環境設定(for Linux Ubuntu)

1. Ubuntuインストール（LiveUSB/DVDを作成し，BIOSからBoot）
2. 学校のPC（デスクトップ）はプロキシを使用しているため，Ubuntuにもプロキシ設定をする．（わからなければ，proxy環境下ubuntuとかでググれカス）[1]

その他の問題なら，だいたいはググればいい．端末のエラーをコピってググカス

2-1. 端末上で，sudo nano ~/.bashrc　を行い，一番下に

export https\_proxy=”http://usersproxy : proxy.port”

export http\_proxy=”http://usersproxy : proxy.port”

export ftp\_proxy=”http://usersproxy : proxy.port”

ここで，関大なら”http://proxy.itc.kansai-u.ac.jp:8080”

2-2. /etc/apt/apt.conf（なければ作成）にて，（端末上で，sudo nano /etc/apt/apt.conf）

Acquire::http::proxy <http://proxy.itc.kansai-u.ac.jp:8080/>;

Acquire::https::proxy <https://proxy.itc.kansai-u.ac.jp:8080/>;

Acquire::ftp::proxy <ftp://proxy.itc.kansai-u.ac.jp:8080/>;

1. GPUがあるなら，CUDAとcuDNNを入れていく．
2. . nvidia-driverを入れる．参考ページ[2]通りに進める．ただし参考ページ[2]はnvidia-smiを見るところまでを参照とし，cudaについては[3]を参考に進める．

注意ポイント①

　＄sudo add-apt-repository ppa:graphics-drivers/ppa

を実行する場合，もしもproxy環境下であるなら，

　＄sudo -E add-apt-repository ppa:graphics-drivers/ppa

で実行すること．-Eは環境変数を使用することを指し，proxyに対応可能．

注意ポイント②

　＄sudo apt-get install nvidia-390

を実行するとき，nvidia-390でなくてもいいので，**新しいもの**を入れるとよい．

20190306現在では，nvidia-415を入れました！

1. . cudaを入れる．上述した通り，参考ページ[3]通りに進める．気を付けるのは，記事内でも言われているが，

　＄sudo apt-get install cuda-toolkit-9-0

のように，***cuda-9-0*や*cuda*を入れないように！**

あと，**確認もしておくこと**（サンプルプログラムを通すやつ）

1. .cuDNNについても，参考ページ[3]通りに進める．バージョンについては使用できる一番新しいものを入れる．
2. ここでいったんバックアップを取る．PCのシステムバックアップにはtimeshift，PCのホーム・ダウンロード等のバックアップにはubuntu標準装備の物を使用する．今後，逐一バックアップを取って管理しておく．表などをつくって，どのような環境でバックアップを取っていたかを書いておくといい．使い方はググればすぐ出てくるので割愛
3. 今度は，ROSを入れる．参考ページ[4]に従うだけ．20190306現在ならkineticでよかったけど，時代が進めばねぇってかんじだわねぇえ．あと，参考ページ[4’]のように，

$ source ~/catkin\_ws/devel/setup.bash

を~/.bashrc に入れること．

1. Ubuntuに標準で入っているpythonのversionについて，変更したければ，参考ページ[5]に従う．もし，vertualenvを用いた環境構築ならやっておいて損はないはず．思うに，変更内容としては，python3じゃないかと思うので，参考ページ途中にある，シンボリックリンクを作成するときに，

　＄ln -s /usr/bin/python3.n $HOME/bin/python3

てな感じにしておくこと．（nはpythonのversion）

※もしかすると，python2.7にpipを入れてから変更した方がいいかも．しなくてもうまく入っているが。。。

1. 環境については，python2.7ではvirtualenvを，python3.6以降ではvenvを使用する．Python2.7については，参考ページ[6]に従ってインストールする．Pipについてのエラー（バ―ジョンが古いわ！的な奴）は無視していいってか無視して．Pipはpythonのパッケージの管理を行うもので重要である．そのため少し説明する．Pipはpython2系のものか3系のものかによって，同じように使っても，目的の系に入らないことがある．Pipはversionが大切なのだ．基本的にはpython3.Nに対しては，pip3を入れればいい．

$pip –version

これを端末上でおこなうと，pipのversionと，どこにあるかがわかる．これが，pipならpython2.7もしくはpython3.N，pip2ならpython2.7，pip3ならpython3.Nになっていればよい．各パッケージは，どのpipで入れるかでどこに入るかが決まるので，対応するversionをしっかり確認しておこう．なので，python2.7についてはvirtualenvをpip2（or pip (pip -Vがpython2.7なら)でinstall．Python3.Nについてはpip3（or pip (pip -Vがpython3.Nなら)でinstall．

python3.Nについては，venvを入れる．参考ページ[7]を見ながらやればいい．たしか，

$sudo apt install python3.N-venv

だった気がしないでもない．（笑）先にこれをやってみてダメなら，記事通りどうぞ．

Venvでの環境の作り方は，参考ページ[8]を見ながらやってみて．

　$python3.N -m venv –system-site-packages <環境名>

ってうてば，グローバルな環境でのパッケージも引き継げるお．コマンドは，virtualenvと同じだと思う．

1. あとは，適当にディレクトリをつくって（python2系と3系を分けるように作ること）そのなかで，仮想環境を作ればいい．あとは，$source <環境名>/bin/activate で環境に入って，パッケージを入れていけばいい．ただ，仮想環境を作った時に，pythonのversionとpip listとかをして，系に間違いがないか、いらないパッケージを入れちゃってないかだけは要チェック．もし，環境がいらなくなったら，ディレクトリごと削除で終わり．あとは，tensorでもchainerでも入れるとええ．

ただ，tensorflowだけ注意で，

　$pip install tensorflow-gpu==version

Gpuがないなら，tensorflow-gpuの代わりにtensorflowでいい．また，cudaによってtensorflowのversionが違うから，そこもチェックしてから入れる事．

1. 仮想環境作成とか，作業スペースへの移動とかは，シェル関数[9]を作るといい．このように[10]シェル関数とエイリアス(alias)と組み合わせて，~/.bashrcとかの中に書いておくといい．おまけ程度なので分からなくても別にいい．
2. Atom(プログラムを書くときに便利な奴)を入れる．これはググれば出てくる．設定も必要なので，それもググレカス
3. 次は，Caffe．使うかわかんないけど．参考ページは2つで，少しずつ参考にする．順番に説明していく．また，**参考ページが多いため，この文をすべて読んでから進めた方が，間違いが少ないと思う．**まず，参考ページ[12]の，ページ内の**2.OpenCVの導入**のインストール部分を使う．Versionは任せる．mkdir buildのところから，参考ページ[11]のOpenCV Downloadのmkdir buildに移動して進めていく．最後の再起動までやってしまうこと．**特に，$sudo /bin/bash -c ‘echo “/usr/local/lib” > /etc/ld.so.conf.d/opencv.conf’　は必ず行うこと．**次は，Caffeを実際に入れていく．入れる前に参考ページ[12]のページ内の**1.パッケージインストール**部分のapt installを行う．Installは，上から3,4段目のapt installで，pipやらpythonやらは無視するとよい．Apt Installできたら，ページ内の**5.Caffeのダウンロード**に移る．Caffeのconfigについては，参考ページ[11]のMakefile.configのコピーらへんから進めていく．このとき，cudnnなどのために，Makefile.configのCUDNN=1をアンコメントしておく．設定ファイルの編集が終わったら，参考ページ[12]の**Makeでコンパイル**以下を進めていく．Pythonについては，仮想環境内（python2.7）にて，必要になるpkgを，pipを用いて入れてからimport caffeで通ることを確認すればいい．そのため，間違ってもグローバルにて強引にpipしないようにすること．Makeエラーについては，両サイト参考にしつつ，端末からerror文をコピペしてググれ
4. Yoloは簡単なので，割愛

参考文献

1. proxy環境下の設定(Ubuntu14.04)，< https://qiita.com/showsuzu/items/9ee031208d38ff8ac889>, 20190306
2. Ubuntu 16.04LTSにNVIDIAドライバ(nvidia-390)とCUDA 9.1を入れた時のメモ，

< http://swytel.hatenablog.com/entry/2018/05/08/174943>, 20190306

1. 自作PCにUbuntu16.04 LTS、NVIDIAドライバ、CUDA9.0、cuDNN 7.1をインストール（ディープラーニング用），<https://qiita.com/min9813/items/90a1ef62b3dc37d0cc33>, 20190306
2. Ubuntu 16.04LTSにROS Kineticをインストールする，<https://qiita.com/proton\_lattice/items/6b26ee3a4f84776db7b7>, 20190306

[4’] ロボットプログラミングⅡ：第３週　はじめてのROSプログラミング，

< http://demura.net/lecture/12202.html>, 20190321

1. Ubuntu16.04でPythonのバージョンを2.7から3.6にバージョンアップする，<https://tetechi.com/python3-6/>, 20190313
2. virtualenvでpython環境を管理する，<https://qiita.com/caad1229/items/325ca5c8ad198b0ebce7>, 20190313
3. python3 venvで環境作成，< https://qiita.com/ajitama/items/875224bdb85926a3aa05>, 20190313
4. venv: Python 仮想環境管理，< https://qiita.com/fiftystorm36/items/b2fd47cf32c7694adc2e>, 20190313
5. シェルスクリプトで関数を利用する，< https://qiita.com/kaw/items/034bc4221c4526fe8866>, 20190313
6. aliasにbash関数を使おう：レンタルサーバ借り直した時の.bashrcを晒してみる，< https://qiita.com/mrpepper/items/9b36714bb473aed4c085>, 20190313
7. Ubuntu 16.04 with CUDA にCaffeをインストール，<https://robotics4society.com/2016/06/15/ubuntu1604-caffe/>, 20190321
8. Caffeをインストールしたった，< https://qiita.com/yoyoyo\_/items/ed723a6e81c1f4046241 >, 20190321