

深層学習ライブラリ TensorFlowで犬猫判定にトライ

鈴木 量三朗

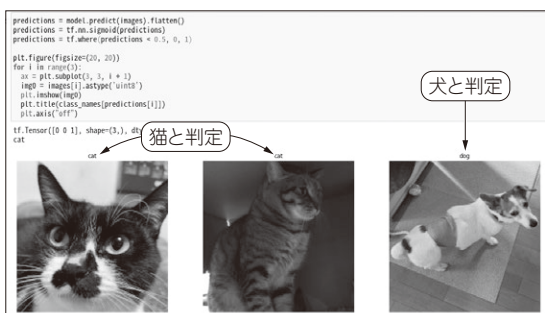


図1 筆者の猫は正しく猫と判定された

GPUは機械学習の場で広く使われています。オープンソースで公開されている機械学習のソフトウェア・ライブラリであるTensorFlowでもGPUを利用してその威力を見ることができます。筆者が使用しているバージョン2.8では、環境変数によってGPUを使う/使わないを選択できます。

なお、TensorFlowは内部でPythonのライブラリcuDNNを使用することから、CUDAをインストールするだけではインストールされないcuDNNを、追加でPCにインストールしておく必要があります。

学習済みモデルの制作

TensorFlowの公式チュートリアル^{注1}を参考に、猫と犬の識別をします(図1)。

● ステップ1…データセットの入手

数千枚の犬と猫の画像を含むデータセットをダウンロードします。GPUとCPUとの性能差が出やすいように、バッチ・サイズを512にします^{注2}(リスト1)。

注1: https://www.tensorflow.org/tutorials/images/transfer_learning

注2: GPUのテストのために512にしましたが、うまく動かない場合があります。通常はバッチ・サイズは32にしたほうがよいでしょう。

リスト1 データセットの入手

```
_URL = 'https://storage.googleapis.com/mledu-
      datasets/cats_and_dogs_filtered.zip'
path_to_zip = tf.keras.utils.get_file('cats_and_
      dogs.zip', origin=_URL, extract=True)
PATH = os.path.join(os.path.dirname(path_to_zip),
                    'cats_and_dogs_filtered')

train_dir = os.path.join(PATH, 'train')
validation_dir = os.path.join(PATH, 'validation')

BATCH_SIZE = 512 ← バッチ・サイズ
IMG_SIZE = (160, 160)

train_dataset = image_dataset_from_directory(
    train_dir, shuffle=True, batch_size=BATCH_SIZE,
    image_size=IMG_SIZE)
```

リスト2 学習の際にはランダムに変換するフィルタをはさむ

```
data_augmentation = tf.keras.Sequential([
    tf.keras.layers.experimental.preprocessing.
        RandomFlip('horizontal'),
    tf.keras.layers.experimental.preprocessing.
        RandomRotation(0.2),
])
```

リスト3 MobileNet V2の基本モデルを使用する

```
IMG_SHAPE = IMG_SIZE + (3,)
base_model = tf.keras.applications.MobileNetV2(
    input_shape=IMG_SHAPE, include_top=False,
    weights='imagenet')
base_model.trainable = False
```

● ステップ2…学習

学習の際には、ランダムに変換するフィルタをはさみます(リスト2)。ここではMobileNet V2の基本モデルを使用します(リスト3)。このモデルでは160×160×3(つまり160×160のRGB画像)を、5×5×1280の特徴ブロックに変換します。今回のプログラムではベースとして学習済みのモデルを利用し、その部分についてはそのまま使うようになっています。

● ステップ3…レイヤの追加

出力が5×5×1280のままだと、今回の犬か猫かを