

Progress

Mizuno Yasuaki

September 18, 2022

1. プログラムの整理
2. 少ないデータでの学習
3. まとめ

画像生成と機械学習のプログラムが散乱していたので整理した。

- generate_img
 - common
 - prc
- train_img
 - common
 - prc

改良

- 千個の画像データごとに保存し直した
- `model` を生成し、コンパイルして返す関数の作成

最も単純な FNN モデル生成

Listing 1: Simple_FNN.py

```
1 def Simple_FNN(input_shape, output_shape):
2     model = keras.Sequential([
3         layers.Flatten(input_shape=input_shape),
4         layers.Dense(128, activation='relu'),
5         layers.Dense(output_shape)
6     ])
7     model.compile(
8         optimizer='adam',
9         loss=keras.losses.
10             SparseCategoricalCrossentropy(
11                 from_logits=True),
12         metrics=['accuracy']
13     )
14     return model
```

少ないデータでの学習¹

- 使用データ
 - 訓練データ：20000
 - テストデータ：6000
- 使用したモデル
 - 順伝播ニューラルネットワーク
 - 畳み込みニューラルネットワーク

¹多くのデータを使うと使用メモリをオーバーするため

- 使用モデル

Listing 2: generate_simple_fnn.py

```
1         model = keras.Sequential([
2             keras.layers.Flatten(input_shape=
3                 input_shape),
4             keras.layers.Dense(128, activation
5                 ='relu'),
6             keras.layers.Dropout(dropout_ratio),
7             keras.layers.Dense(output_shape)
8         ])
```

- ハイパーパラメータ
 - optimizer='adam'
 - dropout_ratio=0.0
 - epochs=30

FNN による学習学習結果

Test Accuracy: 0.28896206

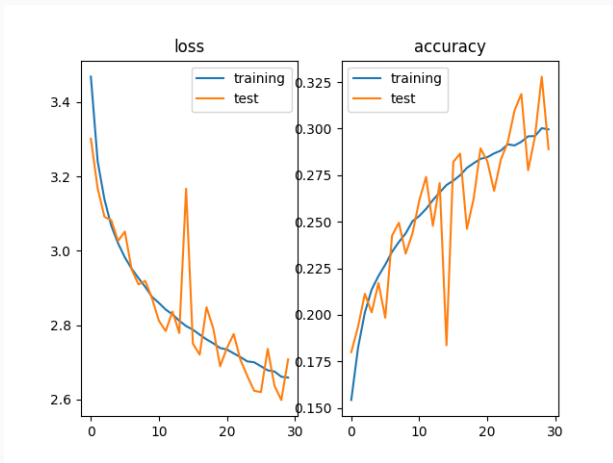


Figure 1: result_fnn.png

- 使用モデル

Listing 3: generate_simple_fnn.py

```
1         model = keras.models.Sequential([
2             keras.layers.Conv2D(32, (3, 3),
3                                   activation='relu', input_shape=
4                                   input_shape),
5             keras.layers.MaxPooling2D((2, 2)),
6             keras.layers.Flatten(),
7             keras.layers.Dense(64, activation='
8                                   relu'),
9             keras.layers.Dropout(dropout_ratio),
10            keras.layers.Dense(output_shape)
11        ])
```

- ハイパーパラメータ
 - optimizer='adam'
 - dropout_ratio=0.0
 - epochs=10

CNN による学習学習結果

Test Accuracy: 0.87836062

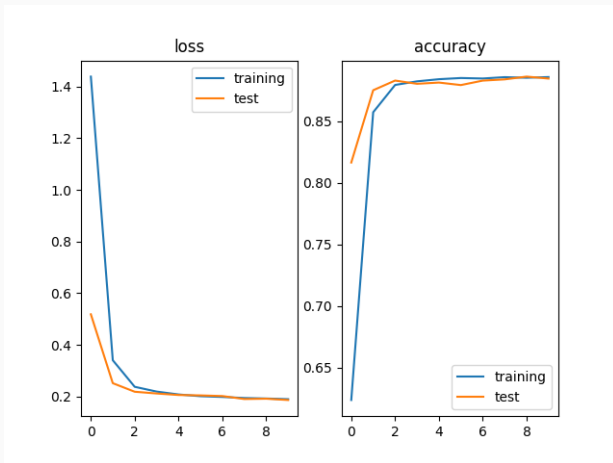


Figure 2: result_cnn.png

- CNN を使ったほうが精度
より複雑な CNN を使って学習を進める
- 過学習はあまり見られなかった
- より最適なハイパーパラメータを探す
- 多くのデータを使用して学習する方法を模索する