# checkstyle: noqa

import tensorflow.compat.v1 as tf

from .constants import INDEX\_BY\_LABEL, LABEL\_NAMES

# TODO: Read these from command line arguments, since they specify the existing example weights in the input data.

DEFAULT\_WEIGHT\_BY\_LABEL = {

"is\_clicked": 0.3,

"is\_favorited": 1.0,

"is\_open\_linked": 0.1,

"is\_photo\_expanded": 0.03,

"is\_profile\_clicked": 1.0,

"is\_replied": 9.0,

"is\_retweeted": 1.0,

"is\_video\_playback\_50": 0.01

}

def add\_weight\_arguments(parser):

for label\_name in LABEL\_NAMES:

parser.add\_argument(

\_make\_weight\_cli\_argument\_name(label\_name),

type=float,

default=DEFAULT\_WEIGHT\_BY\_LABEL[label\_name],

dest=\_make\_weight\_param\_name(label\_name)

)

def make\_weights\_tensor(input\_weights, label, params):

'''

Replaces the weights for each positive engagement and keeps the input weights for negative examples.

'''

weight\_tensors = [input\_weights]

for label\_name in LABEL\_NAMES:

index, default\_weight = INDEX\_BY\_LABEL[label\_name], DEFAULT\_WEIGHT\_BY\_LABEL[label\_name]

weight\_param\_name =\_make\_weight\_param\_name(label\_name)

weight\_tensors.append(

tf.reshape(tf.math.scalar\_mul(getattr(params, weight\_param\_name) - default\_weight, label[:, index]), [-1, 1])

)

return tf.math.accumulate\_n(weight\_tensors)

def \_make\_weight\_cli\_argument\_name(label\_name):

return f"--weight.{label\_name}"

def \_make\_weight\_param\_name(label\_name):

return f"weight\_{label\_name}"