# pylint: disable=no-member, invalid-name, attribute-defined-outside-init

"""

Contains the Isotonic Layer

"""

from .layer import Layer

import libtwml

import numpy as np

class Isotonic(Layer):

"""

This layer is created by the IsotonicCalibrator.

Typically it is used intead of sigmoid activation on the output unit.

Arguments:

n\_unit:

number of input units to the layer (same as number of output units).

n\_bin:

number of bins used for isotonic calibration.

More bins means a more precise isotonic function.

Less bins means a more regularized isotonic function.

xs\_input:

A tensor containing the boundaries of the bins.

ys\_input:

A tensor containing calibrated values for the corresponding bins.

Output:

output:

A layer containing calibrated probabilities with same shape and size as input.

Expected Sizes:

xs\_input, ys\_input:

[n\_unit, n\_bin].

Expected Types:

xs\_input, ys\_input:

same as input.

"""

def \_\_init\_\_(self, n\_unit, n\_bin, xs\_input=None, ys\_input=None, \*\*kwargs):

super(Isotonic, self).\_\_init\_\_(\*\*kwargs)

self.\_n\_unit = n\_unit

self.\_n\_bin = n\_bin

self.xs\_input = np.empty([n\_unit, n\_bin], dtype=np.float32) if xs\_input is None else xs\_input

self.ys\_input = np.empty([n\_unit, n\_bin], dtype=np.float32) if ys\_input is None else ys\_input

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 build(self, input\_shape): # pylint: disable=unused-argument

"""Creates the variables of the layer."""

self.built = True

def call(self, inputs, \*\*kwargs): # pylint: disable=unused-argument

"""The logic of the layer lives here.

Arguments:

inputs: input tensor(s).

Returns:

The output from the layer

"""

calibrate\_op = libtwml.ops.isotonic\_calibration(inputs, self.xs\_input, self.ys\_input)

return calibrate\_op