

マルチIoT・AIエンジニア科 第6期

AIプログラミング導入実習 機械学習課題

内容：猫と犬の判別

作成：2023/10/3

担当：一瀬 貴士

■ 概要

- CNN学習、転移学習の各々で猫と犬の画像を学習し、各自で用意した猫画像、犬画像から何れかを判別する

■ 提供されたファイル

- 学習データ：cat画像x499枚、dog画像x500枚
- CNN環境
 - 前処理 実行generate_data.py→出力imagefiles.npy
 - 学習 実行CNN.py→出力CNN.h5
 - 推論 実行predict.py
- 転移学習環境
 - 前処理 実行generate_data_244.py→出力imagefiles_244.npy
 - 学習 実行vgg16_transfer.py→出力vgg16_transfer.h5
 - 推論 実行predict0.py

1-1. CNNを使った前処理と学習

■ 前処理

授業にて、前処理中に学習データとテストデータを分けていることを学ぶ。次ステップの学習にて良い結果を得られない場合は再度前処理の実行にて好転する場合があった。

1-1. CNNを使った前処理と学習

■ 学習

① オリジナルCNN.pyでの学習

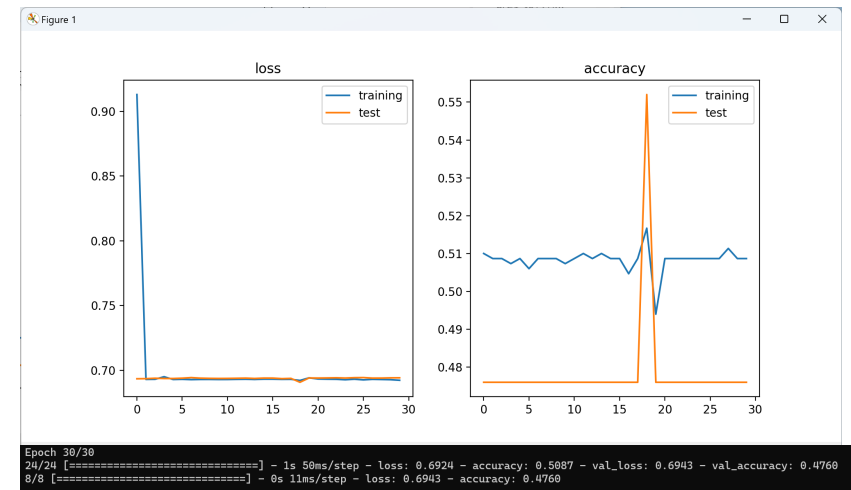
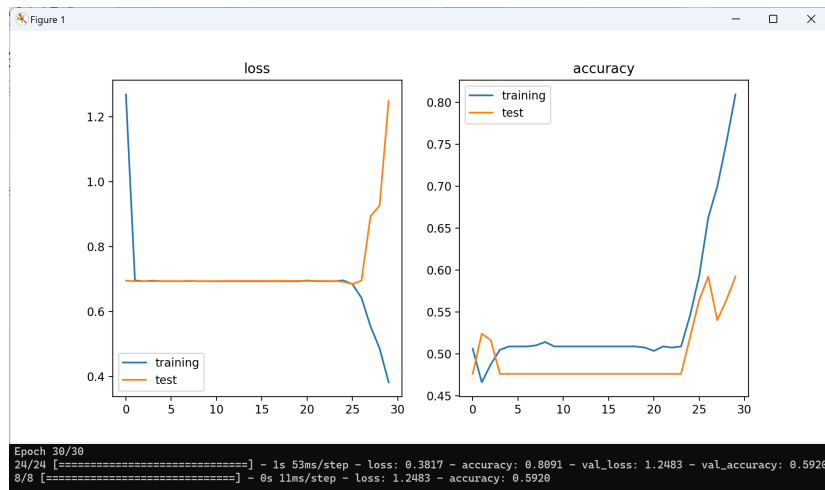
疑問

オリジナルCNN.pyにおいて同一前処理ファイルを使っても、**実行毎に学習結果が大きく異なる**ことが気になった。

以降LossとAccuracyをプロットして確認。

【ケース 1 : **Loss:1.2483/Accuracy:0.5920**】

【ケース 2 : **Loss:0.6943/Accuracy:0.4760**】

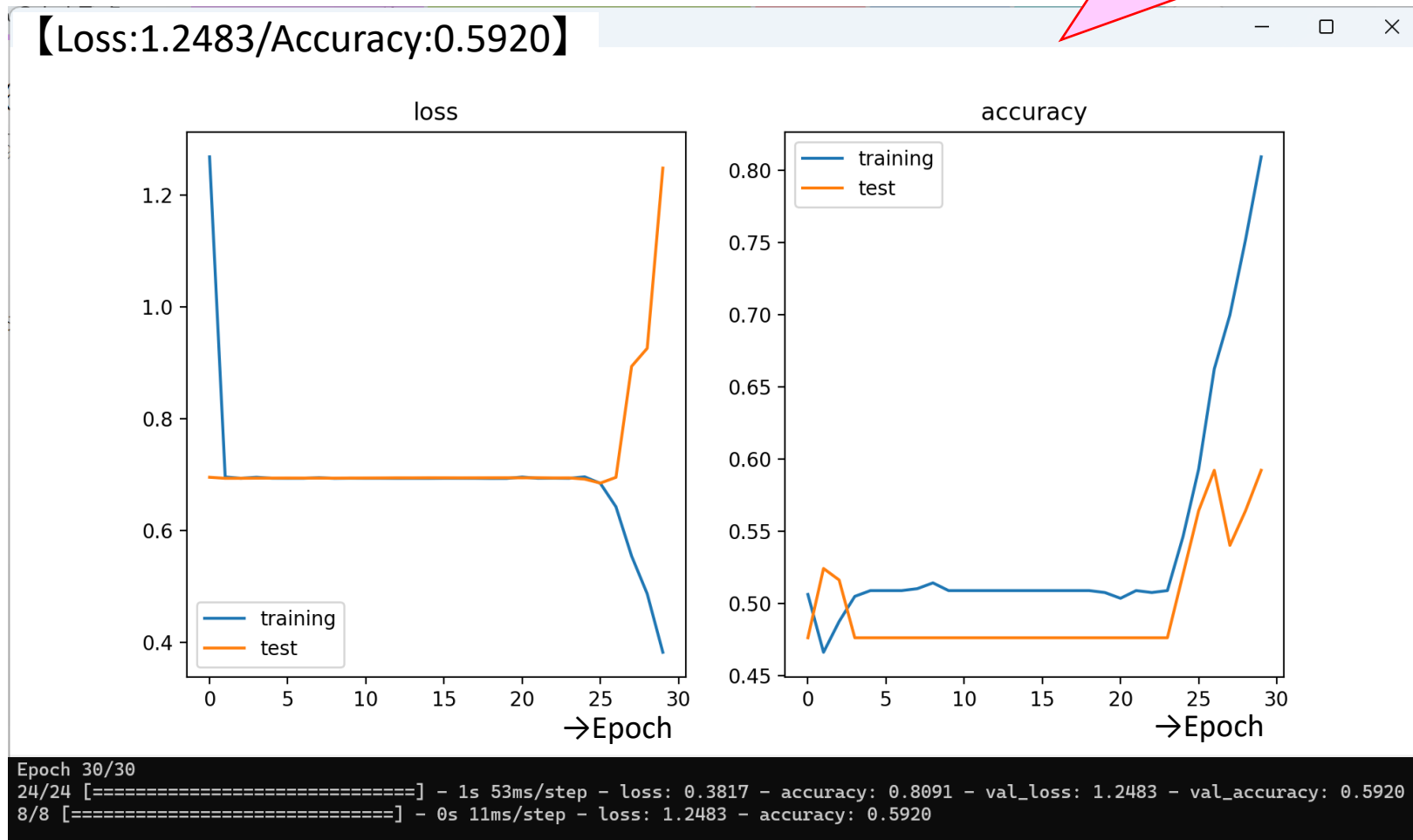


1-1. CNNを使った前処理と学習

■ 学習

① オリジナルCNN.pyでの学習 ～ケース1～

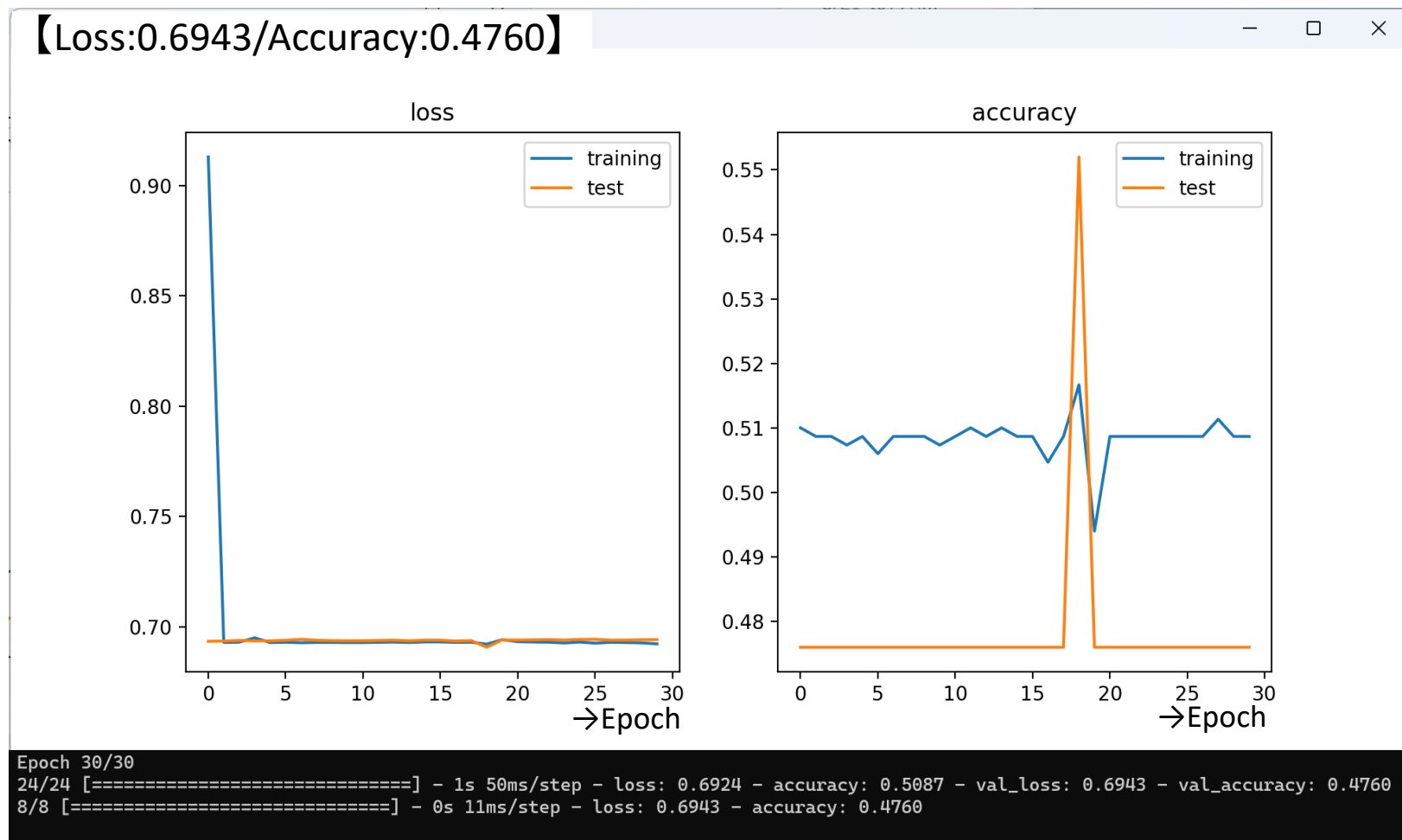
次頁ケース2と比較
学習経過（プロット
形状）が異なる



1-1. CNNを使った前処理と学習

■ 学習

① オリジナルCNN.pyでの学習 ～ケース2～



1-1. CNNを使った前処理と学習

■ 学習

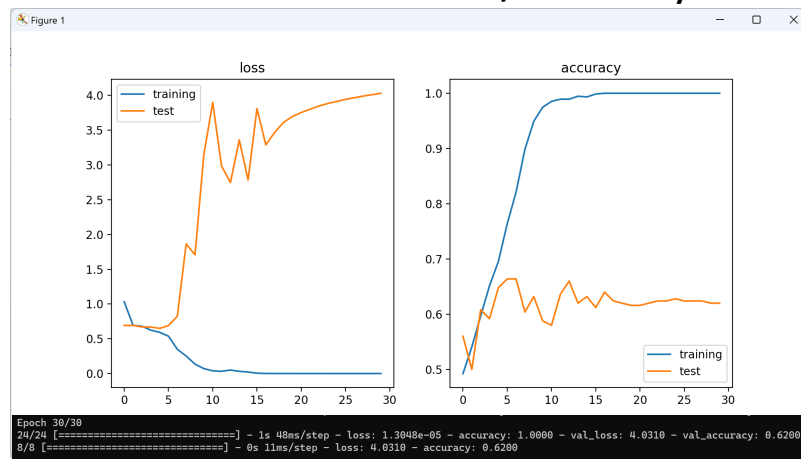
② オリジナルCNN.pyのDropOut無効化での学習

オリジナルCNN.pyの検証・考察

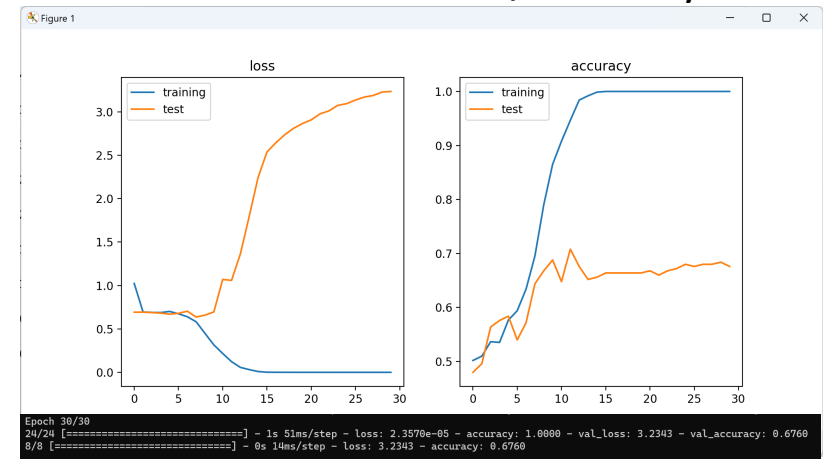
オリジナルCNN.pyのDropOutをすべて無効化して確認。結果、DropOut無効時は学習結果のバラツキは少なく、これはDropOut対象はランダムに選ばれるため実行毎に学習結果が異なると推測。

➡学習結果は10回程度の学習中から最良値を得るのが良い。

【ケース 1 : Loss:4.0310/Accuracy:0.6200】



【ケース 2 : Loss:3.2343/Accuracy:0.6760】



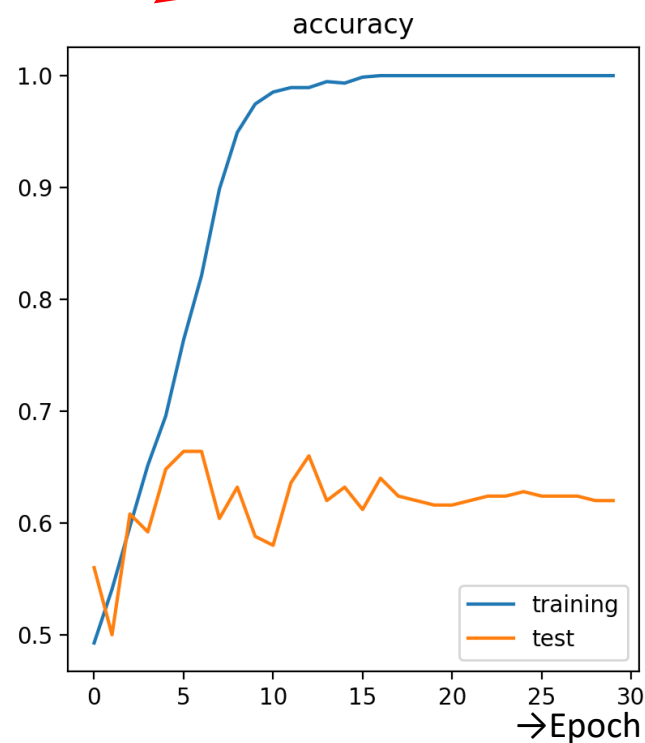
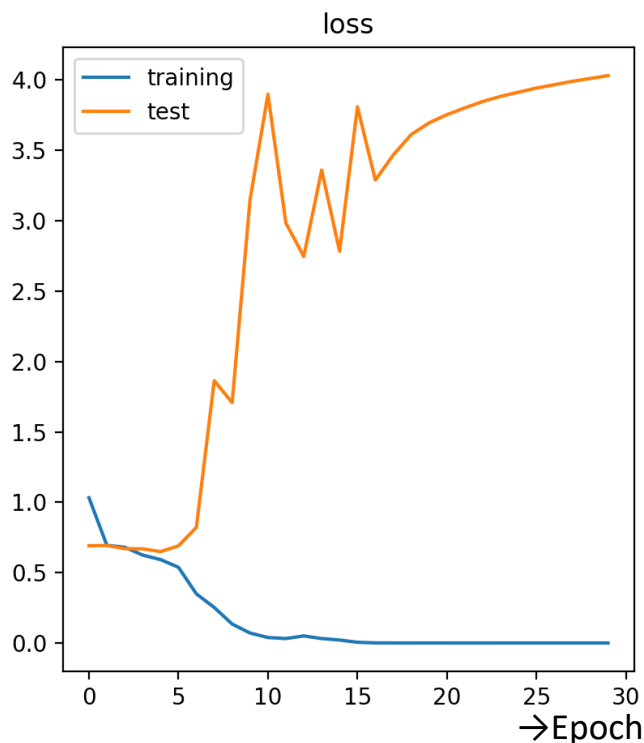
1-1. CNNを使った前処理と学習

■ 学習

② オリジナルCNN.pyのDropOut無効化での学習 ～ケース1～

次頁ケース2と比較
数値は悪いが双方似た学習経過を示す

【Loss:4.0310/Accuracy:0.6200】

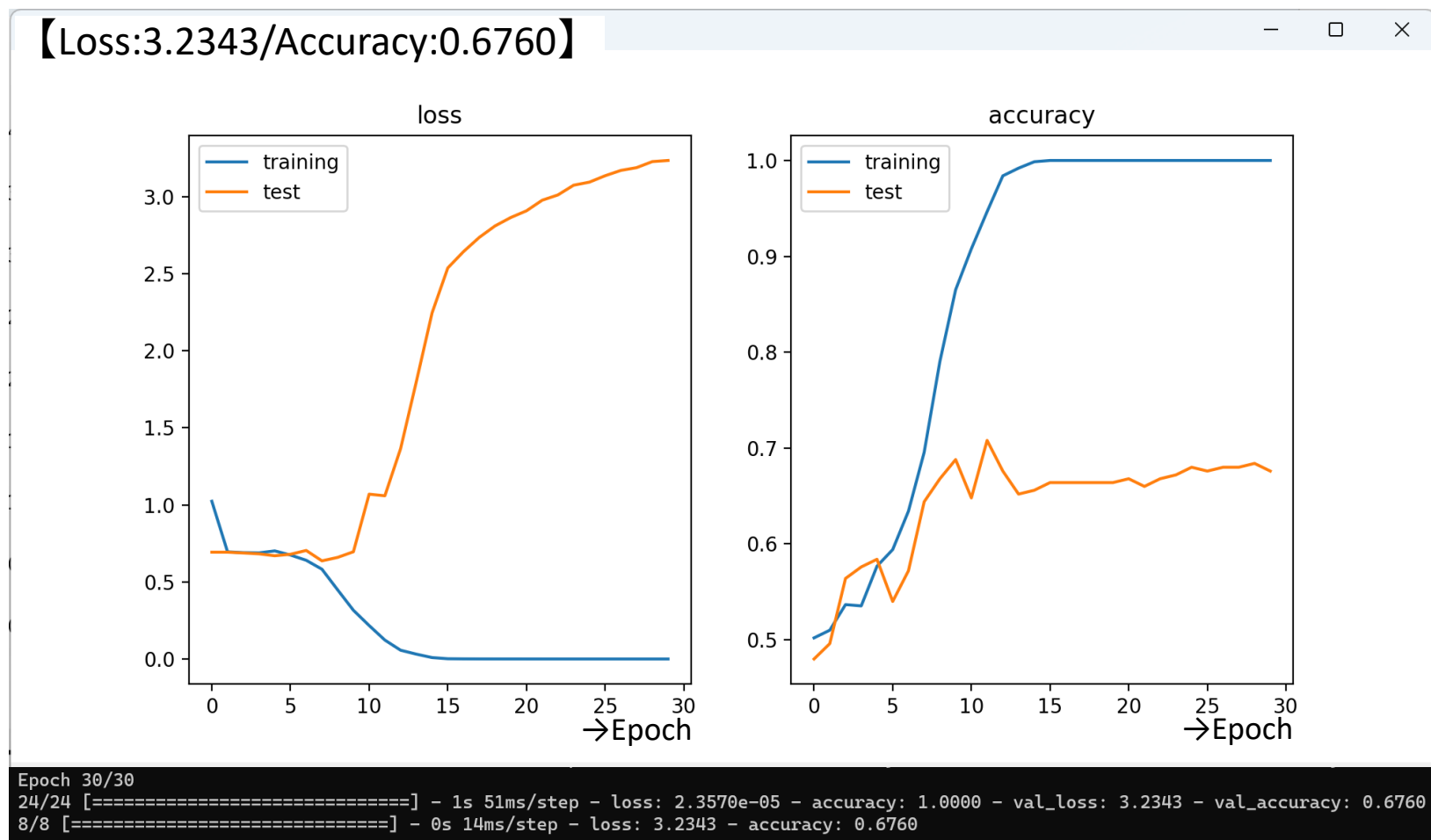


```
Epoch 30/30  
24/24 [=====] - 1s 48ms/step - loss: 1.3048e-05 - accuracy: 1.0000 - val_loss: 4.0310 - val_accuracy: 0.6200  
8/8 [=====] - 0s 11ms/step - loss: 4.0310 - accuracy: 0.6200
```


1-1. CNNを使った前処理と学習

■ 学習

② オリジナルCNN.pyの**DropOut無効化**での学習 ～ケース2～



1-1. CNNを使った前処理と学習

■ 学習

③ 学習結果の改善

授業での改善試行から得た自分なりの改善手法を試す。

改善イメージはEpoch初期回数の段階で**Test結果**が**Training結果**を超えるような学習状態は避ける。結果、1回の学習結果は期待できないが**Epoch数を多くすることで好転**する。

改善実施例

- ・ 折り畳み時のフィルター数を小さくする。
- ・ Pooling Sizeを大きく取る。
- ・ Dropoutを大きく取る。
- ・ Epoch数を多く取る。



cnn_py.txt

1-1. CNNを使った前処理と学習

■ 学習

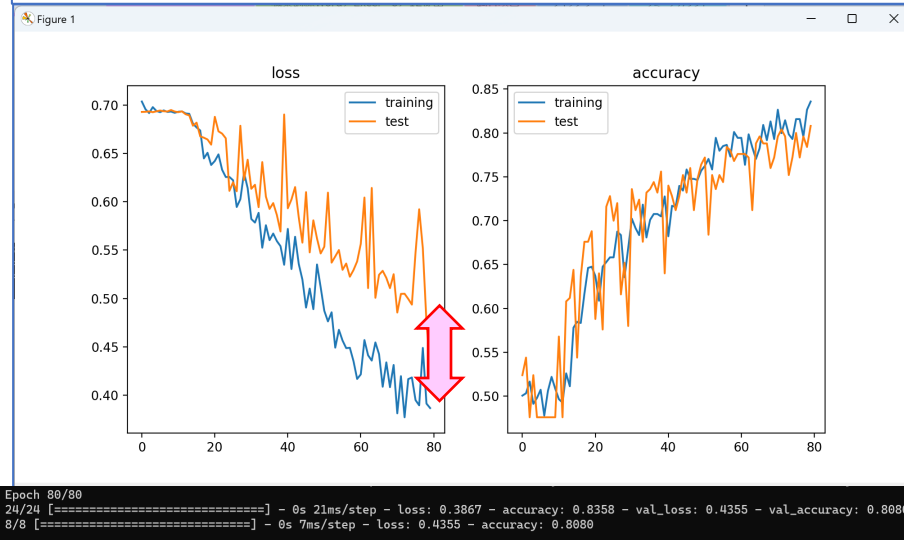
③ 学習結果の改善

試行の結果、下記 2 点の学習結果にて推論をおこなう。

学習結果 1 (CNN1)

Loss : 0.4355(**Champ**)/Accuracy : 0.8080 (**Champ**)

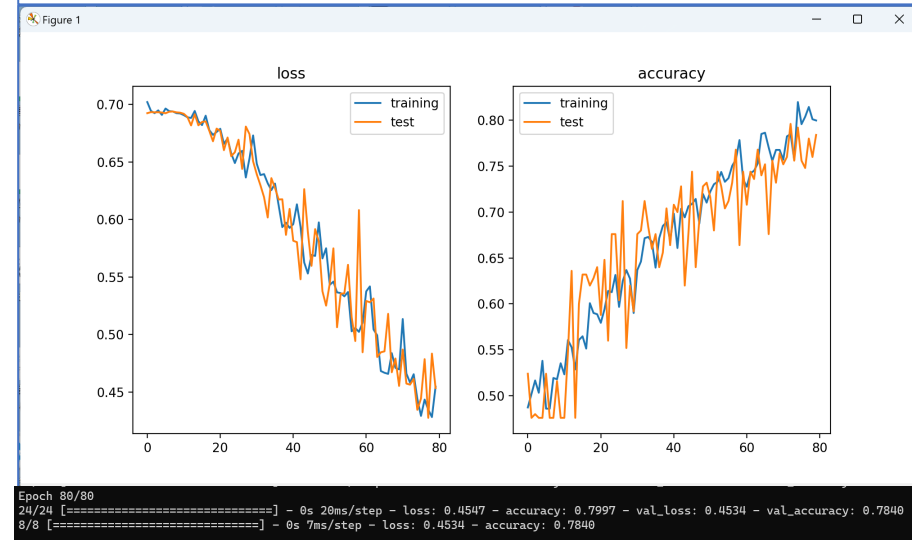
特徴：Lossグラフの**TrainingとTestが徐々に離れる**。前頁で示した対策の結果、細かく乱れながら（ギザギザ）学習を進めている。



学習結果 2 (CNN2)

Loss : 0.4534/Accuracy : 0.7840

特徴：Lossグラフの**TrainingとTestは大きく離れていない**。前頁で示した対策の結果、細かく乱れながら（ギザギザ）学習を進めている。



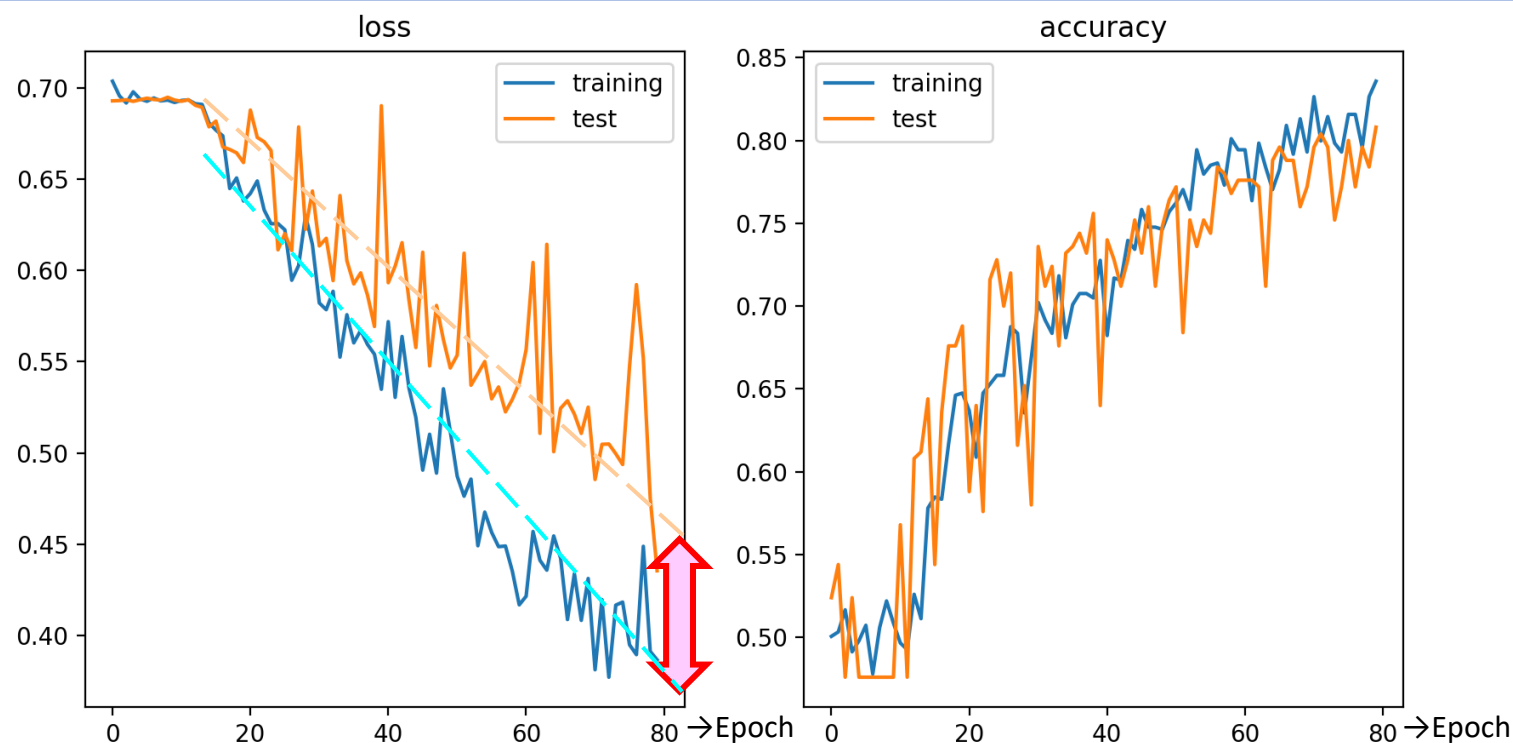
1-1. CNNを使った前処理と学習

■ 学習

③ 学習結果の改善 ～学習結果 1 (CNN 1)～

Loss : 0.4355(チャンピオン)/Accuracy : 0.8080 (チャンピオン)

特徴：Lossグラフの**TrainingとTestが徐々に離れる**。前頁で示した対策の結果、細かく乱れながら（ギザギザ）学習を進めている。



Epoch 80/80
24/24 [=====] - 0s 21ms/step - loss: 0.3867 - accuracy: 0.8358 - val_loss: 0.4355 - val_accuracy: 0.8080
8/8 [=====] - 0s 7ms/step - loss: 0.4355 - accuracy: 0.8080

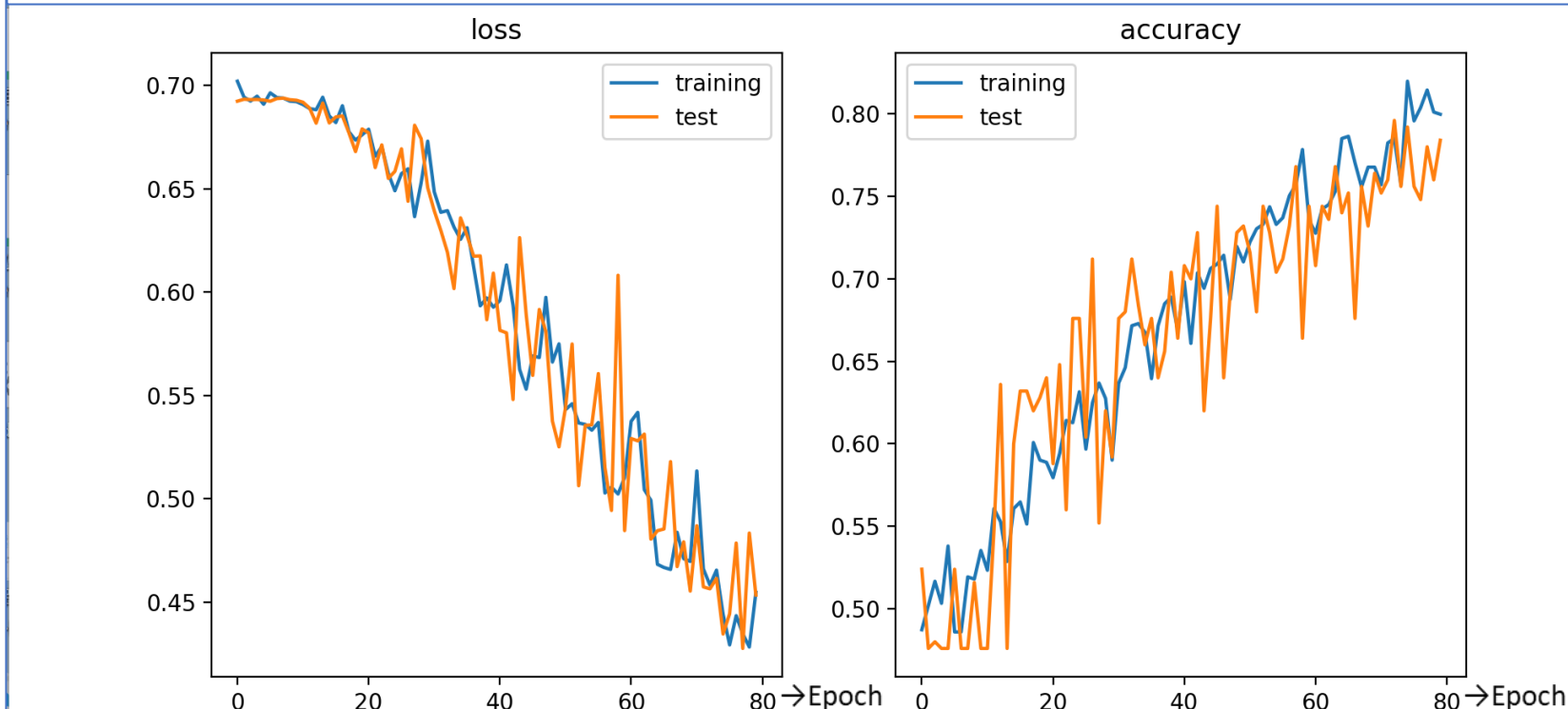
1-1. CNNを使った前処理と学習

■ 学習

③ 学習結果の改善 ～学習結果 2 (CNN2)～

Loss : 0.4534/Accuracy : 0.7840

特徴：Lossグラフの**結果 1** ほど**Training**と**Test**は離れていない。前頁で示した対策の結果、細かく乱れながら（ギザギザ）学習を進めている。



Epoch 80/80

24/24 [=====] - 0s 20ms/step - loss: 0.4547 - accuracy: 0.7997 - val_loss: 0.4534 - val_accuracy: 0.7840



8/8 [=====] - 0s 7ms/step - loss: 0.4534 - accuracy: 0.7840

1-2. CNNを使った推論

■ 推論

3点の推論に判定ミスがあり考察しました。

◆Dog5.jpg



推論データ		学習データCNN1	学習データCNN2
	Loss	0.4355	0.4534
	Accuracy	0.808	0.7840
dog5.jpg	推論結果	cat 57	dog 84
	学習データCNN1にて cat57 と判断したのは治療用の首巻き、又はその色である緑が原因だと思われる。		
dog5w.jpg	推論結果	dog 98	dog 82
	首巻き部を白色へ変換したところ、学習データCNN1にて推論が dog98 と良化した。これにて学習データにおける背景の影響は少ないことが分かる。また、緑色は 草木の色 なので注意が必要。		

※転移学習を使った推論では上記両データ共にdog 100

1-2. CNNを使った推論

■ 推論

◆Cat6,7.jpg

推論データ	学習データCNN1		学習データCNN2	
	Loss	0.4355	Accuracy	0.4534
cat6.jpg	推論結果	dog 56	推論結果	cat 56
	学習データCNN1における判別ミスは猫の学習データに躍動感のあるものが略無いことが原因と思われる。一方、学習データCNN2を猫と判断している原因は不明だが、P12,P13で示した学習進行とTrainingとTestの開きが少ないことが原因の一つと予想。			
cat7.jpg	推論結果	dog 72	推論結果	cat 71
	同上			

2-1. 転移学習を使った前処理と学習

■ 前処理

CNNでの前処理同様に、次ステップの学習にて良い結果を得られない場合は再度前処理の実行にて好転する場合があった。

2-1. 転移学習を使った前処理と学習

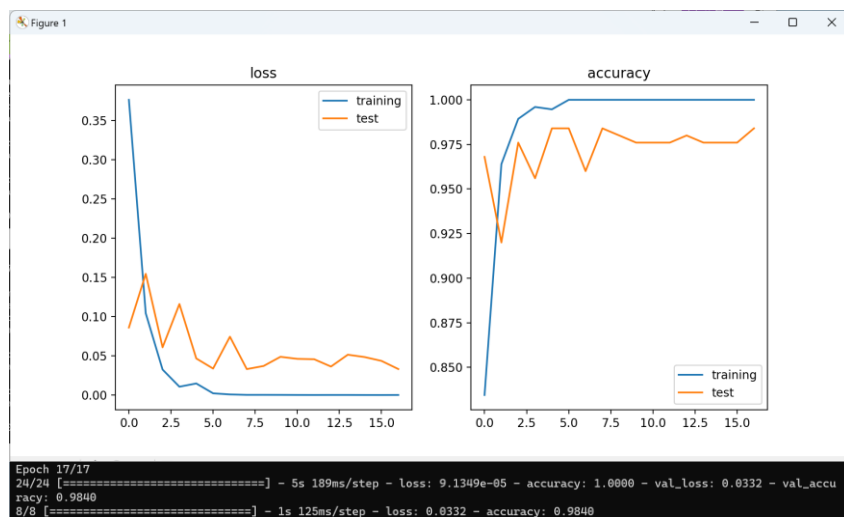
■ 学習

① オリジナルvgg16_transfer.pyでの学習

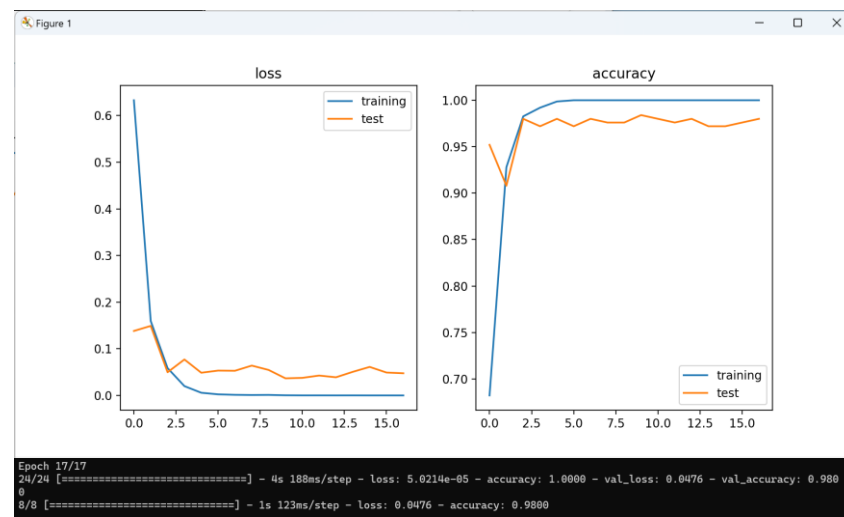
考察

Epoch数17と少ないが、Loss、Accuracy共に、これ以上の改善は難しい。次ページに改善試行を示すが、有効な改善は得られなかった。

【Loss:0.0332/Accuracy:0.9840】



【Loss:0.0476/Accuracy:0.9800】



2-1. 転移学習を使った前処理と学習

■ 学習

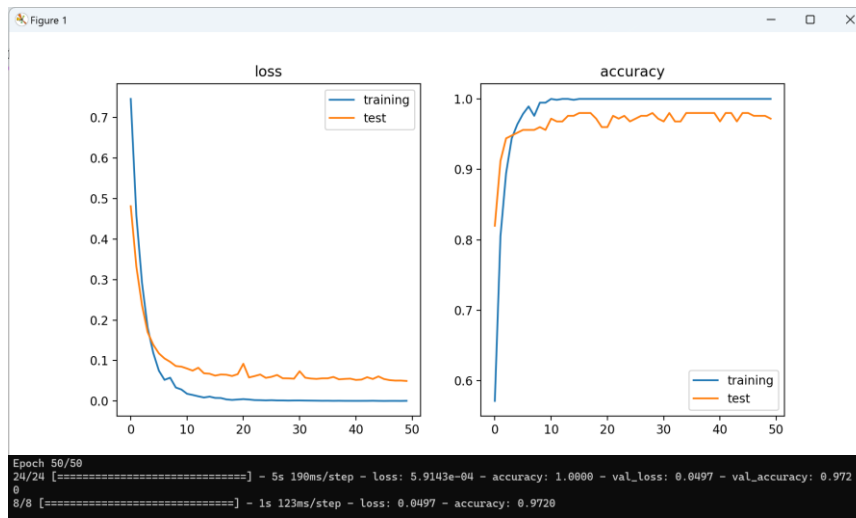
② vgg16_transfer.pyの改善試行

最適化アルゴリズムAdamのパラメータである学習率(lr)を小さくして細かな学習を意識。加えてEpoch数を大きくする。

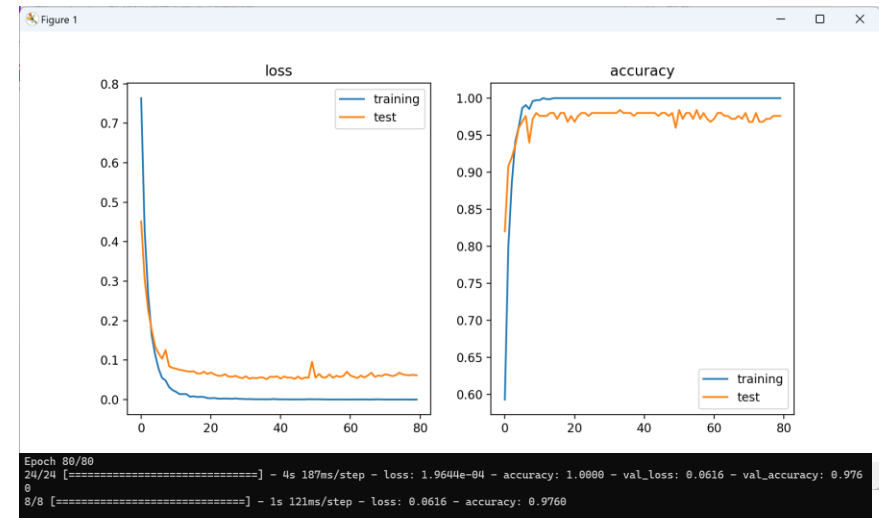
考察

グラフ波形は振れは少ないが、数値的にはオリジナルを超えない。

【Loss:0.0497/Accuracy:0.9720】



【Loss:0.0616/Accuracy:0.9760】





2-2. 転移学習を使った推論

■ 推論

2 点の推論に判定ミスがあり考察しました。

◆cat5,6.jpg(cat7)

推論データ		champion1	champion2
	Loss	0.0332	0.0476
	Accuracy	0.9840	0.9800
cat5.jpg	推論結果	dog 54	dog 98
	判別ミスは猫の学習データに躍動感のあるものが略無いことが原因と思われる。転移学習であっても学習データの網羅率が大切と実感。		
cat6.jpg	推論結果	dog 89	dog 89
	同上		

参考
cat7.jpg



両学習データとも推論判別結果は**cat 99**。
上記 2 点との差分は体模様が猫らしいからか。

3. まとめ

(1) 学習データ（画像）の背景処理

CNN学習によるdog5(w).jpgの推論結果から、学習は**画像背景に影響される**ことを理解した。推論スコア向上には学習データの背景処理を試すこと。

(2) 学習データの網羅

今回、躍動感ある猫の判別にミスがあった。（CNN、転移ともに）これは学習データに躍動感ある画像が少なかったことが原因の一つと思われる。用意する**学習データの網羅率**に注意すること。

(3) 学習経過を示すプロット

学習データの考察を行う上で本プロットは参考になる。

(4) 推論結果ファイル



AI課題_推論結果.xlsx