elmo

# example code

|  |
| --- |
| %tensorflow\_version 1.x  import tensorflow\_hub as hub  import tensorflow as tf  from keras import backend as K  import urllib.request  import pandas as pd  import numpy as np  elmo = hub.Module("https://tfhub.dev/google/elmo/1", trainable=True)  # 텐서플로우 허브로부터 ELMo를 다운로드  sess = tf.Session()  K.set\_session(sess)  sess.run(tf.global\_variables\_initializer())  sess.run(tf.tables\_initializer())  urllib.request.urlretrieve("https://raw.githubusercontent.com/mohitgupta-omg/Kaggle-SMS-Spam-Collection-Dataset-/master/spam.csv", filename="spam.csv")  data = pd.read\_csv('spam.csv', encoding='latin-1')  data[:5]  data['v1'] = data['v1'].replace(['ham','spam'],[0,1])  y\_data = list(data['v1'])  X\_data = list(data['v2'])  X\_data[:5]  print(len(X\_data))  n\_of\_train = int(len(X\_data) \* 0.8)  n\_of\_test = int(len(X\_data) - n\_of\_train)  print(n\_of\_train)  print(n\_of\_test)  X\_train = np.asarray(X\_data[:n\_of\_train]) #X\_data 데이터 중에서 앞의 4457개의 데이터만 저장  y\_train = np.asarray(y\_data[:n\_of\_train]) #y\_data 데이터 중에서 앞의 4457개의 데이터만 저장  X\_test = np.asarray(X\_data[n\_of\_train:]) #X\_data 데이터 중에서 뒤의 1115개의 데이터만 저장  y\_test = np.asarray(y\_data[n\_of\_train:]) #y\_data 데이터 중에서 뒤의 1115개의 데이터만 저장  def ELMoEmbedding(x):      return elmo(tf.squeeze(tf.cast(x, tf.string)), as\_dict=True, signature="default")["default"]  # 데이터의 이동이 케라스 → 텐서플로우 → 케라스가 되도록 하는 함수  from keras.models import Model  from keras.layers import Dense, Lambda, Input  input\_text = Input(shape=(1,), dtype=tf.string)  embedding\_layer = Lambda(ELMoEmbedding, output\_shape=(1024, ))(input\_text)  hidden\_layer = Dense(256, activation='relu')(embedding\_layer)  output\_layer = Dense(1, activation='sigmoid')(hidden\_layer)  model = Model(inputs=[input\_text], outputs=output\_layer)  model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])  history = model.fit(X\_train, y\_train, epochs=1, batch\_size=60)  print("\n 테스트 정확도: %.4f" % (model.evaluate(X\_test, y\_test)[1])) |

# testing result

|  |
| --- |
| Epoch 1/1 4457/4457 [==============================] - 52s 12ms/step - loss: 0.1095 - accuracy: 0.9668  1115/1115 [==============================] - 11s 10ms/step 테스트 정확도: 0.9776 |