

**11주. Keras DNN**

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

Q1 (3점) 10장의 예제 소스코드를 활용하여 liver.csv 데이터셋에 대한 classification 모델을 만들고 테스트 하시오. (train:test = 6:4) 첫번째 컬럼이 class label

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
from keras.models import Sequential
from keras.layers import Dense
from keras.utils import np_utils
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split

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

# load dataset
dataframe = pd.read_csv("./liver.csv")
dataset = dataframe.values
X = dataset[:,1:].astype(float)
Y = dataset[:,0]

# one hotencoded
encoded_Y = np_utils.to_categorical(Y)

# Divide train, test
train_X, test_X, train_y, test_y = train_test_split(X,
                                                    encoded_Y,
                                                    test_size=0.4,
                                                    random_state=321)

# define model (DNN structure)
epochs = 50
batch_size = 10

model = Sequential()
model.add(Dense(10, input_dim=6, activation='relu'))
model.add(Dense(10, activation='relu'))
```

```

model.add(Dense(2, activation='softmax'))

model.summary() # show model structure

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

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

```

실행화면 캡처:

```

Model: "sequential_12"
-----
Layer (type)                Output Shape              Param #
-----
dense_37 (Dense)            (None, 10)                70
dense_38 (Dense)            (None, 10)                110
dense_39 (Dense)            (None, 2)                 22
-----
Total params: 202
Trainable params: 202
Non-trainable params: 0
-----

```

```

Test loss: 0.6837843656539917
Test accuracy: 0.5797101259231567

```

Q2 (3점) 예제에 hidden layer 를 한층 더 추가 하되 node 수는 8 로 하고, activation 은 relu 함수를 적용하여 테스트 하시오 (liver.csv 데이터셋)

Source code :

```

// source code 의 폰트는 Courier10 BT Bold으로 하시오
# define model (DNN structure)
epochs = 50
batch_size = 10

model = Sequential()
model.add(Dense(10, input_dim=6, activation='relu'))

```

```

model.add(Dense(10, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(2, activation='softmax'))

model.summary() # show model structure

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

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

```

실행화면 캡처:

```

Model: "sequential_21"
-----
Layer (type)                 Output Shape              Param #
-----
dense_70 (Dense)             (None, 10)                70
dense_71 (Dense)             (None, 10)               110
dense_72 (Dense)             (None, 8)                 88
dense_73 (Dense)             (None, 2)                 18
-----
Total params: 286
Trainable params: 286
Non-trainable params: 0

```

```

Test loss: 0.5929567813873291
Test accuracy: 0.7028985619544983

```

Q2 (4점) 2번문제에서 epoch 를 100, 150, 200 으로 변경하여 시행한 뒤 변경전과 결과 (test dataset에 대한 loss, accuracy)를 비교하여 보시오. (epoch 200 의 경우는 학습곡선 그래프 제시)

Source code :

```

// source code 의 폰트는 Courier10 BT Bold으로 하시오
disp
for epochs in [100,150,200]:
    # model fitting (learning)

```

```

disp = model.fit(train_X, train_y, batch_size= batch_size,
                  epochs=epochs, verbose=0, validation_data=(test_X,
test_y))

# Test model
pred = model.predict(test_X)

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

```

100 Test loss: 0.5614426732063293
100 Test accuracy: 0.7101449370384216
150 Test loss: 0.540718138217926
150 Test accuracy: 0.7681159377098083
200 Test loss: 0.5244383811950684
200 Test accuracy: 0.7681159377098083

```

