

## 1-6. 더 좋은 네트워크 만들어 보기

그러면 인식률을 다시 99점대로 만들수 없을까요? 딥러닝 네트워크의 구조 자체는 바꾸지 않으면서도 우리가 해볼 수 있는 것들이 있습니다. Step 3에서 살펴본 하이퍼파라미터들을 바꾸어 보는 것인데요.

**Conv2D** 레이어에서 입력 이미지의 특징 수를 늘리거나 줄여 보거나, **Dense** 레이어에서 뉴런수를 바꾸어 보거나, 학습 반복 횟수인 **epoch** 값을 변경해 볼 수 있을 겁니다.

### [Input]

#바뀌 볼 수 있는 하이퍼파라미터들

n\_channel\_1=16

n\_channel\_2=32

n\_dense=32

n\_train\_epoch=10

model=keras.models.Sequential()

model.add(keras.layers.Conv2D(n\_channel\_1, (3,3), activation='relu', input\_shape=(28,28,1)))

model.add(keras.layers.MaxPool2D(2,2))

model.add(keras.layers.Conv2D(n\_channel\_2, (3,3), activation='relu'))

model.add(keras.layers.MaxPooling2D((2,2)))

model.add(keras.layers.Flatten())

model.add(keras.layers.Dense(n\_dense, activation='relu'))

model.add(keras.layers.Dense(10, activation='softmax'))

model.summary()

model.compile(optimizer='adam',  
loss='sparse\_categorical\_crossentropy',  
metrics=['accuracy'])

# 모델 훈련

model.fit(x\_train\_resaped, y\_train, epochs=n\_train\_epoch)

# 모델 시험

test\_loss, test\_accuracy = model.evaluate(x\_test\_resaped, y\_test, verbose=2)

print("test\_loss: {}".format(test\_loss))

print("test\_accuracy: {}".format(test\_accuracy))

실행 ▶

### [Output]

**Q7. 하이퍼파라미터를 바꾸어가면서 시험용 데이터(x\_test)에 대한 인식률을 확인해 봅시다. 최고로 높은 점수를 얻은 네트워크 모델 코드와 그 때의 시험용 데이터 인식률 값을 올려주세요.**

```
n_channel_1=32  
n_channel_2=64  
n_dense=32
```

Epoch 36/40

1875/1875 [=====] - 2s 1ms/step - loss: 0.0025 - accuracy: 0.9993

제출

### 예시답안

예시답안이 없는 퀴즈입니다. 동료들과 함께 토의해보세요 :)