

12주. Keras DNN (2)

학번	32170578	이름	김산
----	----------	----	----

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

- train/test set을 나누되 test set 은 전체 dataset 의 30% 로 한다.
- hidden layer 의 수는 4개, layer별 노드 수는 각자 정한다.
- hidden layer 의 활성화 함수는 relu, output layer 의 노드수는 softmax 로 한다
- 각 hidden layer 에 dropout을 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 :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
from keras.models import Sequential
from keras.layers import Dense
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

# load dataset
dataframe = pandas.read_csv("./PimaIndiansDiabetes.csv")
dataset = dataframe.values

X = dataset[:,0:-1].astype(float)
Y = dataset[:, -1]

# encode class values as integers
encoder = LabelEncoder()
encoder.fit(Y)
```

```

encoded_Y = encoder.transform(Y)

# convert integers to dummy variables (i.e. one hot encoded)
dummy_y = np_utils.to_categorical(encoded_Y)

# Divide train, test(test set은 전체 dataset의 30%로 한다)
train_X, test_X, train_y, test_y = train_test_split(X,
                                                    dummy_y, test_size=0.3, random_state=321)

def define_model() :
    model = Sequential()
    model.add(Dense(100, input_dim=8, activation='relu'))
    model.add(Dense(100, activation='relu'))
    model.add(Dropout(rate = 0.3))
    model.add(Dense(100, activation='relu'))
    model.add(Dropout(rate = 0.3))
    model.add(Dense(100, activation='relu'))
    model.add(Dropout(rate = 0.3))
    model.add(Dense(100, activation='relu'))
    model.add(Dropout(rate = 0.3))
    model.add(Dense(2, activation='softmax'))

    return model

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

def fit_model(model, epochs):
    disp = model.fit(train_X, train_y,
                     epochs=epochs,
                     verbose=0, # print fitting process
                     validation_data=(test_X, test_y))

    return disp

epoch_list = [20,40,60,80]

for epochs in epoch_list:
    model = define_model()
    compile_model(model)

```

```

disp = fit_model(model, epochs)

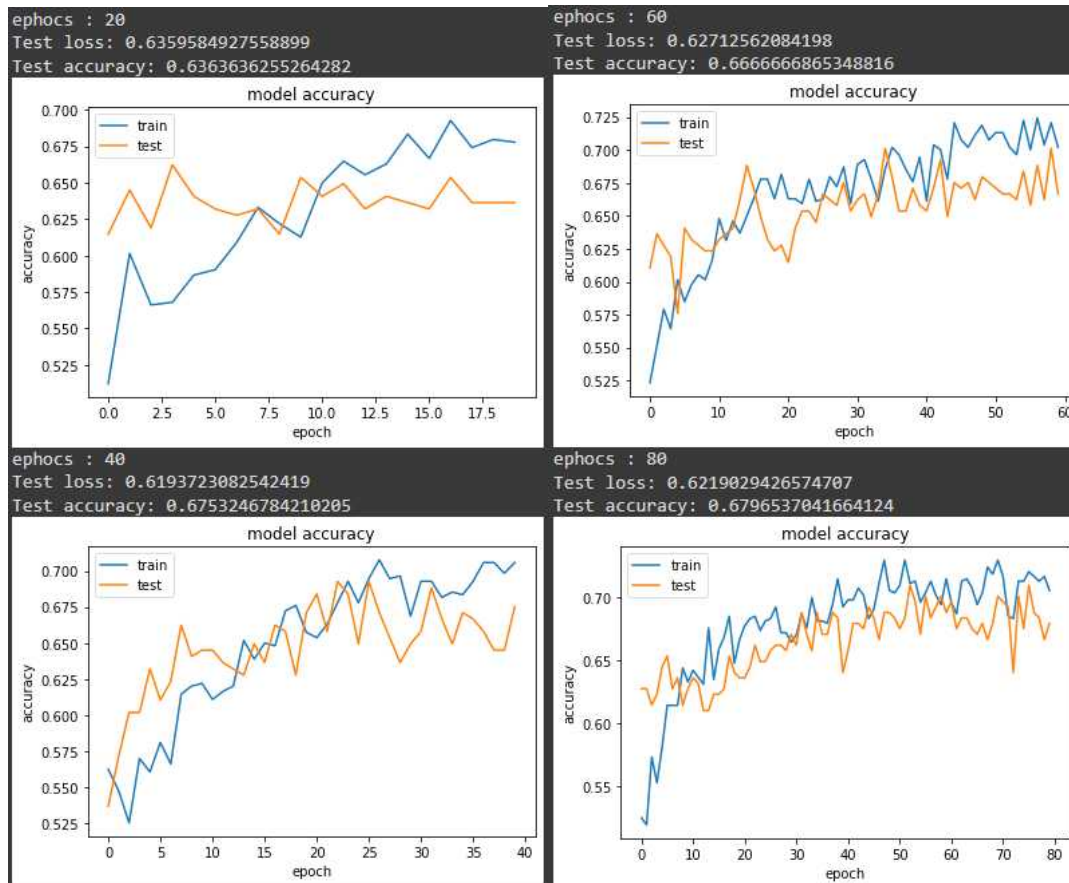
# Test model
pred = model.predict(test_X)

# model performance
score = model.evaluate(test_X, test_y, verbose=0)
print('ephocs :', epochs)
print('Test loss:', score[0])
print('Test accuracy:', score[1])

# summarize history for accuracy
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 캡처:



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

- (1) 컬러 이미지 파일을 화면에 출력한다
- (2) 컬러 이미지를 흑백 이미지로 변환하여 화면에출력한다
- (3) 컬러 이미지의 크기를 1/3 로 축소하여 출력한다
- (4) 컬러 이미지를 좌우 반전시켜 화면에 출력한다
- (5) 컬러 이미지를 상하 반전시켜 화면에 출력한다 (ppt에 없는 내용임)

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
from skimage import io
from tkinter.filedialog import askopenfilename
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

%matplotlib inline

fname = askopenfilename() # choose image file
image = mpimg.imread(fname) # read image
plt.imshow(image)

# color to gray
from skimage import color
gray_image = color.rgb2gray(image)
print(gray_image.shape)
plt.imshow(gray_image)

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

# 좌우반전
from skimage.transform import rotate
import numpy as np
plt.imshow(np.fliplr(image)) # filp

# 상하 반전
plt.imshow((np.fliplr(rotate(image, angle=180))))
```

실행화면 캡처:

