Anaconda3-2020.02+ (2020/5/13)

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/anaconda3-2020.02-May/conda_init.sh
```

csh (tcsh) の場合

```
% source /local/apl/lx/anaconda3-2020.02-May/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) %
```

注意点

大量のファイルにアクセスする初期化の操作は時間を要することがあります。
これはセンターで利用している分散ファイルシステムの性質が原因で、避けられません。
そのため、anaconda 環境の読み込みは必要時にのみ行うようにした方が無難だと思われます。
ただし、一旦読み込めば当面の間は高速に読み込めるようになります(キャッシュされるため)。
同様に実行中に影響が出ることは基本的にありません。

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 3.7.7
  - numpy 1.18.1, scipy 1.4.1, pandas 1.0.3, pyyaml 5.3.1, pillow 7.1.2, opencv 3.4.2, scikit-image 0.16.2, scikit-learn 0.22.1 他
  - tensorflow 2.1.0, keras 2.3.1, pytorch 1.5.0
    - 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)が使えます (chainer は導入していません)
    - 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ほぼまかなえるはずです。
- gpuenv: PyTorch, Tesorflow の GPU 環境
  - python 3.7.7
  - tensorflow 2.1.0, keras(-gpu) 2.3.1, cudatoolkit 10.1.243, cudnn 7.6.5

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 gpuenv
(gpuenv) %
```

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。

```
(gpuenv) % 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/anaconda3-2020.02-May/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。)

pytorch の場合の例:

```
% cp -r /local/apl/lx/anaconda3-2020.02-May/samples_rccs/pytorch/ ~/pytorch_sample_mnist
% cd ~/pytorch_sample_mnist
% jsub -q PN sample-pytorch-cpu.sh
```

(MNIST のデータについてはあらかじめ /local/apl/lx/anaconda3-2020.02-May/samples_rccs/datasets 以下にあり、それを読み込むようになっています。演算ノードで実行されるジョブでは外部ファイルを wget で取得するようなことはできません。ご注意下さい。)

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

2020.02

使用したファイル

- Anaconda3-2020.02-Linux-x86_64.sh (individual 版、BSD 3-clause license)

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大幅に省略しています。

```
% sh Anaconda3-2020.02-Linux-x86_64.sh
...
% /local/apl/lx/anaconda3-2020.02-May/bin/conda shell.bash hook > /local/apl/lx/anaconda3-2020.02-May/conda_init.sh
% /local/apl/lx/anaconda3-2020.02-May/bin/conda shell.csh hook > /local/apl/lx/anaconda3-2020.02-May/conda_init.csh
% . /local/apl/lx/anaconda3-2020.02-May/conda_init.sh
(base) % conda update --all
...
(base) % conda install opencv tensorflow keras
...
(base) % conda install pytorch=1.5.0=py3.7_cpu_0 torchvision cpuonly -c pytorch
...
(base) % conda create --name gpuenv python=3.7 anaconda
...
(base) % conda activate gpuenv
(gpuenv) % conda install cudatoolkit=10.1.243 cudnn opencv
(gpuenv) % conda install tensorflow=2.1.0=gpu_py37h7a4bb67_0 keras
...
(gpuenv) % conda install pytorch torchvision -c pytorch
...
(gpuenv) % conda remove keras
...
(gpuenv) % conda install keras-gpu
...
```

メモ

- (gpuenv の最後で keras を keras-gpu に置き換える操作はただのミス修正です…)
- gpuenv は base を clone するとうまくいかないため新規導入しています。
    - python のバージョンを base と同じにしたかったため、python=3.7 をつけています
    - update --all をしても期待通りパッケージが更新されないため、スキップしています。
- pytorch は公式 channel (pytorch) のものが一番導入が楽です。