

**11 주. Keras DNN**

학번	32191197	이름	김채은
----	----------	----	-----

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

Source code :

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

dataframe = pandas.read_csv("C:/Users/82104/Desktop/deeplearning/dataset/liver.csv")
dataset = dataframe.values
X = dataset[:, 1:7].astype(float)
Y = dataset[:, 0]

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=0.4, random_state=321)

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()
```

```

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]
print(y_classes)

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

```

실행화면 캡처:

```

Model: "sequential_8"

```

Layer (type)	Output Shape	Param #
dense_24 (Dense)	(None, 10)	70
dense_25 (Dense)	(None, 10)	110
dense_26 (Dense)	(None, 2)	22

```

=====
Total params: 202
Trainable params: 202
Non-trainable params: 0

```

```

[0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0,
0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1,
1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0,
1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0,
1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0]
Test loss: 0.7007268071174622
Test accuracy: 0.6449275612831116

```

Q2 (3 점) 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 :

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

dataframe = pandas.read_csv("C:/Users/82104/Desktop/deeplearning/dataset/liver.csv")
dataset = dataframe.values
X = dataset[:, 1:7].astype(float)
Y = dataset[:, 0]

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=0.4, random_state=321)

epochs = 50
batch_size = 10

model = Sequential()
model.add(Dense(8, input_dim=6, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(8, activation='relu'))
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]
print(y_classes)

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

```

**실행화면 캡처:**

```

Model: "sequential_11"

```

Layer (type)	Output Shape	Param #
dense_35 (Dense)	(None, 8)	56
dense_36 (Dense)	(None, 8)	72
dense_37 (Dense)	(None, 8)	72
dense_38 (Dense)	(None, 2)	18

```

=====
Total params: 218
Trainable params: 218
Non-trainable params: 0

```

```

[0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1,
0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1,
0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1]
Test loss: 0.6668049097061157
Test accuracy: 0.5724637508392334

```

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

**Source code :**

```

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
import matplotlib.pyplot as plt
import numpy as np

dataframe =
pandas.read_csv("C:/Users/82104/Desktop/deeplearning/dataset/liver.csv

```

```

")
dataset = dataframe.values
X = dataset[:, 1:7].astype(float)
Y = dataset[:, 0]

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=0.4, random_state=321)

epochs = 200
batch_size = 10

model = Sequential()
model.add(Dense(8, input_dim=6, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(8, activation='relu'))
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]
print(y_classes)

score = model.evaluate(test_X, test_y, verbose=0)
print('Test loss: ', score[0])
print('Test accuracy: ', 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 캡처:

epoch=100

```
Test loss: 0.6271327137947083  
Test accuracy: 0.6376811861991882
```

epoch=150

```
Test loss: 0.6492846012115479  
Test accuracy: 0.6594203114509583
```

epoch=200

```
Test loss: 0.5010697245597839  
Test accuracy: 0.7753623127937317
```

