from keras.models import Sequential from keras.layers import Dense, Activation from keras.utils import np\_utils

import numpy as np

seed = 7 np.random.seed(seed) data = np.loadtxt("data.txt")

X\_train = data[:4646,:12]

Y\_train = data[:4646,12:13]

X\_test = data[4646:6936,:12] Y\_test = data[4646:6936,12:13] len\_y\_train = len(Y\_train)

len\_y\_test = len(Y\_test)

print(len(Y\_train))

print(len(Y\_test))

for i in range(0,len\_y\_train): if(Y\_train[i]==1000): Y\_train[i] = 3 Y\_train[i] = int(Y\_train[i]) elif(Y\_train[i]==100): Y\_train[i] = 2 Y\_train[i] = int(Y\_train[i]) elif(Y\_train[i]==10): Y\_train[i] = 1 Y\_train[i] = int(Y\_train[i]) elif(Y\_train[i]==1): Y\_train[i] = 0

Y\_train[i] = int(Y\_train[i])

for i in range(0,len\_y\_test): if(Y\_test[i]==1000): Y\_test[i] = 3 Y\_test[i] = int(Y\_test[i]) elif(Y\_test[i]==100): Y\_test[i] = 2 Y\_test[i] = int(Y\_test[i]) elif(Y\_test[i]==10): Y\_test[i] = 1 Y\_test[i] = int(Y\_test[i]) elif(Y\_test[i]==1): Y\_test[i] = 0

Y\_test[i] = int(Y\_test[i]) print(Y\_train)

print(Y\_test)

Y\_train = Y\_train.astype('int32')

Y\_train = np\_utils.to\_categorical(Y\_train,4)

Y\_test = Y\_test.astype('int32')

Y\_test = np\_utils.to\_categorical(Y\_test,4) model = Sequential()

model.add(Dense(100, input\_dim=12, init='uniform', activation='relu')) model.add(Dense(80, init='uniform', activation='relu')) model.add(Dense(60, init='uniform', activation='relu')) model.add(Dense(60, init='uniform', activation='relu'))

model.add(Dense(4)) model.add(Activation('softmax'))

model.summary()

model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy']) model.fit(X\_train, Y\_train, nb\_epoch=150, batch\_size=10, verbose=2,

validation\_data=(X\_test,Y\_test))

scores = model.evaluate(X\_test, Y\_test, verbose=0)

print("\n")

print("%s: %.2f%%" % (model.metrics\_names[1], scores[1]\*100))

json\_string = model.to\_json()

open('model\_architecture.json', 'w').write(json\_string)

model.save\_weights('weights.h5',overwrite=True)