190521 영어음성학응용

boston\_housing

None=batch\_size

**None = batch\_size?**

In [22]:model.fit(x\_train, y\_train, epochs=1000)batch\_size=1

batch size는 계산되는 속도 ?

make the highest price

model.predict(np.array([0, 90, 0, 1, 0,

100, 0, 10, 0, 250,

13, 300, 0]).reshape(1,13))

숫자가 총 13개 들어 있는 list를 array로 바꿔 주기

왜 reshape 했는가? (None,13)처럼 형태를 지정하기 위해 reshape 필요함

8만5천 달러

여기까지 regression

<mnist>

classification mnist

이미지를 넣으면 데이터로

흰 부분은 1 검은부분은0

제일 마지막에 softmax 사용

sequential 넣기 시작

dense 두개일때

input shape은 28x28

img\_rows = 28

img\_cols = 28

model = tf.keras.models.Sequential([

tf.keras.layers.Flatten(input\_shape=(img\_rows, img\_cols)),

tf.keras.layers.Dense(512, activation='relu'),

tf.keras.layers.Dropout(0),

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

])

flatten = reshape하는 것

dense

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

Adam(lr=0.01), loss=

model accuracy 그래프에서 train은 올라가고 test는 감소하거나 변화 없는 이유?

underfit /overfit(성능이 좋아지는 시점)

model loss

train은 loss가 점점 떨어지고(fit에 가까워지는 과정) test는 어느 시점부터 overfit(증가한다.)

underfit도 아니고 overfit도 아닌 시점에 팔면 된다.

layer의 node들이 어디서 결정되는지 알아야 함.

model.add(Dense(512, activation='relu'))

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

(여기서)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Layer (type) Output Shape Param # ================================================================= flatten\_6 (Flatten) (None, 784) 0 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_12 (Dense) (None, 512) 401920 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_13 (Dense) (None, 10) 5130 ================================================================= Total params: 407,050 Trainable params: 407,050 Non-trainable params: 0 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

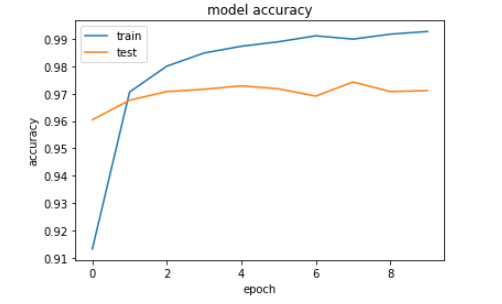
2차원의 데이터를 바꾸기 위해 flatten 사용? (flatten 사용법 알기)

history 하면 모든 기록들이 나옴

modelfit : underfit도 안되고 overfit도 안됨.

epoch 때문에 오차가 생길 수 있음..

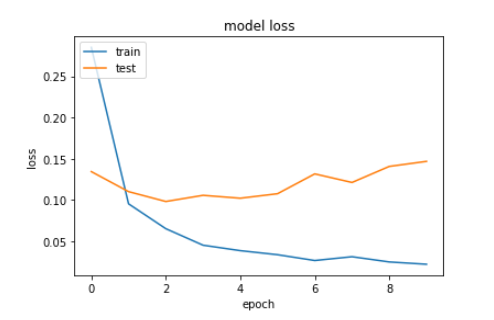
accuracy 그래프



train 데이터(80%), 오렌지색은 validation\_split=0.2에 해당..

validation의 accuracy 변화 추이가 중요

0~2까지는 underfit, 그 이후부터는 overfit이 됨



train의 loss는 계속 낮아짐. validation은 2 에서 완벽히 fit된 후 이후 점점 loss가 늘어남

fashion mnist

plt.figure(figsize=(100,100))

**for** i **in** range(100):

plt.subplot(10,10,i+1)

plt.xticks([])

plt.yticks([])

plt.imshow(x\_train[i], cmap=plt.cm.binary)

plt.title(fashion\_mnist[y\_train[i]], fontsize=100)

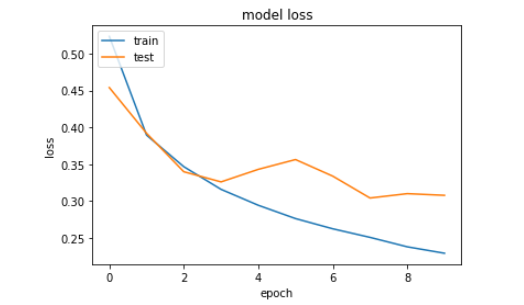
plt.show()

10,10 (10x10으로 plot)

imshow(image를 보여줘라 )

x\_train.shape

(6000,28,28)일 때 28x28이 img\_rows 와 img\_cols 에 들어감



vocab size