
cmsml

Release 0.1.2

CMS Machine Learning Group

Sep 02, 2021

CONTENTS

1	Table of contents	3
2	Docker images	7
3	Testing	9
4	Development	11
	Python Module Index	13
	Index	15



The documentation of this Python package is hosted on [readthedocs](#).

However, note that this documentation only covers the API and technical aspects of the package itself. Usage examples and further techniques for working with machine learning tools in CMS, alongside a collection of useful guidelines can be found in the [general CMS ML group documentation](#).

Click [here](#) to submit a feature suggestion!

TABLE OF CONTENTS

1.1 Command line tools

This document lists and describes the command line tools available in the cmsml package.

1.1.1 cmsml_open_tf_graph

```
> cmsml_open_tf_graph --help
usage: cmsml [-h] [--log-dir LOG_DIR] [--txt] [--binary]
             [--tensorboard-args TENSORBOARD_ARGS]
             graph_path
```

Takes a tensorflow graph that was previously saved to a protobuf file and opens a tensorboard server to visualize it.

positional arguments:

graph_path the path to the graph to open

optional arguments:

-h, --help show this **help** message and **exit**
--log-dir LOG_DIR, -l LOG_DIR
 the tensorboard logdir, temporary when not **set**
--txt, -t force reading the graph as text
--binary, -b force reading the graph as a binary
--tensorboard-args TENSORBOARD_ARGS, -a TENSORBOARD_ARGS
 optional arguments to pass to the tensorboard **command**

1.2 API Reference

1.2.1 cmsml.tensorflow

Classes, functions and tools for efficiently working with TensorFlow.

Functions:

<code>import_tf([log_level, autograph_verbosity])</code>	Imports TensorFlow and returns a 3-tuple containing the module itself, the v1 compatibility API (i.e.
<code>save_graph(path, obj[, ...])</code>	Extracts a TensorFlow graph from an object <i>obj</i> and saves it at <i>path</i> .
<code>load_graph(path[, create_session, ...])</code>	Reads a saved TensorFlow graph from <i>path</i> and returns it.
<code>write_graph_summary(graph, summary_dir, **kwargs)</code>	Writes the summary of a <i>graph</i> to a directory <i>summary_dir</i> using a <code>tf.summary.FileWriter</code> (v1) or <code>tf.summary.create_file_writer</code> (v2).

import_tf(*log_level*='WARNING', *autograph_verbosity*=3)

Imports TensorFlow and returns a 3-tuple containing the module itself, the v1 compatibility API (i.e. the TensorFlow module itself if v1 is the primarily installed version), and the package version as a 3-tuple containing strings. Example:

```
tf, tf1, tf_version = import_tf()
```

At some point in the future, when v1 support might get fully removed from TensorFlow 2 or higher, the second tuple element might be *None*.

The verbosity of logs printed by TensorFlow and AutoGraph can be controlled through *log_level* and *autograph_verbosity*.

save_graph(*path, obj, variables_to_constants=False, output_names=None, *args, **kwargs*)

Extracts a TensorFlow graph from an object *obj* and saves it at *path*. The graph is optionally transformed into a simpler representation with all its variables converted to constants when *variables_to_constants* is *True*. The saved file contains the graph as a protobuf. The accepted types of *obj* greatly depend on the available API versions.

When the v1 API is found (which is also the case when `tf.compat.v1` is available in v2), `Graph`, `GraphDef` and `Session` objects are accepted. However, when *variables_to_constants* is *True*, *obj* must be a session and *output_names* should refer to names of operations whose subgraphs are extracted (usually just one).

For TensorFlow v2, *obj* can also be a compiled keras model, or either a polymorphic or concrete function as returned by `tf.function`. Polymorphic functions either must have a defined input signature (`tf.function(input_signature=(...))`) or they must accept no arguments in the first place. See the TensorFlow documentation on [concrete functions](#) for more info.

args and *kwargs* are forwarded to `tf.train.write_graph` (v1) or `tf.io.write_graph` (v2).

load_graph(*path, create_session=None, session_kwargs=None, as_text=None*)

Reads a saved TensorFlow graph from *path* and returns it. When *create_session* is *True*, a session object (compatible with the v1 API) is created and returned as the second value of a 2-tuple. The default value of *create_session* is *True* when TensorFlow v1 is detected, and *False* otherwise. In case a session is created, *session_kwargs* are forwarded to the session constructor as keyword arguments when set. When *as_text* is either *True* or *None*, and the file extension is ".pbtxt" or ".pb.txt", the content of the file at *path* is expected to be a human-readable text file. Otherwise, it is read as a binary protobuf file. Example:

```
graph = load_graph("path/to/model.pb", create_session=False)

graph, session = load_graph("path/to/model.pb", create_session=True)
```

write_graph_summary(*graph, summary_dir, **kwargs*)

Writes the summary of a *graph* to a directory *summary_dir* using a `tf.summary.FileWriter` (v1) or `tf.summary.create_file_writer` (v2). This summary can be used later on to visualize the graph via tensor-

board. *graph* can be either a graph object or a path to a protobuf file. In the latter case, `load_graph()` is used and all *kwargs* are forwarded.

Note: When used with TensorFlow v1, eager mode must be disabled.

1.2.2 cmsml.keras

Keras callbacks, metrics, losses and other useful tools. If not mentioned otherwise, all objects are based on `tf.keras` rather than plain keras.

cmsml.keras.callbacks

Custom keras callbacks.

1.2.3 cmsml.util

Helpful functions and utilities.

Functions:

<code>is_lazy_iterable(obj)</code>	Returns whether <i>obj</i> is iterable lazily, such as generators, range objects, maps, etc.
<code>make_list(obj[, cast])</code>	Converts an object <i>obj</i> to a list and returns it.
<code>tmp_file([create, delete])</code>	Prepares a temporary file and opens a context yielding its path.
<code>tmp_dir([create, delete])</code>	Prepares a temporary directory and opens a context yielding its path.

`is_lazy_iterable(obj)`

Returns whether *obj* is iterable lazily, such as generators, range objects, maps, etc.

`make_list(obj, cast=True)`

Converts an object *obj* to a list and returns it. Objects of types *tuple* and *set* are converted if *cast* is *True*. Otherwise, and for all other types, *obj* is put in a new list.

`tmp_file(create=False, delete=True, **kwargs)`

Prepares a temporary file and opens a context yielding its path. When *create* is *True*, the file is created before the context is opened, and deleted upon closing if *delete* is *True*. All *kwargs* are forwarded to `tempfile.mkstemp()`.

`tmp_dir(create=True, delete=True, **kwargs)`

Prepares a temporary directory and opens a context yielding its path. When *create* is *True*, the directory is created before the context is opened, and deleted upon closing if *delete* is *True*. All *kwargs* are forwarded to `tempfile.mkdtemp()`.

DOCKER IMAGES

To use the cmsml package via docker, checkour our [DockerHub](#) which contains tags for several Python versions.

TESTING

The tests can be triggered with

```
python -m unittest tests
```

and in general, they should be run for Python 2.7, 3.7 and 3.8. To run tests in a docker container, do

```
# run the tests  
./tests/docker.sh cmsml/cmsml  
  
# or interactively by adding a flag "i" to the command  
./tests/docker.sh cmsml/cmsml i  
> python -m unittest tests
```

In addition, before pushing to the repository, [PEP 8](#) compatibility should be checked with [flake8](#)

```
./tests/lint.sh
```

or via using the docker container

```
# run the tests  
./tests/docker.sh cmsml/cmsml tests/lint.sh
```


DEVELOPMENT

- Source hosted at [GitHub](#)
- Report issues, questions, feature requests on [GitHub Issues](#)

PYTHON MODULE INDEX

C

`cmsml.keras`, 5
`cmsml.keras.callbacks`, 5
`cmsml.tensorflow`, 3
`cmsml.util`, 5

C

`cmsml.keras`
 module, 5
`cmsml.keras.callbacks`
 module, 5
`cmsml.tensorflow`
 module, 3
`cmsml.util`
 module, 5

I

`import_tf()` (in module `cmsml.tensorflow`), 4
`is_lazy_iterable()` (in module `cmsml.util`), 5

L

`load_graph()` (in module `cmsml.tensorflow`), 4

M

`make_list()` (in module `cmsml.util`), 5
module
 `cmsml.keras`, 5
 `cmsml.keras.callbacks`, 5
 `cmsml.tensorflow`, 3
 `cmsml.util`, 5

S

`save_graph()` (in module `cmsml.tensorflow`), 4

T

`tmp_dir()` (in module `cmsml.util`), 5
`tmp_file()` (in module `cmsml.util`), 5

W

`write_graph_summary()` (in module
 `cmsml.tensorflow`), 4