

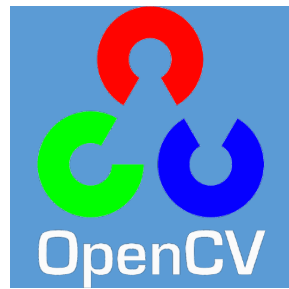
はじめてのAI用パソコン

Chainer/caffe/opencv学習編 抜粋版

(ubuntu版)



Caffe



スペクトラム・テクノロジー株式会社
<https://spectrum-tech.co.jp>
sales@spectrum-tech.co.jp



目次

• ubuntu運用マニュアル	ページ
1. Ubuntuについて	<u>3</u>
2. 基本コマンド	<u>3</u>
3. 基本操作	<u>5</u>
4. 日常運用	<u>9</u>
5. ソフト一覧	<u>12</u>
• Chainer学習プログラム	ページ
① Chainer概要	<u>14</u>
② 操作方法	<u>15</u>
③ Deep learningの例(Cifar)	<u>17</u>
④ Deep learningの例(Mnist)	<u>18</u>
• Caffe学習プログラム	ページ
① Caffe概要	<u>20</u>
② 操作方法	<u>21</u>
③ Deep learningの例(Cifar)	<u>23</u>
④ Deep learningの例(Mnist)	<u>24</u>
• Opencv学習プログラム	ページ
① Opencv概要	<u>26</u>
② 操作方法	<u>27</u>
③ 顔検出	<u>28</u>
④ 顔認識	<u>29</u>

抜粋版

ページと内容は
抜粋版のため異
なります

ubuntu運用マニュアル

1. Ubuntuについて

Ubuntu(ウブントウ)とは、コミュニティにより開発されているオペレーティングシステムです。ラップトップ、デスクトップ、そしてサーバーに利用することができます。Ubuntuには、家庭・学校・職場で必要とされるワープロやメールソフトから、サーバーソフトウェアやプログラミングツールまで、あらゆるソフトウェアが含まれています。Linuxの中でも一番使用されています。

2. Linux基本コマンド

① システム関係

- 起動: 電源を入れると自動で起動します。
- 再起動: # reboot
又は、menu>shutdown>reboot; 左上のメニューから
- 終了: # shutdown
又は、menu>shutdown>shutdown; 左上のメニューから
- ログアウト # exit
又は、menu>shutdown>logout; 左上のメニューから
- **日本語／英語の入力切替**: 半角／全角キー(数字の1の横)、英数と文字の切替は、capital lock(Aの横)

ubuntu運用マニュアル

基本は、masa@Ubuntu:~\$で使用してください

2. Linux基本コマンド

② ディレクトリ操作、コピー、移動、削除

masa@ubuntu:~\$ **cd** /home/masa/Documents ディレクトリの切り替え
masa@ubuntu:~/home/masa/Documents\$ **ls** ファイルとディレクトリの表示(表示したら操作したいファイルを右クリックでコピーして操作します)
root@ubuntu:~# **cp** ファイル名 ディレクトリ 配下のディレクトリのファイルを別のディレクトリへコピー
root@ubuntu:~# **mv** ファイル名 ディレクトリ 配下のディレクトリのファイルを別のディレクトリへ移動
root@ubuntu:~# **rm** ファイル名 ファイルの削除
便利な機能 **rm -help** コマンドのオプションが分からない場合は、ヘルプで問い合わせる。すべてのコマンド共通(マイナスを2個とhelp)

③ ユーザ権限、プロセス他

masa@ubuntu:~ \$ **su -** スーパーユーザ(root)に切り替え、パスワードを入力
root@ubuntu:~# **ps** a 現状の動いているプロセスを表示
root@ubuntu:~# **kill** 特定のプロセスを強制終了
root@ubuntu:~# **apt-get** install pkg パッケージのインストールなどに使用 \$で使用する場合は、sudoを付与
root@ubuntu:~# **date** 日付、時間の設定を行います。
root@ubuntu:~# **leafpad** /etc/network/interfaces インタフェースに記述している内容を変更します。Viよりも使いやすいです。

④ モジュール、usb、メモリ、HDDなどの表示

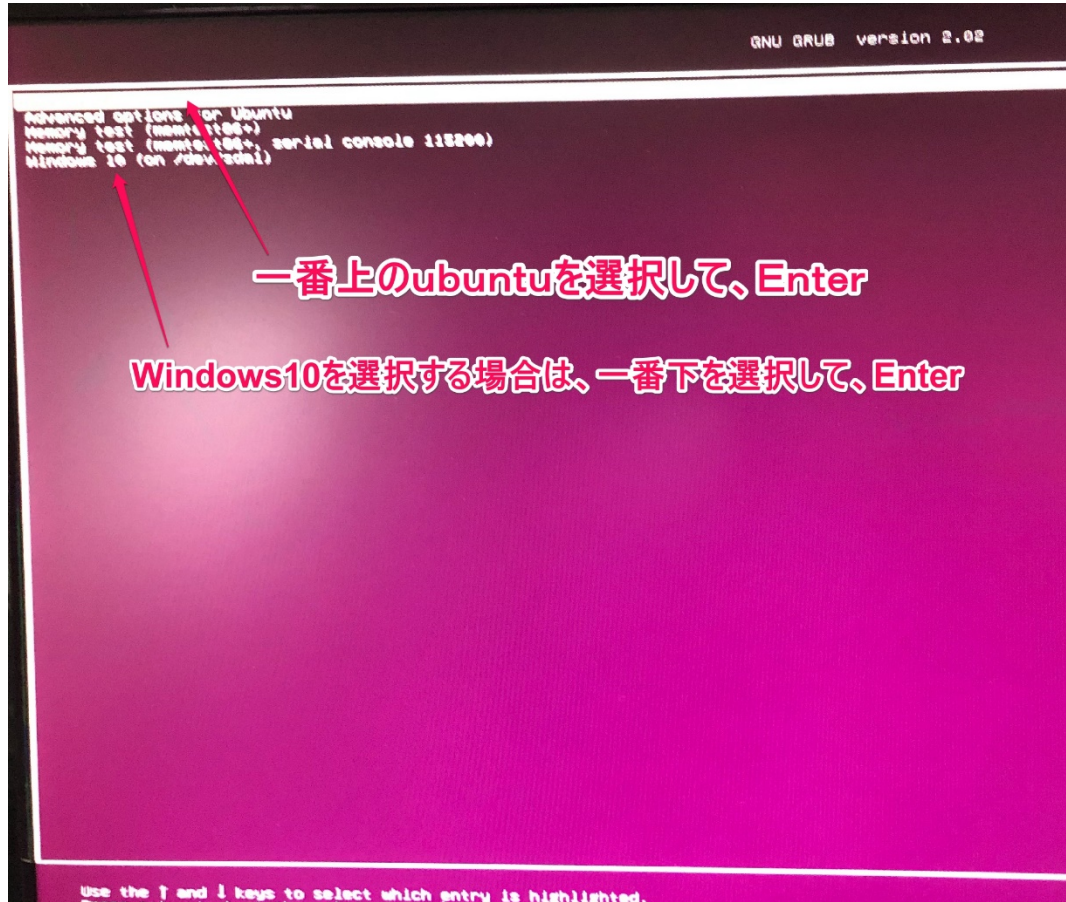
root@ubuntu:~# **lsmod** linuxのモジュールリスト表示
root@ubuntu:~# **lsusb** usbのデバイス表示
root@ubuntu:~# **free -mt** メモリ使用状態表示
root@ubuntu:~# **df -h** SSDの使用状態表示
root@ubuntu:~# **ifconfig** ipアドレスを確認します。



ubuntu運用マニュアル

3. 基本操作

① 起動(電源ON)



一番上のubuntuを選択して、Enter

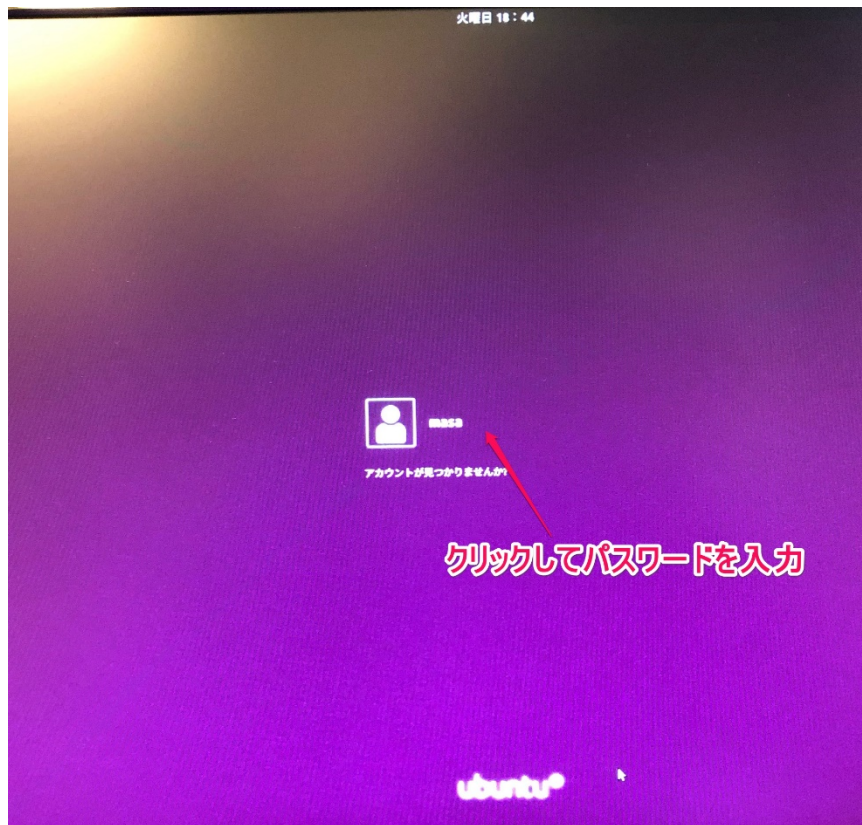
Windows10を選択する場合は、一番下を選択して、Enter

GNU GRUB画面が立ち上がり、Ubuntu, windows10の選択画面がでます。
Ubuntuを選択

ubuntu運用マニュアル

3. 基本操作

② ログオン

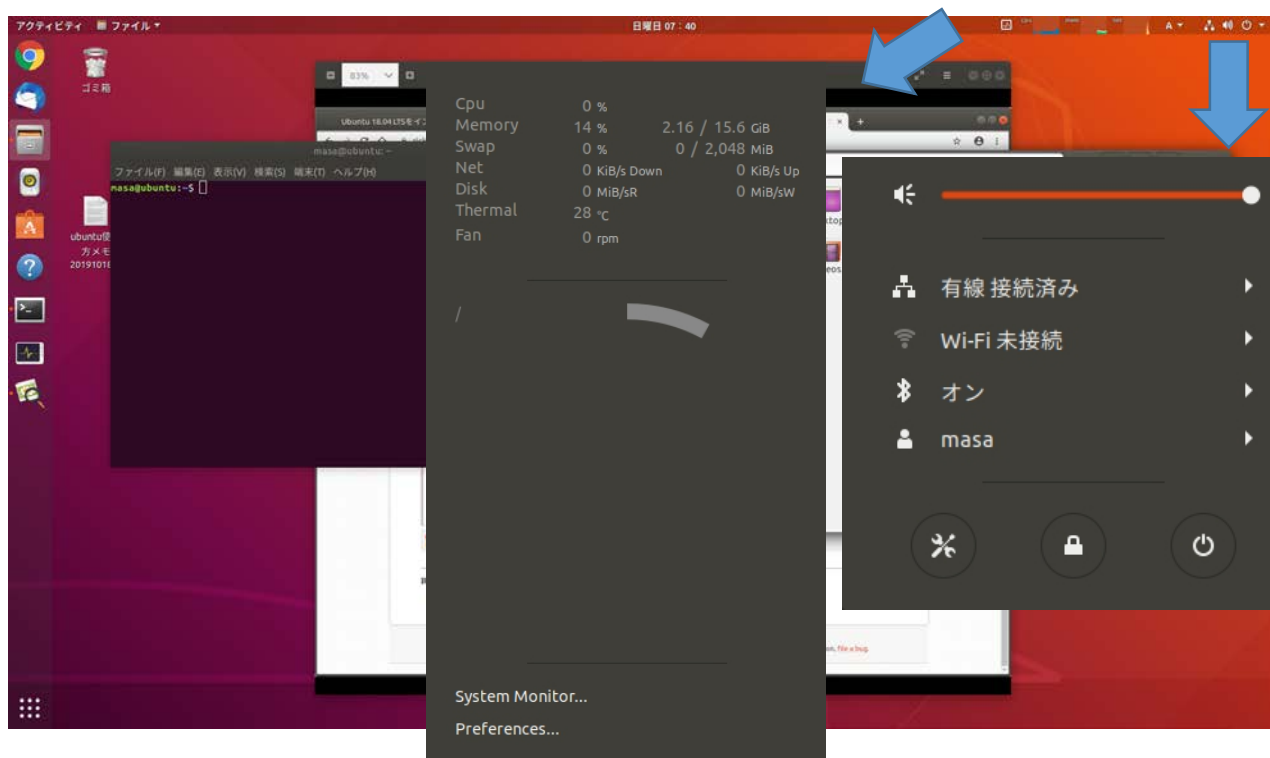


ユーザを選択して、パスワードを入力。

ubuntu運用マニュアル

3. 基本操作

③ 表示画面と内容



トップ画面(左側のアクティビティバー)

- ブラウザ
- メール
- ファイルマネージャ
- カメラ
- アプリ
- ヘルプ
- コマンド
- システムモニタ

トップバー(上側)

- アクティビティ
- 時計
- システムモニタ
- LAN
- スピーカー
- 電源



ubuntu運用マニュアル

3. 基本操作

④ パスワード変更



\$ sudo passwd masa
パスワードを変更してください。
\$ sudo passwd root
rootのパスワードも同様に設定

ubuntu運用マニュアル

4. 日常運用

① セキュリティ対策(アンチウイルス更新、スキャン)

- アンチウイルス対策として無料のclamAVをインストールしてます。
- 手動での運用を基本としています。

```
masa@ubuntu: ~  
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)  
systemd (237-3ubuntu10.31) のトリガを処理しています ...  
man-db (2.8.3-2ubuntu0.1) のトリガを処理しています ...  
dbus (1.12.2-1ubuntu1.1) のトリガを処理しています ...  
ureadahead (0.100.0-21) のトリガを処理しています ...  
ureadahead will be reprofiled on next reboot  
libc-bin (2.27-3ubuntu1) のトリガを処理しています ...  
masa@ubuntu:~$ sudo passwd masa  
新しい UNIX パスワードを入力してください:  
新しい UNIX パスワードを再入力してください:  
passwd: パスワードは正しく更新されました  
masa@ubuntu:~$ sudo clamscan --infected --remove --recursive  
sudo: clamscan: コマンドが見つかりません  
masa@ubuntu:~$ sudo clamscan --infected --remove --recursive  
----- SCAN SUMMARY -----  
Known viruses: 6506172  
Engine version: 0.101.4  
Scanned directories: 10410  
Scanned files: 74827  
Infected files: 0  
Data scanned: 4866.40 MB  
Data read: 27046.70 MB (ratio 0.18:1)  
Time: 510.068 sec (8 m 30 s)  
masa@ubuntu:~$
```

パターンファイル更新

手動スキャン時に更新されます

手動でスキャン

\$ sudo clamscan --infected --remove --recursive
自動化可能ですが、バックグラウンドで重くなる可能性大。コマンド入力後約10分位かかります。

ubuntu運用マニュアル

4. 日常運用

② インストール済パッケージの更新リスト、アップグレード

- Linuxの場合は、頻繁に更新が発生します。アップグレードを定期的実施してください。
- 更新前には、バックアップを取ることをお勧めします。特にアップグレードはまれに動作不良、戻せない状態が発生します。自己責任で実施してください。

```
masa@ubuntu: ~  
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)  
masa@ubuntu:~$ sudo apt-get update  
ヒット:1 http://jp.archive.ubuntu.com/ubuntu bionic InRelease  
取得:2 http://jp.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Metadata [2,468 B]  
無視:3 http://dl.google.com/linux/en [3,972 B]  
取得:4 http://dl.google.com/linux/en [3,972 B]  
取得:5 http://dl.google.com/linux/images [8,092 B]  
取得:6 http://jp.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation metadata [9,896 B]  
取得:7 http://dl.google.com/linux/en [3,972 B]  
取得:45 http://jp.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 DEP-11 Metadata [2,468 B]  
取得:8 http://jp.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 DEP-11 Metadata [7,916 B]  
取得:46 http://jp.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 DEP-11 Metadata [7,916 B]  
ヒット:9 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages 19,896 kB を 51秒 で取得しました (194 kB/s)  
ヒット:10 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages  
取得:11 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages  
取得:12 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages  
取得:13 http://jp.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages  
取得:14 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages  
取得:15 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages  
取得:16 http://jp.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages  
取得:17 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages  
masa@ubuntu:~$ sudo apt-get upgrade  
パッケージリストを読み込んでいます... 完了  
依存関係ツリーを作成しています  
状態情報を読み取っています... 完了  
アップグレードパッケージを検出しています... 完了  
以下のパッケージは保留されます:  
linux-generic-hwe-18.04 linux-headers-generic-hwe-18.04  
linux-image-generic-hwe-18.04  
以下のパッケージはアップグレードされます:  
bluez bluez-cups bluez-obexd libbluetooth3 libexiv2-14 libxkbcommon-dev  
libxkbcommon-x11-0 libxkbcommon0 linux-libc-dev  
アップグレード: 9 個、新規インストール: 0 個、削除: 0 個、保留: 3 個。  
3,319 kB のアーカイブを取得する必要があります。  
この操作後に 889 kB のディスク容量が解放されます。  
続行しますか? [Y/n]
```

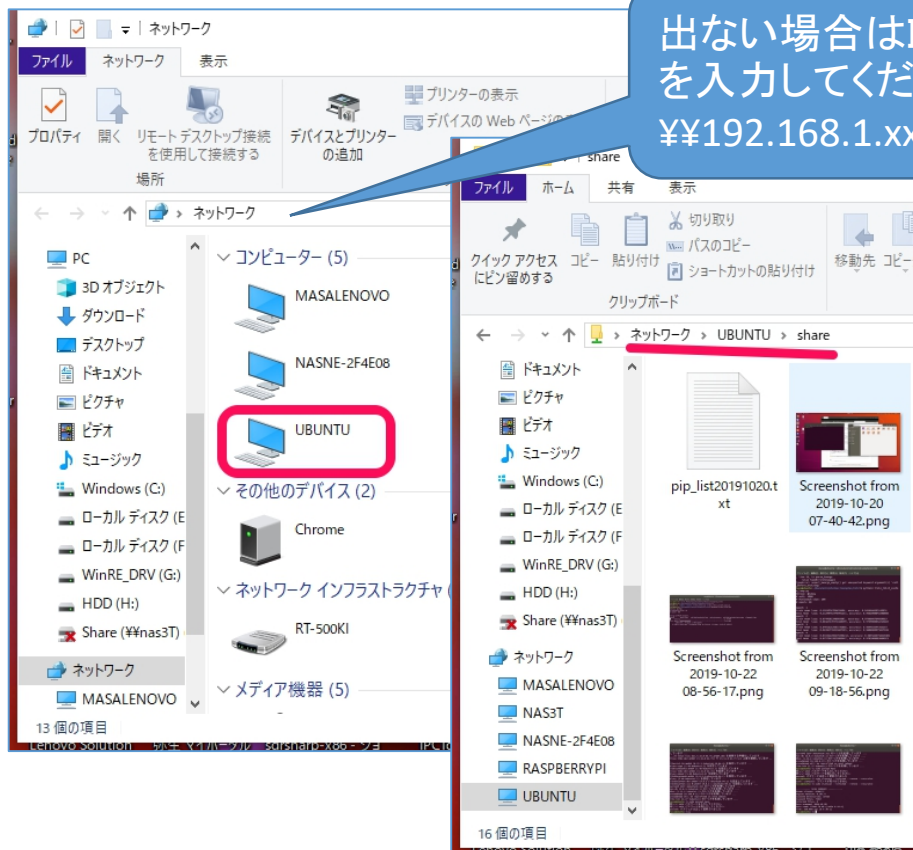
```
更新リスト取得  
$ sudo apt-get update  
アップグレード実施  
$ sudo apt-get upgrade
```

ubuntu運用マニュアル

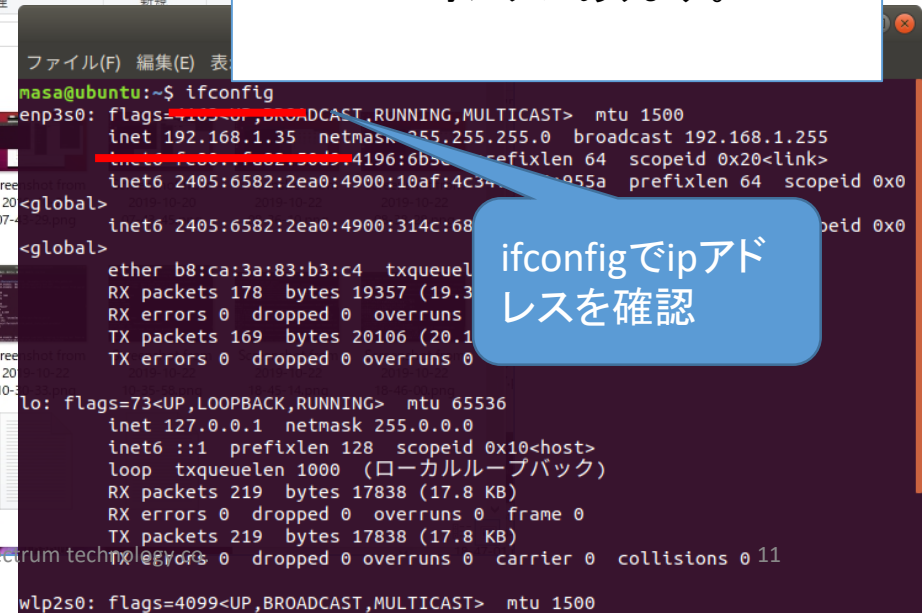
4. 日常運用

③ Samba

- Windowsとネットワークを共有し、ファイルの共有を可能とします。非常に便利です。既に設定済です。



- windowsPCのネットワークを確認します。
 - ① UBUNTUとネットワークに表示されます
 - ② ダブルクリックするとshareが出ます
 - ③ ubuntu上はhome>masa>shareにフォルダがあります。



ubuntu運用マニュアル

5. ソフトウェア一覧

区分	ソフト名	バージョン	備考
OS	ubuntu	18.04.3 LTS	
GPU用	cuDNN	7.6+cuda10	Nvidia用
画像	Opencv	3.4	
プログラム言語	python3	3.6.8	
	python2	2.7.15+	
AI用プログラム	tensorflow	1.14.0	Google
	chainer	6.4.0	Preferred Network
	caffe	1.0.0	berkeley
サンプルプログラム	Mnist,cifar10など		使用AI言語により比較が可能
その他	Jupyter notebook、matplotlibなど多数のpipライブラリ Samba,clamavなどの基本アプリ		

Chainer学習プログラム

①. Chainer概要

• Chainerとは

- Chainerは、Preferred Networkが開発したディープラーニングのオープン・ソース・ソフトウェアで2015年6月に公開されました。
- 特徴は、Pythonベースの深層学習フレームワークとして、“Define-by-Run”の手法を通じてユーザーが簡単かつ直感的に複雑なニューラルネットワークを設計するための高い柔軟性とパフォーマンスを兼ね備えています。

• 比較表

https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

ディープラーニング・ソフト	開発元	プログラム言語	分散対応	対応OS	GPU対応
Tensorflow	Google	Python, c++	○	Linux, mac, windows	○
Chainer	Preferred network(日本)	python	?	linux	○
Caffe	Berkeleyvision	C++	?	Linux, mac, windows	○
Theano	モントリオール大学	python	?	Linux, mac, windows	○

Chainer学習プログラム

③. Deep learningの例(cifar)

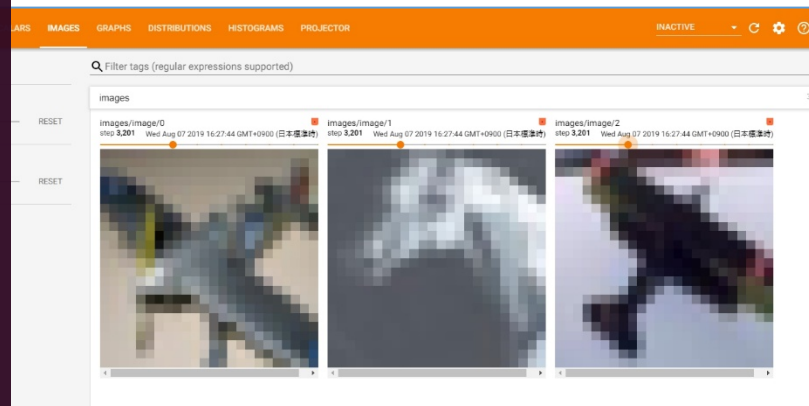
```
$ cd /home/masa/Documents/chainer/examples/cifar
```

```
$ python3 train_cifar.py
```

- cifar10は画像認識のベンチマークテスト用に準備された10分類の6万枚
 - airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.
- CPUの場合は、1 epochで46分かかります。(他に比べて遅いです)
- GPUの場合は、

```
masa@ubuntu: ~/Documents/chainer/examples/cifar
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
or http://127.0.0.1:8888/?token=f314d8b0dc5bc994d701bc579362f078bceb777e245f8de
4
Shutdown this notebook server (y/[n])? y
[C 09:07:53.294 NotebookApp] Shutdown confirmed
[I 09:07:53.295 NotebookApp] Shutting down 1 kernel
[I 09:07:53.496 NotebookApp] Kernel shutdown: 60a613b5-f91e-4f7f-9582-fbb231ef1f
b1
masa@ubuntu:~/Documents/notebooks$ cd /home/masa/Documents/chainer
masa@ubuntu:~/Documents/chainer$ cd /home/masa/Documents/chainer/examples/cifar
masa@ubuntu:~/Documents/chainer/examples/cifar$ ls
README.md  models  result  train_cifar.py  train_cifar_custom_loop.py
masa@ubuntu:~/Documents/chainer/examples/cifar$ python3 train_cifar.py
Device: @numpy
# Minibatch-size: 64
# epoch: 300

Using CIFAR10 dataset.
epoch      main/loss      validation/main/loss      main/accuracy      validation/main/acc
uracy      elapsed_time
█ total [.....] 0.09%
this epoch [#####] 25.60%
200 iter, 0 epoch / 300 epochs
0.24869 iters/sec. Estimated time to finish: 10 days, 21:34:09.791945.
```



Caffe学習プログラム

③. Deep learningの例(mnist)

```
$ cd /home/masa/Documents/caffe
```

```
$ ./data/mnist/get_mnist.sh データ取得
```

```
$ ./examples/mnist/create_mnist.sh 設定
```

```
$ ./examples/mnist/train_lenet.sh 学習
```

- Mnistは、手書き文字の画像認識のベンチマークテスト用に準備された6万枚のデータになります。
- CPUの場合は、1万回で15分かかります。
- GPUの場合は、

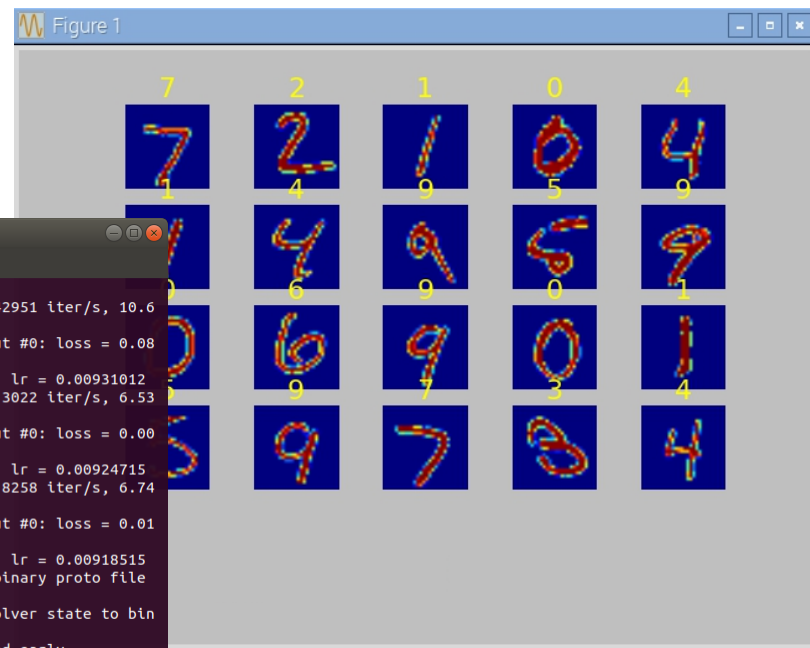
```
masa@ubuntu: ~/Documents/caffe
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)

masa@ubuntu:~/Documents/caffe$ ./examples/mnist/train_lenet.sh
I1023 10:02:17.791770 7269 caffe.cpp:137] Use CPU.
I1023 10:02:17.792075 7269 solver.cpp:45] Initializing solver from parameters:
test_iter: 100
test_interval: 500
base_lr: 0.01
display: 100
max_iter: 10000
lr_policy: "inv"
gamma: 0.0001
power: 0.75
momentum: 0.9
weight_decay: 0.0005
snapshot: 5000
snapshot_prefix: "examples/mnist/lenet"
solver_mode: CPU
net: "examples/mnist/lenet_train_test.prototxt"
train_state {
  level: 0
  stage: ""
}
I1023 10:02:17.792273 7269 solver.cpp:16]
examples/mnist/lenet_train_test.prototxt

masa@ubuntu:~/Documents/caffe
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)

2592 (* 1 = 0.0542592 loss)
I1023 10:03:36.028090 7269 solver.cpp:239] Iteration 1000 (9.42951 iter/s, 10.6
05s/100 iters), loss = 0.0849332
I1023 10:03:36.028123 7269 solver.cpp:258] Train net output #0: loss = 0.08
49332 (* 1 = 0.0849332 loss)
I1023 10:03:36.028131 7269 sgd_solver.cpp:112] Iteration 1000, lr = 0.00931012
I1023 10:03:42.563697 7269 solver.cpp:239] Iteration 1100 (15.3022 iter/s, 6.53
5s/100 iters), loss = 0.00580766
I1023 10:03:42.563733 7269 solver.cpp:258] Train net output #0: loss = 0.00
580767 (* 1 = 0.00580767 loss)
I1023 10:03:42.563742 7269 sgd_solver.cpp:112] Iteration 1100, lr = 0.00924715
I1023 10:03:49.308970 7269 solver.cpp:239] Iteration 1200 (14.8258 iter/s, 6.74
5s/100 iters), loss = 0.0178237
I1023 10:03:49.309114 7269 solver.cpp:258] Train net output #0: loss = 0.01
78237 (* 1 = 0.0178237 loss)
I1023 10:03:49.309126 7269 sgd_solver.cpp:112] Iteration 1200, lr = 0.00918515
^C I1023 10:03:49.509964 7269 solver.cpp:464] Snapshotting to binary proto file
examples/mnist/lenet_iter_1204.caffemodel
I1023 10:03:49.516571 7269 sgd_solver.cpp:284] Snapshotting solver state to bin
ary proto file examples/mnist/lenet_iter_1204.solverstate
I1023 10:03:49.519320 7269 solver.cpp:311] Optimization stopped early.
I1023 10:03:49.519335 7269 caffe.cpp:250] Optimization Done.

masa@ubuntu:~/Documents/caffe$
```





Opencv学習

①. Opencv概要

- Opencvとは

- OpenCVとはインテルが開発・公開したオープンソースのコンピュータビジョン向けライブラリ。画像認識では必須となります。
- ライブカメラ、動画、写真などで使用します。
- Pythonのライブラリとして使用します。

- USBカメラ

- ライブ動画を処理するためにライブ用のUSBカメラが必要になります。

Opencv学習

③. 顔検出(カメラ)

コマンド

```
$ cd
```

```
/home/masa/Documents/opencv/sample/OpenCV-  
Face-Recognition/FacialRecognition
```

- Opencvを使った、webカメラでの登録した顔を検出します。
- <https://www.instructables.com/id/Real-time-Face-Recognition-an-End-to-end-Project/>
- 顔の登録
- python3 01_face_dataset.py
 - Face id:0-3までを入力してください。
 - 自動で顔のスキャンが始まります。数分で終了します
- python3 02_face_training.py
- python3 03_face_recognition.py
 - カメラで認識します
 - プログラム内でidと名前を紐づけています。

