"""

Wrappers around tf.estimator.Exporters to export models and save checkpoints.

"""

import os

import tensorflow.compat.v1 as tf

from tensorflow.python.estimator import exporter

import twml

class \_AllSavedModelsExporter(tf.estimator.Exporter):

"""Internal exporter class to be used for exporting models for different modes."""

def \_\_init\_\_(self,

name,

input\_receiver\_fn\_map,

backup\_checkpoints,

assets\_extra=None,

as\_text=False):

"""

Args:

name: A unique name to be used for the exporter. This is used in the export path.

input\_receiver\_fn\_map: A map of tf.estimator.ModeKeys to input\_receiver\_fns.

backup\_checkpoints: A flag to specify if backups of checkpoints need to be made.

assets\_extra: Additional assets to be included in the exported model.

as\_text: Specifies if the exported model should be in a human readable text format.

"""

self.\_name = name

self.\_input\_receiver\_fn\_map = input\_receiver\_fn\_map

self.\_backup\_checkpoints = backup\_checkpoints

self.\_assets\_extra = assets\_extra

self.\_as\_text = as\_text

@property

def name(self):

return self.\_name

def export(self, estimator, export\_path, checkpoint\_path, eval\_result,

is\_the\_final\_export):

del is\_the\_final\_export

export\_path = twml.util.sanitize\_hdfs\_path(export\_path)

checkpoint\_path = twml.util.sanitize\_hdfs\_path(checkpoint\_path)

if self.\_backup\_checkpoints:

backup\_path = os.path.join(export\_path, "checkpoints")

# Ensure backup\_path is created. makedirs passes if dir already exists.

tf.io.gfile.makedirs(backup\_path)

twml.util.backup\_checkpoint(checkpoint\_path, backup\_path, empty\_backup=False)

export\_result = estimator.experimental\_export\_all\_saved\_models(

export\_path,

self.\_input\_receiver\_fn\_map,

assets\_extra=self.\_assets\_extra,

as\_text=self.\_as\_text,

checkpoint\_path=checkpoint\_path)

return export\_result

class BestExporter(tf.estimator.BestExporter):

"""

This class inherits from tf.estimator.BestExporter with the following differences:

- It also creates a backup of the best checkpoint.

- It can export the model for multiple modes.

A backup / export is performed everytime the evaluated metric is better

than previous models.

"""

def \_\_init\_\_(self,

name='best\_exporter',

input\_receiver\_fn\_map=None,

backup\_checkpoints=True,

event\_file\_pattern='eval/\*.tfevents.\*',

compare\_fn=exporter.\_loss\_smaller,

assets\_extra=None,

as\_text=False,

exports\_to\_keep=5):

"""

Args:

name: A unique name to be used for the exporter. This is used in the export path.

input\_receiver\_fn\_map: A map of tf.estimator.ModeKeys to input\_receiver\_fns.

backup\_checkpoints: A flag to specify if backups of checkpoints need to be made.

Note:

Check the following documentation for more information about the remaining args:

https://www.tensorflow.org/api\_docs/python/tf/estimator/BestExporter

"""

serving\_input\_receiver\_fn = input\_receiver\_fn\_map.get(tf.estimator.ModeKeys.PREDICT)

super(BestExporter, self).\_\_init\_\_(

name, serving\_input\_receiver\_fn, event\_file\_pattern, compare\_fn,

assets\_extra, as\_text, exports\_to\_keep)

if not hasattr(self, "\_saved\_model\_exporter"):

raise AttributeError(

"\_saved\_model\_exporter needs to exist for this exporter to work."

" This is potentially broken because of an internal change in Tensorflow")

# Override the saved\_model\_exporter with SaveAllmodelsexporter

self.\_saved\_model\_exporter = \_AllSavedModelsExporter(

name, input\_receiver\_fn\_map, backup\_checkpoints, assets\_extra, as\_text)

class LatestExporter(tf.estimator.LatestExporter):

"""

This class inherits from tf.estimator.LatestExporter with the following differences:

- It also creates a backup of the latest checkpoint.

- It can export the model for multiple modes.

A backup / export is performed everytime the evaluated metric is better

than previous models.

"""

def \_\_init\_\_(self,

name='latest\_exporter',

input\_receiver\_fn\_map=None,

backup\_checkpoints=True,

assets\_extra=None,

as\_text=False,

exports\_to\_keep=5):

"""

Args:

name: A unique name to be used for the exporter. This is used in the export path.

input\_receiver\_fn\_map: A map of tf.estimator.ModeKeys to input\_receiver\_fns.

backup\_checkpoints: A flag to specify if backups of checkpoints need to be made.

Note:

Check the following documentation for more information about the remaining args:

https://www.tensorflow.org/api\_docs/python/tf/estimator/LatestExporter

"""

serving\_input\_receiver\_fn = input\_receiver\_fn\_map.get(tf.estimator.ModeKeys.PREDICT)

super(LatestExporter, self).\_\_init\_\_(

name, serving\_input\_receiver\_fn, assets\_extra, as\_text, exports\_to\_keep)

if not hasattr(self, "\_saved\_model\_exporter"):

raise AttributeError(

"\_saved\_model\_exporter needs to exist for this exporter to work."

" This is potentially broken because of an internal change in Tensorflow")

# Override the saved\_model\_exporter with SaveAllmodelsexporter

self.\_saved\_model\_exporter = \_AllSavedModelsExporter(

name, input\_receiver\_fn\_map, backup\_checkpoints, assets\_extra, as\_text)