環境構築

2020/07/19

まずは本稿のコードを実行できるための環境構築について、説明は windows 10 に基づくが、Linux や Mac に関して、必要なソフトウェアが基本同じですので、本稿も参考になると思います。

keras は本来、python というプログラミングのライブラリなので、r 上の keras は r のコードを python のコードに変換して動かします。そのため r keras には python と python のライブラリ TensorFlow が必要です。TensorFlow が必要の理由は keras 自身は高水準のニューラルネットワークライブラリですので、TensorFlow がそのバックエンドとして低水準のサポートを提供しています。(keras のバックエンドは TensorFlow のほかに、CNTK と Theano も使えますが、本稿は TensorFlow を使用します)。ですので、python のライブラリ TensorFlow もインストールする必要があります。

また、r、python、keras も TensorFlow もまだ更新中のツールですので、バッジョの更新とともに、古いコードが使えなくなる可能性があります。それを避けるために、r keras は conda という環境管理システムに使用しています。conda を利用することによって、様々なバッジョの python や、keras が共存することが可能となります。古いコードは古いバッジョのソフトウェア環境を使えば、この問題を避けられます。

さらに、GPU (Graphics Processing Unit, グラフィックスプロセッシングユニット) というプロセッサはディープラーニングに関する演算は、通常の CPU より 10 倍以上の計算力を持つため、keras のフルパワーを発揮するために、TensorFlow の GPU バッジョを使用するのがおすすめです。ただし、残念なことは TensorFlow が支持する GPU は NVIDIA 社制の一部しかない、サポートされる GPU はこちらの URLhttps://developer.nvidia.com/cuda-gpusに確認できます。また、GPU を動かすには、NVIDIA 社が開発する GPU プログラミングためのツール、CUDA と GPU がディープラーニング演算を加速するためにライブラリ、cuDNN が必要です。TensorFlow の GPU バッジョを使用するために、CUDA と cuDNN もインストールしましょう。ここで注意して欲しいことがあります、2019 年 11 月から、NVIDIA 社は新しい CUDA バッジョが macn いサポートしないと公表しましたので、新しい TensorFlow GPU バッジョが mac に使えないと考えたほうがいいです。

表1では、必要のツールをまとめていました。

表1　必要のツール一覧

ツール	バッジョ	備考
R	4.02	
rtools	4.0	r keras が必要のツール
r keras	2.3.0.0	
Miniconda/Anaconda		conda 環境
python	3.7.7	
Tensorflow	2.20	keras のバックエンド
GPU バッジョ追加ツール		
CUDA	10.1	GPU プログラミングツール
cuDNN	7.6.4	ディープラーニング演算 を加速するための CUDA のライブラリ

0.1　TensorFlow GPU 環境の構築

まずは keras GPU バッジョの環境構築について説明します。CPU バッジョを使いますなら、この節を飛ばして結構です。前節に述べた通り、TensorFlow GPU を使うには、CUDA と cudnn が必要です。ただし、TensorFlow GPU のバッジョと CUDA、cudnn のバッジョが厳密に対応しなければなりません。間違ったバッジョでは動きませんので、気をつけてください。TensorFlow の公式サイトのソースからのビルドページ (https://www.tensorflow.org/install/source_windows)では TensorFlow GPU と CUDA、cudnn 対応バッジョの関係が書かれていますので、参照してください (日本語のページの更新が遅

Version	Python version	Compiler	Build tools	cuDNN	CUDA
tensorflow_gpu-2.2.0	3.5-3.8	MSVC 2019	Bazel 2.0.0	7.4	10.1
tensorflow_gpu-2.1.0	3.5-3.7	MSVC 2019	Bazel 0.27.1-0.29.1	7.4	10.1
tensorflow_gpu-2.0.0	3.5-3.7	MSVC 2017	Bazel 0.26.1	7.4	10
tensorflow_gpu-1.15.0	3.5-3.7	MSVC 2017	Bazel 0.26.1	7.4	10
tensorflow_gpu-1.14.0	3.5-3.7	MSVC 2017	Bazel 0.24.1-0.25.2	7.4	10
tensorflow_gpu-1.13.0	3.5-3.7	MSVC 2015 update 3	Bazel 0.19.0-0.21.0	7.4	10
tensorflow_gpu-1.12.0	3.5-3.6	MSVC 2015 update 3	Bazel 0.15.0	7	9
tensorflow_gpu-1.11.0	3.5-3.6	MSVC 2015 update 3	Bazel 0.15.0	7	9
tensorflow_gpu-1.10.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	7	9
tensorflow_gpu-1.9.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	7	9
tensorflow_gpu-1.8.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	7	9
tensorflow_gpu-1.7.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	7	9
tensorflow_gpu-1.6.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	7	9
tensorflow_gpu-1.5.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	7	9
tensorflow_gpu-1.4.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	6	8
tensorflow_gpu-1.3.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	6	8
tensorflow_gpu-1.2.0	3.5-3.6	MSVC 2015 update 3	Cmake v3.6.3	5.1	8
tensorflow_gpu-1.1.0	3.5	MSVC 2015 update 3	Cmake v3.6.3	5.1	8
tensorflow_gpu-1.0.0	3.5	MSVC 2015 update 3	Cmake v3.6.3	5.1	8

図1　TensorFlow GPU と CUDA、cudnn の対応関係

いので、新しい TensorFlow の対応バッジョが書かれていない場合は英語のページをご覧ください)。

図1から、今回使用する tensorflow_gpu-2.2.0 に対応する python のバッジョは 3.5-3.8、CUDA が 10.1、cuDNN が 7.4 と確認できましたので、NVIDIA のページから CUDA 10.1（https://developer.nvidia.com/cuda-10.1-download-archive-base）と cuDNN 7.6（https://developer.nvidia.com/rdp/cudnn-archive）をダウンロードします。cuDNN をダウンロードする際に、まず NVIDIA 社のアカウントを作る必要がありますでの、アカウントを作ってから、ダウンロードしましょう。

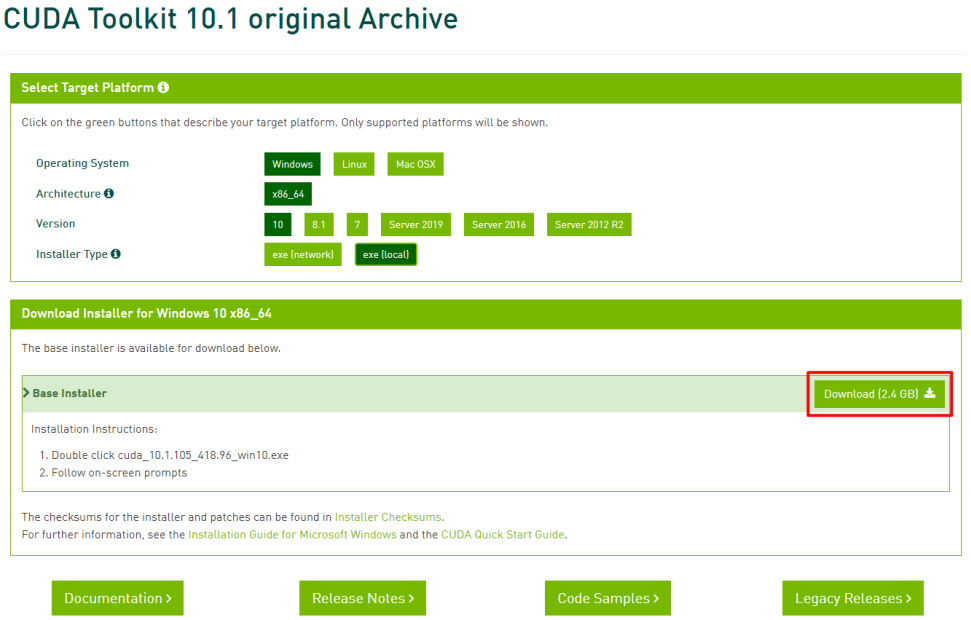

図2　CUDA 10.1 のダウンロードページ

ダウンロードしたら、まず CUDA をインストールします、途中の選択肢に関して、すべてデフォルトにしてもかまいません。CUDA のインストールが終わたら、ダウンロードの cuDNN の zip ファイルを解凍してください。すると、「CUDA」というフィルターが得られる、ファイル内には「bin」、「include」、「lib」という名前にフィルターと「NVIDIA_SLA_cuDNN_Support.txt」という名前のファイルがありますはず。この 4 つを先 CUDA のインストールフィルターにコピーしてください、筆者の CUDA は「`C:\ProgramFiles\NVIDIAGPUComputingToolkit\CUDA\v10.1`」にありますので、ここにコピーします、コピーする際に同じ名前のファイルが存在するという提示にした、ファイルを書き換えを選択してください。

そして重要のステップ、システムの環境変数に CUDA と cuDNN を導入します。スタートメニューの「windows システムツール」の「コントロールパネル」を開いて、システムを選択してください、開いたページの左のシステム詳細設定をクリックしてください。そしてさらに開いたウィンドウの環境変数ボタンをクリックすると、環境変数編集ウィンドウが開きます。下部分のシステム環境変数のところに、新規をクリックし、変数名を「CUDA_PATH」

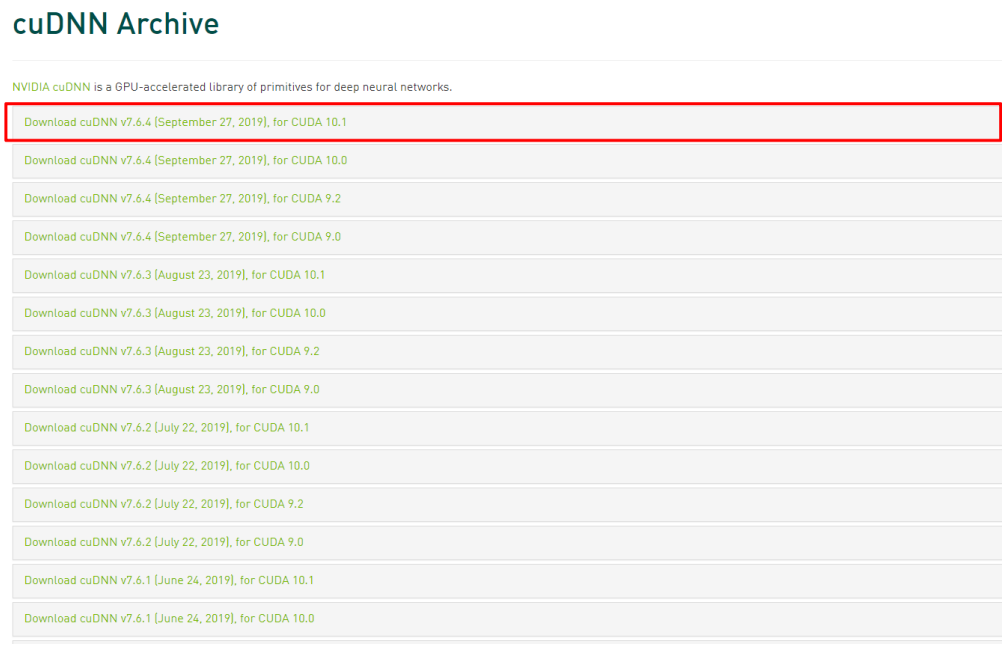

図3　cuDNN 7.6.4 のダウンロードページ

にして、ディレクトリの参照を開いて、CUDA のインストールパスを選択してください。続きはシステム環境変数にある「path」をダブルクリックして、開いたウィンドウに新規をクリックし、参照をクリックしたら、CUDA のインストールパスの下の「bin」というファイルに選択してください。

パス追加の成功を確認するために、「コントロールパネル」と同じく、「windows システムツール」下の「コマンドプロンプト」を開いて、「where cudnn64_7.dll」と入力し、「cudnn64_7.dll」のパスが戻ったら成功です。

0.2　python 環境の構築

まずは python の環境管理システム「conda」のインストールです、「conda」はメインに「Miniconda」と「Anaconda」二つのバッジョがあります。「Anaconda」は「conda」以外、デフォルトの「python」環境にデータ分析のための python のライブラリ多数が付属しているに対して、「Miniconda」は「conda」と「python」以外にライブラリを何もつけていません。ですので、「python」にも勉強したいなら「Anaconda」、r keras を動かすためだけなら、「Miniconda」があれば十分です。「Miniconda」を例に説明します。

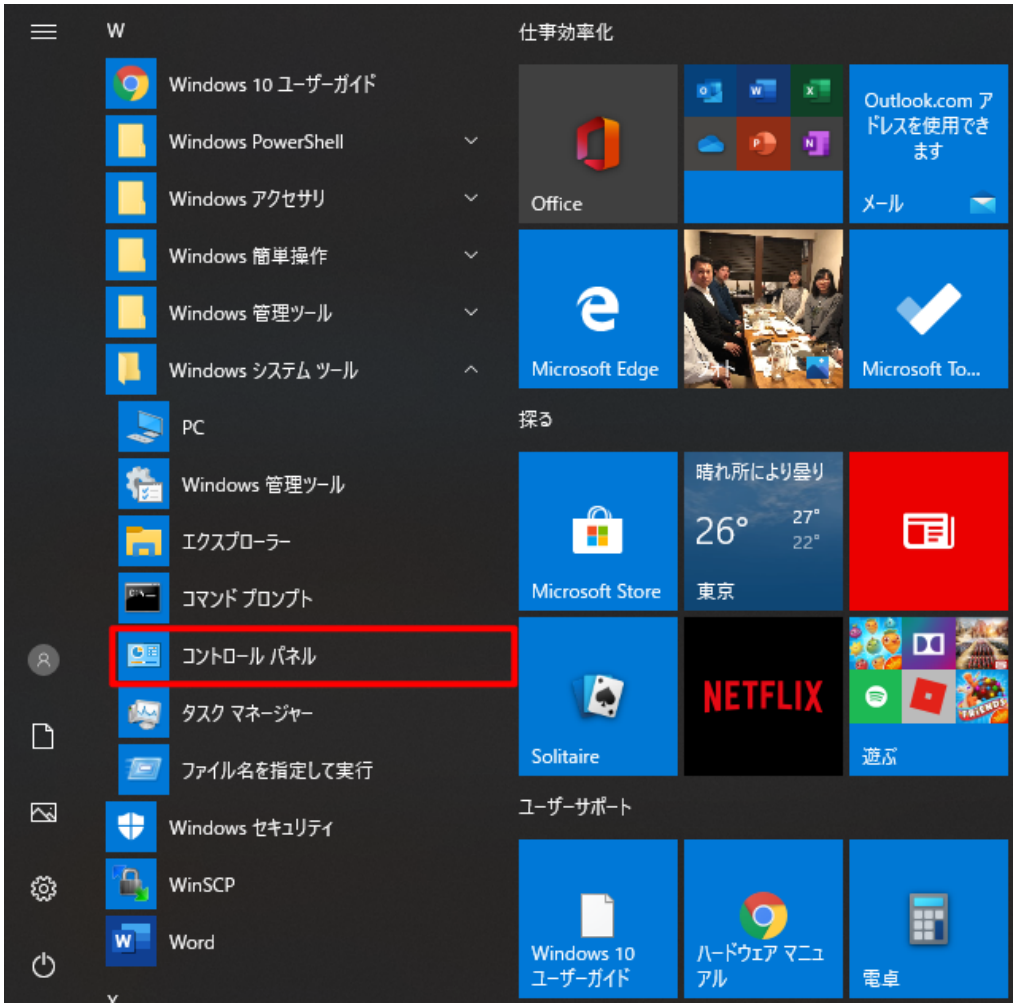

図4

「Miniconda」のダウンロードページに Python 3.7 バッジョが選択し、ダウンロードし、インストールします、インストールするときに注意するところは、Install For のところに「Just Me」を選択するのがをすすめと、Advanced Options のページに、必ず「Add Miniconda3 to PATH environment variable」にチャックを入れてください。最後に「コマンドプロン

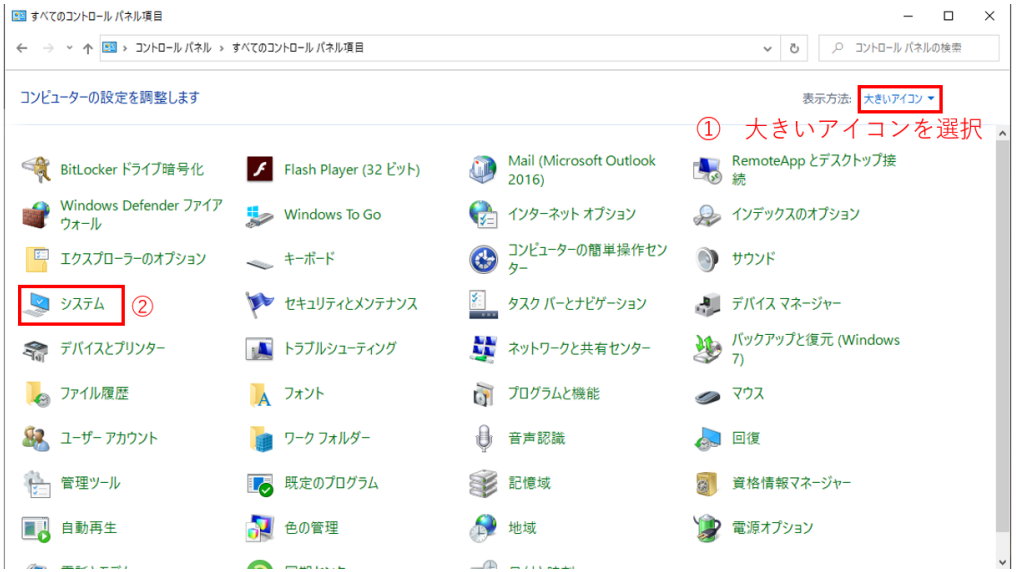

図5

プト」に「conda」というコマンドを入れて、「usage: conda-script.py [-h] [-V] command ...」のような提示があればインストールが成功です。

これからは r keras パッケージの「install_keras()」という関数を利用すれば、残りの環境が自動的に整ます、しかし、時に、GPU バッジョを使う場合に、未知のトラブルが多いので、「install_keras()」を利用すると、r、conda、TensorFlow どちらにトラブルがあるのが判断しつらくなるため、本稿はまず python と TensorFlow の環境を手動に整えて、上手くいったら r keras の環境に導入するのがおすすめです。

「conda」が無事にインストールできたら、「コマンドプロンプト」に「conda create -n r-reticulate python=3.7」を実行してください、このコマンドは「conda」に「r-reticulate」という名前の環境の新規作成し、その環境の python のバッジョを 3.7 と指定するという意味です。そして conda はインストールするものを提示し、「y」と確認したらインストールされます。そして「conda info -e」というコマンドを執行すると、今ある環境の情報を表示します、「r-reticulate」があれば成功です。成功したら、コマンド「activate r-reticulate」で「r-reticulate」環境に入ります。そこで「pip install tensorflow」コマンドで TensorFlow をインストールします。注意してほしいのが、TensorFlow が 2.2 から GPU と CPU が区別

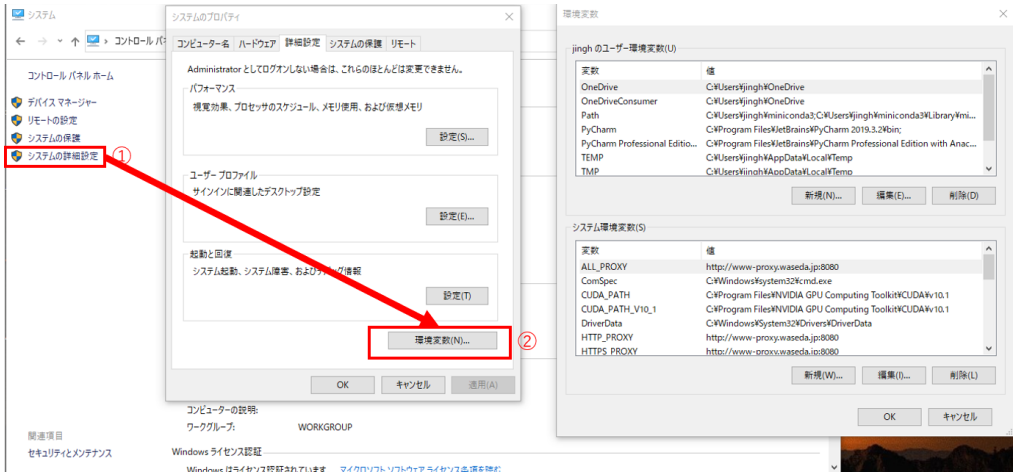

図6

しないことになりました、もしそれ以前の GPU バッジョの TensorFlow をインストールする際に、「pip install tensorflow-gpu」と実行してください。無事インストールするができたら、コマンド「python」を執行し、以下のコードを執行してください。

```
import tensorflow as tf
tf.random.normal([4,4])
```

エラーが表示せず、図17のように、マトリックスが返すなら、TensorFlow のインストールが成功します。

0.3 r 環境の構築

r と rtools のインストール方法が省略しますが、r に「install.packages(“keras”)」を実行すると r keras がインストールされます、そして「keras::use_condaenv(“r-reticulate”)」コマンドで r keras が使用する python の環境を先インストールした「r-reticulate」にしてします。何もエラーが表示されませんでしたら、これで無事に r keras と python、TensorFlow の連携が成功しました。

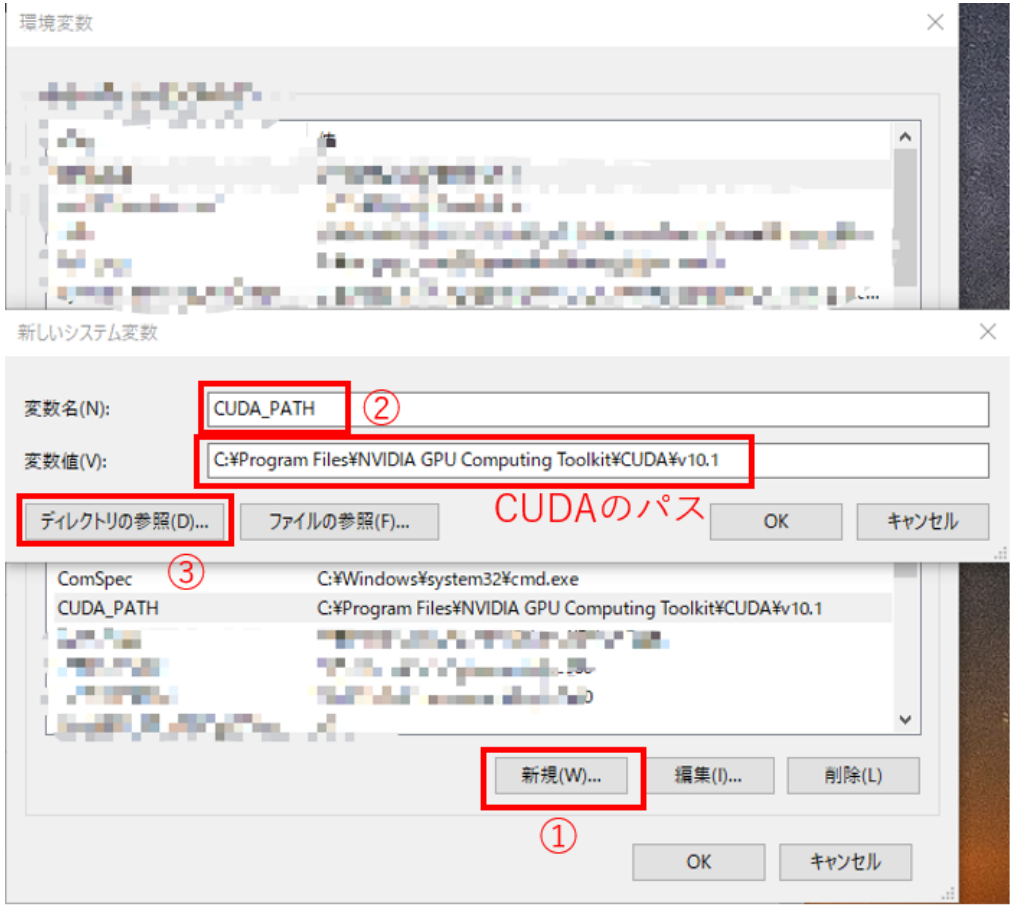
環境変数
新しいシステム変数
変数名(N):
CUDA_PATH
②
変数値(V):
C:¥Program Files¥NVIDIA GPU Computing Toolkit¥CUDA¥v10.1
CUDAのパス
ディレクトリの参照(D)...
ファイルの参照(F)...
OK
キャンセル
③
ComSpec
C:¥Windows¥system32¥cmd.exe
CUDA_PATH
C:¥Program Files¥NVIDIA GPU Computing Toolkit¥CUDA¥v10.1
新規(W)...
編集(I)...
削除(L)
①
OK
キャンセル

図7

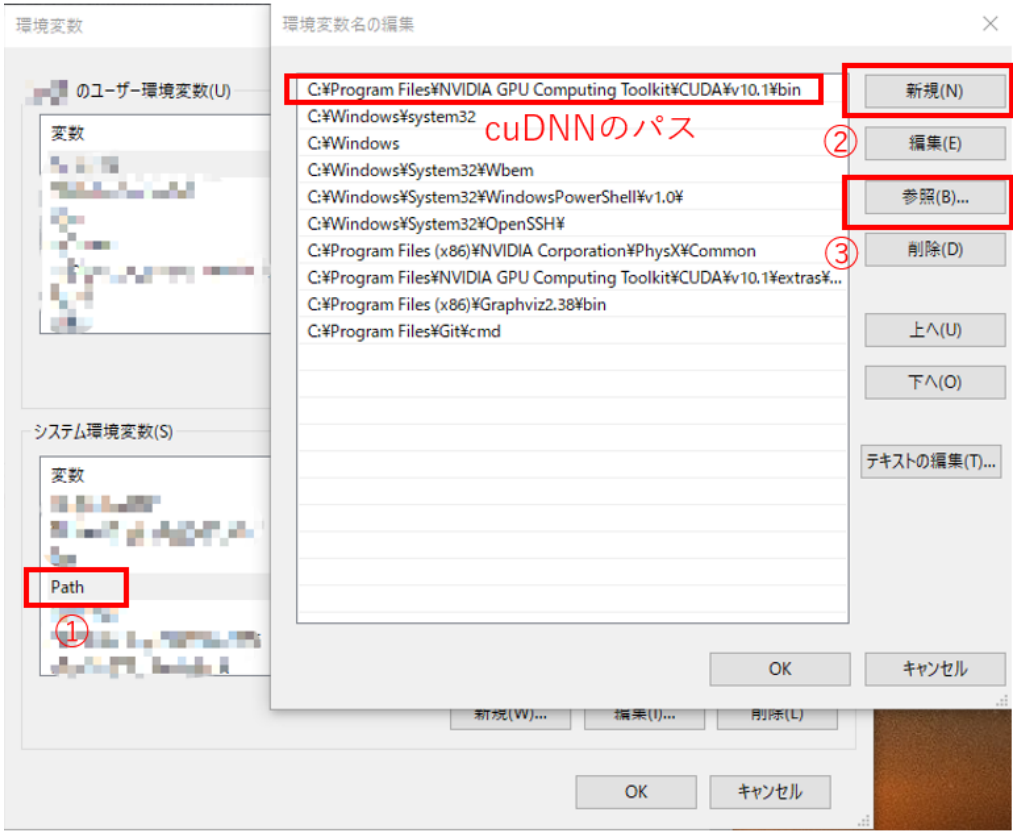
環境変数
のユーザー環境変数(U)
変数
システム環境変数(S)
変数
Path
①
環境変数名の編集
C:¥Program Files¥NVIDIA GPU Computing Toolkit¥CUDA¥v10.1¥bin
C:¥Windows¥system32
C:¥Windows
C:¥Windows¥System32¥Wbem
C:¥Windows¥System32¥WindowsPowerShell¥v1.0¥
C:¥Windows¥System32¥OpenSSH¥
C:¥Program Files (x86)¥NVIDIA Corporation¥PhysX¥Common
C:¥Program Files¥NVIDIA GPU Computing Toolkit¥CUDA¥v10.1¥extras¥...
C:¥Program Files (x86)¥Graphviz2.38¥bin
C:¥Program Files¥Git¥cmd
cuDNNのパス
②
③
新規(N)
編集(E)
参照(B)...
削除(D)
上へ(U)
下へ(O)
テキストの編集(T)...
OK
キャンセル
OK
キャンセル

図8

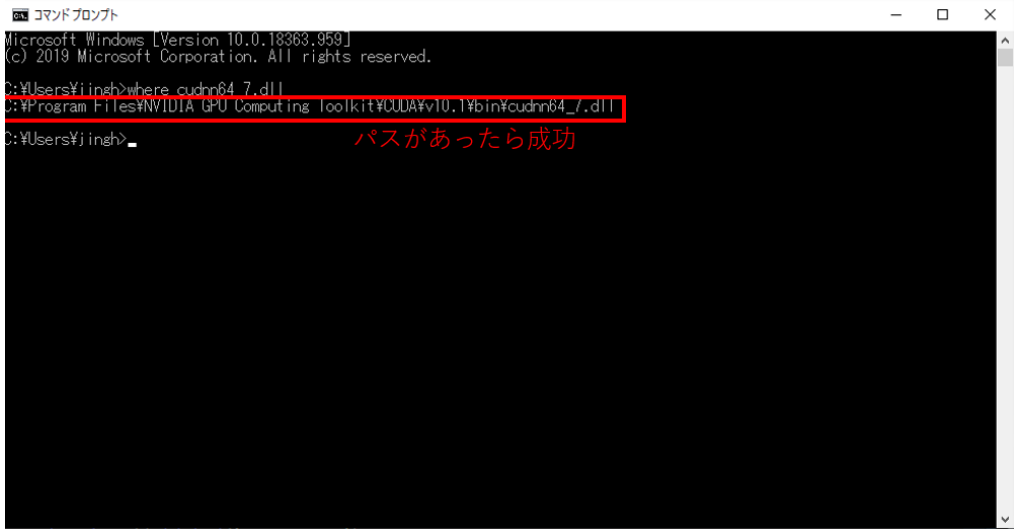
コマンドプロンプト
Microsoft Windows [Version 10.0.18363.959]
(c) 2019 Microsoft Corporation. All rights reserved.
C:¥Users¥jingh>where cudnn64_7.dll
C:¥Program Files¥NVIDIA GPU Computing Toolkit¥CUDA¥v10.1¥bin¥cudnn64_7.dll
C:¥Users¥jingh>
パスがあったら成功

図9

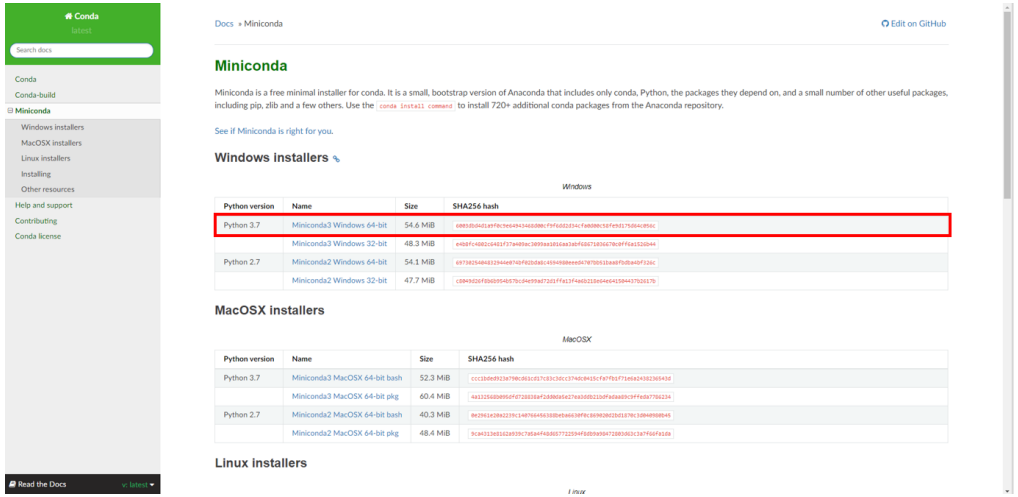
Conda
Miniconda
Windows installers
MacOSX installers
Linux installers
Docs » Miniconda
Edit on GitHub
Miniconda
Windows installers
MacOSX installers
Linux installers
Python version
Name
Size
SHA256 hash
Python 3.7
Miniconda3 Windows 64-bit
54.6 MiB
Miniconda3 Windows 32-bit
48.3 MiB
Python 2.7
Miniconda2 Windows 64-bit
54.1 MiB
Miniconda2 Windows 32-bit
47.7 MiB
Miniconda3 MacOSX 64-bit bash
52.3 MiB
Miniconda3 MacOSX 64-bit pkg
60.4 MiB
Miniconda2 MacOSX 64-bit bash
40.3 MiB
Miniconda2 MacOSX 64-bit pkg
48.4 MiB

図10

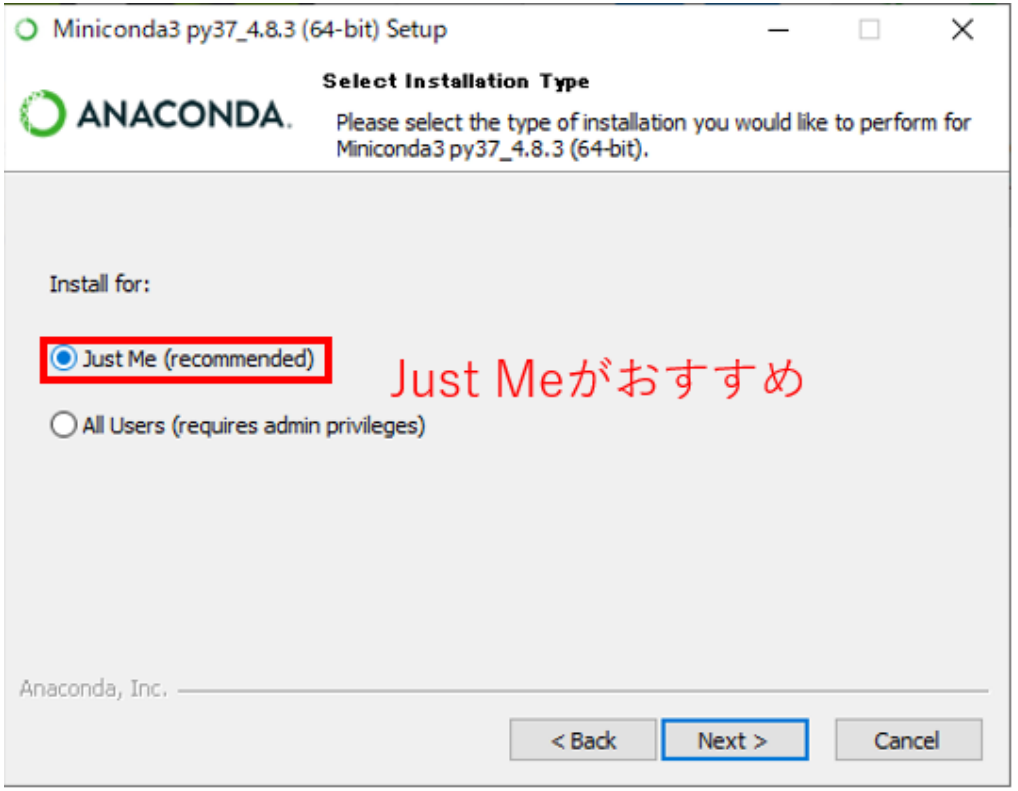
Miniconda3 py37_4.8.3 (64-bit) Setup
ANACONDA.
Select Installation Type
Please select the type of installation you would like to perform for Miniconda3 py37_4.8.3 (64-bit).
Install for:
Just Me (recommended)
Just Meがおすすめ
All Users (requires admin privileges)
Anaconda, Inc.
< Back
Next >
Cancel

図11

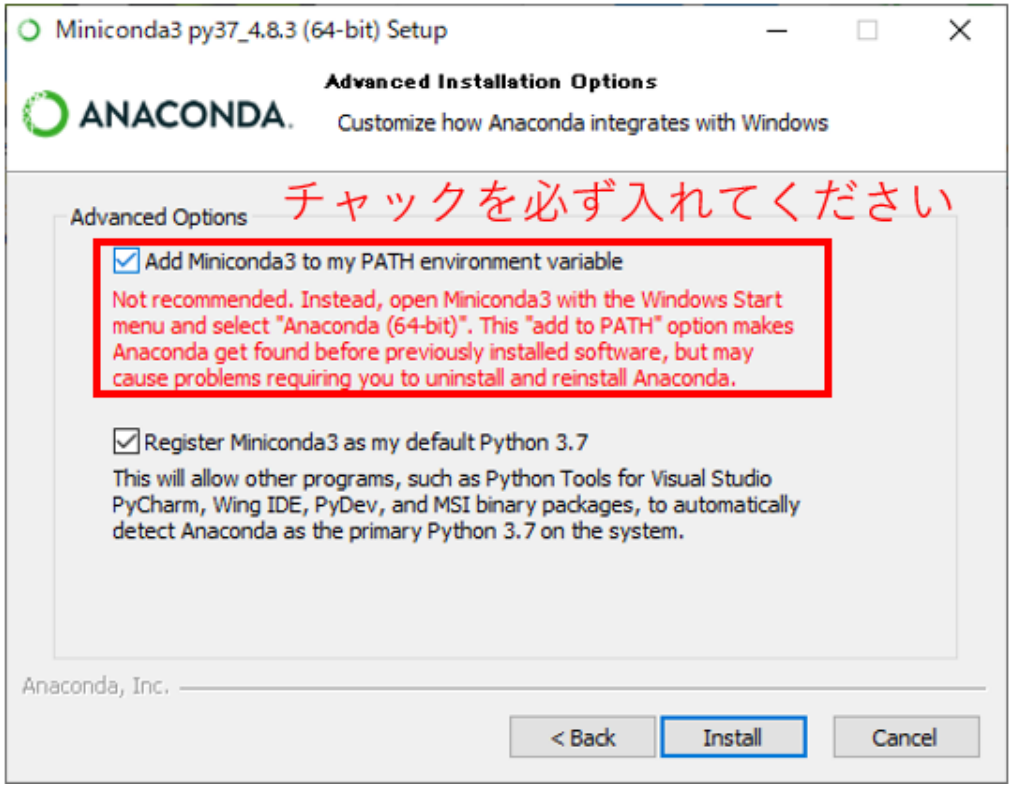
Miniconda3 py37_4.8.3 (64-bit) Setup
ANACONDA.
Advanced Installation Options
Customize how Anaconda integrates with Windows
チャックを必ず入れてください
Advanced Options
Add Miniconda3 to my PATH environment variable
Not recommended. Instead, open Miniconda3 with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.
Register Miniconda3 as my default Python 3.7
This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.7 on the system.
Anaconda, Inc.
< Back
Install
Cancel

図12

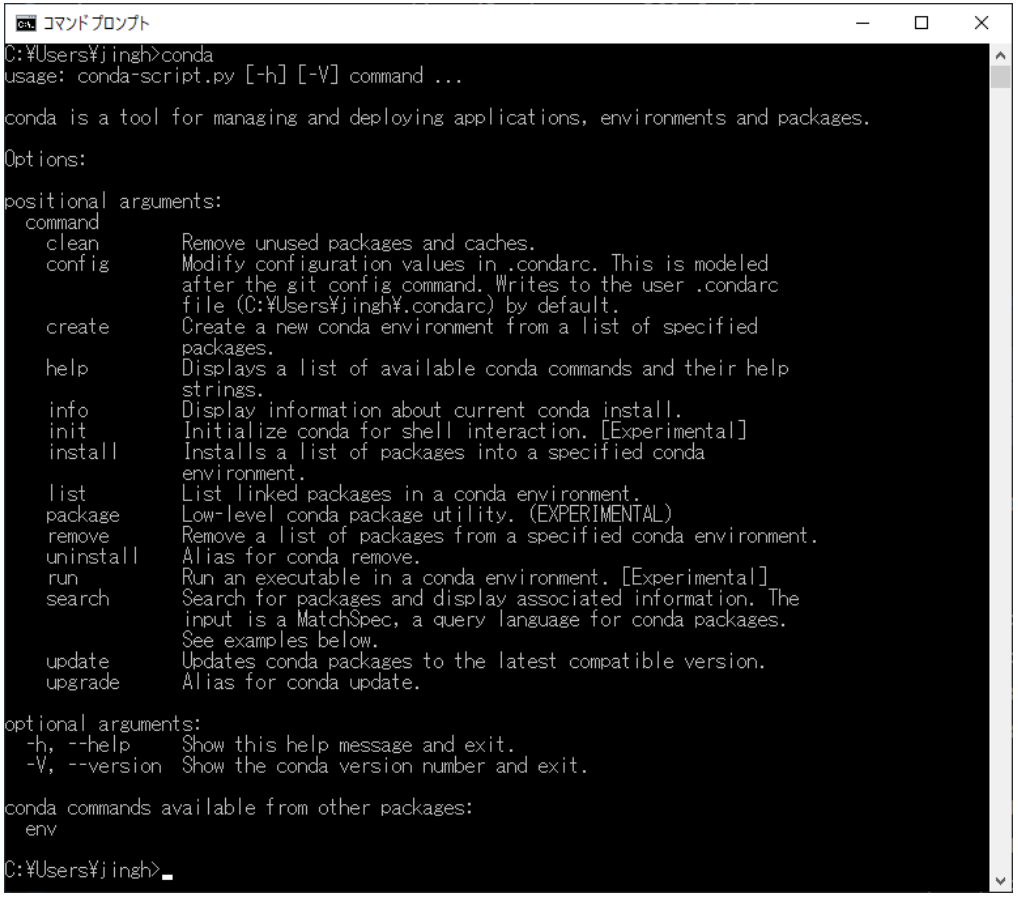
コマンド プロンプト
C:¥Users¥jingh>conda
usage: conda-script.py [-h] [-V] command ...
conda is a tool for managing and deploying applications, environments and packages.
Options:
positional arguments:
command
clean Remove unused packages and caches.
config Modify configuration values in .condarc. This is modeled after the git config command. Writes to the user .condarc file (C:¥Users¥jingh¥.condarc) by default.
create Create a new conda environment from a list of specified packages.
help Displays a list of available conda commands and their help strings.
info Display information about current conda install.
init Initialize conda for shell interaction. [Experimental]
install Installs a list of packages into a specified conda environment.
list List linked packages in a conda environment.
package Low-level conda package utility. (EXPERIMENTAL)
remove Remove a list of packages from a specified conda environment.
uninstall Alias for conda remove.
run Run an executable in a conda environment. [Experimental]
search Search for packages and display associated information. The input is a MatchSpec, a query language for conda packages. See examples below.
update Updates conda packages to the latest compatible version.
upgrade Alias for conda update.
optional arguments:
-h, --help Show this help message and exit.
-V, --version Show the conda version number and exit.
conda commands available from other packages:
env
C:¥Users¥jingh>

図13

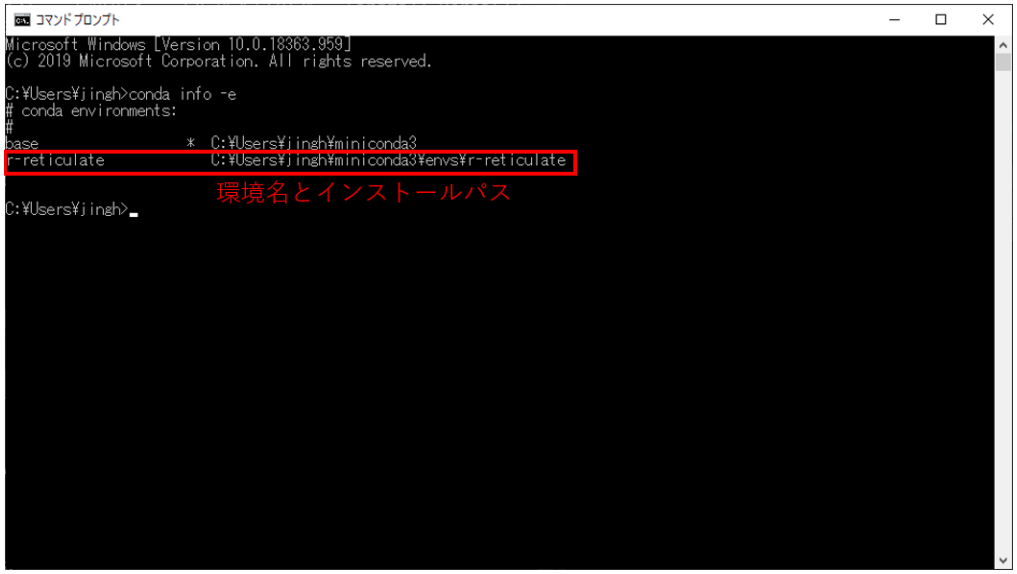

図14

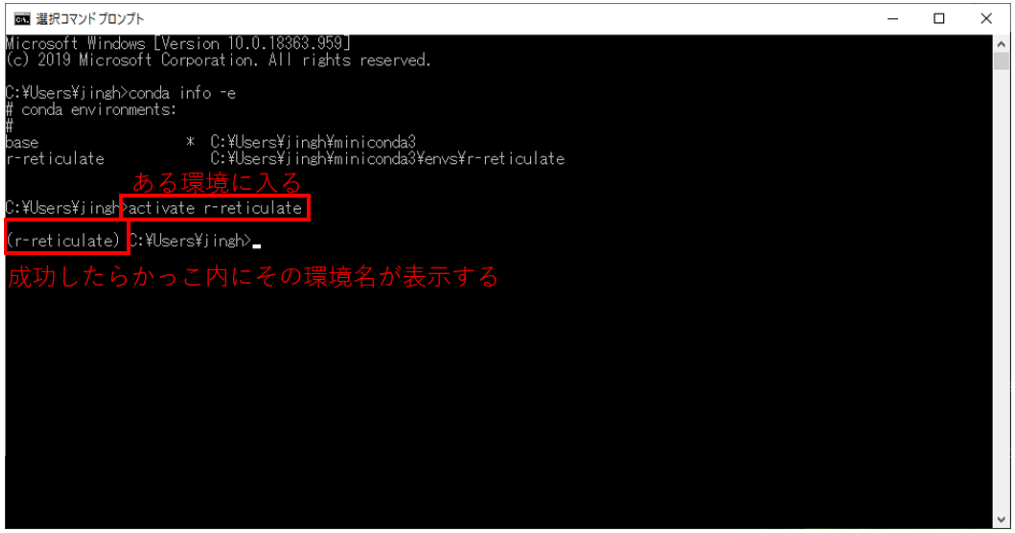

図15

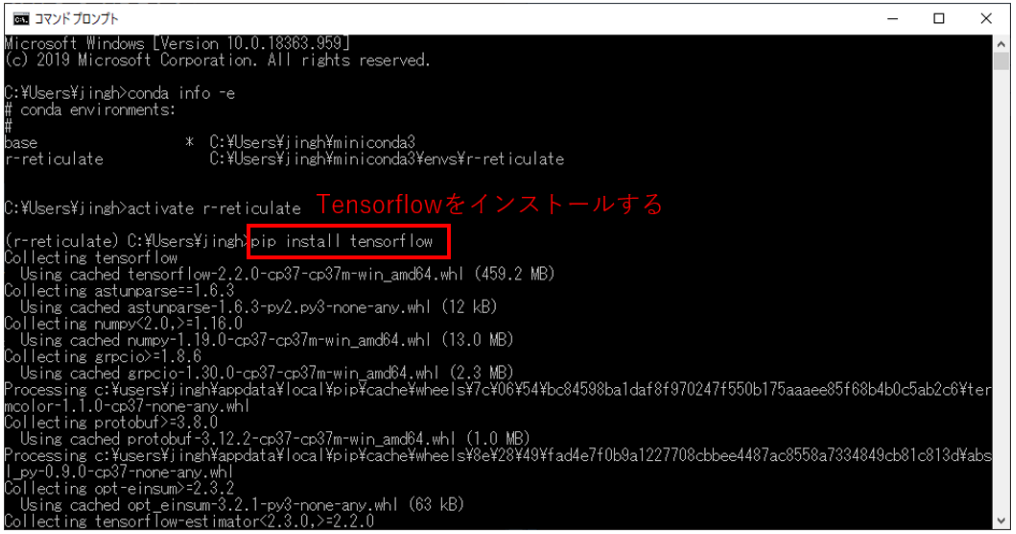

図16

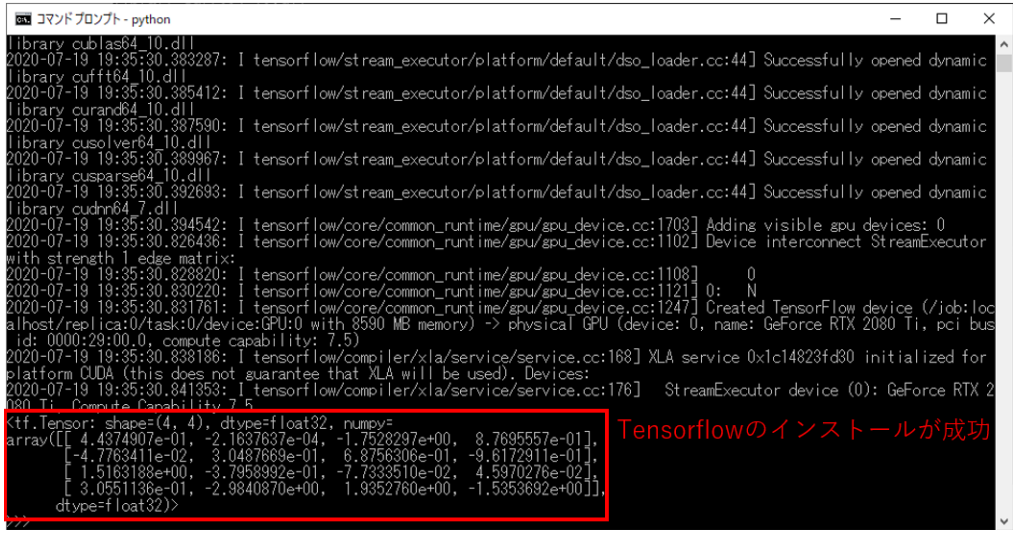

図17