# pylint: disable=useless-super-delegation

"""

Implementing Stitch Layer

"""

from .layer import Layer

import tensorflow.compat.v1 as tf

class Stitch(Layer):

"""

This layer is responsible for stitching a partioned layer together.

Output:

A layer that performs stitching

"""

def compute\_output\_shape(self, input\_shape):

"""Computes the output shape of the layer given the input shape.

Args:

input\_shape: A (possibly nested tuple of) `TensorShape`. It need not

be fully defined (e.g. the batch size may be unknown).

Raises NotImplementedError.

"""

raise NotImplementedError

def call(self, partioned\_val, partioned\_keys,

partioned\_indices, \*\*kwargs): # pylint: disable=unused-argument, arguments-differ

"""

This layer is responsible for stitching a partioned layer together.

Input:

partioned\_val:

a list of partioned Tensors which represent the vals of the hashmap

partioned\_keys:

a list of partioned Tensors which represent the keys of the hashmap

partioned\_indices:

a list of partioned Tensors which represent the indices of the hashmap

Output:

List which contains: [output\_vals, output\_keys]

output\_vals:

Values of the HashMap (float)

output\_keys:

Keys of HashMap (float)

"""

indices = [tf.to\_int32(index) for index in partioned\_indices]

concat\_keys = tf.dynamic\_stitch(indices, partioned\_keys)

concat\_vals = tf.dynamic\_stitch(indices, partioned\_val)

return [concat\_vals, concat\_keys]