12 주. Keras DNN (2)			
학번	32191197	이름	김채은

Q1 (7 점) 제공된 PimaIndiansDiabetes.csv 파일에 대해 Keras 를 이용한 classification 모델을 개발하고 테스트 하시오

- train/test set 을 나누되 test set 은 전체 dataset 의 30% 로 한다.
- hidden layer 의 수는 4 개, layer 별 노드 수는 각자 정한다.
- hidden layer 의 활성화 함수는 relu, output layer 의 노드수는 softmax 로 한다
- 각 hidden layer 에 dropout 을 적용한다. (적용률을 각자 알아서)
- 기타 필요한 매개변수들은 각자 정한다.
- epoch 는 20,40,60,80 으로 변화시켜 가면서 테스트한다.

* 각 training accuracy 와 validation accuracy 학습곡선 그래프를 제시한다

Source code :

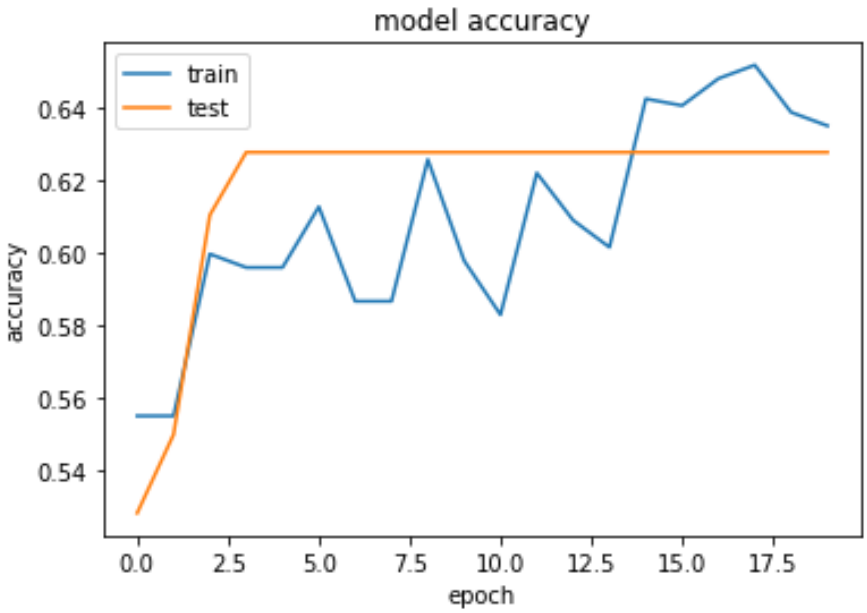
```
from keras import optimizers
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers import Dropout
from keras.utils import np_utils
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split

import pandas
import matplotlib.pyplot as plt
import numpy as np

dataframe                                                              =
pandas.read_csv("C:/Users/82104/Desktop/deeplearning/dataset/PimaIndia
nsDiabetes.csv")
dataset = dataframe.values
X = dataset[:, 0:8].astype(float)
Y = dataset[:, 8]

encoder = LabelEncoder()
encoder.fit(Y)
encoded_Y = encoder.transform(Y)
```

```
dummy_y = np_utils.to_categorical(encoded_Y)

train_X, test_X, train_y, test_y = train_test_split(X, dummy_y,
test_size=0.3, random_state=321)

epochs = 20
batch_size = 10

model = Sequential()
model.add(Flatten(input_shape=(8, )))
model.add(Dense(10, activation='relu'))
model.add(Dropout(rate=0.4))
model.add(Dense(10, activation='relu'))
model.add(Dropout(rate=0.4))
model.add(Dense(10, activation='relu'))
model.add(Dropout(rate=0.4))
model.add(Dense(2, activation='softmax'))

model.summary()

model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

disp = model.fit(train_X, train_y,
                 batch_size=batch_size,
                 epochs=epochs,
                 verbose=1,
                 validation_data=(test_X, test_y))

pred = model.predict(test_X)
y_classes = [np.argmax(y, axis=None, out=None) for y in pred]

train_score = model.evaluate(train_X, train_y, verbose=0)
print('Train loss: ', train_score[0])
print('Train accuracy: ', train_score[1])
test_score = model.evaluate(test_X, test_y, verbose=0)
print('Test loss: ', test_score[0])
print('Test accuracy: ', test_score[1])

plt.plot(disp.history['accuracy'])
plt.plot(disp.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

실행화면 캡쳐:

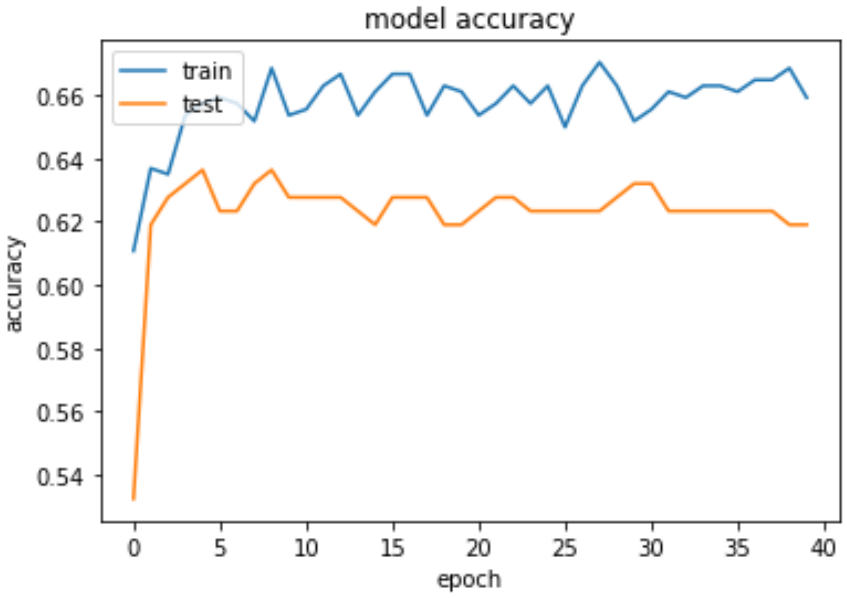
epoch=20

```
Train loss:  0.6405866742134094
Train accuracy:  0.6610800623893738
Test loss:  0.6610632538795471
Test accuracy:  0.6277056336402893
```



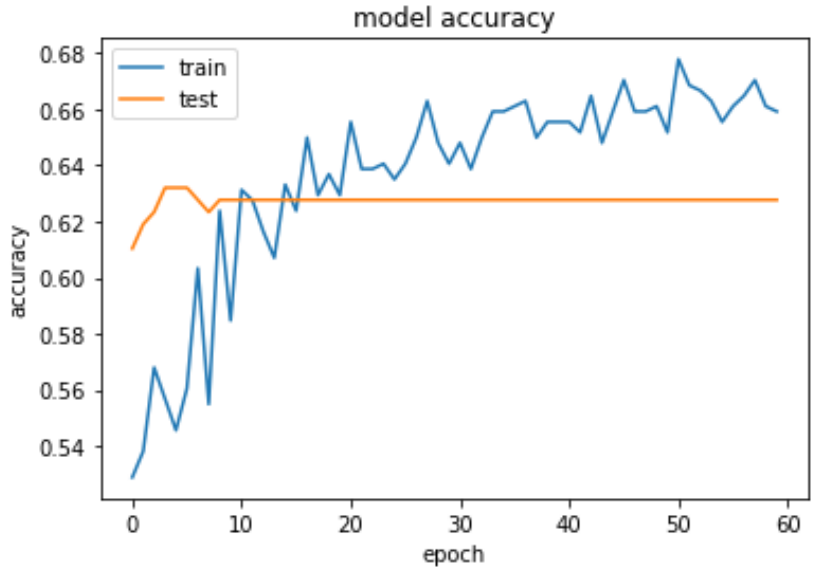
epoch=40

```
Train loss:  0.6361882090568542
Train accuracy:  0.6648044586181641
Test loss:  0.667256236076355
Test accuracy:  0.6190476417541504
```



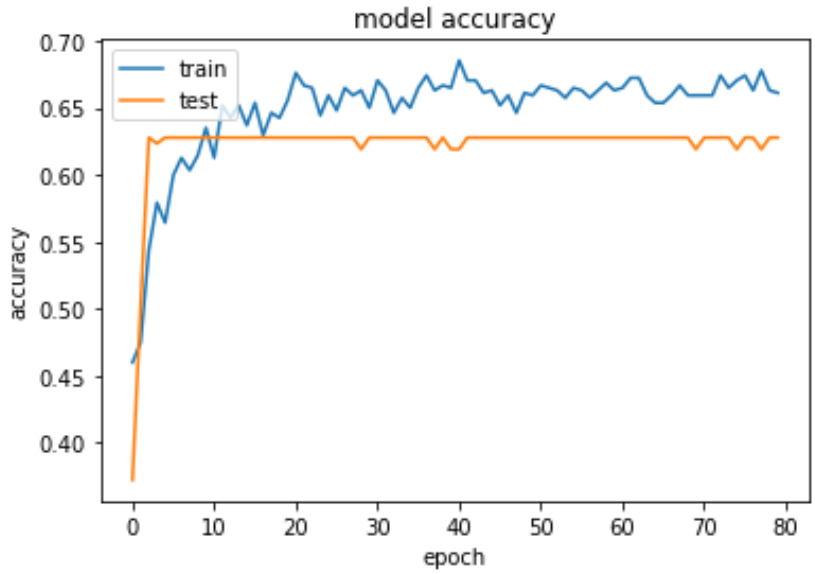
epoch=60

```
Train loss:  0.6393991112709045
Train accuracy:  0.6610800623893738
Test loss:  0.664125382900238
Test accuracy:  0.6277056336402893
```



epoch=80

```
Train loss:  0.6333968043327332
Train accuracy:  0.6610800623893738
Test loss:  0.6612846851348877
Test accuracy:  0.6277056336402893
```



Q2 (3 점) 컬러 이미지 파일을 하나 지정하여 다음의 작업을 수행하고 결과를 제시하시오

(1) 컬러 이미지 파일을 화면에 출력한다

(2) 컬러 이미지를 흑백 이미지로 변환하여 화면에 출력한다

(3) 컬러 이미지의 크기를 1/3 로 축소하여 출력한다

(4) 컬러 이미지를 좌우 반전시켜 화면에 출력한다

(5) 컬러 이미지를 상하 반전시켜 화면에 출력한다 (ppt 에 없는 내용임)

Source code :

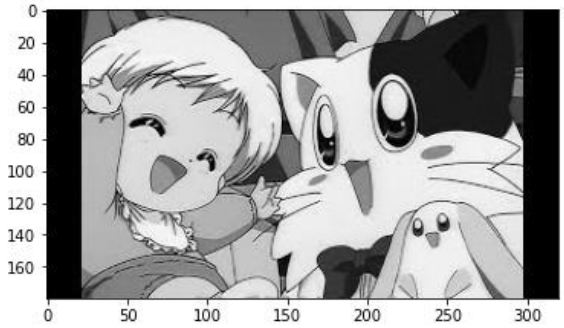
```
from skimage import io
from tkinter.filedialog import askopenfilename

from skimage import color
from skimage import transform
from skimage.transform import rotate
import numpy as np

import matplotlib.pyplot as plt
import matplotlib.image as mpimg
%matplotlib inline


# (1) 화면에 출력
image                                                                    =
mpimg.imread('C:/Users/82104/Desktop/deeplearning/mqdefault.jpg')
plt.imshow(image)

# (2) 흑백 변환
gray_image = color.rgb2gray(image)
plt.imshow(gray_image, cmap=plt.cm.gray)

# (3) 1/3 축소
new_shape = (image.shape[0]//3, image.shape[1]//3, image.shape[2])
small = transform.resize(image=image, output_shape=new_shape)
plt.imshow(small)

# (4) 좌우 반전
plt.imshow(np.fliplr(image))

# (5) 상하 반전
plt.imshow(np.flipud(image))
```

실행화면 캡쳐:

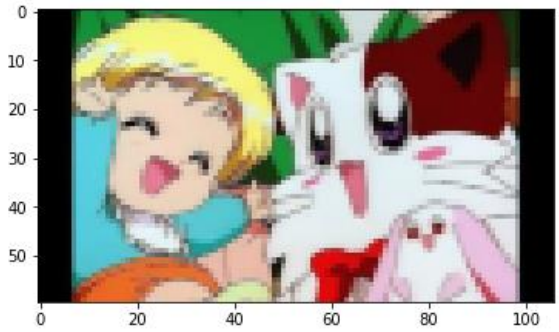
(1)


0
20
40
60
80
100
120
140
160
0
50
100
150
200
250
300

(2)


0
20
40
60
80
100
120
140
160
0
50
100
150
200
250
300

(3)


0
10
20
30
40
50
0
20
40
60
80
100

(4)


0
20
40
60
80
100
120
140
160
0
50
100
150
200
250
300

(5)