M1 MacでPythonと機械学習をやろう

Unagi.py 勉強会37枚目～【祝4周年】機械学習・データサイエンスLT会～

2020/11/20 Hiroshi Sano

お前誰よ

- Hiroshi Sano @hrs_sano645 🏠:静岡の🗻見えるところ
- Job💼
  - 佐野設計事務所🚗⚙️📏: 自動車系機械の3D設計事務所
  - 米農家🌾
- Community👨‍💻
  - 🗻🐍: shizuoka.py, unagi.py, Python駿河
  - 🗻🐍: PyCon mini Shizuokaスタッフ
  - 🐍: PyCon JP 2020 チュートリアル講師

M1 Macbook Air買いました🎉

おススメ！
今日は~~自慢~~布教しに来ました。

まとめ

- M1 Macめちゃくちゃ快適
- Python使うならRosseta モードが基本的におすすめ
- M1最適化のTensorflowを試してみた

M1 Macめちゃくちゃ快適です

M1 Macめちゃくちゃ快適です

買ったもの

- M1 Macbook Air: 124,800円
- CPU 8core / GPU 7 Core
- 256GB: そんなに入れるものがなければ十分
    - 必要なら外付けSSD: 1TB1万円ぐらいで買える
- メモリ 16GB

M1 Mac快適なこと一覧

- Macbook Airはファンレス。他のものも通常はファン回らない
- 4K動画流していても平然としてる
- よく使うアプリはほぼ動く: chrome, vscode, office, などなど

おすすめ理由: 処理早い

- OS自体の挙動も早い
- Intelバイナリの変換（Rosseta）をしてもほとんど遅いと感じない
  - アプリの初回起動のバイナリ変換にもたつくけどそれ以降は至って普通
  - ほとんど多くのアプリをRossetaモードで使ってるけど不具合らしいものに当たってない
- 4K動画回しながら普通に作業していても気にならない
- しかも不安定さも感じない
  - Apple初物感が全くない

おすすめ理由: バッテリー持ち

ブラウザたくさん開いて作業していても二日ぐらいバッテリーが持ってしまう

- Chrome系ブラウザのタブを多数開いても
- Yotuube垂れ流ししていても

iPad感覚で使えてしまう（ちなみにiOSアプリもある程度入ります

おすすめ理由: ファンレス / ファンがほぼ回らない

- Macbook Airはファンレス。よほどのことをしない限り熱を持たない
- Mac mini , Macbook Proはファン搭載だけど、いろんなレビューでもファンがほぼ回らないらしい

そのほか、Macbook Airだとキーボードもシザーに戻って~~Touchbarもないので~~非常に快適

まだこれから部分

- armネイティブのアプリが出揃うのはこれから
- 夏場はよくわからない（冬場だから熱くないのかも？

正直悪い要素がないので、気になったら買いです

気になったら今すぐ買おう！💸

(mac miniはもっと安いので、そっちもおすすめ)

M1 Macめちゃくちゃ快t...

冬場は冷えたアルミの板...🥶

Python使うならRosseta モード

Pythonのインストール

macOS Bug SurにはPythonは標準で入ってない

- XCode or Command line tools 必須
  - -> Command line toolsを入れることでPython3.8のUniversal Binaryが入ります
- 3.9は最新版でUniversal Binary対応（3.9.1）
- それ意外のバージョン(2.7はCommand Line toolsに入ってる)はx86バイナリが使えるはず（そもそも使わない🤔）

Universal Binaryとは

- M1 Macはarmとx86両方のバイナリが動く
  - armバイナリはネイティブで動く
  - x86バイナリはRosseta2（というバイナリトランスレーター）で動く
- armとx86（Rosseta）両方対応のバイナリをUnivseral Binaryと呼ぶ

両方動くので、どちらで動いているか確認が必要

バイナリの対応アーキテクチャ確認方法

fileコマンド

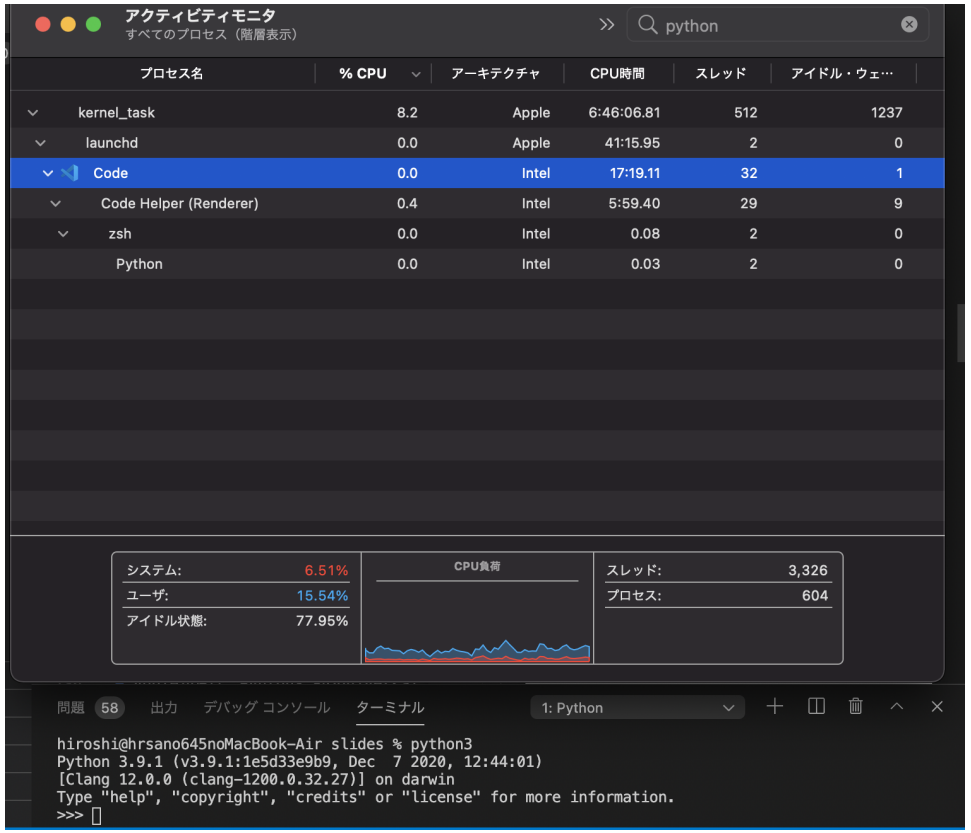
```
% file /usr/local/bin/python3.9
/usr/local/bin/python3.9: Mach-O universal binary with 2 architectures: [x86_64:Mach-O 64-bit executable x86_64] [arm64:Mach-O 64-bit executable arm64]
/usr/local/bin/python3.9 (for architecture x86_64):     Mach-O 64-bit executable x86_64
/usr/local/bin/python3.9 (for architecture arm64):      Mach-O 64-bit executable arm64
```

lipoコマンド

```
% lipo -archs /usr/local/bin/python3.9
x86_64 arm64
```

バイナリの対応アーキテクチャ確認方法

動いてるプロセスなら、アクティビティモニタを使う（一番簡単）



ターミナルでは(zsh)でどう動くか

- Bug Sur標準のzsh（デフォルトのシェル）はユニバーサルバイナリ
- ターミナル上で起動するコマンドはターミナルのアーキテクチャに従う（らしい
  - ユニバーサルバイナリのアプリはアプリの右クリックから見れる `情報を見る` からRosseta モードを起動させるか設定で変更可能
- サードパーティのターミナルアプリでも利用しているバイナリの種類で起動する
  - 例: x86のVS Codeからターミナル起動 -> x86のzshが起動

ターミナルでは(zsh)でどう動くか

ターミナルのシェルがどのアーキテクチャで動いているかを確認するには`uname -m` or `arch`コマンドが良い。

```
% arch
i386
% uname -m
x86_64
```

```
# armのzsh
(arm64) % arch
arm64
(arm64) % uname -m
arm64
```

ターミナルでは(zsh)でどう動くか

archコマンドを使ってzshのプロンプトカスタマイズするとさらにわかりやすい

```
## .zshrc に記載
export PROMPT="%n@%m(`uname -m`) %1~ %## "

## sourceコマンドで読み込む
hiroshi@hrsano645noMacBook-Air slides % source ~/zshrc
hiroshi@hrsano645noMacBook-Air(x86_64) slides %
```

ターミナルでは(zsh)でどう動くか

ターミナルで別のアーキテクチャのzshを動かしたい時は、archコマンドを使う

```
(x86_64) % arch -arm64e zsh # ここまでx86
(arm64) %  # <- ここからarmのzshで動く
```

そのPythonどっちのバイナリで動いてるの？

起動するターミナルのアーキテクチャで変わるので、意識して確認しよう。

ArmのPythonって使える？

ArmのPythonって使える？

- 標準では至って普通に動く
- venvも普通に動く。Pure Pyhtonなら問題ない（と思われる
- pipが問題

pipのアップグレードの様子

```
hiroshi@hrsano645noMacBook-Air(x86_64) test_arm_venv % arch -arm64e zsh
hiroshi@hrsano645noMacBook-Air(arm64) test_arm_venv % python3 -m venv .env
hiroshi@hrsano645noMacBook-Air(arm64) test_arm_venv % source .env/bin/activate
(.env) hiroshi@hrsano645noMacBook-Air(arm64) test_arm_venv % pip install -U pip
Collecting pip
  Using cached pip-20.3.3-py2.py3-none-any.whl (1.5 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 20.2.3
    Uninstalling pip-20.2.3:
      Successfully uninstalled pip-20.2.3
Successfully installed pip-20.3.3
(.env) hiroshi@hrsano645noMacBook-Air(arm64) test_arm_venv %
```

Rosseta(x86)モードの標準ターミナルからarmバイナリのzshを起動してvenv上で試す。

none-anyなら大丈夫そう

pipがmacos armに対応してない

- pipの仕様にmacOSのArmアーキテクチャを考慮してないからインストールもwheel生成もできない（まだらしい？
- 依存関係でちょっとでも引っかかると色々面倒（手動で解決しないといけない）

Python 3.9.1の macOS Big Sur/Apple Siliconサポート - python.jp

pip 20.3 release (Q4 2020) · Issue #8936 · pypa/pip

macos armのpipでnumpyを入れると

```
(.env) hiroshi@hrsano645noMacBook-Air(arm64) test_arm_venv % pip install numpy
Collecting numpy
  Using cached numpy-1.19.5.zip (7.3 MB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
    Preparing wheel metadata ...
```

ソースビルド走っちゃう（そもそもwheelのバイナリないし）

現時点ではM1 MacでArmネイティブのPythonの常用は難しい

意欲ある方はコントリビュートのチャンスです！M1 Mac買いま（略

Rosseta モードだとどうなの？

- 至って普通に動く。venv, pip などなど
- ただpipは最新版にする必要あり:
  - 正しくバイナリ判定がされなくてソースビルドが走ること
  - pipの最新: pip-20.3.3
  - `pip install -U pip` これ絶対
- 性能はRosseta依存: GPUとかの扱いよくわからん

Web開発ぐらいなら問題なく使えると思います。

Rossetaモードだとどうなの？

```
(.env) hiroshi@hrsano645noMacBook-Air(x86_64) test_86_venv % pip install numpy
Collecting numpy
  Using cached numpy-1.19.5-cp39-cp39-macosx_10_9_x86_64.whl (15.6 MB)
Installing collected packages: numpy
Successfully installed numpy-1.19.5
```

Rossetaモードだとどうなの？

（やっていて気がついたこと）

- 現時点でpandasをpipenvで入れるとpip lockがちゃんとできない
    - 今のところ、素直にvenv, pipを使ったほうがいいと思う

Rosseteaモードでnumpyベンチマークしてみる

提供: オーイシさん (@oec014)さん

ベンチ内容: numpyの内積計算のループ（ソースはできたら公開）

比較対象は

- Python3.8.2 / rosseta / numpy 1.19.5
- Python3.8.2 / arm native / tfmac numpy
    - tensorflow-macosのnumpy利用
- Python3.9.0 / Ryzen7 5800X / numpy 1.19.5

Rosseteモードでnumpyベンチマーク 結果

環境	ベンチ結果: 5 回測定/最小値 (s)
Python3.8.2 / rosseta / numpy 1.19.5	38.50
Python3.8.2 / arm native / tfmac include numpy 1.18.5	20.62
Python3.9.0 / Ryzen7 5800X / numpy 1.19.5	306.63

所感

- arm nativeやばい（震え
- rossetaが振るわないのがCPUのトランスコードの問題なのか、メモリなのかは不明
    - CPUはベンチ回している間、各コアは90%は超えていたがずっと100%ではなかった。その辺なる

armネイティブ環境は夢がある

Ryzenが振るわない理由は最適化されていないかも（pypiのnumpyだから？）

気になったら今すぐ買おう！💸

（いやMac Pro待ったほうがいいかもだけど）

M1最適化のTensorflowを試してみた

tensorflow-macosがある

- Appleが専用のビルドをしたものを公開
  - apple/tensorflow_macos: TensorFlow for macOS 11.0+ accelerated using Apple's ML Compute framework.
- numpyもビルド済み
  - PyPI側からarmネイティブでビルドしてみたけど動かず
  - armネイティブベンチはこちらのnumpyを使ってます
- pandasは入ってないらしい

勢いに任せてインストールしてみたものの、ネタがない。

（インストールでなんかあるだろうと思ったらさっくり入ってしまった）

当人DLやっていないのである（やりたい

なんかネタはないものか...

そこに救世主が！（ババーン


人気ブロガー
からあげ先生の
とにかく楽しい
AI自作教室
1週間であなたも
日経BP
じゃんけん画像を識別
実在しない人の顔を生成
ルンバを自分で制御
ドグラ・マグラ風の文を作る、Vチューバーを動かす…ほか！
Pythonでサクッと作れる
からあげ著
AIプログラマー！
全Pythonコードを Google Colab用のノートブック形式で用意

からあげ先生のAI本のコードを試してみた

からあげ先生本とは

- 「AI」に関する技術を楽しく紹介
- 「じゃんけん画像を自動認識する」「実在しない人の顔画像を生成する」「ルンバをAIで制御する」
- 画像処理、自然言語、画像生成、姿勢推定、エッジコンピューティングといった多数の技術の紹介と実装
- ソースコードはGitHubに公開済み。Google Colabで試せる

盛りだくさんすぎる...

tensorflow-macosで動かしてみる

２章の画像分類(tensorflowバージョン）を試しました。

元のノートブック: karaage-ai-book/02_karaage_ai_book_image_classification_tf2_x.ipynb at master · karaage0703/karaage-ai-book

tensorflow-macosで動かしてみる

動かす環境を用意する

- armバイナリのPython3.8でjupyterlabは動く
- わけあってminiforgeのcondaで環境を作り直す
  - pypiのmatplotlibはインストールでない
  - miniforgeはM1 Mac向けのインストーラーあり。ライブライのインストールも同じく可能
- notebookをコピーして、動かせられるように改造
  - 必要なパッケージをインストール
  - Google CoLab向けのパッケージやコマンド実行を省く

改造したjpnbファイル ->
tfmac_ver__02_karaage_ai_book_image_classification_tf2_x.ipynb

試した様子

- いとも簡単に動いてしまった…（ネタとして
- モデルの構築時の時間はTensorflow-macosの方が早かった
  - Google Colabの無料環境: 約17.5s
  - Tensorflow-macos: 約9.5s

気になったら今すぐ買おう！💸

M1 Macを買ったらからあげ先生の本も買いましょう！

まとめ

- M1 Macめちゃくちゃ快適
- Python使うならRosseta モードが基本的におすすめ
- M1最適化のTensorflowを試してみた

気になったら今すぐ買おう！💸

冬場はアルミの板で冷えます🧊

参考

- M1 Mac
- M1 Macの開発環境 - Qiita
- 人気ブロガーからあげ先生のとにかく楽しいAI自作教室｜日経BPブックナビ【公式サイト】