TensorFlow の概要

（AIカメラの準備）

URL: https://www.kkaneko.jp/ai/aicamera/index.html

金子邦彦

TensorFlow

- 機械学習のソフトウエア
- Python, C/C++ 言語から利用可能．機械学習のアプリケーションを簡単に作成できるもの
- プロセッサ（CPU）, GPU, Google TPU で動く
- Google 社のディープラーニング研究プロジェクト
- 2015年11月に最初のリリース

  ソースコード等の公開（オープンソース）．

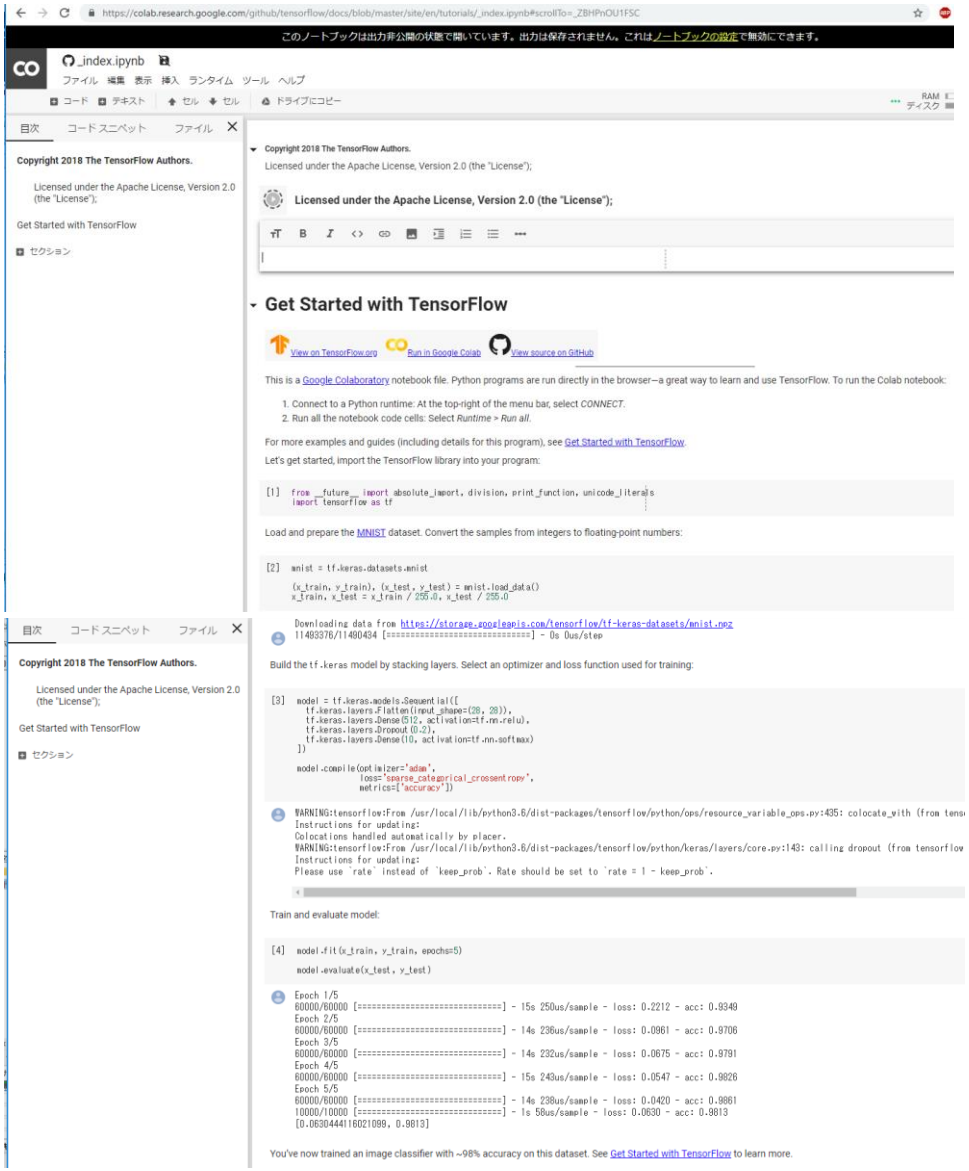
  ダウンロードは簡単
- オンラインのデモもある

TensorFlow のサイトで公開されている プログラム例


```python
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(512, activation=tf.nn.relu),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

https://www.tensorflow.org/tutorials

オンラインのデモの例



https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/_index.ipynb#scrollTo=_ZBHPnOU1FSC

このノートブックは出力非公開の状態で開いています。出力は保存されません。これはノートブックの設定で無効にできます。

_index.ipynb

ファイル 編集 表示 挿入 ランタイム ツール ヘルプ

コード テキスト セル セル ドライブにコピー

目次 コードスニペット ファイル

Copyright 2018 The TensorFlow Authors.

Licensed under the Apache License, Version 2.0 (the "License");

Get Started with TensorFlow

セクション

Copyright 2018 The TensorFlow Authors.

Licensed under the Apache License, Version 2.0 (the "License");

Licensed under the Apache License, Version 2.0 (the "License");

Get Started with TensorFlow

View on TensorFlow.org Run in Google Colab View source on GitHub

This is a Google Colaboratory notebook file. Python programs are run directly in the browser—a great way to learn and use TensorFlow. To run the Colab notebook:

1. Connect to a Python runtime: At the top-right of the menu bar, select *CONNECT*.
2. Run all the notebook code cells: Select *Runtime* > *Run all*.

For more examples and guides (including details for this program), see Get Started with TensorFlow.

Let's get started, import the TensorFlow library into your program:

```
[1]  from __future__ import absolute_import, division, print_function, unicode_literals
     import tensorflow as tf
```

Load and prepare the MNIST dataset. Convert the samples from integers to floating-point numbers:

```
[2]  mnist = tf.keras.datasets.mnist

     (x_train, y_train), (x_test, y_test) = mnist.load_data()
     x_train, x_test = x_train / 255.0, x_test / 255.0
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [==============================] - 0s 0us/step
```

Build the `tf.keras` model by stacking layers. Select an optimizer and loss function used for training:

```
[3]  model = tf.keras.models.Sequential([
       tf.keras.layers.Flatten(input_shape=(28, 28)),
       tf.keras.layers.Dense(512, activation=tf.nn.relu),
       tf.keras.layers.Dropout(0.2),
       tf.keras.layers.Dense(10, activation=tf.nn.softmax)
     ])

     model.compile(optimizer='adam',
                   loss='sparse_categorical_crossentropy',
                   metrics=['accuracy'])
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/resource_variable_ops.py:435: colocate_with (from tens
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/layers/core.py:143: calling dropout (from tensorflow
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
```

Train and evaluate model:

```
[4]  model.fit(x_train, y_train, epochs=5)

     model.evaluate(x_test, y_test)
```

```
Epoch 1/5
60000/60000 [==============================] - 15s 250us/sample - loss: 0.2212 - acc: 0.9349
Epoch 2/5
60000/60000 [==============================] - 14s 236us/sample - loss: 0.0961 - acc: 0.9706
Epoch 3/5
60000/60000 [==============================] - 14s 232us/sample - loss: 0.0675 - acc: 0.9791
Epoch 4/5
60000/60000 [==============================] - 15s 243us/sample - loss: 0.0547 - acc: 0.9826
Epoch 5/5
60000/60000 [==============================] - 14s 238us/sample - loss: 0.0420 - acc: 0.9861
10000/10000 [==============================] - 1s 58us/sample - loss: 0.0630 - acc: 0.9813
[0.06304441160210991, 0.9813]
```

You've now trained an image classifier with ~98% accuracy on this dataset. See Get Started with TensorFlow to learn more.

用語集

- **Cloud TPU** は，Google が提供する機械学習のクラウドサービス，そこで使用されている TensorFlow 処理用専用の機器（チップ）．TensorFlow を用いて，高速に機械学習できるとされる．
- TensorFlow の**データセット**として，音声，画像，テキスト，ビデオのデータが多数公開されており，訓練（学習）に利用できる

  https://github.com/tensorflow/datasets
- **Keras** は，TensorFlow を用いてのディープラーニング（深層学習）での**モデル**の構築と，その訓練（構築）を簡単に行えるようにするソフトウエア

用語集

- **Keras** の**モデル**は，複数の層が組み合わさったもの．単純に層を積み重ねたもの（シーケンシャル）や，複雑な構成のもの（グラフ）がある
- **Keras** の**層**には，**活性化関数**，**層の重み**（カーネル，バイアスなど）を設定できる
- **Keras** の**層**には，**全結合**，**畳み込み（コンボリューション）**などの種類がある
- 訓練（学習）のために，**オプティマイザ**の設定を行う
- **Keras** の**モデル**の構成や**オプティマイザ**の設定は，保存できる
- 訓練（学習）の結果は，**モデルの重み**になる．**モデルの重み**も保存できる

用語集

- TensorFlow の**データフローグラフ**は，データの流れを定義するもの
- データフローグラフの中には，TenforFlow が定める**オペレーション**を置く．
- **オペレーション**と**オペレーション**の間を，**テンソル**と呼ばれるデータがやりとりされる

- 以上のことは，Keras を介しても行うことができる．そのとき，**オペレーション**や**テンソル**を意識することは少ない