

인공지능 과제 3: CNN 기반 CIFAR-10 분류모델 개발 및 실험 보고서

```
history1 = model1.fit(x_train_partial, y_train_partial, epochs=10, batch_size=16, validation_data=(x_val, y_val))
```

```
Epoch 1/10
2813/2813 [=====] - 35s 12ms/step - loss: 1.9732 - accuracy: 0.2520 - val_loss: 1.8460 - val_accuracy: 0.2996
Epoch 2/10
2813/2813 [=====] - 33s 12ms/step - loss: 1.8443 - accuracy: 0.3083 - val_loss: 1.7934 - val_accuracy: 0.3306
Epoch 3/10
2813/2813 [=====] - 32s 11ms/step - loss: 1.7976 - accuracy: 0.3260 - val_loss: 1.7483 - val_accuracy: 0.3440
Epoch 4/10
2813/2813 [=====] - 34s 12ms/step - loss: 1.7423 - accuracy: 0.3452 - val_loss: 1.6895 - val_accuracy: 0.3670
Epoch 5/10
2813/2813 [=====] - 33s 12ms/step - loss: 1.7190 - accuracy: 0.3523 - val_loss: 1.6993 - val_accuracy: 0.3616
Epoch 6/10
2813/2813 [=====] - 32s 12ms/step - loss: 1.7048 - accuracy: 0.3577 - val_loss: 1.6690 - val_accuracy: 0.3720
Epoch 7/10
2813/2813 [=====] - 34s 12ms/step - loss: 1.6925 - accuracy: 0.3621 - val_loss: 1.6532 - val_accuracy: 0.3744
Epoch 8/10
2813/2813 [=====] - 33s 12ms/step - loss: 1.6855 - accuracy: 0.3680 - val_loss: 1.6444 - val_accuracy: 0.3792
Epoch 9/10
2813/2813 [=====] - 33s 12ms/step - loss: 1.6779 - accuracy: 0.3704 - val_loss: 1.6599 - val_accuracy: 0.3762
Epoch 10/10
2813/2813 [=====] - 32s 11ms/step - loss: 1.6716 - accuracy: 0.3712 - val_loss: 1.6581 - val_accuracy: 0.3776
```

모델1의 학습

에폭 10를 기준으로 모델을 돌렸습니다. 모델1의 에폭1는 Loss 1.9732, accyuracy 0.2520

검증손실은 1.8460, 검증 정확도는 0.2996으로 나오고 에폭 10은 Loss 1.6716, accyuracy 0.3712

검증손실은 1.6591, 검증 정확도는 0.3776입니다 초기 훈련에서 훈련 손실 및 검증 손실이 높지만 에폭을 진행될수록 손실이 감소하고 정확도가 올라가는 것을 확인할 수 있었습니다. 검증 정확도가 0.29에서 0.37로 상승하는 것을 확인할 수 있습니다.

```
# 모델2 학습
model2.compile(optimizer='adam',
               loss='categorical_crossentropy',
               metrics=['accuracy'])

history2 = model2.fit(x_train_partial, y_train_partial, epochs=10, batch_size=16, validation_data=(x_val, y_val))
```

```
Epoch 1/10
2813/2813 [=====] - 56s 20ms/step - loss: 1.6122 - accuracy: 0.4048 - val_loss: 1.3981 - val_accuracy: 0.4854
Epoch 2/10
2813/2813 [=====] - 55s 19ms/step - loss: 1.3379 - accuracy: 0.5192 - val_loss: 1.3594 - val_accuracy: 0.5210
Epoch 3/10
2813/2813 [=====] - 53s 19ms/step - loss: 1.2283 - accuracy: 0.5596 - val_loss: 1.1860 - val_accuracy: 0.5720
Epoch 4/10
2813/2813 [=====] - 55s 20ms/step - loss: 1.1556 - accuracy: 0.5867 - val_loss: 1.1388 - val_accuracy: 0.5938
Epoch 5/10
2813/2813 [=====] - 55s 20ms/step - loss: 1.0953 - accuracy: 0.6091 - val_loss: 1.1607 - val_accuracy: 0.5836
Epoch 6/10
2813/2813 [=====] - 54s 19ms/step - loss: 1.0524 - accuracy: 0.6234 - val_loss: 1.0805 - val_accuracy: 0.6200
Epoch 7/10
2813/2813 [=====] - 54s 19ms/step - loss: 1.0075 - accuracy: 0.6387 - val_loss: 1.0518 - val_accuracy: 0.6372
Epoch 8/10
2813/2813 [=====] - 54s 19ms/step - loss: 0.9684 - accuracy: 0.6557 - val_loss: 1.1032 - val_accuracy: 0.6156
Epoch 9/10
2813/2813 [=====] - 59s 21ms/step - loss: 0.9343 - accuracy: 0.6653 - val_loss: 1.0021 - val_accuracy: 0.6416
Epoch 10/10
2813/2813 [=====] - 54s 19ms/step - loss: 0.8993 - accuracy: 0.6794 - val_loss: 1.0700 - val_accuracy: 0.6332
```

테스트 결과 출력

모델2의 학습

에폭 10를 기준으로 모델을 돌렸습니다. 모델1의 에폭1는 Loss 1.6122, accyuracy 0.4048

검증손실은 1.3981, 검증 정확도는 0.4854이로 나오고 에폭 10은 Loss 0.8993, accyuracy 0.6794

검증손실은 1.0700, 검증 정확도는 0.6332입니다 초기 훈련에서 훈련 손실 및 검증 손실이 높지만 에폭을 진행될수록 손실이 감소하고 정확도가 올라가는 것을 확인할 수 있었습니다.

즉 검증 정확도가 0.48에서 0.63으로 상승했다는 것을 확인할 수 있습니다.

```
[14] last_epoch_loss1 = history1.history['val_loss'][-1]
last_epoch_accuracy1 = history1.history['val_accuracy'][-1]

last_epoch_loss2 = history2.history['val_loss'][-1]
last_epoch_accuracy2 = history2.history['val_accuracy'][-1]

print(f"Last Epoch Loss for Model1: {last_epoch_loss1}")
print(f"Last Epoch Accuracy for Model1: {last_epoch_accuracy1}")
print(f"Last Epoch Loss for Model2: {last_epoch_loss2}")
print(f"Last Epoch Accuracy for Model2: {last_epoch_accuracy2}")
```

```
➡ Last Epoch Loss for Model1: 1.658138632774353
Last Epoch Accuracy for Model1: 0.3776000142097473
Last Epoch Loss for Model2: 1.0699748992919922
Last Epoch Accuracy for Model2: 0.6331999897956848
```

```
[15] test_loss1, test_acc1 = model1.evaluate(x_test, y_test, verbose=2)
test_loss2, test_acc2 = model2.evaluate(x_test, y_test, verbose=2)

print(f'Test accuracy for Model1: {test_acc1}')
print(f'Test accuracy for Model2: {test_acc2}')
CIFAR_10_CLASSES = ["airplane", "automobile", "bird", "cat", "deer", "dog", "frog", "horse", "ship", "truck"]

313/313 - 2s - loss: 1.6743 - accuracy: 0.3741 - 2s/epoch - 6ms/step
313/313 - 3s - loss: 1.0848 - accuracy: 0.6243 - 3s/epoch - 8ms/step
Test accuracy for Model1: 0.374099999666214
Test accuracy for Model2: 0.6243000030517578
```

모델1의 마지막 에폭에서 손실 값은 1.6581이요 마지막 정확도는 0.3776입니다.

모델2의 마지막 에폭에서 손실 값은 1.0699이며 마지막 정확도는 0.6332입니다.

두 모델을 비교해보면 모델2의 마지막 에폭에서 더 낮은 손실과 더 높은 정확도를 보입니다.

즉 모델2가 모델1보다 성능이 우수하다는 것을 확인할 수 있습니다.

313는 모델이 테스트 데이터를 처리하는데 걸리는 시간이며 각각 모델의 로스와 정확도를 나타냅니다. 모델1의 로스율은 1.6743, 정확도는 0.3741이고 모델2의 로스율은 1.0848, 정확도는 0.6243입니다

테스트 데이트에 대한 정확도 확인에서 모델1은 0.3741이고 모델2의 정확도는 0.6243입니다.

▶ # 모델1의 오분류된 예제 출력

```
print_misclassified_examples(model1, "Model1", x_test, y_test)
```

313/313 [=====] - 2s 6ms/step

Model Model1 - Number of Misclassifications: 6259

(3) airplane (true) -> ship (predicted) (4) frog (true) -> bird (predicted) (6) automobile (true) -> truck (predicted) (9) automobile (true) -> ship (predicted)

Class: airplane, Misclassifications: 606

Class: automobile, Misclassifications: 466

Class: bird, Misclassifications: 813

Class: cat, Misclassifications: 737

Class: deer, Misclassifications: 771

Class: dog, Misclassifications: 735

Class: frog, Misclassifications: 459

Class: horse, Misclassifications: 614

Class: ship, Misclassifications: 525

Class: truck, Misclassifications: 533

모델1는 정확도가 약 37퍼로 나왔습니다.

결과를 보면 10000개의 데이터 중 에서 6259개가 오분류 되었다고 알 수 있습니다.

밑에는 (예측번호)(실제클래스)(true) -> (예측클래스)(predicted)로 표현했습니다.

오분류된 통계를 보면 비행기는 606, 자동차 466, 새 813, 고양이 737, 사슴 771, 강아지 735, 개
구리 459, 말 614, 배 525, 트럭 533개라는 오분류된 횟수 통계를 알 수 있습니다.

▶ #모델2의 오분류된 예제 출력

```
print_misclassified_examples(model2, "Model2", x_test, y_test)
```

313/313 [=====] - 3s 9ms/step

Model Model2 - Number of Misclassifications: 3757

(1) ship (true) -> automobile (predicted) (3) airplane (true) -> ship (predicted) (4) frog (true) -> deer (predicted)

Class: airplane, Misclassifications: 309

Class: automobile, Misclassifications: 135

Class: bird, Misclassifications: 524

Class: cat, Misclassifications: 558

Class: deer, Misclassifications: 536

Class: dog, Misclassifications: 639

Class: frog, Misclassifications: 146

Class: horse, Misclassifications: 305

Class: ship, Misclassifications: 218

Class: truck, Misclassifications: 387

모델2는 정확도가 약 63퍼로 나왔습니다.

결과를 보면 10000개의 데이터 중 에서 3757개가 오분류 되었다고 알 수 있습니다.

오분류된 통계를 보면 비행기는 309, 자동차 135, 새 524, 고양이 558, 사슴 536, 강아지 639, 개
구리 146, 말 305, 배 218, 트럭 387개라는 오분류된 횟수 통계를 알 수 있습니다.

이러한 결과들을 통해서 모델1은 37퍼라는 낮은 정확도를 가지고 있고 성능이 낮으며,
모델2는 63퍼의 정확도와 모델1보다 우수한 성능을 가지고 있다고 알 수 있습니다.