Anaconda3-2019.03 (2019/7/3)

利用方法

conda, anaconda に関する情報はウェブ上にたくさんありますので、それらも参考にしてください。

初期設定読み込み

以下のように設定ファイルを読み込むことで conda が有効になります。

bash, zsh の場合

```
% . /local/apl/lx/anaconda3-2019Jul/conda_init.sh
```

csh (tcsh) の場合

```
% source /local/apl/lx/anaconda3-2019Jul/conda_init.csh
```

読み込んだ時点では "base"
と呼ばれる「環境(environment)」になっています。環境名は以下のようにプロンプトの左側に表示されます。

```
(base) %
```

環境リスト

- base: 上記コマンドで読み込んだ時点での標準環境
  - python 3.7.3
  - numpy 1.16.4, scipy 1.2.1, pandas 0.24.2, pyyaml 5.1.1, pillow 6.0.0
  - tensorflow 1.14.0, keras 2.2.4, chainer 5.3.0, pytorch 1.1.0
    - GPU 用のパッケージはインストールされていません。各種基本的なパッケージと CPU 版のディープラーニング環境(tensorflow, keras, pytorch, chainer)が使えます
    - 研究で用いる標準的なパッケージ(pandas等)についてはこの環境で大抵のものはまかなえるはずです。
- chainer_gpu: Chainer の GPU 環境
  - python 3.7.3
  - chainer 5.3.0, cupy 6.0.0, cudatoolkit 9.0
- pytorch_gpu: PyTorch の GPU 環境
  - python 3.7.3
  - pytorch(-gpu) 1.1.0, torchvision(-gpu) 0.3.0, cudatoolkit 9.0
- tensorflow_gpu: TensorFlow + Keras の GPU 環境
  - python 3.7.3
  - tensorflow(-gpu) 1.13.1, keras(-gpu) 2.2.4, cudatoolkit 9.0

以上の既存環境についてはインストール済みパッケージのバージョンを変えるような変更は原則行いません。
新しいバージョンのパッケージを導入する際には別の環境を作るか、別の Anaconda を新規にインストールします。

環境切り替え

conda 設定を読み込んだ後、conda activate コマンドで環境を切り替えることができます。

```
(base) % conda activate pytorch_gpu
(pytorch_gpu) %
```

標準設定ならば、csh でも bash でも現在の環境名がプロンプト左側に表示されます。
逆に conda deactivate コマンドを実行すると環境から抜けることができます。

```
(pytorch_gpu) % conda deactivate
(base) % conda deactivate
%
```

独自環境の構築

この Anaconda 環境は共有領域に作られているため、ユーザがパッケージを直接導入することはできません。
独自にパッケージを導入したい場合は環境(のクローン)を作る必要があります。
新しい環境のファイルはデフォルトでは ~/.conda 以下に置かれます。

新規 anaconda 環境を作る場合の例(myenv は新しい環境名です):

```
(base) % conda create -n myenv anaconda
```

base 環境のクローンを作る場合(myenv は新しい環境名です):

```
(base) % conda create -n myenv --clone base
```

(メモ: conda_init.sh, conda_init.csh, samples_rccs/ もクローンされます。必要なければ後で消去してください)

以上の方法を使わず、ホームディレクトリ以下に独自の conda 環境(anaconda or miniconda)を作る方法もあります。
特殊な環境を必要とする方はそちらの方が使いやすいかもしれません。

MNIST サンプルの実行

MNIST を実行するサンプルが /local/apl/lx/anaconda3-2019Jul/samples_rccs 以下にあります。
単に実行するだけならば、サンプルのディレクトリを自分のディレクトリにコピーし、中の sh を jsub で投入するだけです。
conda の読み込みや環境選択はスクリプト内で行うようになっています。

なお、ログインシェルが csh で GPU 版を使う場合、スクリプト内部の module コマンドがエラーになるため、修正が必要かもしれません。(修正点はスクリプト中に記述があります。)

chainer の場合の例:

```
% cp -r /local/apl/lx/anaconda3-2019Jul/samples_rccs/chainer/ ~/chainer_sample_mnist
% cd ~/chainer_sample_mnist
% jsub -q PN sample-chainer-cpu.sh
```

(MNIST のデータについてはあらかじめ /local/apl/lx/anaconda3-2019Jul/samples_rccs/datasets
以下にコピーしてあり、それを読み込むようになっています。)

番外: conda を使わずに環境を作る場合

conda (anaconda, miniconda) を使わずに以下のように pip
コマンドでホームディレクトリ以下にパッケージを導入し、環境を構築することもできます。

例 (--user を付けると、ホームディレクトリ以下にパッケージが導入されます):

```
% pip install cupy-cuda91 chainer --user
```

(Python3 系を使う場合は pip3 としてください)

インストール詳細

ウェブページ

https://www.anaconda.com/

バージョン

2019.03

使用したファイル

- Anaconda3-2019.03-Linux-x86_64.sh

インストール手順

共用領域へのインストールのため、root
権限で実行しています。また、ログが非常に長いため、出力部分は大幅に省略しています。

base 環境 (非GPU環境として設定)

```
% sh Anaconda3-2019.03-Linux-x86_64.sh
...
Anaconda3 will now be installed into this location:
/root/anaconda3

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below
```

```
[/root/anaconda3] >>> /local/apl/lx/anaconda3-2019Jul
...
Do you wish the installer to initialize Anaconda3
by running conda init? [yes|no]
[no] >>> no
% /local/apl/lx/anaconda3-2019Jul/bin/conda shell.bash hook > /local/apl/lx/anaconda3-2019Jul/conda_init.sh
% /local/apl/lx/anaconda3-2019Jul/bin/conda shell.csh hook > /local/apl/lx/anaconda3-2019Jul/conda_init.csh
% . /local/apl/lx/anaconda3-2019Jul/conda_init.sh
(base) % conda update --all
...
The following packages will be DOWNGRADED:

  anaconda                          2019.03-py37_0 --> custom-py37_0
...
(base) % conda install tensorflow keras chainer opencv
...

The following packages will be REMOVED:

  pytables-3.5.2-py37h71ec239_1

The following packages will be DOWNGRADED:

  h5py                          2.9.0-py37h7918eee_0 --> 2.8.0-py37h989c5e5_3
  hdf5                            1.10.4-hb1b8bf9_0 --> 1.10.2-hba1933b_1

...
(base) % conda install -c pytorch pytorch-cpu torchvision-cpu
...
(base) % conda install pytables
...

The following NEW packages will be INSTALLED:

  pytables          pkgs/main/linux-64::pytables-3.4.4-py37ha205bf6_0
```

Chainer GPU用環境

```
% . /local/apl/lx/anaconda3-2019Jul/conda_init.sh
(base) % conda create --name chainer_gpu --clone base
...
(base) % conda activate chainer_gpu
(chainer_gpu) % conda remove chainer tensorflow keras pytorch-cpu torchvision-cpu
...
(chainer_gpu) % conda install cudatoolkit=9.0 cupy chainer
...

The following NEW packages will be INSTALLED:

  chainer           pkgs/main/linux-64::chainer-5.3.0-py37_0
```

```
  cudatoolkit       pkgs/main/linux-64::cudatoolkit-9.0-h13b8566_0
  cudnn             pkgs/main/linux-64::cudnn-7.6.0-cuda9.0_0
  cupy              pkgs/main/linux-64::cupy-6.0.0-py37h686fdb1_0
  fastrlock         pkgs/main/linux-64::fastrlock-0.4-py37he6710b0_0
  nccl              pkgs/main/linux-64::nccl-1.3.5-cuda9.0_0

...
```

PyTorch GPU用環境

```
% . /local/apl/lx/anaconda3-2019Jul/conda_init.sh
(base) % conda create --name pytorch_gpu --clone base
...
(base) % conda activate pytorch_gpu
(pytorch_gpu) % conda remove chainer tensorflow keras pytorch-cpu torchvision-cpu
...
(pytorch_gpu) % conda install -c pytorch cudatoolkit=9.0 pytorch torchvision cudnn
...

The following NEW packages will be INSTALLED:

  cudatoolkit       pkgs/main/linux-64::cudatoolkit-9.0-h13b8566_0
  cudnn             pkgs/main/linux-64::cudnn-7.6.0-cuda9.0_0
  pytorch           pytorch/linux-64::pytorch-1.1.0-py3.7_cuda9.0.176_cudnn7.5.1_0
  torchvision       pytorch/linux-64::torchvision-0.3.0-py37_cu9.0.176_1

...
```

(cudnn は pytorch には必要無いかもしれません)

TensorFlow + Keras GPU用環境

```
% . /local/apl/lx/anaconda3-2019Jul/conda_init.sh
(base) % conda create --name tensorflow_gpu anaconda
..
(base) % conda activate tensorflow_gpu
(tensorflow_gpu) % conda install cudatoolkit=9.0 tensorflow-gpu keras-gpu
...

  keras-applications pkgs/main/noarch::keras-applications-1.0.8-py_0
  keras-base         pkgs/main/linux-64::keras-base-2.2.4-py37_0
  keras-gpu          pkgs/main/linux-64::keras-gpu-2.2.4-0
  keras-preprocessi~ pkgs/main/noarch::keras-preprocessing-1.1.0-py_1
  tensorboard        pkgs/main/linux-64::tensorboard-1.13.1-py37hf484d3e_0
  tensorflow         pkgs/main/linux-64::tensorflow-1.13.1-gpu_py37hd37c573_0
  tensorflow-base    pkgs/main/linux-64::tensorflow-base-1.13.1-gpu_py37h8f37b9b_0
  tensorflow-estima~ pkgs/main/noarch::tensorflow-estimator-1.13.0-py_0
  tensorflow-gpu     pkgs/main/linux-64::tensorflow-gpu-1.13.1-h0d30ee6_0

...
(tensorflow_gpu) % conda install anaconda=custom=py37_0
```

```
...
(tensorflow_gpu) % conda install opencv
...

The following packages will be REMOVED:

  pytables-3.5.1-py37h71ec239_0

The following packages will be DOWNGRADED:

  h5py                       2.9.0-py37h7918eee_0 --> 2.8.0-py37h989c5e5_3
  hdf5                         1.10.4-hb1b8bf9_0 --> 1.10.2-hba1933b_1

...
(tensorflow_gpu) % conda install pytables
...
```

(base を clone した環境では依存関係でうまくいかなかったため、以上の手順を採用しています)