Что нужно сделать чтобы собрать этот проект еще раз

0) Бросить это делать на python и идти делать на Java / C#

Это намного проще и надежнее.

Если все же Вы решили сделать это на Python, Вам определенно помогут ссылки:

* Пошаговая инструкция по установке от разработчика на Windows (native): <https://www.tensorflow.org/install/pip#windows-native_1> ,
* Пошаговая инструкция по установке от разработчика на Windows WSL: <https://www.tensorflow.org/install/pip#windows-wsl2_1> .

1) Определиться с версией Tensorflow

Последняя версия Tensorflow, которая умеет работать с GPU на Windows без бубна – 2.10.0 (в WSL может быть затруднительно установить и интегрировать CUDA с разрабатываемым программным средством).

2) Определиться с версией Embedded Python

Протестированные разработчиками билды находятся здесь:

* Linux / macOS: <https://www.tensorflow.org/install/source> ,
* Windows: <https://www.tensorflow.org/install/source_windows> .

Для версии Tensorflow 2.10.0:



Я выбрал python 3.9.

3) Скачать Python выбранной версии

Релизы выбранного мною python находятся здесь:

* <https://www.python.org/downloads/release/python-390>/ .

Мой выбранный Python может быть скачан со страницы по ссылке:

* https://www.python.org/ftp/python/3.9.0/ .

В данной инструкции будет установлен и настроен Embeddable Python.

4) Определиться с версией CUDA и cuDNN

Протестированные разработчиками билды c GPU находятся здесь:

* Linux / macOS: <https://www.tensorflow.org/install/source#gpu> ,
* Windows: <https://www.tensorflow.org/install/source_windows#gpu> .

Для версии Tensorflow 2.10.0 Windows:



5) Скачать CUDA и cuDNN

Для скачивания cuDNN из архива NVIDIA потребуется авторизация. CUDA 11.2 предназначена для работы под управлением ОС Windows 10 и совместима с Windows 11. Ссылки для скачивания:

* CUDA 11.2: <https://developer.download.nvidia.com/compute/cuda/11.2.0/local_installers/cuda_11.2.0_460.89_win10.exe> ,
* cuDNN 8.1.0 для CUDA 11.2: <https://developer.nvidia.com/compute/machine-learning/cudnn/secure/8.1.0.77/11.2_20210127/cudnn-11.2-windows-x64-v8.1.0.77.zip> .

6) Установка и настройка Embeddable Python

Embedded Python не содержит в себе модуля pip для загрузки необходимых зависимостей. Исправить это можно, если исполнить скрипт get\_pip.py на этом интерпретаторе. Сам скрипт может быть найден по ссылке:

* get-pip.py: <https://bootstrap.pypa.io/get-pip.py> .

После этого в папке с Python должны появиться папки

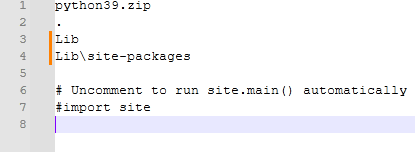
* Libs,
* Scripts.

В папке Libs хранятся установленные извне пакеты. Далее необходимо прописать новые пути источники пакетов в sys.path Python. Проще всего это сделать изменением файла python{some\_version}.\_pth в папке с Python.

Это текстовый файл, в который нужно дописать следующие строки:

* Lib ,
* Lib/site-packages .

Должно получиться примерно так:



Теперь можно установить дополнительные модули. В качестве теста можно установить, например, модуль rx. Для этого используйте команду консольной строки

python -m pip install rx

Установите любые необходимые зависимости таким образом.

7) Установка CUDA и cuDNN

CUDA достаточно просто устанавливается из GUI. В одном окружении не может быть установлено одновременно несколько версий CUDA, поэтому при возникновении необходимости установить раннюю версию CUDA текущую следует удалить (через “Установка и удаление программ” в Windows).

Установке cuDNN на Windows посвящена страница на сайте NVIDIA:

* <https://docs.nvidia.com/deeplearning/cudnn/latest/installation/windows.html>

Но проще всего переместить файлы из архива cuDNN в папку с CUDA.

В результате установки должны быть определены переменные среды:

* CUDA\_PATH,
* CUDA\_PATH\_V{some\_version}.

В переменной среды PATH должны появиться следующие сегменты:

* %CUDA\_PATH%/bin,
* %CUDA\_PATH%/libnvvp.

Если cuDNN был установлен не в одной папке с CUDA, то путь к cuDNN аналогочно нужно прописать в PATH.

8) Установка Tensorflow версии 2.10.0

Для установки Tensorflow версии 2.10.0 необходимо выполнить следующую команду на Embedded Python:

python -m pip install tensorflow==2.10.0

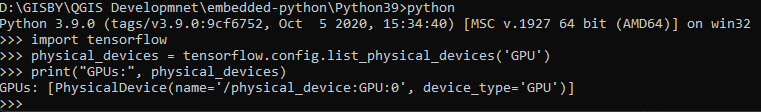
После чего pip установит Tensorflow со всеми его зависимостями (10.07.2024 на Windows 11 это почти 1.5 Гб).

Убедитесь, что Tensorflow правильно установлен. Для этого выполните следующий код:

import tensorflow

physical\_devices = tensorflow.config.list\_physical\_devices('GPU')

print("GPUs:", physical\_devices)

Результат выполнения программы должен быть примерно таким:

Может возникнуть ошибка, связанная с тем, что билд Tensorflow откажется работать с Numpy версии 2+, а на Python установлена именно такая версия. В этом случае необходимо установить пакет Numpy более ранней версии. Для этого выполните консольную команду:

python -m pip install --force-reinstall -v "numpy<2"