

補助資料

Windowsユーザ向け Python環境の構築方法

担当: 井尻 敬

はじめに

- この資料はPythonを講義で初めて利用する情報工学科学生向けに環境構築方法を紹介するものです。
- 自身で環境を構築できる方、仮想環境を利用したい方などは対象にはしていません（自分で好みの環境を作って講義・演習を実施してください）


手順

1. 準備
2. pythonのインストール
3. 必要なライブラリのインストール

4. Dlibのインストール（顔認識をしたい人向け）
5. PyTorchのインストール（深層学習をしたい人向け）

準備 - 既存のPythonの削除

※すでにPythonをインストールし、正しく動作している人はこの作業は不要です

- 左下のウィンドウズスタートボタン  をクリック
- 少し上にある歯車  をクリック
- 『アプリ』 をクリック
- 『このリストを検索』 窓内に"Python"と記入
 - 「Python *.*」や「Python Launcher」があればクリックしアンインストール

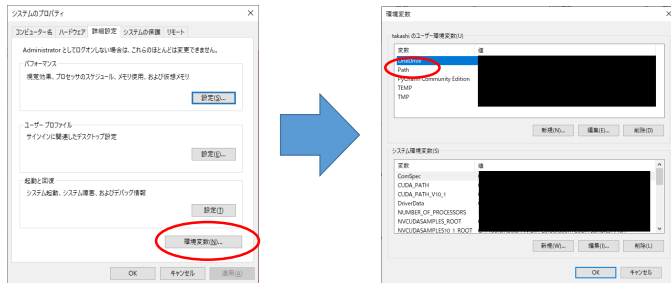
準備 -環境変数: Pathの確認

※コマンドプロンプトにコマンドを打ち込むと、Windowsはそのコマンドに対応する実行ファイルを検索する。検索対象となるフォルダへのpathが、「環境変数のPath」という項目に書かれている。

※以前にPythonをインストールし現在利用していない場合、過去のPathが残っており不具合が起きる可能性があるため確認する必要がある

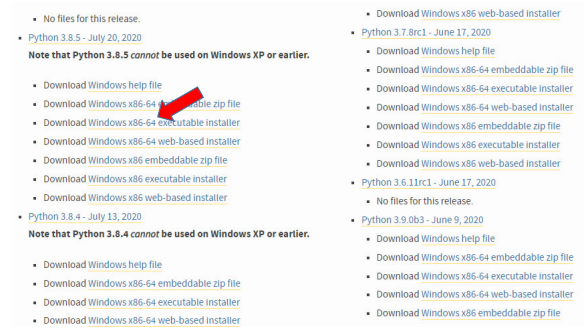
※不具合が起きる可能性があるためよくわからなければ何もしない

- Windowsのスタートボタンを右クリックし、システムを選択 → 右側の「システム情報」を選択 → 「システムの詳細設定」を選択 → 「環境変数」を選択
- 「Path」を編集状態にして、もしPythonやAnacondaに関するものがあり、過去にインストールしたものなら削除する



Pythonのインストール

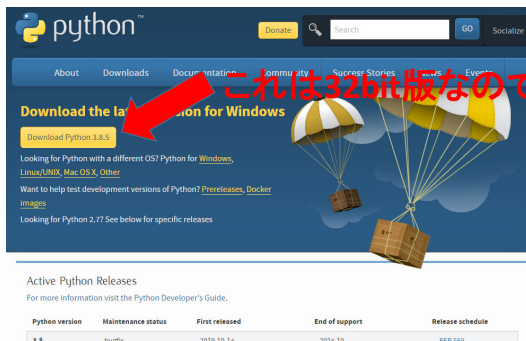
1. Pythonのページ (<https://www.python.org/>) を開く
2. Downloadにカーソルを置き → windowをクリック
3. 下の通り、「Download Windows x86-64 executable installer」をクリックしダウンロード
ここではPython 3.8.5を選択



Pythonのインストール (注意)

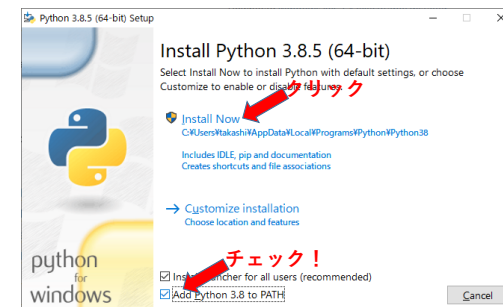
以下のリンクからダウンロードできるのは32bit版。

動けばよい方はそれでもOK。機械学習をやる方は、後に問題が起きる（入れられないライブラリがあるとか）ので64bit版を強く推奨。



Pythonのインストール

4. インストーラーを起動する
5. インストーラー起動画面でPath設定 (下図参照) にチェックを入れる
6. Install Nowをクリックし、インストールを進める



Pythonの動作確認

1. コマンドプロンプトを起動
- Windowsの検索ボックス（左下の「ここに入力して検索」）にcmdと打ち込む
2. コマンドプロンプトに「\$python -V」と打ち込みpythonのバージョンが表示されればOK
3. コマンドプロンプトに「\$pytno」と打ち込みpythonの対話モードが起動するのを確認してもOK（対話モードはCtrl+zで終了できる）

```
C:\Users\takashi>python -V
Python 3.8.5

C:\Users\takashi>python
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:57:54) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> -
```

必要なライブラリのインストール

- Pythonでは、「pip」コマンドによりライブラリパッケージの管理（インストール・アンインストール）ができる
- コマンドプロンプトに「\$pip list」と打ち込むと現在インストールされているパッケージが表示される

```
D:\Users\takashi>pip list
Package          Version
-----
certifi          2020.6.20
cycler           0.10.0
future          0.18.2
joblib           0.16.0
kiwisolver       1.2.0
Pillow           7.2.0
pip              20.2.3
pyparsing        2.4.7
python-dateutil 2.8.1
scikit-learn     0.23.2
scipy            1.5.2
setuptools       49.6.0
six              1.15.0
threadpoolctl    2.1.0
```

- ※pipのバージョンが古いとwarningが出る
- ※初期状態はもう少し入っているパッケージは少ないはず

必要なライブラリのインストール

1. pipのアップグレード. コマンドラインに以下のコマンドを打つ
\$python -m pip install --upgrade pip
2. numpy/matplotlib/opencvのインストール. コマンドラインに以下のコマンドを打つ
\$pip install numpy
\$pip install matplotlib
\$pip install opencv-python

3. \$pip list としてパッケージが入ったことを確認

```
D:\Users\takashi>pip list
Package          Version
-----
certifi          2020.6.20
cycler           0.10.0
future          0.18.2
joblib           0.16.0
kiwisolver       1.2.0
matplotlib       3.3.2
numpy            1.19.2
opencv-python    4.4.0.42
Pillow           7.2.0
pip              20.2.3
pyparsing        2.4.7
python-dateutil 2.8.1
scikit-learn     0.23.2
scipy            1.5.2
setuptools       49.6.0
six              1.15.0
threadpoolctl    2.1.0
```

OpenCVの動作確認

- テスト用ディレクトリを用意
- 画像ファイルを用意し「img.png」という名前でこのディレクトリに保存
※異なる形式・ファイル名を利用するときには右のコードの該当部分を書き換える
- ファイル「test.py」を作成し、右のコードをコピー
- このディレクトリがカレントのコマンドプロンプトを起動（エクスプローラのアドレスバーに「cmd」と書く）
- 「\$ python test.py」とコマンドプロンプトに打ち込み画像が表示されたらOK

```
# -*- coding: utf-8 -*-
import cv2
```

```
img = cv2.imread("img.png")
print(img.shape)
cv2.imshow("sample", img)
cv2.waitKey(0)
```

Dlibのインストール

1. Cmakeをインストール : \$pip install cmake
2. Visual studio 2019のC++によるデスクトップ開発環境をインストール
3. Dlibをインストール : \$pip install dlib

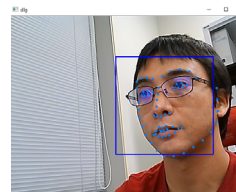
※ 参考 : <https://qiita.com/taungyeon/items/0afa3a5580c7521d54d1>

※ 2については、テストした環境で初めから入っていたので本当に必要か不明

Dlibのテスト

1. imutils をインストール : \$pip install imutils
2. webcameraが繋がったPCを用意
3. <http://dlib.net/files/>から「shape_predictor_68_face_landmarks.dat」をダウンロードし解凍
4. 右のコードをwebcam.pyという名前で上の.datファイルと同じ場所に保存
5. 実行 \$python webcam.py

Webcamの動画が表示され、顔の特徴点にマーカが置かれる



```
#!/usr/bin/env python
import cv2
import dlib
from imutils import face_utils

cap = cv2.VideoCapture(0)
detector = dlib.get_frontal_face_detector()
predictor_path = 'shape_predictor_68_face_landmarks.dat'
face_predictor = dlib.shape_predictor(predictor_path)

while 1:
    ret, frame = cap.read()

    #顔検出
    dets, score, idx = detector.run(frame, 0)
    for r in dets:
        frame = cv2.rectangle(frame, (r.left(), r.top()), (r.right(), r.bottom()), (255, 0, 0), 2)

    #顔ランドマーク検出
    landmark = face_predictor(frame, r)
    landmark = face_utils.shape_to_np(landmark)
    for (i, (x, y)) in enumerate(landmark):
        cv2.circle(frame, (x, y), 3, (255, 128, 0), -1)

    cv2.imshow('dlib', frame)

    k = cv2.waitKey(1)
    if k == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

Torchのインストール

1 Cudaのインストール

- GPUのドライバの更新
 - nvidiaのページから最新のものをDLしてインストール
- Cudaのインストール
 - 本家ページに行く (<https://developer.nvidia.com/cuda-toolkit-archive>)
 - CUDA Toolkit 10.1 update2 を選択
 - OS等を選んでダウンロードしてインストール
 - コマンドプロンプトで \$nvcc -V でcuda のバージョン確認できる

2 PyTorchのインストール

- 本家ページに行く <https://pytorch.org/>
- ページの少し下のほうで、OS, cudaのバージョン情報を指定するとコマンドプロンプトからpipで打ち込むべきコマンドが出てくる
- 出力されたコマンドをコマンドプロンプトで実行

3. PyTorchのテスト

右のコードを実行できれ、 cuda is availableと表示され

3x2の乱数列が表示されれば、 torchとcudaのインストールはおそらく成功している

```
import torch
```

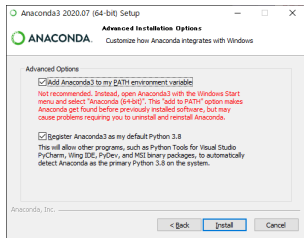
```
if torch.cuda.is_available():
    print ("cuda is available ")
else :
    print ("cuda is NOT available ")
```

```
x = torch.rand(3,2)
print(x)
```

Anacondaを利用する場合

Anacondaのインストール

1. Anaconda (<https://www.anaconda.com/products/individual>) のページを開く
2. ページ内の「Download」をクリック
3. Python 3.8の「64Bit Graphical Installer」をクリックしインストール用ファイルをダウンロード
4. ダウンロードしたファイルをダブルクリックしインストールを進める
 - 規約をよく読みインストールを進める
 - 以下の「Add Anaconda3 to my PATH environment variable」にチェックを入れる



※自分で環境変数のPATHを編集する方はチェック不要

※ここをチェックした場合、自動で「環境変数：PATH」に自動でAnacondaへのpathが追加されるので、アンインストール時には自分で削除する必要があるかもしれない（おそらくアンインストール時に自動削除されると思うが未確認）

Anacondaの動作確認

1. コマンドプロンプトを起動
 - Windowsの検索ボックス（左下の「ここに入力して検索」）にcmdと打ち込む
2. コマンドプロンプトに「\$python -V」と打ち込みpythonのバージョンが表示されればOK
3. コマンドプロンプトに「\$pytno」と打ち込みpythonの対話モードが起動するのを確認してもOK（対話モードはCtrl+zで終了できる）

```
C:\Users\takashi>python -V
Python 3.8.3

C:\Users\takashi>python
Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> _
```

必要なライブラリのインストール

- Anaconda では、「conda」コマンドによりライブラリパッケージの管理（インストール・アンインストール）ができる
- コマンドプロンプトに「\$conda list」と打ち込むと現在インストールされているパッケージが表示される

```
C:\Users\takashi>conda list
# packages in environment at C:\Users\takashi\Anaconda3:
#
# Name                    Version            Build                Channel
#-----
_ipyw_lab_ext_conf        0.1.0              py38_0
alabaster                 0.7.12            py_0
anaconda                  2020.07           py38_0
anaconda-client           1.7.2             py38_0
anaconda-navigator        0.7.12            py38_0
anaconda-project          0.8.4             py_0
arab                       0.26.2            py38_0
astropy                   3.3.0             py38_0
astroquery                2.4.2             py38_0
astroquery.post1         4.0.1.post1       py38he774522_1
astronomy                 1.4.0             py_0
attrs                     19.3.0            py_0
autocpe8                  1.5.3             py_0
babel                     2.8.0             py_0
backcall                  0.2.0             py_0
backports                 1.0               py_2
backports.functionools_1 1.8.1             py_0
backports.shutil_get_terminal_size 1.0.0            py38_2
backports.teefile         1.1               py_1
backports.weakref         1.0.post1         py_1
certifi                   2020.6.20         py38_0
beautifulsoup4            4.8.1             py38_0
bitarray                  1.4.0             py38he774522_0
bktarrs                   0.2               py38_0
blas                      1.0               mkl
bleach                    3.1.5             py_0
```

必要なライブラリのインストール

- Anacondaには最初からnumpyやmatplotlibが入っているので、「OpenCV」のみインストールする
- コマンドプロンプトに以下を打ち込み,
『\$conda install -c conda-forge opencv』
- Proceed?と聞かれるのでyキーを押してリターンキーを押す

OpenCVの動作確認

- テスト用ディレクトリを用意
- 画像ファイルを用意し「img.png」という名前でこのディレクトリに保存

※異なる形式・ファイル名を利用するときには右のコードの該当部分を書き換える

- ファイル「test.py」を作成し、右のコードをコピー
- このディレクトリがカレントのコマンドプロンプトを起動（エクスプローラのアドレスバーに「cmd」と書く）
- 「\$ python test.py」とコマンドプロンプトに打ち込み画像が表示されたらOK

```
# -*- coding: utf-8 -*-  
import cv2  
  
img = cv2.imread("img.png")  
print(img.shape)  
cv2.imshow("sample", img)  
cv2.waitKey(0)
```