**1.**

**Source code :**

|  |
| --- |
| // source code 의 폰트는 Courier10 BT Bold으로 하시오  import numpy as np  import pandas  from tensorflow.keras.models import Sequential  from tensorflow.keras.layers import Dense  from keras.utils import np\_utils  from sklearn.preprocessing import LabelEncoder  from sklearn.model\_selection import train\_test\_split  import matplotlib.pyplot as plt  import tensorflow as tf  tf.random.set\_seed(2)  # load dataset  dataframe= pandas.read\_csv("C:/Users/user/PycharmProjects/deepLearning/data/liver.csv")  dataset = dataframe.values  X = dataset[:,1::].astype(float)  Y = dataset[:,0]  # encode class values as integers  encoder = LabelEncoder()  encoder.fit(Y)  encoded\_Y= encoder.transform(Y)  # one hot encoding  dummy\_y= np\_utils.to\_categorical(encoded\_Y)  # Divide train, test  train\_X, test\_X, train\_y, test\_y= train\_test\_split(X, dummy\_y, test\_size=0.4, random\_state=321)  # define model (DNN structure)  epochs = 200 #50 #100 #150 #200  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,# print fitting process  validation\_data=(test\_X, test\_y))  # Test model  pred= model.predict(test\_X)  print(pred)  y\_classes= [np.argmax(y, axis=None, out=None) for y in pred]  print(y\_classes)  # result of prediction# model performance  score = model.evaluate(test\_X, test\_y, verbose=0)  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 캡쳐:**

**Epoch : 50**

****

****

**Epoch : 100**

****

****

**Epoch : 150**

****

****

**Epoch : 200**

****

****

**Epoch 이 200일 때 정확도가 가장 높지만 150일때와 비교했을 때 loss가 줄어들지 않았고, 학습 및 테스트 곡선이 상당히 불안정하게 흔든리는 것으로 관찰된다. 모맨텀을 추가하여 안정적인 모델이 될 수 있도록 개선할 수 있을 것 같다.**