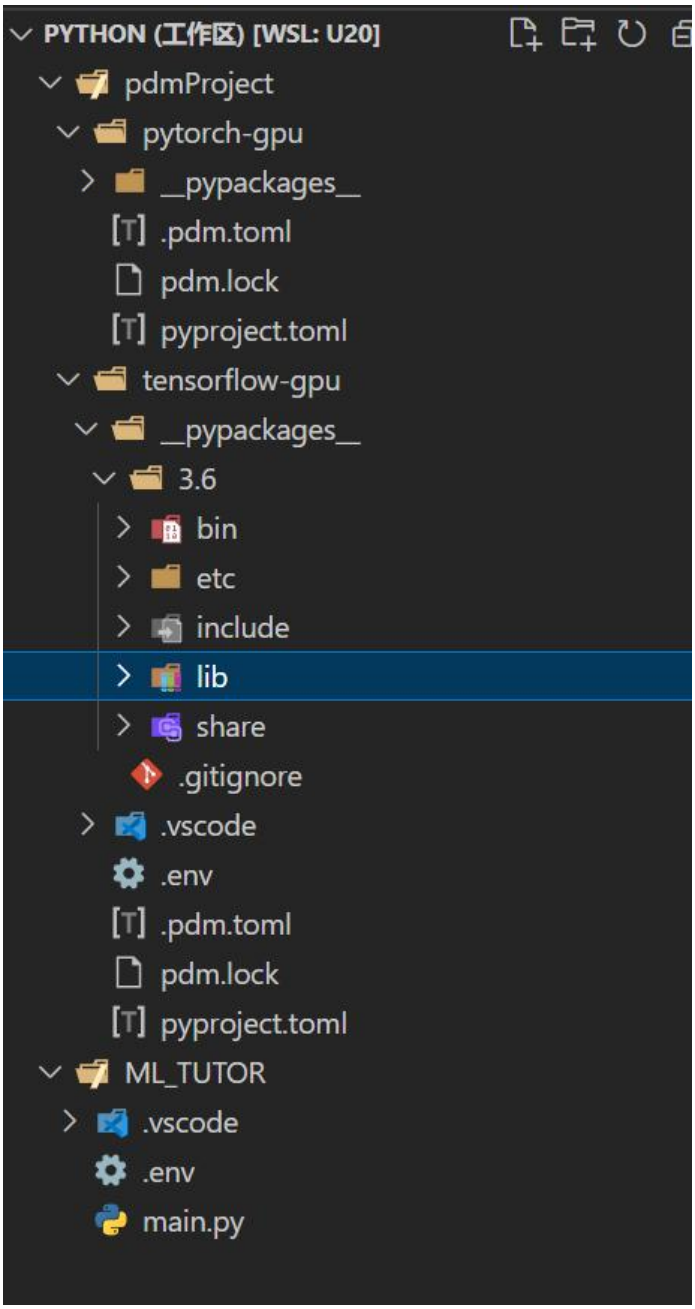
1. 节约空间，因为经常要从网上下载库，总会需要添加环境。如果是用 conda 的环境管理，每创建个虚拟环境就会多拷贝一份 python 二进制文件、标准库。以及在多数项目中存在相同依赖包，在虚环境中，都是存在冗余的。

pdm 并不创建虚拟环境（除了 pdm 运行本身会创建虚拟环境），并且模块库是通过 PEP582 本地接的方式，可以极大节省硬盘空间（需要开启配置 install.cache = True 和 install.cache_method = ymlink）。(主要是我的 wsl2 装在固态硬盘上，能省则省)

```
# 可以看到下载的模块文件基本都是软连接
 ls -al your_path/PDM-Project/tensorflow-gpu/__pypackages__/3.6/lib

# 使用pdm后，依赖包的总空间占用大小
 du -sh ~/codeEnv/pyCache/packages/
1.4G    /home/bingo/codeEnv/pyCache/packages/
```

2. 有没有好方法复用当前配置好的项目开发环境?

pdm2.0 推出了虚拟环境（本质上还是基于 virtualenv/venv/conda），我并不是很推荐这个。如果有更改依赖的话，推荐将配置好的 .vscode 目录和 .env 目录拷贝到 PDM-Project 对应的项目目录。后续要复用直接将这两个 copy 到新的项目目录下即可。



原文链接：核显 DirectML 深度学习环境