

● 「5-10 転移学習とファインチューニング」のプログラムについて

本書の「5-10 転移学習とファインチューニング」において作成したプログラムは、TensorFlow2.0 で作成されたものですが、その後バージョンアップされた TensorFlow2.1 においては、うまく学習が進まない場合があることがわかりました。

この場合、初期の学習率を 0.0001 に引き下げると、正常に学習が行われます。つきましては、これに該当する場合、368 ページのスケジューラーの設定において、以下のように学習率の初期値を 0.0001 に設定して試してみてください。

▼368 ページ記載の学習率のスケジューラー (step_decay()関数)

```
def step_decay(epoch):
    initial_lrate = 0.0001 # 学習率の初期値
    drop = 0.5             # 減衰率は 50%
    epochs_drop = 10.0    # 10 エポック毎に減衰する
    lrate = initial_lrate * math.pow(
        drop,
        math.floor((epoch)/epochs_drop)
    )
    return lrate
```

● 「5-10 転移学習とファインチューニング」を Google Colab で実行した場合、学習が停滞する現象について

「5-10 転移学習とファインチューニング」においては、損失関数にスパースラベル対応クロスエントロピー誤差として、

```
loss='sparse_categorical_crossentropy'
```

を指定しています。本書で使用している Jupyter Notebook では（『「5-10 転移学習とファインチューニング」のプログラムについて』の場合を除く）問題なく学習できますが、該当のプログラムを Google Colab のような外部の Web サービスで提供されるノートブックで実行した場合、学習が進まないことがあります。この場合は、以下の措置を行って試してみてください。

・ 360 ページ サンプルコード 5-10-1 セル 1 のコード

末尾に One-hot 化を行うコードを追加して、正解ラベルを One-hot 化します。

```
from tensorflow.keras import datasets, utils
import tensorflow as tf
```

```
(x_train, t_train), (x_test, t_test) = datasets.cifar10.load_data()
x_train = tf.image.resize(x_train, [64, 64])
```

```
# One-Hot 化を行うコードを追加
```

```
t_train = utils.to_categorical(t_train, 10)
```

```
t_test = utils.to_categorical(t_test, 10)
```

・ サンプルコード 5-10-2 の 362 ページ、モデルのコンパイルの箇所

'sparse_categorical_crossentropy'を'categorical_crossentropy'に変更します。

```
model.compile(
    # 損失関数はクロスエントロピー誤差
    loss='categorical_crossentropy',
    # オプティマイザーは Adam
    optimizer=optimizers.Adam(lr=0.001),
    metrics=['accuracy'])
```