

石原 幸治 <ichara.com@gmail.com>

2018/03/25

DL4US 最終課題レポート

私が勤めている会社が先日、高精度な位置情報と時刻情報を活用したソリューションを発表し

た。空港車両の自動運転の実現を目指したさまざまな実験も行われる予定である。私の将来の

業務において、道路標識等の画像分類技術を活用できる機会があるかもしれないと思い、

German Traffic Sign Recognition Benchmark (GTSRB)に取り組んだ。実装において

は、自分の興味がある Capsules Network(CapsNet)を使用した。

GTSRB、CapsNet についてはそれぞれ下記資料を参考にし、ソースコードも流用した。

[1] Keras Tutorial - Traffic Sign Recognition

<<https://chsasank.github.io/keras-tutorial.html>>

[2] keras/examples/cifar10_cnn_capsule.py

<https://github.com/keras-team/keras/blob/master/examples/cifar10_cnn_capsule.py>

以下における<初版>とは、資料[1]の GTSRB データを読み込む処理を、単純に資料[2]のモ

デルに接続したものである。<改善版>とは、<初版>のテスト結果を見て改善を試みた中で、最

もテスト結果が良かったものである。<改善版>のテスト結果(精度)は<初版>に対して 0.2%

向上した(96.7%→96.9%)。この取り組みの過程を以下に記す。

<初版>の学習過程、結果

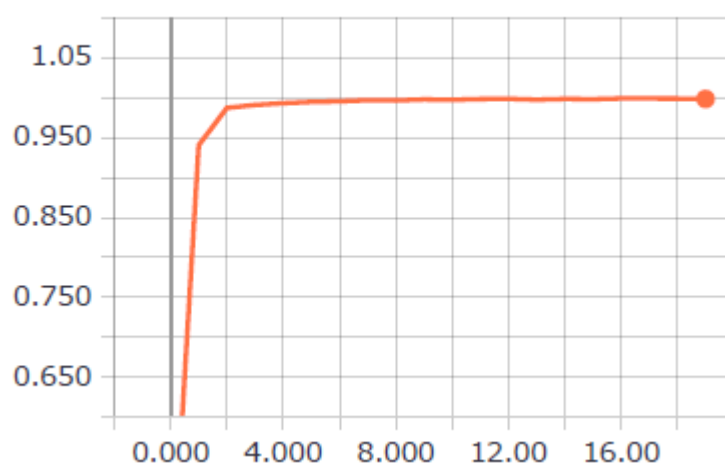
テスト結果(精度) : 96.7%

テストデータ(バリデーションとは別)は official GTSRB test set を使用した。

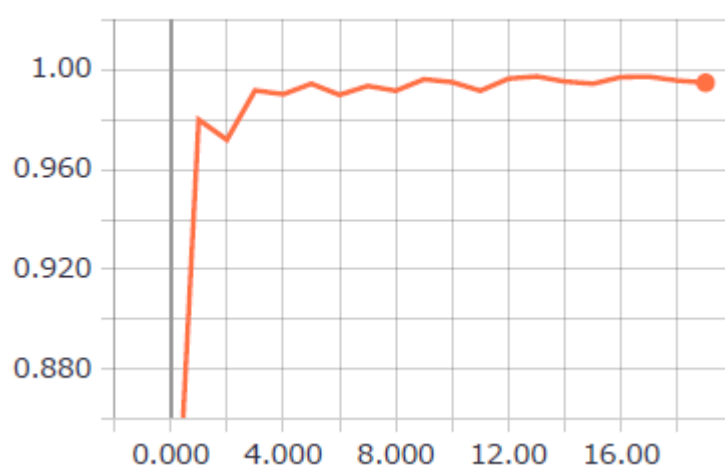
学習過程 :

epoch 数は<初版>、<改善版>共に 20 である。

acc



val_acc



epoch=12 前後でバリデーションの精度が頭打ちになっている。

結果分析：

<初版>のテストで誤答数が多い分類の上位 1 位、2 位を以下に示す。

図は左から順に、テストの画像データ(の一例)、正解の標識、誤答の標識である。



誤分類(1-2)から、トレーニングデータの拡張(回転)が有効と予想した。また、学習過程のグラフから、過学習が発生している可能性が考えられた。

<改善版>の改善内容、学習過程、結果

改善内容：

- ImageDataGenerator によるデータ拡張

下記パラメータを設定

```
rotation_range=20, # randomly rotate images in 0 to 180 degrees
width_shift_range=0.1, # randomly shift images horizontally
height_shift_range=0.1, # randomly shift images vertically
```

- ドロップアウト層の追加

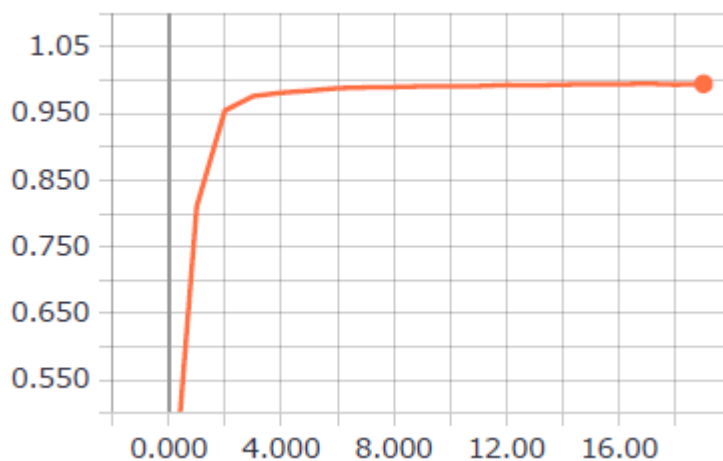
下記黄色部分を追加

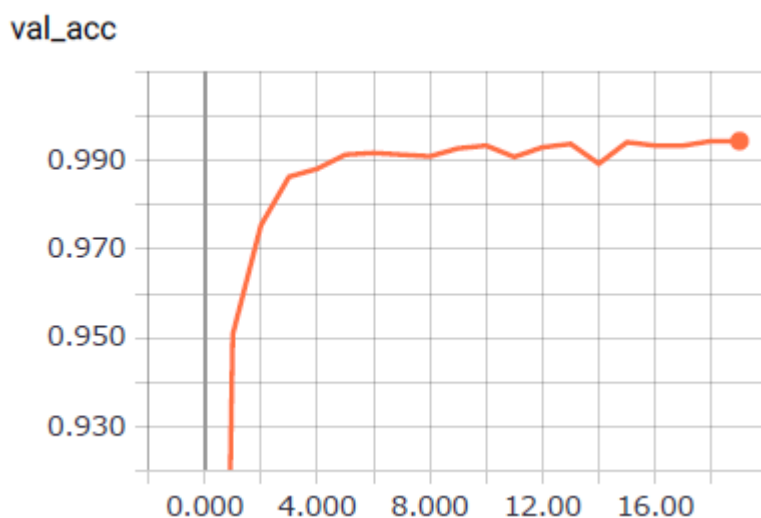
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, None, None, 3)	0
conv2d_1 (Conv2D)	(None, None, None, 64)	1792
conv2d_2 (Conv2D)	(None, None, None, 64)	36928
average_pooling2d_1 (Average)	(None, None, None, 64)	0
dropout_1 (Dropout)	(None, None, None, 64)	0
conv2d_3 (Conv2D)	(None, None, None, 128)	73856
conv2d_4 (Conv2D)	(None, None, None, 128)	147584
reshape_1 (Reshape)	(None, None, 128)	0
dropout_2 (Dropout)	(None, None, 128)	0
capsule_1 (Capsule)	(None, 43, 16)	88064
lambda_1 (Lambda)	(None, 43)	0
Total params: 348,224		
Trainable params: 348,224		
Non-trainable params: 0		

テスト結果(精度) : 96.9% (0.2%向上)

学習過程 :

acc





epoch=12 以後もバリデーションの精度の微増が見られる。

結果分析：

<改善版>のテストで誤答数が多い分類の上位 1 位、2 位を以下に示す。



誤分類(2-1)の誤答数が<初版>より増えた原因については、データ拡張(回転)の影響

が考えられるが、有効な対策は見つからなかった。

トライ&エラーした点等

- CapsNet の DigitCaps 16(ベクトル)x43(分類) を 64x43 に変更したが、改善されなかった(テスト結果 : 96.1%)。

- <改善版>の出力付近への Dense、Dropout の追加も効果が無かった。
- ModelCheckpoint コールバックを使って、トレーニング中の一番良いモデルをセーブ /ロードしようとしたが、Keras の問題によりロード出来なかった。
- 学習過程の可視化には TensorBoard、テスト結果の分析には GTSRB の Result analysis application をビルドして利用した。Deep Learning に取り組む上でツールを含む環境整備が重要であることを認識した。

所感

本講座によって Keras を使った Deep Learning を学習する機会/動機付けが得られたことを感謝いたします。講座の高度な内容によって視野が広がりました。ありがとうございました。