

2020年9月22日 開催

ハンズオンセミナー

「外科領域における医療用画像の深層学習」

事前準備手順

2020年8月18日版

作成：名古屋大学 小田昌宏

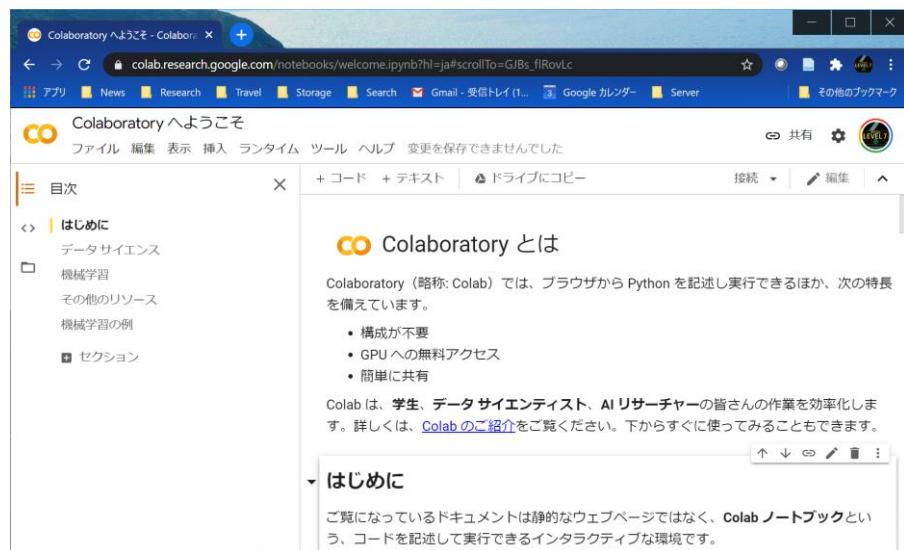
2020年3月開催を予定して配布した handson_install3.pdf に沿ってご準備された方も、お手数ですが再度本資料に沿って作業を行ってください。
現在の最新の深層学習実行環境を準備するためです。

本セミナーはクラウド環境及びローカル環境での深層学習実行を行います。クラウド環境及びローカル環境それぞれ準備が必要ですので、セミナー参加前に本資料に従いご準備お願いします。

クラウド環境準備手順

Google アカウントを作成してください。既存のアカウントがあれば作成不要です。セミナー中に Google Colaboratory を使用する際に必要です。

セミナー中はなるべく高速なインターネット接続を使用してください。



Google Colaboratory の画面例

□一カル必要環境

- Windows10 がインストールされた PC
- メモリ 8GB 以上必須, 16GB 以上推奨
- インターネット接続
- Windows のユーザ名に日本語等の全角文字を使用していないこと. 使用していたら, 半角文字だけからなるユーザ名のアカウントを作成してサインインしてください.

上記環境の PC をご用意ください. CPU は速いものの方が良いです.

Windows 以外の OS を使用する場合は自己責任でインストール等行ってください.

□一カル環境準備手順

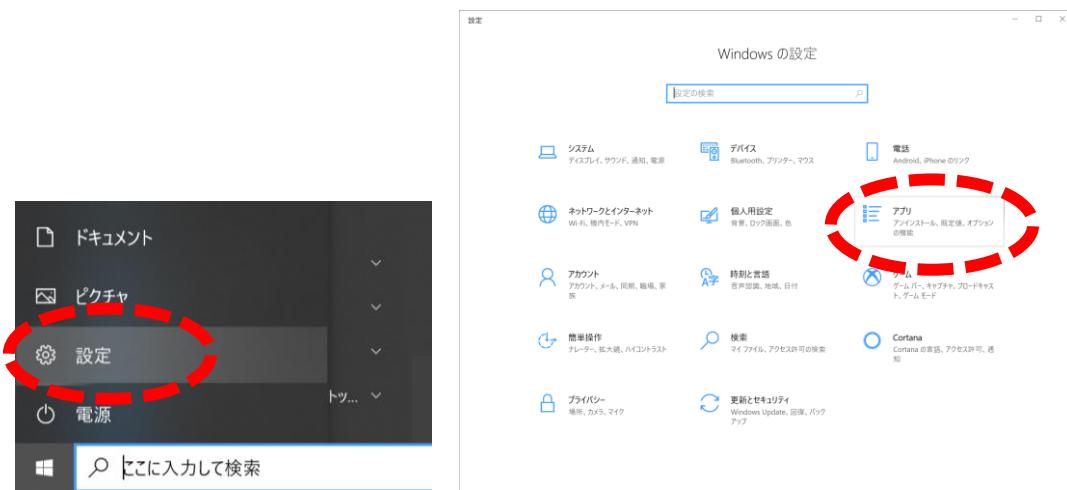
ハンズオンセミナーの日までに次ページから示す手順でインストールを行ってください.

様々なファイルをインターネットからダウンロードしインストールしますので, やや時間を要します.

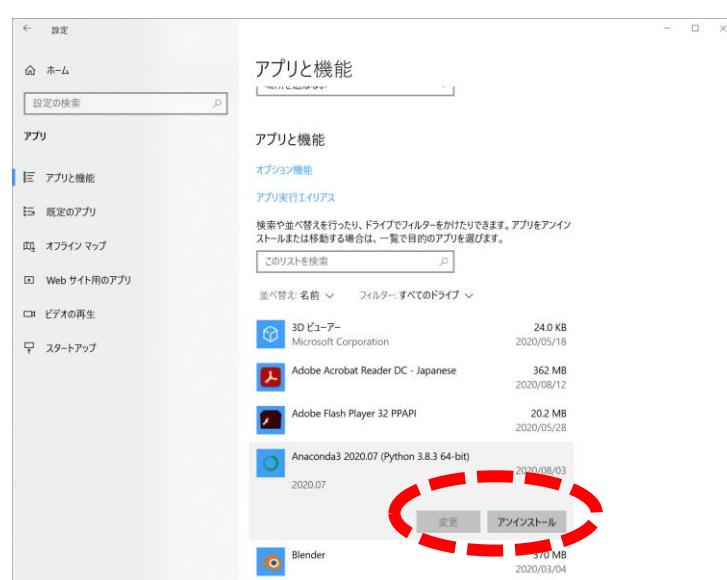
①古い環境のアンインストール

この手順は、2020年3月開催を予定して配布した `handson_install3.pdf` に沿ってインストールを既に行った方のみ実施してください。

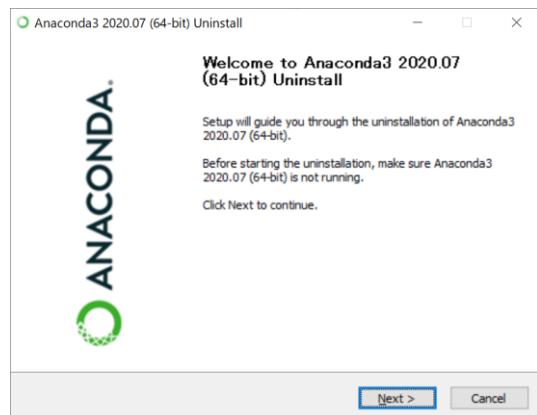
Windows のスタートメニューの「設定」を選択し、Windows の設定の中にある「アプリ」を選択します。



表示されたアプリと機能ウィンドウで「Anaconda3・・・」を見つけ、「アンインストール」を押してください（以下の画像はバージョンが違いますが気にしないで）。



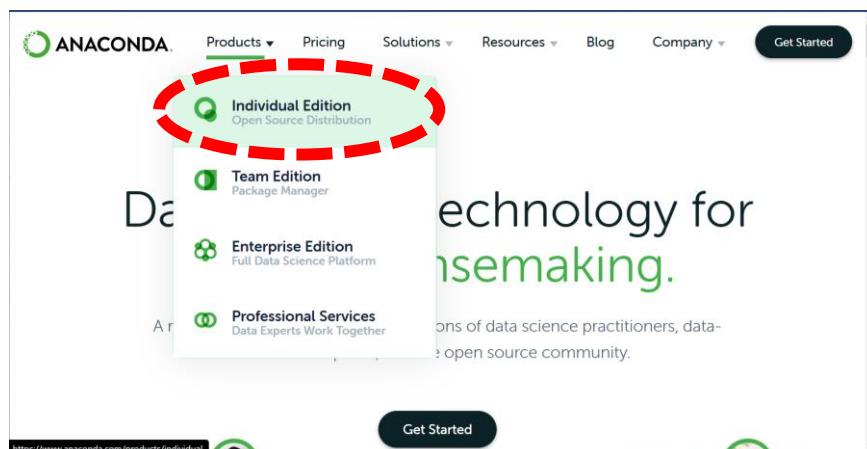
Anaconda のアンインストーラが表示されるので、「Next」を押してアンインストールを完了してください。完了したらパソコンを再起動してください（念のため）。



① Anaconda のインストール

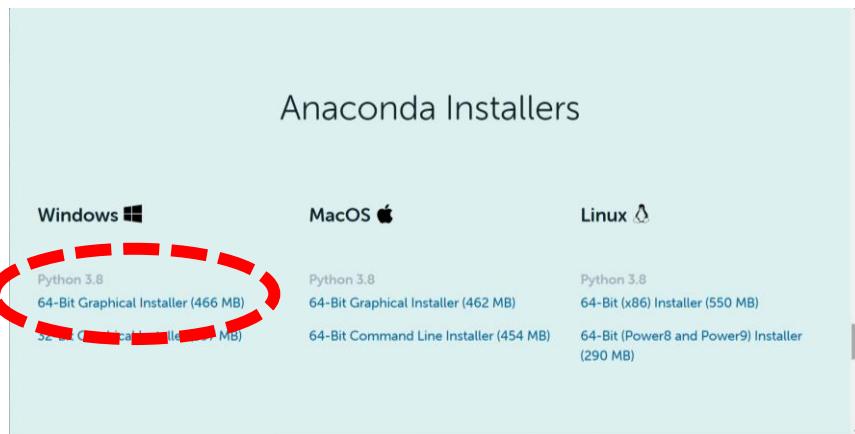
<https://www.anaconda.com/>

にアクセスし、メニューの Products 内にある Individual Edition を選択します。



出てきたページを下の方にスクロールすると Anaconda Installers と書かれた部分があります。Windows の 64-bit Graphical Installer を選択して Anaconda インストーラをダウ

ンロードします。



ダウンロードした exe ファイル (Anaconda3-2020.07-Windows-x86_64.exe, 日付が少し違っても問題ありません) を実行しインストールを行ってください。インストール中にいくつか質問ダイアログが表示されますが、基本的にデフォルトのまま Next ボタンを押してください。以下の質問ではこの選択肢を選択してください。

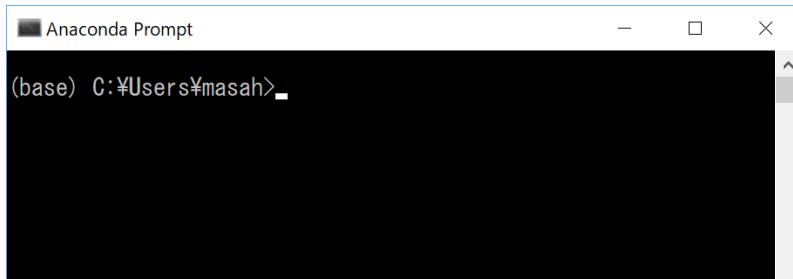
- 「Select Installation Type」の質問には「Just Me (recommended)」を選択する。

② Anaconda Prompt の起動

スタートメニューのアプリリストの中にある「Anaconda3 (64-bit)」グループ内にある「Anaconda Prompt (anaconda3)」をクリックし起動してください。以下のような画面が出ます。（ユーザ名が masah の場合の例です）

ここでユーザ名に日本語等の全角文字が含まれている場合は今後の作業に問題が発生する可能性があります。半角文字だけからなるアカウントを作成して Windows にサインイン

し、最初から作業してください。



③ 仮想環境の作成

ハンズオン用に py37 という名前の仮想環境を作成します。Anaconda Prompt 上で次のコマンドを入力し Enter を押してください。今後はコマンド入力を青字で示します。

```
conda create -n py37 python=3.7
```

新たなパッケージインストール確認が表示されたら y を入力して Enter または単に Enter を押してください。

A screenshot of the Anaconda Prompt window showing the output of the "conda create -n py37 python=3.7" command. The terminal shows the package collection process, the package plan, and the list of new packages to be installed. It ends with a prompt asking if the user wants to proceed with "y" or "n".

```
(base) C:\Users\masah>conda create -n py37 python=3.7
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\masah\anaconda3\envs\py37

added / updated specs:
- python=3.7

The following NEW packages will be INSTALLED:

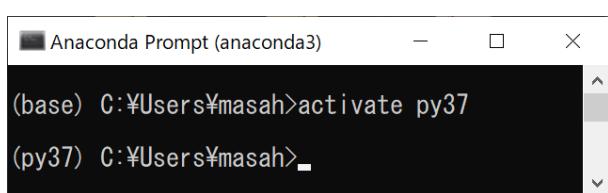
ca-certificates      pkgs/main/win-64::ca-certificates-2020.6.24-0
certifi               pkgs/main/win-64::certifi-2020.6.20-py37_0
openssl              pkgs/main/win-64::openssl-1.1.1g-he774522_0
pip                  pkgs/main/win-64::pip-20.1.1-py37_1
python                pkgs/main/win-64::python-3.7.7-h81c818b_4
setuptools            pkgs/main/win-64::setuptools-49.2.0-py37_0
sqlite               pkgs/main/win-64::sqlite-3.32.3-h2a8f88b_0
vc                   pkgs/main/win-64::vc-14.1-h0510ff6_4
vs2015_runtime        pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_3
wheel                pkgs/main/win-64::wheel-0.34.2-py37_0
wincertstore         pkgs/main/win-64::wincertstore-0.2-py37_0
zlib                 pkgs/main/win-64::zlib-1.2.11-h62dcd97_4

Proceed ([y]/n)?
```

作成した仮想環境を起動します。今後 Anaconda Prompt を新たに起動するたびにこのコマンドを入力してください。

```
conda activate py37
```

入力するとコマンドプロンプト部分に現在有効な仮想環境名がカッコ内に(**py37**)と表示されています。これが分かります。



A screenshot of the Anaconda Prompt window titled "Anaconda Prompt (anaconda3)". The window shows two lines of text: "(base) C:\Users\masah>activate py37" and "(py37) C:\Users\masah>". The prompt "(py37)" indicates that the "py37" environment is currently active.

④ TensorFlow のインストール

Anaconda Prompt 上で以下のコマンドを実行し、TensorFlow をインストールします。後でインストールするソフトウェアとの互換性のあるバージョンを指定してインストールしています。ハイフンが 2 個連続することに気を付けてください。

```
pip install --upgrade tensorflow
```

以下のような表示が出ます。やや時間がかかります。

```
Anaconda Prompt (anaconda3)
Stored in directory: c:\Users\masah\appdata\local\pip\cache\wheels\62\76\4c\aa25851149f3
f6d9785f6c869387ad82b3fd37582fa8147ac6
Building wheel for termcolor (setup.py) ... done
Created wheel for termcolor: filename=termcolor-1.1.0-py3-none-any.whl size=4835 sha256=
93f3713369fbdb57c9b82b445071a741a24876ff27b7225f3fbcb81e0a71800a03
Stored in directory: c:\Users\masah\appdata\local\pip\cache\wheels\3f\ec\8a8336ff1960
23622fbcb36de0c5a5c218ccb24111d1d4c7f2
Building wheel for absl-py (setup.py) ... done
Created wheel for absl-py: filename=absl_py-0.9.0-py3-none-any.whl size=121935 sha256=eb
12742a04b970d4bc251ef9e295a393847a305eff423d2d99de92c23c66749
Stored in directory: c:\Users\masah\appdata\local\pip\cache\wheels\cc\af\1a\498a24d0730e
f484019e007bb9e8cef3ac00311a672c049a3e
Successfully built wrapt termcolor absl-py
Installing collected packages: wrapt, termcolor, six, grpcio, gast, numpy, keras-preprocess
sing, scipy, zipp, importlib-metadata, markdown, pyasn1, pyasn1-modules, rsa, cachetools,
google-auth, werkzeug, absl-py, idna, chardet, urllib3, requests, oauthlib, requests-oauth
lib, google-auth-oauthlib, tensorflow-plugin-wit, protobuf, tensorflow, google-pasta, op
t-einsum, astunparse, tensorflow-estimator, h5py, tensorflow
Successfully installed absl-py-0.9.0 astunparse-1.6.3 cachetools-4.1.1 chardet-3.0.4 gast-
0.3.3 google-auth-1.20.0 google-auth-oauthlib-0.4.1 google-pasta-0.2.0 grpcio-1.30.0 h5py-
2.10.0 idna-2.10 importlib-metadata-1.7.0 keras-preprocessing-1.1.2 markdown-3.2.2 numpy-1
.18.5 oauthlib-3.1.0 opt-einsum-3.3.0 protobuf-3.12.4 pyasn1-0.4.8 pyasn1-modules-0.2.8 re
quests-2.24.0 requests-oauthlib-1.3.0 rsa-4.6 scipy-1.4.1 six-1.15.0 tensorflow-2.3.0 ten
sorboard-plugin-wit-1.7.0 tensorflow-2.3.0 tensorflow-estimator-2.3.0 termcolor-1.1.0 urll
ib3-1.25.10 werkzeug-1.0.1 wrapt-1.12.1 zipp-3.1.0
(py37) C:\Users\masah>
```

⑤ Python と TensorFlow のバージョン確認

Anaconda Prompt 上で python と入力して起動します。

```
python
```

起動メッセージを確認してください。Python 3.7. …と表示されるのを確認してください。

Python コンソール (>>>となる) に

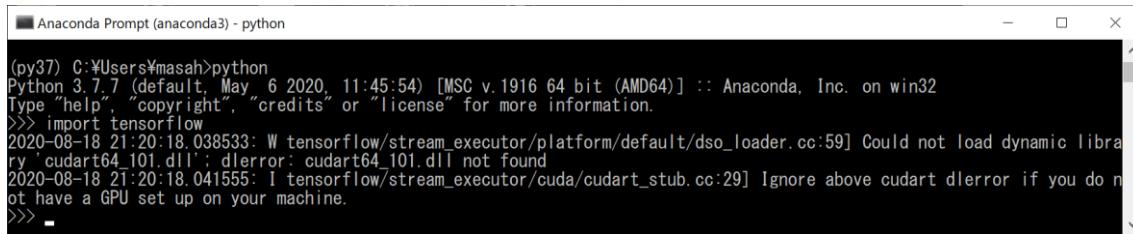
```
import tensorflow
```

と入力してください。ここでエラーが出たら TensorFlow インストールができていません

(下記「TensorFlow インストールに失敗している場合」を参照)。ただし、以下の画面の

ようなメッセージが表示されても問題ありませんので続行してください。これは GPU 使用のためのファイルが無いという警告ですが、今回は GPU を使用しない環境を構築するので

問題ありません。



```
Anaconda Prompt (anaconda3) - python
(py37) C:\Users\masah>python
Python 3.7.7 (default, May 6 2020, 11:45:54) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow
2020-08-18 21:20:18.038533: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'cudart64_101.dll'; dlsym: cudart64_101.dll not found
2020-08-18 21:20:18.041555: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlsym if you do not have a GPU set up on your machine.
>>> -
```

次に Python コンソールに

`tensorflow.__version__`

と入力してください。「__」は半角アンダースコア 2 個連続です。するとバージョン 2.3.0

と表示されます。確認したら

`quit()`

と入力して Python を終了してください。

TensorFlow インストールに失敗している場合

再度「TensorFlow のインストール」の作業を行ってからバージョン確認してください。

それでもだめなら

`deactivate`

で仮想環境を無効化し、

`conda remove -n py37 --all`

で仮想環境を削除してください。再度「仮想環境の作成」から作業を行ってください。

⑥ ライブラリ類のインストール

必要なライブラリをインストールします。以下のコマンドを 1 行ずつ入力し実行してください。
Proceed ([y]/n)?と聞かれたら y 押して Enter または単に Enter を押してください。

```
conda install matplotlib
```

```
conda install scikit-learn
```

Python 開発環境 Spyder をインストールします。インストール中に Winwods のインストール確認ダイアログが表示されたら、インストールを許可してください。

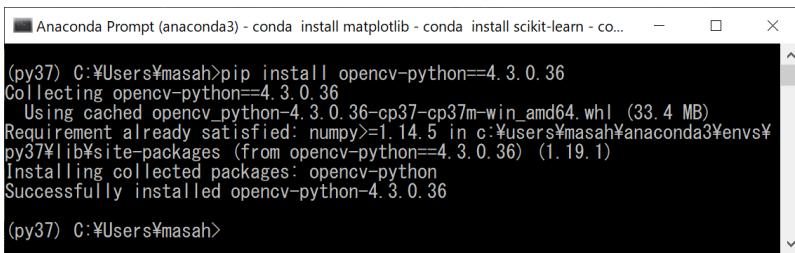
```
conda install spyder
```

OpenCV をインストールします。「==」はイコールが 2 個あることに気を付けてください。

```
pip install opencv-python==4.3.0.36
```

インストール中に色々な文字や警告が表示される場合がありますが、最後に「Successfully installed ソフトウェア名」と表示されればインストール成功です。例えば OpenCV インストールでは最後に以下の文字が表示されます（バージョン番号は少し異なっていても問題ありません）。

```
Successfully installed opencv-python-4.3.0.36
```



The screenshot shows a terminal window titled 'Anaconda Prompt (anaconda3)'. The command entered was 'pip install opencv-python==4.3.0.36'. The output indicates that the package was successfully collected and installed, with a warning about a cached file being used. It also shows that a requirement for numpy version 1.14.5 was already satisfied. The final message 'Successfully installed opencv-python-4.3.0.36' confirms the successful installation.

```
(py37) C:\Users\masah>pip install opencv-python==4.3.0.36
Collecting opencv-python==4.3.0.36
  Using cached opencv_python-4.3.0.36-cp37-cp37m-win_amd64.whl (33.4 MB)
Requirement already satisfied: numpy>=1.14.5 in c:\users\masah\anaconda3\envs\py37\lib\site-packages (from opencv-python==4.3.0.36) (1.19.1)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.3.0.36
(py37) C:\Users\masah>
```

これでインストールは終わりです。

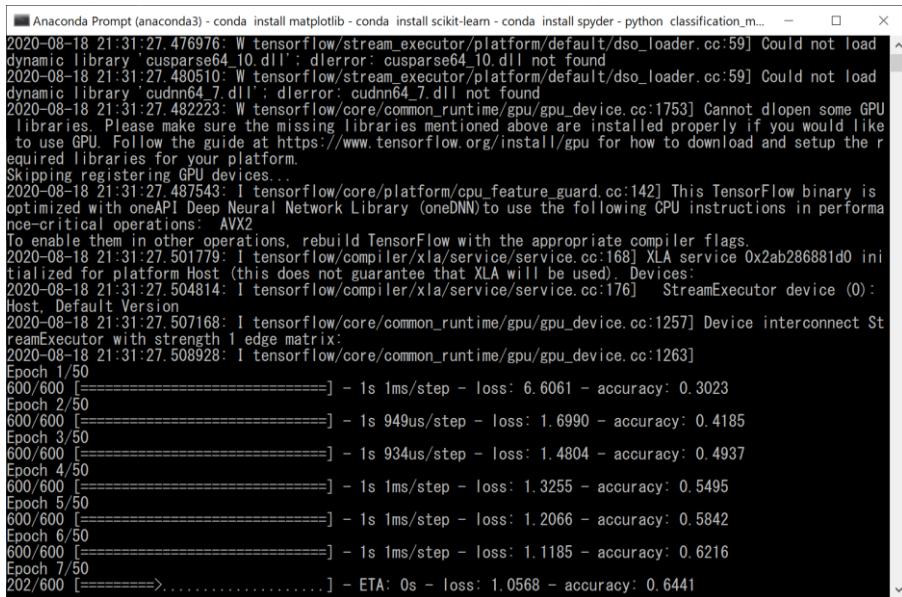
⑦ 動作確認

別途配布した `classification_mnist_nn.py` をホームディレクトリに保存してください。インターネットに接続した状態である必要があります（実行中に mnist データをインターネットからダウンロードするため）。Anaconda Prompt 上で

`python classification_mnist_nn.py`

と入力し Enter を押すと、Keras と TensorFlow を用いた学習と識別処理が始まります。

Epoch が 1 つずつ増加していきます。



```
[■ Anaconda Prompt (anaconda3) - conda install matplotlib - conda install scikit-learn - conda install spyder - python classification_m... - □ X
2020-08-18 21:31:27.476976: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'cusparsse64_10.dll'; derror: cusparsse64_10.dll not found
2020-08-18 21:31:27.480510: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'cudnn64_7.dll'; derror: cudnn64_7.dll not found
2020-08-18 21:31:27.482223: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1753] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2020-08-18 21:31:27.487543: 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
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2020-08-18 21:31:27.501779: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x2ab286881d0 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2020-08-18 21:31:27.504814: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host. Default Version
2020-08-18 21:31:27.507168: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1257] Device interconnect StreamExecutor with strength 1 edge matrix;
2020-08-18 21:31:27.508928: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1263]
Epoch 1/50
600/600 [=====] - 1s 1ms/step - loss: 6.6061 - accuracy: 0.3023
Epoch 2/50
600/600 [=====] - 1s 949us/step - loss: 1.6990 - accuracy: 0.4185
Epoch 3/50
600/600 [=====] - 1s 934us/step - loss: 1.4804 - accuracy: 0.4937
Epoch 4/50
600/600 [=====] - 1s 1ms/step - loss: 1.3255 - accuracy: 0.5495
Epoch 5/50
600/600 [=====] - 1s 1ms/step - loss: 1.2066 - accuracy: 0.5842
Epoch 6/50
600/600 [=====] - 1s 1ms/step - loss: 1.1185 - accuracy: 0.6216
Epoch 7/50
202/600 [=====>.....] - ETA: 0s - loss: 1.0568 - accuracy: 0.6441
```

最後まで実行すると以下のような表示になります。

```
Anaconda Prompt (anaconda3) - conda install matplotlib - conda install scikit-learn - conda install spyder
600/600 [=====] - 1s 1ms/step - loss: 0.4790 - accuracy: 0.8555
Epoch 49/50
600/600 [=====] - 1s 1ms/step - loss: 0.4801 - accuracy: 0.8553
Epoch 50/50
600/600 [=====] - 1s 1ms/step - loss: 0.4829 - accuracy: 0.8547
313/313 [=====] - 0s 422us/step

Classification accuracy: 0.9429
[[ 963   0   0   0   0    0    3    1   12    1]
 [ 0 1123   3   1   0   1    2    0    5    0]
 [ 3   2  969   7   6   0    5   14   25    1]
 [ 5   0  17  936   1   11   0   12   26    2]
 [ 2   0   3   0  896   0    8    2    8   63]
 [ 4   1   2   29   1  796   9    1   43    6]
 [ 9   3   0   0   4   10  918   0   14    0]
 [ 1   2   16   4   4   0    0   979    6   16]
 [ 6   2   10   6   5   15   10    5  909    6]
 [ 5   6   0   11  14   2    0   10   21  940]]]

(py37) C:\Users\masah\Documents\research\Event\2020CAOS_handson\files\1_classification>
```

ここまで表示されればインストールが正常に行われています。（画像中の数値と少々異なる

値が表示されても気にしないでください）