import numpy as np

import tensorflow.compat.v1 as tf

def create\_sparse\_tensor(batch\_size, input\_size, num\_values, dtype=tf.float32):

random\_indices = np.sort(np.random.randint(batch\_size \* input\_size, size=num\_values))

test\_indices\_i = random\_indices // input\_size

test\_indices\_j = random\_indices % input\_size

test\_indices = np.stack([test\_indices\_i, test\_indices\_j], axis=1)

test\_values = np.random.random(num\_values).astype(dtype.as\_numpy\_dtype)

return tf.SparseTensor(indices=tf.constant(test\_indices),

values=tf.constant(test\_values),

dense\_shape=(batch\_size, input\_size))

def create\_reference\_input(sparse\_input, use\_binary\_values):

if use\_binary\_values:

sp\_a = tf.SparseTensor(indices=sparse\_input.indices,

values=tf.ones\_like(sparse\_input.values),

dense\_shape=sparse\_input.dense\_shape)

else:

sp\_a = sparse\_input

return sp\_a