#accuracy 0.689309

#\_\*\_coding: utf-8\_\*\_

import tensorflow as tf

import numpy as np

dirname = '/gdrive/My Drive/조아쓰/자료조사/18.11.07\_녹음/'

data = np.load(dirname + 'numpy/data\_.npz')

x\_data = data['X']

y\_data = data['Y']

tf.reset\_default\_graph()

with tf.Session() as sess:

saver = tf.train.import\_meta\_graph('/gdrive/My Drive/조아쓰/자료조사/18.11.07\_녹음/190128\_2/model.ckpt.meta')

saver.restore(sess, tf.train.latest\_checkpoint('/gdrive/My Drive/조아쓰/자료조사/18.11.07\_녹음/190128\_2/'))

graph = tf.get\_default\_graph()

learning\_rate = 0.1

training\_epochs = 10000000

#batch\_size = 100

n\_dim = 39

n\_classes = 9

n\_hid1 = 300

n\_hid2 = 200

n\_hid3 = 100

dirname = '/gdrive/My Drive/조아쓰/자료조사/18.11.07\_녹음/'

X = graph.get\_tensor\_by\_name("X:0")

Y = graph.get\_tensor\_by\_name("Y:0")

stddev = 0.1 #standard deviation

#WEIGHTS

W1 = graph.get\_tensor\_by\_name("w1:0")

W2 = graph.get\_tensor\_by\_name("w2:0")

W3 = graph.get\_tensor\_by\_name("w3:0")

W = graph.get\_tensor\_by\_name("w:0")

#BIASES

b1 = graph.get\_tensor\_by\_name("b1:0")

b2 = graph.get\_tensor\_by\_name("b2:0")

b3 = graph.get\_tensor\_by\_name("b3:0")

b = graph.get\_tensor\_by\_name("b:0")

#laters

layer1 = tf.nn.sigmoid(tf.matmul(X, W1) + b1)

layer2 = tf.nn.relu(tf.matmul(layer1, W2) + b2)

layer3 = tf.nn.relu(tf.matmul(layer2, W3) + b3)

#dropout

keep\_prob = graph.get\_tensor\_by\_name("keep\_prob:0")

dropout = tf.nn.dropout(layer3, keep\_prob)

k = tf.matmul(dropout, W) + b

y = tf.nn.softmax(tf.matmul(dropout, W) + b)#추측값

cross\_entropy = tf.reduce\_mean(-Y\*tf.log(tf.clip\_by\_value(y, 1e-10, 1.0)))#tf.clip\_by\_value(t, min, max, name = None): t를 min과 max 사이의 값으로 지정

train\_step = tf.train.GradientDescentOptimizer(learning\_rate).minimize(cross\_entropy)

correct\_prediction = tf.equal(tf.argmax(y, 1), tf.argmax(Y, 1))

accuracy = tf.reduce\_mean(tf.cast(correct\_prediction, tf.float32))

#init = tf.global\_variables\_initializer()

#sess.run(tf.global\_variables\_initializer())

epoch = 0

train\_accuracy = 0

for epoch in range(training\_epochs):

sess.run(train\_step, feed\_dict = {X: x\_data, Y: y\_data, keep\_prob: 0.5})

if epoch%1000 == 0:

train\_accuracy, hypo = sess.run([accuracy, y], feed\_dict = {X: x\_data, Y: y\_data, keep\_prob: 1.0})

print("step: %d, "%(epoch))

print("training accuracy: %f, "%(train\_accuracy))

print("hypo: ", hypo)

print("\n")

if epoch%1000 == 0:

saver.save(sess, dirname + '190129/model.ckpt')

print("save %d"%epoch)

if 0.9 < train\_accuracy:

break

if hypo[0][0] == 'nan':

break

print(sess.run([accuracy, y], feed\_dict = {X: x\_data, Y: y\_data, keep\_prob: 1.0}))

saver.save(sess, "/gdrive/My Drive/조아쓰/자료조사/18.11.07\_녹음/190129/model.ckpt")

data.close()

print("save the model")