

[멋사 13기 이지훈] [AI_이지훈]

📅 날짜	@2025년 5월 29일
🌟 상태	진행 중

[멋사 13기] [AI_이지훈]

정확도를 높이기 위한 방법(batch_size, EarlyStopping, epochs)

batch_size를 기존 5회에서 → 10회로 변경

▼ 4. 이미지 데이터 전처리

- rescale: 픽셀값 정규화 (0~1)
- target size: (64,64)로 이미지 크기 통일(리사이징)
- batch_size: 한 번에 몇 번씩 학습할건지 정의
- class mode: (=binary) 이진 분류

```
[ ] train_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    "/content/data",
    target_size=(64, 64),
    batch_size=10,
    class_mode='binary'
)
```

```
# 학습용 제너레이터
train_generator = train_datagen.flow_from_directory(
    '/content/data/',
    target_size=(64, 64),
    batch_size=10,
    class_mode='binary',
    subset='training'
)

# 검증용 제너레이터
validation_generator = train_datagen.flow_from_directory(
    '/content/data/',
    target_size=(64, 64),
    batch_size=10,
    class_mode='binary',
    subset='validation'
)
```

Early Stopping 사용

```
# early stopping 정의
early_stopping = tf.keras.callbacks.EarlyStopping(
    monitor='val_loss',
    patience=3,
    restore_best_weights=True
)
```

EarlyStopping 설명

- **monitor='val_loss'** : 검증 손실을 모니터링하며 검증 손실이 더 이상 감소하지 않으면 학습 종료
- **patience=3** : 검증 손실이 3번의 에포크(반복) 동안 개선되지 않으면 학습을 종료
- **restore_best_weights=True** : 학습 종료 후 가장 낮은 검증 손실을 기록한 가중치를 복원하여 과적합된 모델 대신 최적의 모델을 사용
- 에포크 수를 15회로 늘리고 배치 크기도 10회로 늘려냈기에 과적합 우려가 있어 EarlyStopping 으로 과적합 방지를 기대해 볼 수 있음

epochs 횟수 늘리기

▽ 6. 모델 학습시키기

- epochs: 전체 학습 반복 수 정의

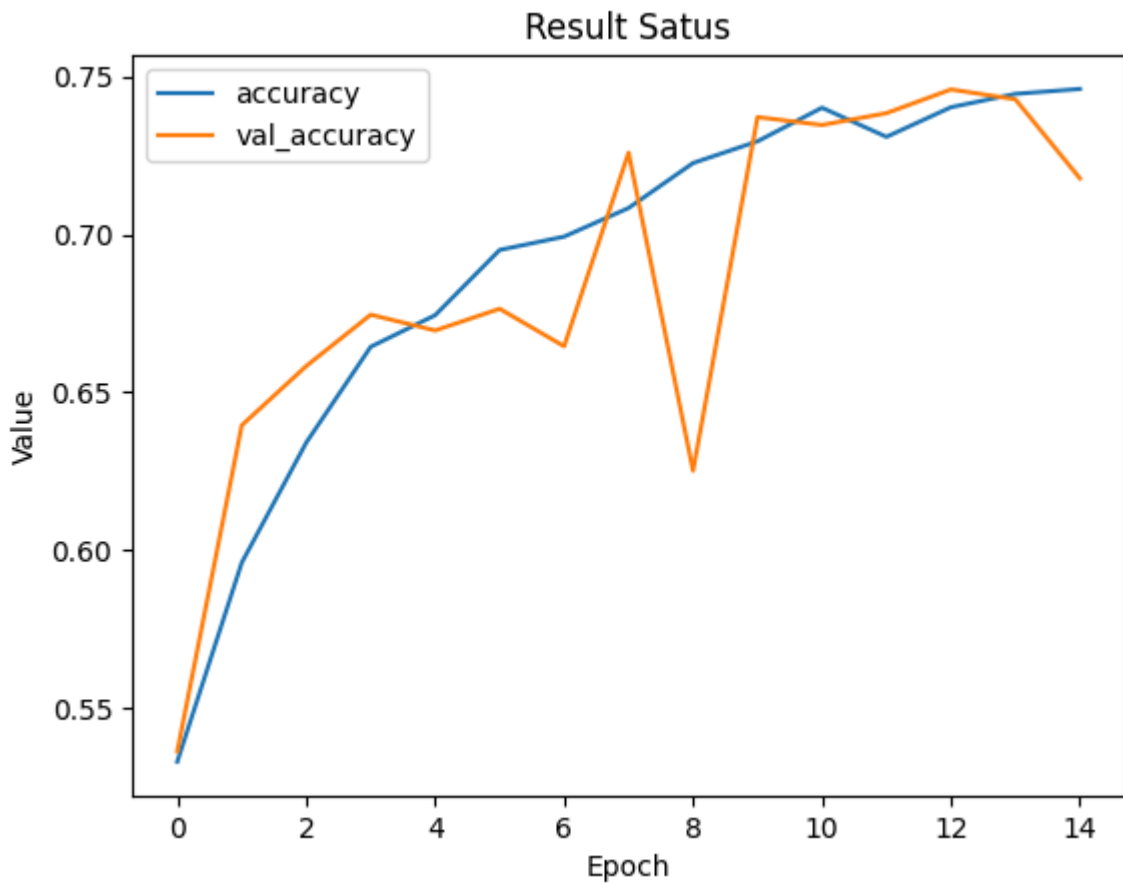
```
[ ] history = model.fit(
    train_generator,
    epochs=15,
    validation_data=validation_generator
)
```

기존 10회였던 epochs 횟수를 15회로 늘려 진행

```
history = model.fit(
    train_generator,
    epochs=15,
    validation_data=validation_generator
)

/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your 'PyDataset' class should call 'super().__init__(**kwargs)' in its construct
self._warn_if_super_not_called()

Epoch 1/15: 63s 94ms/step - accuracy: 0.5207 - loss: 0.6981 - val_accuracy: 0.5365 - val_loss: 0.6871
Epoch 2/15: 57s 88ms/step - accuracy: 0.5723 - loss: 0.6744 - val_accuracy: 0.6396 - val_loss: 0.6510
Epoch 3/15: 57s 89ms/step - accuracy: 0.6376 - loss: 0.6342 - val_accuracy: 0.6583 - val_loss: 0.6135
Epoch 4/15: 59s 92ms/step - accuracy: 0.6597 - loss: 0.6159 - val_accuracy: 0.6746 - val_loss: 0.6025
Epoch 5/15: 61s 95ms/step - accuracy: 0.6820 - loss: 0.5824 - val_accuracy: 0.6696 - val_loss: 0.5990
Epoch 6/15: 57s 89ms/step - accuracy: 0.6983 - loss: 0.5761 - val_accuracy: 0.6765 - val_loss: 0.5923
Epoch 7/15: 56s 88ms/step - accuracy: 0.6931 - loss: 0.5676 - val_accuracy: 0.6646 - val_loss: 0.6326
Epoch 8/15: 59s 93ms/step - accuracy: 0.7061 - loss: 0.5636 - val_accuracy: 0.7258 - val_loss: 0.5490
Epoch 9/15: 56s 88ms/step - accuracy: 0.7268 - loss: 0.5439 - val_accuracy: 0.6252 - val_loss: 0.6744
Epoch 10/15: 60s 94ms/step - accuracy: 0.7251 - loss: 0.5469 - val_accuracy: 0.7370 - val_loss: 0.5327
Epoch 11/15: 55s 86ms/step - accuracy: 0.7331 - loss: 0.5370 - val_accuracy: 0.7345 - val_loss: 0.5217
Epoch 12/15: 61s 95ms/step - accuracy: 0.7376 - loss: 0.5242 - val_accuracy: 0.7383 - val_loss: 0.5165
Epoch 13/15: 55s 86ms/step - accuracy: 0.7433 - loss: 0.5199 - val_accuracy: 0.7458 - val_loss: 0.5112
Epoch 14/15: 61s 94ms/step - accuracy: 0.7436 - loss: 0.5140 - val_accuracy: 0.7427 - val_loss: 0.5216
Epoch 15/15: 55s 86ms/step - accuracy: 0.7466 - loss: 0.5117 - val_accuracy: 0.7177 - val_loss: 0.5727
```



1회 반복 시 accuracy: 0.5207 / val_loss: 0.6871

15회 반복 시 accuracy: 0.7466 / val_loss: 0.5727

정확도는 22%가량 상승, 검증 손실률은 11% 하락해 심각한
과적합 없이 학습이 완료된 것으로 보인다

만약 반복 8회쯤에서 검증 손실률이 다시 복구되지 않았다면,
EarlyStopping으로 학습을 종료 후 검증 손실률이 가장 낮은
모델로 다시 학습할 수 있다

느낀점

요즘 AI를 안 쓰는 분야가 거의 없는 만큼 AI가 일상생활에 깊숙히 들어와
있다고 생각하는데, 마침 AI의 구조를 체험해보고 수정해볼 수 있는 기회가
생긴 것 같아 의미 있던 수업과 과제였던 것 같다.