

1.

CNN\_train\_Fashion.py は 2 層の 2xConvolution+Pooling、2 層の全結合層により Fashion MNIST データセット ( 28x28 グレース ケールの画像, 0:T-シャツ/トップ, 1:ズボン, 2:プルオーバー, 3:ドレス, 4:コート, 5: サンダル, 6: シャツ, 7: スニーカー, 8: バッグ, 9: アンクルブーツ, ) を分類する CNN を構成し, 学習を行うスクリプトである。

(a)

配布しているスクリプトは全データ (60,000) のうち 50,000 データを訓練, 10,000 データをテストに分けている。さらに, 50,000 データのうち, 111 行目で 20% しか訓練に使わない設定としている。この状態で学習し, どのような学習結果になるか示せ。なお, 学習回数は 40 epoch を最小 epoch 数とし, 時間が許す範囲で学習すること。(Y 軸の値に注意すること) 学習誤差, 検証 誤差, 学習データに対する精度, 検証データに対する精度について, 考察すること。

(b)

次に dropout の効果を見るため, モデルの 1 段目の全結合層 (Dense) に Dropout し, Dropout の効果について考察せよ。元のファイルをコピーしておき, 1 段目の Dence(), Activation('relu') の後に Dropout(0.5) を入れる。また, 学習結果と学習誤差を違うファイル名で保存するようにしておくと, あとで比較するときに便利である。

(c)

さらに, データを 100% 使うようにした場合に, どうなるか試し, データ数と学習について考察すること。(可能ならば, Dropout の有無とも関連付けよ)

(a)

デフォルト : epoch:40にて実行

Epoch 40/40

```
9000/9000 [=====] - 1s 76us/step - loss: 0.0109 - acc: 0.9992 - val_loss: 0.7134 - val_acc: 0.8650
```



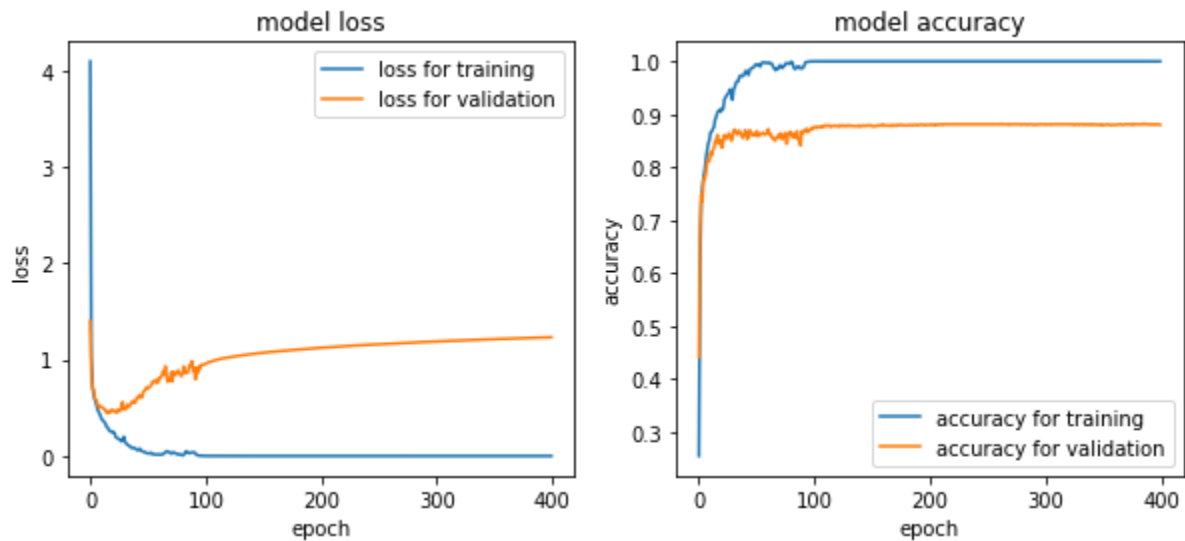
トレーニングデータに対しては十分な判別を行っているが、検証データに対しては

、十分とは言えない(正答率86.50%)結果のため、過学習を起こしていると考えられる。損失のグラフに至っては、これ以上エポック数を大きくすると増えるようにも考えられる。

epoch:400にて実行

Epoch 400/400

9000/9000 [=====] - 1s 75us/step - loss: 6.4489e-06 - acc: 1.0000 - val\_loss: 1.2317 - val\_acc: 0.8800

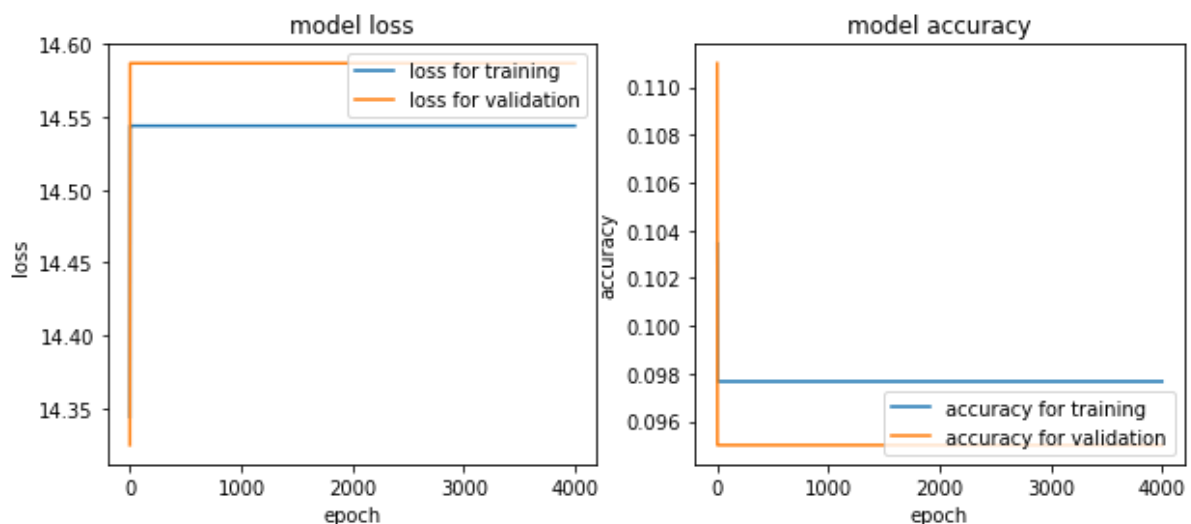


トレーニングデータに対しては完全に判別ができている。一方で、検証データに関しては0.88%の正答率であり、エポック数40よりも上がってはいるが、エポック数に対する正答率と、エポック数に対する損失共にグラフが一定ちに収束しているように見られる。

epoch:4000にて実行

Epoch 4000/4000

9000/9000 [=====] - 1s 74us/step - loss: 14.5439 - acc: 0.0977 - val\_loss: 14.5869 - val\_acc: 0.0950



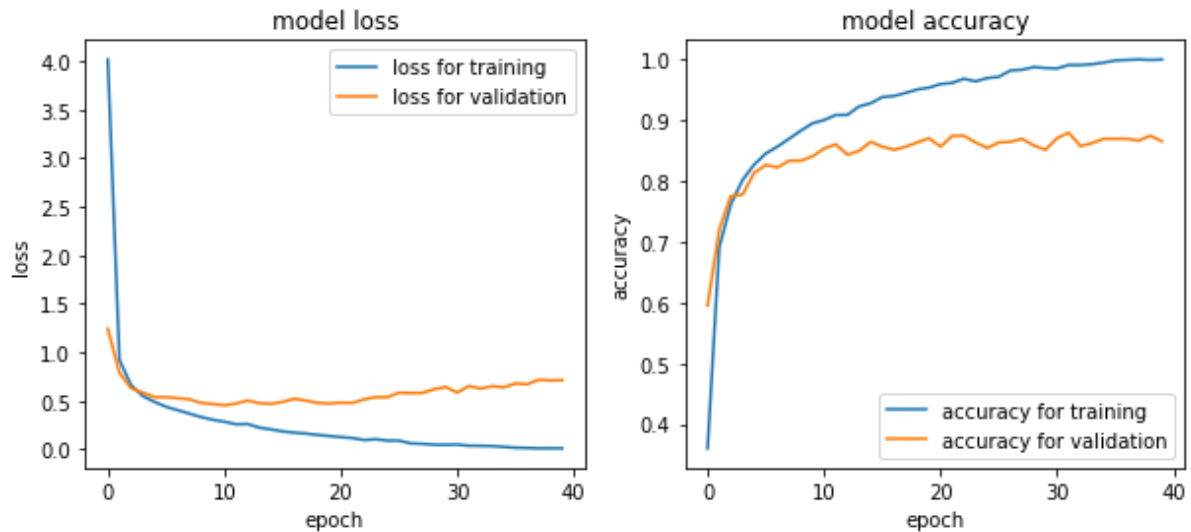
ほぼ損失も精度もほぼ完全に収束している。テストデータ・検証データ共に正答率が低い。誤差もこれより少ないエポック数に比べ、増大している。エポック数は400周辺に最大値をとると考えられる。

(b)

Dropoutなし

Epoch 40/40

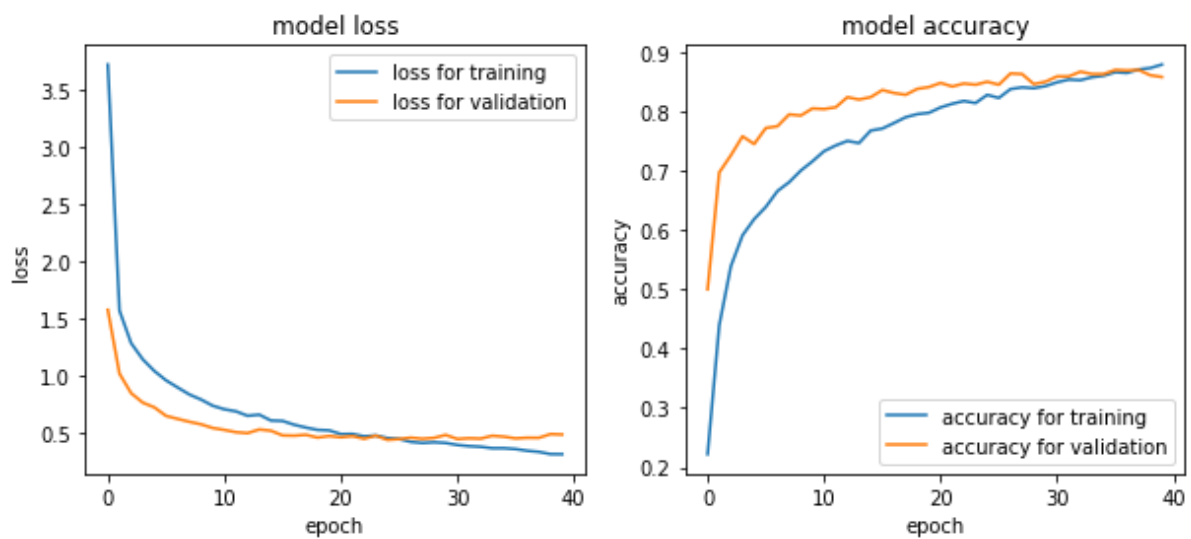
9000/9000 [=====] - 1s 76us/step - loss: 0.0109 - acc: 0.9992 - val\_loss: 0.7134 - val\_acc: 0.8650



Dropoutあり

Epoch 40/40

9000/9000 [=====] - 1s 75us/step - loss: 0.3203 - acc: 0.8790 - val\_loss: 0.4916 - val\_acc: 0.8580



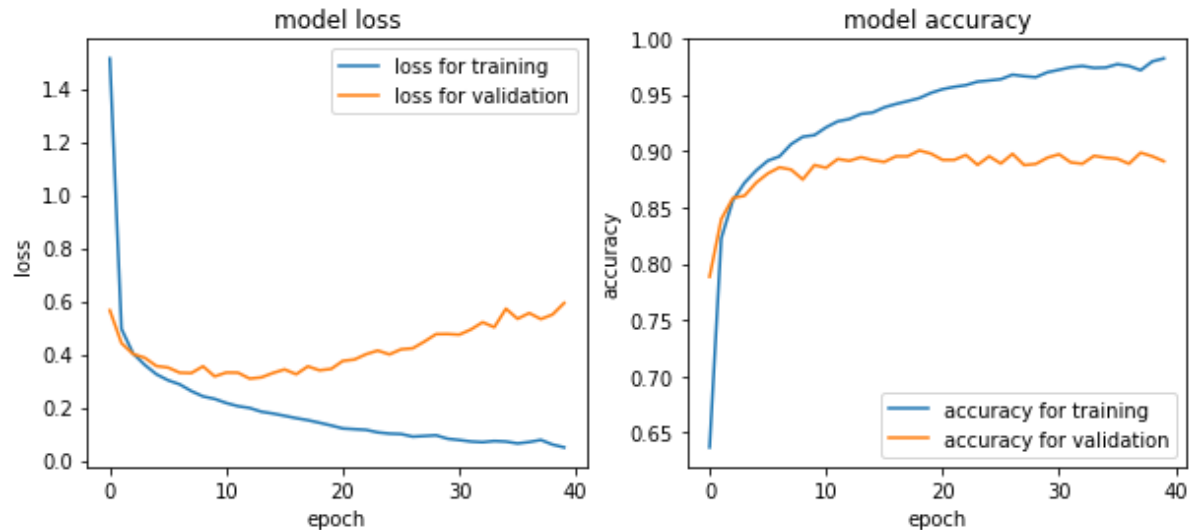
Dropoutがあると、トレーニングデータに対する誤差と正答率が落ちるが、検証データに対する損失が下がり、正答率は上がる。よって、テストデータと検証データの差が縮まって近い値を出力するようになっている。

(c)

Dropoutなし

Epoch 40/40

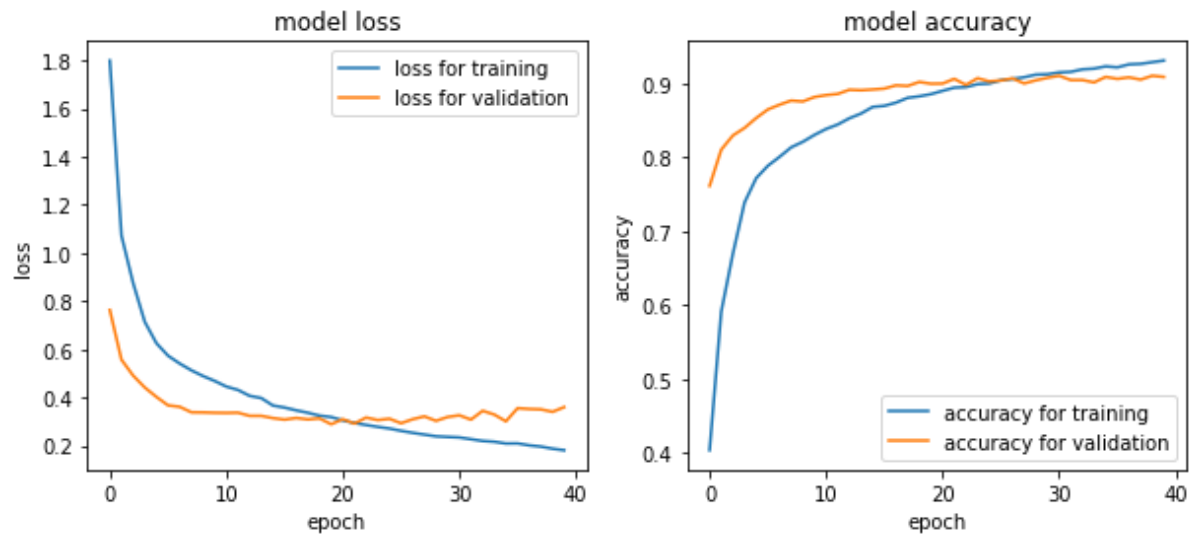
45000/45000 [=====] - 3s 73us/step -  
loss: 0.0499 - acc: 0.9820 - val\_loss: 0.5928 - val\_acc: 0.8908



Dropoutあり

Epoch 40/40

45000/45000 [=====] - 3s 74us/step - loss:  
0.1813 - acc: 0.9308 - val\_loss: 0.3602 - val\_acc: 0.9086



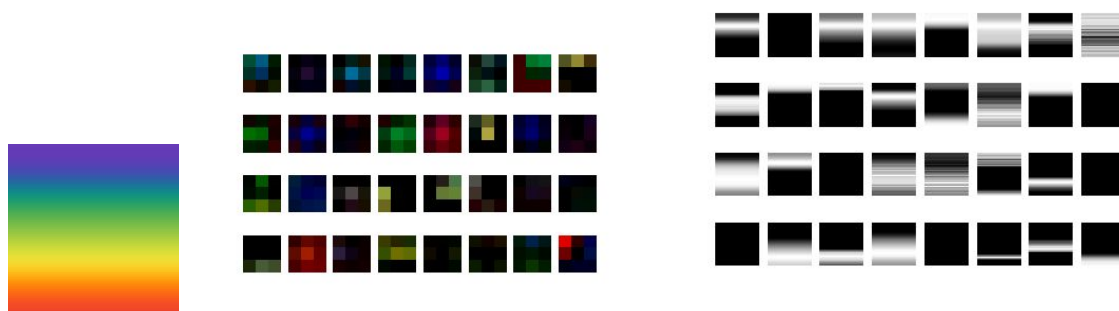
Dropoutなしでは、テストデータ・検証データに対して、(a)の40epochsに近い正答率、低い損失を出している。従って、このfashion mnist場合は50,000データ学習しなくともその2割の10,000データで十分特徴を掴むことが可能であると考えられる。

Dropoutを採用すると、先ほどのDropoutと同様、精度が上がり、これまでのモデルの中で最も検証データに対して高精度(正答率90%を超えている)なモデルとなっている。

2.

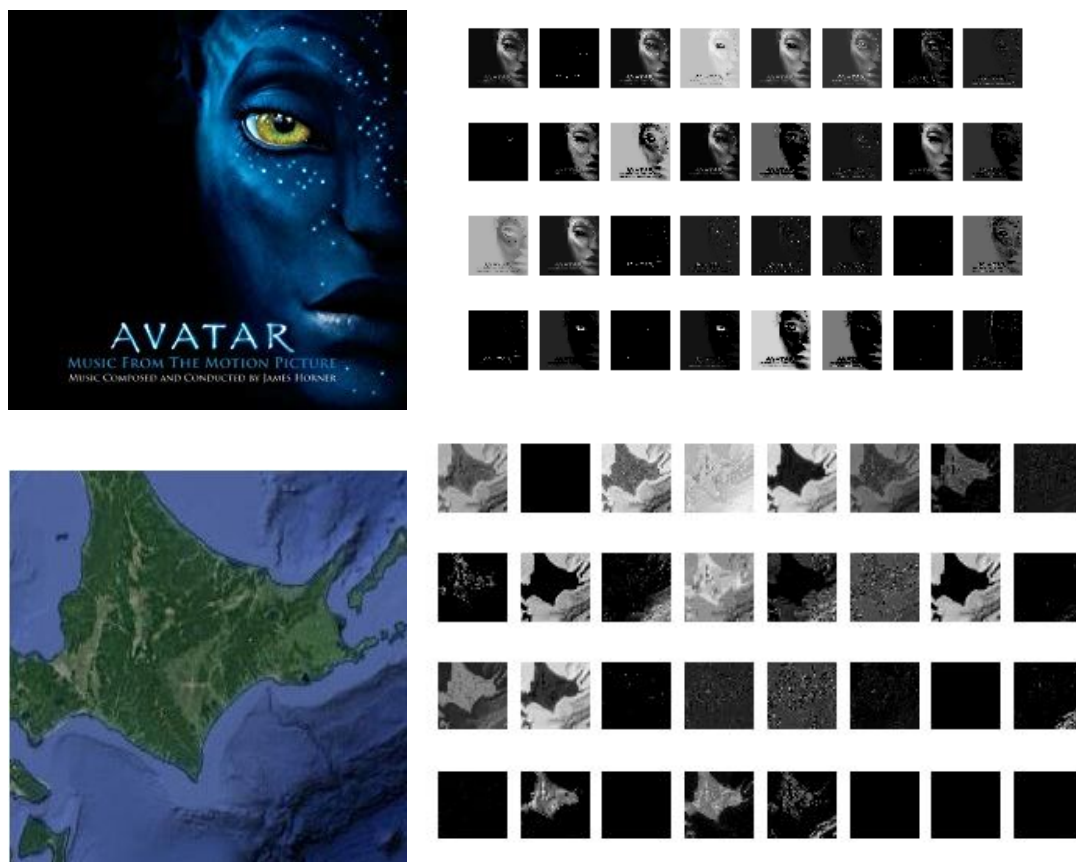
inceptionv3.py は inception v3 モデルの最初の畳み込み層 (conv2d\_1) の特徴フィルタと、その出力を活性化関数層 (activation\_1) を通したあとの特徴マップを可視化するスクリプトである。inception v3 で識別可能な適当な画像をいくつか用意し、それらの画像を入力した時の特徴マップと特徴フィルタから、第一層の畳み込みがどのような特徴を抽出しているかを考察せよ。

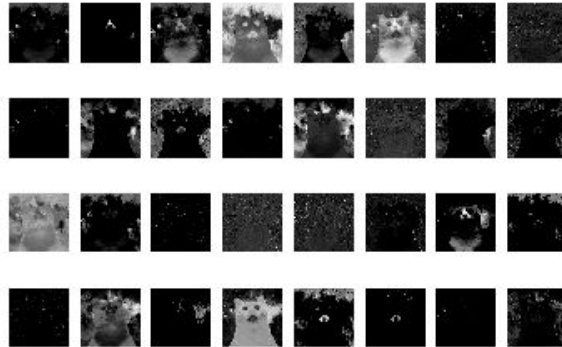
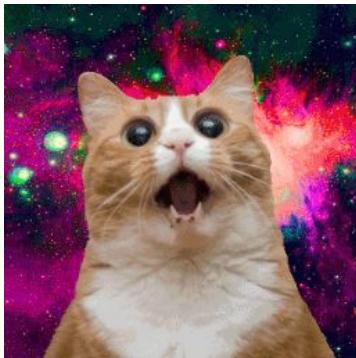
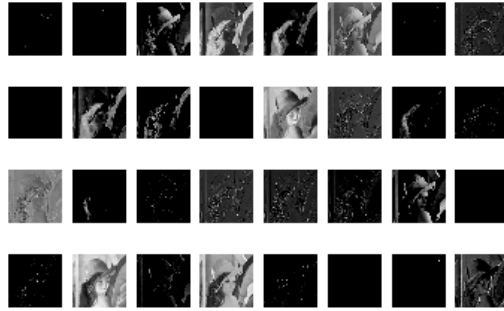
色への反応を調べるため、虹色画像を入力する



(左：入力画像 中：特徴フィルタの色 右：通過後の特徴マップ)  
各々のフィルタは、各々の色を特徴として抽出していると考えられる。

続いて、形状に対して何か反応があるかを調べる。





1行8列目や、3行6列目のといった、片側に要素の重みが偏ったフィルタを通すと、エンボス加工がかかったような画像が抽出されていることが見て取れる。1行4列目のようなフィルタでは、暗い印象の色をフィルタで重みを大きくしていると考えられ、ネガ写真のような画像が抽出される。