TensorFlow での画像分類

1　今回の授業の目的

- TensorFlow というフレームワークを使って画像認識を行う方法を学びます。
- python のバージョンや module 等の環境を切り替える方法を学びます。
- ノートパソコンにおける TensorFlow のインストール方法を学びます。
- TensorFlow のプログラムの特徴を学びます。
- WEB カメラからの画像の取得方法を学びます。

2　概要

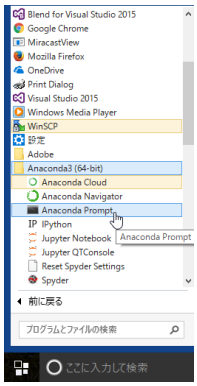
今までの授業では、ディープラーニングのフレームワークとして主に Chainer を使ってきました。今回の授業では、TensorFlow というフレームワークを使って画像認識を行います。TensorFlow は、Google が開発したディープラーニングのフレームワークです。Computational Graph を作成して、それを実行することにより、モデルの訓練や計算を行います。ノートパソコンに WEB カメラを接続して、画像認識を行うプログラムを作成します。

3　tensorenv 環境の作成

ここでは、TensorFlow を自分のＰＣ上で使うために必要な環境設定を行います。

第 1 回目の授業を欠席した人は、このテキストの最後の章「Widnows の環境設定」を参照して、Anaconda と Visual C++ 2015 Build Tools をインストールしてから、以下の作業を行って下さい。

3.1 Anaconda Prompt の立ち上げ



Anaconda Prompt を選択します。

3.2 tensorenv 環境の作成

作業ディレクトリへ移動します。ホームディレクトリの下に「Lesson」というディレクトリがあることを前提に作業します。ない場合には作成して下さい。

```
cd %homepath%
cd Lesson
```

python3.5 を使う環境を作成します。環境の名前を「tensorenv」とします。環境の作成が終ったら、その環境に入ります。

```
conda create --name=tensorenv python=3.5
activate tensorenv
```

環境に入ると、コマンドプロンプトの先頭に環境名（tensorenv）が表示されます。

3.3 TensorFlow のインストール

作成した環境に、TensorFlow やその他、プログラムの実行に必要なモジュール等をインストールします。この TensorFlow をバックエンドに、容易に素早くディープラーニングプロトタイプの迅速な実験を可能にすることに重点を置いて開発されたライブラリが Keras です。Keras もあわせてインストールしておきます。インストール中に、「Proceed ([y]/n)?」と聞かれたら「y」を入力します。

```
pip install --upgrade pip
CPU版
pip install tensorflow
conda install scipy
pip install keras
conda install matplotlib
conda install -c menpo opencv3
```

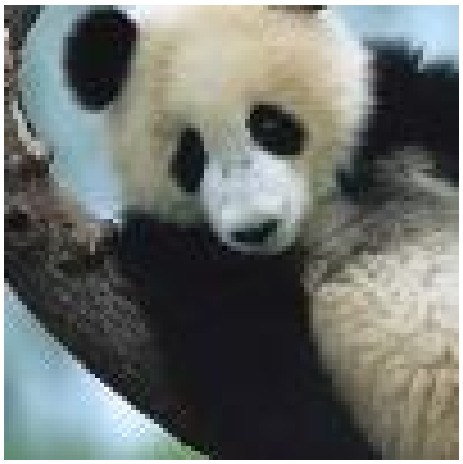
3.4 実行用のツールのダウンロード

実行用のツールをダウンロードします。そして、関連ファイルの入っているディレクトリに移動します。

```
git clone https://github.com/k1nk/ch7.git
cd ch7
```

3.5 画像の認識

classify_image.py を実行すると、訓練済みのモデルを/tmp/imagenet にダウンロードします。また、classify_image.py では、デフォルトでパンダの画像 (/tmp/imagenetcropped_panda.jpg 100 × 100 ピクセル) が用意されています。パラメータを与えずに実行すると、それを認識します。実際に、認識させてみましょう。



cropped_panda.jpg

```
python classify_image.py

giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca (
    score = 0.89632)
indri, indris, Indri indri, Indri brevicaudatus (score = 0.00766)
lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens (
    score = 0.00266)
custard apple (score = 0.00138)
earthstar (score = 0.00104)
```

3.6 自分の用意した画像の認識

–image_file 引数を編集することで、他の JPEG 画像を与えることができます。現在のディレクトにある「what_is_this.jpg」というファイルを認識させてみます。画像を 100 × 100 ピクセルに切りとってから、認識

させます。

what_is_this.jpg

画像を 100 × 100 ピクセルに切り取ります。

```
python cropimg.py what_is_this.jpg what_is_this_cropped.jpg
```

```
python classify_image.py --image what_is_this_cropped.jpg

Samoyed, Samoyede (score = 0.83239)
keeshond (score = 0.01841)
Pomeranian (score = 0.00555)
Japanese spaniel (score = 0.00443)
Eskimo dog, husky (score = 0.00177)
```

サモエド (Samoyed) は、ロシアのシベリアを原産地とする犬の品種のひとつです。典型的なスピッツ系の体型をしており、シベリアン・スピッツとも呼ばれます。これ以外にも、適当な画像を現在のディレクトリに配置して、クロップした上で、認識させて見ましょう。

答えを日本語で表示するように修正したバージョンもあります。

```
python classify_image_jp.py --image what_is_this_cropped.jpg

サモエド, (score = 0.83239)
keeshond (score = 0.01841)
ポメラニアン, ポメラニアけん, ポメラニア犬, スピッツ, (score = 0.00555)
狆, ちん, (score = 0.00443)
エスキモー犬, エスキモーけん, (score = 0.00177)
```

3.7 カメラからの画像認識

ノートパソコンに WEB カメラを接続して、画像の認識を行ってみましょう。

```
copy classify_image.py classify_camera.py
```

main 関数は以下の通りとなっています。

```
1 def main(_):
2   maybe_download_and_extract()
3   image = (FLAGS.image_file if FLAGS.image_file else
4            os.path.join(FLAGS.model_dir, 'cropped_panda.jpg'))
5   run_inference_on_image(image)
```

それを下記のように書き換えます。

```
1  import cv2
2  def main(_):
3    maybe_download_and_extract()
4    cam = cv2.VideoCapture(0)
5    #cam = cv2.VideoCapture(1)
6    while True:
7      ret, img = cam.read()
8      h = img.shape[0]
9      w = img.shape[1]
10     sq_len = min(h,w)
11     h_offset = (h - sq_len) // 2
12     w_offset = (w - sq_len) // 2
13     sq_img = img[h_offset:h_offset+sq_len,w_offset:w_offset+sq_len]
14     cv2.imshow('cam',sq_img)
15
16     sq_img_sm = cv2.resize(sq_img,(100,100))
17
18     k = cv2.waitKey(25) & 0xFF
19     if k == 27 or k == ord('q'):
20       print("--bye--")
21       break
22     if k == ord('s'):
23       print("--run_inference--")
24       cv2.imwrite('cam.jpg',sq_img_sm)
25       run_inference_on_image('cam.jpg')
26       print("--end run_inference--")
```

```
27
28    cam.release()
29    cv2.destroyAllWindows()
```

cam = cv2.VideoCapture(0) の「0」はデバイスの番号です。複数のビデオ入力用デバイスが接続されている場合には、cv2.VideoCapture(1) 等として、入力デバイスをを切り替えて下さい。

以下でプログラムを実行します。

```
python classify_camera.py
```

「s」キーを押すと、画像を認識し、結果を返します。「q」キーを押すとプログラムを終了します。キー入力は、表示されているビデオを前面にした状態で行って下さい。