import tensorflow.compat.v1 as tf

from twml.contrib.utils import math\_fns

def mean\_max\_normalizaiton(dense\_tensor):

"""

In-batch normalization

Args:

dense\_tensor: A dense `Tensor`.

Returns:

(dense\_tensor - mean) / abs(max value)

Note:

when dense\_tensor is of size [1, ?] it will give 0

If this is not what you want handle it outside the function

"""

dense\_mean = tf.reduce\_mean(dense\_tensor, reduction\_indices=[0])

dense\_abs\_max = tf.abs(tf.reduce\_max(dense\_tensor, reduction\_indices=[0]))

dense\_tensor = math\_fns.safe\_div(dense\_tensor - dense\_mean, dense\_abs\_max,

'mean\_max\_normalization\_in\_batch')

return dense\_tensor

def standard\_normalizaiton(dense\_tensor):

"""

In-batch normalization

z-normalization or standard\_normalization in batch

Args:

dense\_tensor: A dense `Tensor`.

Returns:

(dense\_tensor - mean) / variance

Note:

when dense\_tensor is of size [1, ?] it will give 0

If this is not what you want handle it outside the function

"""

epsilon = 1E-7

dense\_mean, dense\_variance = tf.nn.moments(dense\_tensor, 0)

# using epsilon is safer than math\_fns.safe\_div in here

dense\_tensor = (dense\_tensor - dense\_mean) / (dense\_variance + epsilon)

return dense\_tensor