

## 인공지능 MNIST과제

프로그램1개로 훈련 테스트 및 모델 2개를 구현 했습니다.

```
✓ 2분 # 6. 모델 훈련
batch_size = 32
epochs = 10

history = model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs,
                    validation_data=(x_val, y_val))

# 7. 모델 평가
test_loss, test_accuracy = model.evaluate(x_test, y_test, verbose=2)
print(f'Test accuracy: {test_accuracy * 100:.2f}%')
```

Epoch 1/10  
1500/1500 [=====] - 11s 7ms/step - loss: 0.2274 - accuracy: 0.9327 - val\_loss: 0.1207 - val\_accuracy: 0.9643  
Epoch 2/10  
1500/1500 [=====] - 11s 7ms/step - loss: 0.0913 - accuracy: 0.9720 - val\_loss: 0.1190 - val\_accuracy: 0.9652  
Epoch 3/10  
1500/1500 [=====] - 12s 8ms/step - loss: 0.0613 - accuracy: 0.9808 - val\_loss: 0.0883 - val\_accuracy: 0.9736  
Epoch 4/10  
1500/1500 [=====] - 11s 7ms/step - loss: 0.0456 - accuracy: 0.9854 - val\_loss: 0.0984 - val\_accuracy: 0.9732  
Epoch 5/10  
1500/1500 [=====] - 9s 6ms/step - loss: 0.0375 - accuracy: 0.9884 - val\_loss: 0.0780 - val\_accuracy: 0.9781  
Epoch 6/10  
1500/1500 [=====] - 11s 8ms/step - loss: 0.0268 - accuracy: 0.9911 - val\_loss: 0.0979 - val\_accuracy: 0.9753  
Epoch 7/10  
1500/1500 [=====] - 11s 7ms/step - loss: 0.0238 - accuracy: 0.9916 - val\_loss: 0.1229 - val\_accuracy: 0.9743  
Epoch 8/10  
1500/1500 [=====] - 10s 7ms/step - loss: 0.0219 - accuracy: 0.9926 - val\_loss: 0.1189 - val\_accuracy: 0.9746  
Epoch 9/10  
1500/1500 [=====] - 10s 7ms/step - loss: 0.0181 - accuracy: 0.9939 - val\_loss: 0.0993 - val\_accuracy: 0.9786  
Epoch 10/10  
1500/1500 [=====] - 11s 8ms/step - loss: 0.0153 - accuracy: 0.9949 - val\_loss: 0.1350 - val\_accuracy: 0.9735  
313/313 - 1s - loss: 0.1180 - accuracy: 0.9755 - 792ms/epoch - 3ms/step  
Test accuracy: 97.55%

훈련 테스트에서 Epoch를 10를하고 훈련시킨 결과입니다.

가장 좋은 epoch는 6이며 최종 epoch의 validation 성능은 97.86%이며, 테스트 데이터를 이용하여 테스트한 테스트 성능은 97.55%입니다. 최종적인 결과는 97.55%라는 결과를 얻어냈습니다.

## 084\_전유성MNIST.ipynb ☆

도구 도움말 모든 변경사항이 저장됨

X

+ 코드 + 텍스트

✓

2분

```
# 모델 요약
model.summary()

# 모델 1 훈련
model.fit(x_train, y_train, batch_size=32, epochs=30, validation_data=(x_val, y_val))

# 모델 1 평가
model.evaluate(x_test, y_test)
```

```
└─ dense_8 (Dense)          (None, 16)           12560
    dense_9 (Dense)          (None, 10)            170
```

```
=====
Total params: 12730 (49.73 KB)
Trainable params: 12730 (49.73 KB)
Non-trainable params: 0 (0.00 Byte)
=====
```

```
Epoch 1/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.4801 - accuracy: 0.8673 - val_loss: 0.2766 - val_accuracy: 0.9213
Epoch 2/30
1500/1500 [=====] - 5s 3ms/step - loss: 0.2735 - accuracy: 0.9229 - val_loss: 0.2469 - val_accuracy: 0.9316
Epoch 3/30
1500/1500 [=====] - 3s 2ms/step - loss: 0.2381 - accuracy: 0.9327 - val_loss: 0.2230 - val_accuracy: 0.9388
Epoch 4/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.2146 - accuracy: 0.9386 - val_loss: 0.2079 - val_accuracy: 0.9426
Epoch 5/30
1500/1500 [=====] - 4s 3ms/step - loss: 0.1976 - accuracy: 0.9430 - val_loss: 0.2045 - val_accuracy: 0.9423
Epoch 6/30
1500/1500 [=====] - 4s 3ms/step - loss: 0.1845 - accuracy: 0.9469 - val_loss: 0.1931 - val_accuracy: 0.9457
Epoch 7/30
1500/1500 [=====] - 3s 2ms/step - loss: 0.1747 - accuracy: 0.9484 - val_loss: 0.1919 - val_accuracy: 0.9457
Epoch 8/30
1500/1500 [=====] - 3s 2ms/step - loss: 0.1652 - accuracy: 0.9514 - val_loss: 0.1879 - val_accuracy: 0.9483
Epoch 9/30
1500/1500 [=====] - 5s 3ms/step - loss: 0.1597 - accuracy: 0.9535 - val_loss: 0.1809 - val_accuracy: 0.9489
Epoch 10/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.1530 - accuracy: 0.9555 - val_loss: 0.1803 - val_accuracy: 0.9498
Epoch 11/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.1470 - accuracy: 0.9565 - val_loss: 0.1774 - val_accuracy: 0.9480
Epoch 12/30
1500/1500 [=====] - 4s 3ms/step - loss: 0.1417 - accuracy: 0.9583 - val_loss: 0.1720 - val_accuracy: 0.9517
Epoch 13/30
1500/1500 [=====] - 5s 3ms/step - loss: 0.1368 - accuracy: 0.9601 - val_loss: 0.1723 - val_accuracy: 0.9513
Epoch 14/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.1332 - accuracy: 0.9605 - val_loss: 0.1752 - val_accuracy: 0.9498
Epoch 15/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.1298 - accuracy: 0.9612 - val_loss: 0.1843 - val_accuracy: 0.9485
```

용 가능

+ 코드 + 텍스트

✓

2분

```
Epoch 13/30
1500/1500 [=====] - 5s 3ms/step - loss: 0.1368 - accuracy: 0.9601 - val_loss: 0.1723 - val_accuracy: 0.9513
Epoch 14/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.1332 - accuracy: 0.9605 - val_loss: 0.1752 - val_accuracy: 0.9498
Epoch 15/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.1298 - accuracy: 0.9612 - val_loss: 0.1843 - val_accuracy: 0.9485
Epoch 16/30
1500/1500 [=====] - 5s 4ms/step - loss: 0.1262 - accuracy: 0.9626 - val_loss: 0.1749 - val_accuracy: 0.9512
Epoch 17/30
1500/1500 [=====] - 3s 2ms/step - loss: 0.1229 - accuracy: 0.9630 - val_loss: 0.1751 - val_accuracy: 0.9501
Epoch 18/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.1212 - accuracy: 0.9638 - val_loss: 0.1774 - val_accuracy: 0.9512
Epoch 19/30
1500/1500 [=====] - 4s 3ms/step - loss: 0.1184 - accuracy: 0.9647 - val_loss: 0.1762 - val_accuracy: 0.9493
Epoch 20/30
1500/1500 [=====] - 5s 3ms/step - loss: 0.1162 - accuracy: 0.9653 - val_loss: 0.1771 - val_accuracy: 0.9489
Epoch 21/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.1139 - accuracy: 0.9648 - val_loss: 0.1777 - val_accuracy: 0.9489
Epoch 22/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.1125 - accuracy: 0.9659 - val_loss: 0.1791 - val_accuracy: 0.9494
Epoch 23/30
1500/1500 [=====] - 5s 3ms/step - loss: 0.1094 - accuracy: 0.9672 - val_loss: 0.1758 - val_accuracy: 0.9517
Epoch 24/30
1500/1500 [=====] - 4s 3ms/step - loss: 0.1078 - accuracy: 0.9676 - val_loss: 0.1764 - val_accuracy: 0.9511
Epoch 25/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.1059 - accuracy: 0.9678 - val_loss: 0.1816 - val_accuracy: 0.9488
Epoch 26/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.1044 - accuracy: 0.9684 - val_loss: 0.1767 - val_accuracy: 0.9513
Epoch 27/30
1500/1500 [=====] - 5s 3ms/step - loss: 0.1024 - accuracy: 0.9689 - val_loss: 0.1774 - val_accuracy: 0.9517
Epoch 28/30
1500/1500 [=====] - 3s 2ms/step - loss: 0.1019 - accuracy: 0.9698 - val_loss: 0.1848 - val_accuracy: 0.9487
Epoch 29/30
1500/1500 [=====] - 4s 2ms/step - loss: 0.1001 - accuracy: 0.9697 - val_loss: 0.1880 - val_accuracy: 0.9497
Epoch 30/30
1500/1500 [=====] - 4s 3ms/step - loss: 0.0977 - accuracy: 0.9704 - val_loss: 0.1871 - val_accuracy: 0.9488
313/313 [=====] - 1s 2ms/step - loss: 0.1799 - accuracy: 0.9519
[0.17993777990341187, 0.9519000053405762]
```

모델1에서 Epoch를 30를하고 훈련시킨 결과입니다.

가장 좋은 epoch는 22이며 최종 epoch의 validation 성능은 95.17%이며, 테스트 데이터를 이용하여 테스트한 테스트 성능은 95.19%입니다. 최종적인 결과는 95.19%라는 결과를 얻어냈습니다.

전유성MNIST.ipynb ☆

도움말 모든 변경사항이 저장됨

+ 코드 + 텍스트

5분

```
metrics=[ 'accuracy' ] )

# 모델 2 요약
model2.summary()

# 모델 2 훈련
model2.fit(x_train, y_train, batch_size=32, epochs=30, validation_data=(x_val, y_val))

# 모델 2 평가
model2.evaluate(x_test, y_test)
```

Model: "model\_3"

Layer (type)	Output Shape	Param #
input_5 (InputLayer)	[(None, 784)]	0
dense_10 (Dense)	(None, 256)	200960
dense_11 (Dense)	(None, 128)	32896
dense_12 (Dense)	(None, 10)	1290

Total params: 235146 (918.54 KB)  
Trainable params: 235146 (918.54 KB)  
Non-trainable params: 0 (0.00 Byte)

Epoch 1/30  
1500/1500 [=====] - 12s 8ms/step - loss: 0.2324 - accuracy: 0.9305 - val\_loss: 0.1363 - val\_accuracy: 0.9613  
Epoch 2/30  
1500/1500 [=====] - 9s 6ms/step - loss: 0.0945 - accuracy: 0.9703 - val\_loss: 0.0946 - val\_accuracy: 0.9707  
Epoch 3/30  
1500/1500 [=====] - 12s 8ms/step - loss: 0.0631 - accuracy: 0.9801 - val\_loss: 0.0927 - val\_accuracy: 0.9730  
Epoch 4/30  
1500/1500 [=====] - 11s 8ms/step - loss: 0.0468 - accuracy: 0.9849 - val\_loss: 0.0892 - val\_accuracy: 0.9750  
Epoch 5/30  
1500/1500 [=====] - 10s 7ms/step - loss: 0.0358 - accuracy: 0.9884 - val\_loss: 0.0920 - val\_accuracy: 0.9754  
Epoch 6/30  
1500/1500 [=====] - 10s 7ms/step - loss: 0.0288 - accuracy: 0.9905 - val\_loss: 0.1011 - val\_accuracy: 0.9743  
Epoch 7/30  
1500/1500 [=====] - 11s 7ms/step - loss: 0.0238 - accuracy: 0.9926 - val\_loss: 0.1037 - val\_accuracy: 0.9737  
Epoch 8/30  
1500/1500 [=====] - 11s 8ms/step - loss: 0.0208 - accuracy: 0.9930 - val\_loss: 0.1317 - val\_accuracy: 0.9697  
Epoch 9/30  
1500/1500 [=====] - 9s 6ms/step - loss: 0.0176 - accuracy: 0.9940 - val\_loss: 0.1079 - val\_accuracy: 0.9764  
Epoch 10/30  
1500/1500 [=====] - 10s 7ms/step - loss: 0.0169 - accuracy: 0.9945 - val\_loss: 0.1017 - val\_accuracy: 0.9788

```

Epoch 9/30
1500/1500 [=====] - 9s 6ms/step - loss: 0.0176 - accuracy: 0.9940 - val_loss: 0.1079 - val_accuracy: 0.9764
Epoch 10/30
1500/1500 [=====] - 10s 7ms/step - loss: 0.0169 - accuracy: 0.9945 - val_loss: 0.1017 - val_accuracy: 0.9788
Epoch 11/30
1500/1500 [=====] - 11s 7ms/step - loss: 0.0163 - accuracy: 0.9945 - val_loss: 0.1234 - val_accuracy: 0.9746
Epoch 12/30
1500/1500 [=====] - 9s 6ms/step - loss: 0.0138 - accuracy: 0.9953 - val_loss: 0.1495 - val_accuracy: 0.9712
Epoch 13/30
1500/1500 [=====] - 11s 7ms/step - loss: 0.0125 - accuracy: 0.9957 - val_loss: 0.1755 - val_accuracy: 0.9665
Epoch 14/30
1500/1500 [=====] - 11s 8ms/step - loss: 0.0131 - accuracy: 0.9956 - val_loss: 0.1155 - val_accuracy: 0.9788
Epoch 15/30
1500/1500 [=====] - 11s 7ms/step - loss: 0.0112 - accuracy: 0.9967 - val_loss: 0.1159 - val_accuracy: 0.9799
Epoch 16/30
1500/1500 [=====] - 9s 6ms/step - loss: 0.0123 - accuracy: 0.9960 - val_loss: 0.1349 - val_accuracy: 0.9779
Epoch 17/30
1500/1500 [=====] - 11s 8ms/step - loss: 0.0085 - accuracy: 0.9974 - val_loss: 0.1388 - val_accuracy: 0.9789
Epoch 18/30
1500/1500 [=====] - 11s 8ms/step - loss: 0.0134 - accuracy: 0.9958 - val_loss: 0.1291 - val_accuracy: 0.9809
Epoch 19/30
1500/1500 [=====] - 10s 7ms/step - loss: 0.0065 - accuracy: 0.9979 - val_loss: 0.1254 - val_accuracy: 0.9802
Epoch 20/30
1500/1500 [=====] - 10s 7ms/step - loss: 0.0084 - accuracy: 0.9972 - val_loss: 0.1426 - val_accuracy: 0.9789
Epoch 21/30
1500/1500 [=====] - 11s 8ms/step - loss: 0.0090 - accuracy: 0.9974 - val_loss: 0.1292 - val_accuracy: 0.9787
Epoch 22/30
1500/1500 [=====] - 12s 8ms/step - loss: 0.0112 - accuracy: 0.9969 - val_loss: 0.1511 - val_accuracy: 0.9779
Epoch 23/30
1500/1500 [=====] - 9s 6ms/step - loss: 0.0098 - accuracy: 0.9971 - val_loss: 0.1677 - val_accuracy: 0.9781
Epoch 24/30
1500/1500 [=====] - 11s 7ms/step - loss: 0.0091 - accuracy: 0.9974 - val_loss: 0.1500 - val_accuracy: 0.9771
Epoch 25/30
1500/1500 [=====] - 11s 7ms/step - loss: 0.0080 - accuracy: 0.9979 - val_loss: 0.1649 - val_accuracy: 0.9772
Epoch 26/30
1500/1500 [=====] - 11s 7ms/step - loss: 0.0084 - accuracy: 0.9977 - val_loss: 0.1405 - val_accuracy: 0.9805
Epoch 27/30
1500/1500 [=====] - 10s 7ms/step - loss: 0.0084 - accuracy: 0.9975 - val_loss: 0.1833 - val_accuracy: 0.9753
Epoch 28/30
1500/1500 [=====] - 12s 8ms/step - loss: 0.0076 - accuracy: 0.9978 - val_loss: 0.1856 - val_accuracy: 0.9773
Epoch 29/30
1500/1500 [=====] - 12s 8ms/step - loss: 0.0076 - accuracy: 0.9978 - val_loss: 0.1856 - val_accuracy: 0.9773
Epoch 29/30
1500/1500 [=====] - 11s 7ms/step - loss: 0.0083 - accuracy: 0.9977 - val_loss: 0.1733 - val_accuracy: 0.9788
Epoch 30/30
1500/1500 [=====] - 10s 7ms/step - loss: 0.0080 - accuracy: 0.9981 - val_loss: 0.1622 - val_accuracy: 0.9799
313/313 [=====] - 1s 3ms/step - loss: 0.1561 - accuracy: 0.9794
[0.15606315433979034, 0.9793999791145325]

```

모델2에서 Epoch를 30를하고 훈련시킨 결과입니다.

가장 좋은 epoch는 20이며 최종 epoch의 validation 성능은 97.88%이며, 테스트 데이터를 이용하여 테스트한 테스트 성능은 97.94%입니다. 최종적인 결과는 97.94%라는 결과를 얻어냈습니다.