1.

Maxpooling을 쓰되, channel 수를 64로 고정을 하면, 모수와 test set에 적용시켰을 때의 test set의 loss와 accuracy가 다음과 같이 나온다.

Total params: 80,266

loss: 0.0269 - accuracy: 0.9921

Maxpooling을 쓰면서 channel의 수를 두 배를 시키면, 모수 수가 channel이 증가함에 따라 증가하게 되고, test set의 Accuracy가 channel을 늘리지 않은 것보다 약간 좋게 나왔다.

model\_1.add(Conv2D(filters=64,kernel\_size=3,activation='relu',input\_shape=(28,28,1)))

model\_1.add(MaxPooling2D(2))

model\_1.add(Conv2D(filters=128,kernel\_size=3,activation='relu'))

model\_1.add(MaxPooling2D(2))

model\_1.add(Conv2D(filters=256,kernel\_size=3,activation='relu')

………

Total params: 392,714

loss: 0.0265 - accuracy: 0.9929

2.

from tensorflow.keras.datasets import boston\_housing

(x\_train, y\_train),(x\_test,y\_test)=boston\_housing.load\_data()

model.fit([x\_train1,x\_train2],[y\_train,y\_train],epochs=20,batch\_size=32,validation\_data=([x\_test1,x\_test2],[y\_test,y\_test]),callbacks=[EarlyStopping(monitor='val\_loss',patience=2),ModelCheckpoint(filepath=’c:/model\_c.h5’,monitor=’val\_loss’),ReduceLROnPlateau(monitor='val\_loss',factor=0.1,patience=5)])

13/13 [==============================] - 0s 6ms/step - loss: 99.8003 - dense\_2\_loss: 151.9697 - dense\_3\_loss: 86.7580 - val\_loss: 102.1665 - val\_dense\_2\_loss: 163.7645 - val\_dense\_3\_loss: 86.7670

Epoch 11/20

13/13 [==============================] - 0s 9ms/step - loss: 97.9738 - dense\_2\_loss: 147.1122 - dense\_3\_loss: 85.6893 - val\_loss: 100.1426 - val\_dense\_2\_loss: 161.6136 - val\_dense\_3\_loss: 84.7749

Epoch 12/20

13/13 [==============================] - 0s 9ms/step - loss: 98.3752 - dense\_2\_loss: 148.2748 - dense\_3\_loss: 85.9003 - val\_loss: 107.5574 - val\_dense\_2\_loss: 153.3907 - val\_dense\_3\_loss: 96.0990

Epoch 13/20

13/13 [==============================] - 0s 9ms/step - loss: 96.4015 - dense\_2\_loss: 142.4137 - dense\_3\_loss: 84.8984 - val\_loss: 103.2043 - val\_dense\_2\_loss: 153.1711 - val\_dense\_3\_loss: 90.7125

결과를 보면, val\_loss가 연속해서 3번 이상 감소하지 않았기에 earlystopping callback에 의해 epoch을 멈추게 된다.