

Package ‘tensorflow’

January 17, 2018

Type Package

Title R Interface to 'TensorFlow'

Version 1.5

Description Interface to 'TensorFlow' <<https://www.tensorflow.org/>>, an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more 'CPUs' or 'GPUs' in a desktop, server, or mobile device with a single 'API'. 'TensorFlow' was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

License Apache License 2.0

URL <https://github.com/rstudio/tensorflow>

BugReports <https://github.com/rstudio/tensorflow/issues>

SystemRequirements TensorFlow (<https://www.tensorflow.org/>)

Encoding UTF-8

LazyData true

Depends R (>= 3.1)

Imports config, jsonlite (>= 1.2), processx, reticulate (>= 1.4),
tfruns (>= 1.0), utils, yaml, rstudioapi (>= 0.7)

Suggests testthat, keras, tfestimators

RoxygenNote 6.0.1

NeedsCompilation no

Author JJ Allaire [aut, cre],
RStudio [cph, fnd],
Yuan Tang [aut, cph],
Dirk Eddelbuettel [ctb, cph],

Nick Golding [ctb, cph],
 Google Inc. [ctb, cph] (Examples and Tutorials)

Maintainer JJ Allaire <jj@rstudio.com>

Repository CRAN

Date/Publication 2018-01-17 15:02:31 UTC

R topics documented:

evaluate	2
export_savedmodel	3
install_tensorflow	3
install_tensorflow_extras	4
parse_arguments	5
parse_flags	5
shape	6
tensorboard	7
tensorflow	8
tf	8
tfe_enable_eager_execution	9
train	10
train_and_evaluate	10
use_session_with_seed	11
view_savedmodel	12
Index	13

evaluate	<i>Evaluate a Model</i>
----------	-------------------------

Description

Evaluate a model object. See implementations in the [keras](#) and [tfestimators](#) packages.

Usage

```
evaluate(object, ...)
```

Arguments

object	An evaluatable R object.
...	Optional arguments passed on to implementing methods.

Implementations

- [keras](#)
- [tfestimators](#)

export_savedmodel	<i>Export a Saved Model</i>
-------------------	-----------------------------

Description

Serialize a model to disk. See implementations in the [keras](#) and [tfestimators](#) packages.

Usage

```
export_savedmodel(object, export_dir_base, ...)
```

Arguments

object	An R object.
export_dir_base	A string containing a directory in which to export the SavedModel.
...	Optional arguments passed on to implementing methods.

Value

The path to the exported directory, as a string.

Implementations

- [keras](#)
- [tfestimators](#)

install_tensorflow	<i>Install TensorFlow and it's dependencies</i>
--------------------	---

Description

Install TensorFlow and it's dependencies

Usage

```
install_tensorflow(method = c("auto", "virtualenv", "conda", "system"),  
  conda = "auto", version = "default", extra_packages = NULL,  
  restart_session = TRUE)
```

Arguments

method	Installation method. By default, "auto" automatically finds a method that will work in the local environment. Change the default to force a specific installation method. Note that the "virtualenv" method is not available on Windows (as this isn't supported by TensorFlow). Note also that since this command runs without privilege the "system" method is available only on Windows.
conda	Path to conda executable (or "auto" to find conda using the PATH and other conventional install locations).
version	TensorFlow version to install. Specify "default" to install the CPU version of the latest release. Specify "gpu" to install the GPU version of the latest release. You can also provide a full major.minor.patch specification (e.g. "1.1.0"), appending "-gpu" if you want the GPU version (e.g. "1.1.0-gpu"). Alternatively, you can provide the full URL to an installer binary (e.g. for a nightly binary).
extra_packages	Additional Python packages to install along with TensorFlow.
restart_session	Restart R session after installing (note this will only occur within RStudio).

```
install_tensorflow_extras
```

Install additional Python packages alongside TensorFlow

Description

Install additional Python packages alongside TensorFlow

Usage

```
install_tensorflow_extras/packages, conda = "auto")
```

Arguments

packages	Python packages to install
conda	Path to conda executable (or "auto" to find conda using the PATH and other conventional install locations). Only used when TensorFlow is installed within a conda environment.

Details

This function requires a version of TensorFlow previously installed via the `install_tensorflow()` function.

For virtualenv and conda installations, the specified packages will be installed into the "r-tensorflow" environment. For system installations on Windows the specified packages will be installed into the system package library.

parse_arguments	<i>Parse Command Line Arguments</i>
-----------------	-------------------------------------

Description

Parse command line arguments of the form `--key=value` and `--key value`. The values are assumed to be valid `yaml` and will be converted using `yaml.load()`.

Usage

```
parse_arguments(arguments = NULL)
```

Arguments

arguments	A vector of command line arguments. When <code>NULL</code> (the default), the command line arguments received by the current R process are used.
-----------	--

parse_flags	<i>Parse Configuration Flags for a TensorFlow Application</i>
-------------	---

Description

Parse configuration flags for a TensorFlow application. Use this to parse and unify the configuration(s) specified through a `flags.yml` configuration file, alongside other arguments set through the command line.

Usage

```
parse_flags(config = Sys.getenv("R_CONFIG_ACTIVE", unset = "default"),
            file = "flags.yml", arguments = commandArgs(TRUE))
```

Arguments

config	The configuration to use. Defaults to the active configuration for the current environment (as specified by the <code>R_CONFIG_ACTIVE</code> environment variable), or <code>default</code> when unset.
file	The configuration file to read.
arguments	The command line arguments (as a character vector) to be parsed.

Value

A named R list, mapping configuration keys to values.

Examples

```
## Not run:
# examine an example configuration file provided by tensorflow
file <- system.file("examples/config/flags.yml", package = "tensorflow")
cat(readLines(file), sep = "\n")

# read the default configuration
FLAGS <- tensorflow::parse_flags("default", file = file)
str(FLAGS)

# read the alternate configuration: note that
# the default configuration is inherited, but
# we override the 'string' configuration here
FLAGS <- tensorflow::parse_flags("alternate", file = file)
str(FLAGS)

# override configuration values using command
# line arguments (normally, these would be
# passed in through the command line invocation
# used to start the process)
FLAGS <- tensorflow::parse_flags(
  "alternate",
  file = file,
  arguments = c("--foo=1")
)
str(FLAGS)

## End(Not run)
```

 shape

Tensor shape

Description

Tensor shape

Usage

```
shape(...)
```

Arguments

... Tensor dimensions

tensorboard	<i>TensorBoard Visualization Tool</i>
-------------	---------------------------------------

Description

TensorBoard is a tool inspecting and understanding your TensorFlow runs and graphs.

Usage

```
tensorboard(log_dir, action = c("start", "stop"), host = "127.0.0.1",
  port = "auto",
  launch_browser = getOption("tensorflow.tensorboard.browser", interactive()),
  reload_interval = 5, purge_orphaned_data = TRUE)
```

Arguments

log_dir	Directories to scan for training logs. If this is a named character vector then the specified names will be used as aliases within TensorBoard.
action	Specify whether to start or stop TensorBoard (TensorBoard will be stopped automatically when the R session from which it is launched is terminated).
host	Host for serving TensorBoard
port	Port for serving TensorBoard. If "auto" is specified (the default) then an unused port will be chosen automatically.
launch_browser	Open a web browser for TensorBoard after launching. Defaults to TRUE in interactive sessions. When running under RStudio uses an RStudio window by default (pass a function e.g. <code>utils::browseURL()</code> to open in an external browser). Use the <code>tensorflow.tensorboard.browser</code> option to establish a global default behavior.
reload_interval	How often the backend should load more data.
purge_orphaned_data	Whether to purge data that may have been orphaned due to TensorBoard restarts. Disabling <code>purge_orphaned_data</code> can be used to debug data disappearance.

Details

When TensorBoard is passed a `logdir` at startup, it recursively walks the directory tree rooted at `logdir` looking for subdirectories that contain tfevents data. Every time it encounters such a subdirectory, it loads it as a new run, and the frontend will organize the data accordingly.

The TensorBoard process will be automatically destroyed when the R session in which it is launched exits. You can pass `action = "stop"` to manually terminate TensorBoard.

Value

URL for browsing TensorBoard (invisibly).

tensorflow

TensorFlow for R

Description

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.

Details

The **TensorFlow API** is composed of a set of Python modules that enable constructing and executing TensorFlow graphs. The tensorflow package provides access to the complete TensorFlow API from within R.

For additional documentation on the tensorflow package see <https://tensorflow.rstudio.com>

tf

Main TensorFlow module

Description

Interface to main TensorFlow module. Provides access to top level classes and functions as well as sub-modules (e.g. tf\$nn, tf\$contrib\$learn, etc.).

Usage

tf

Format

TensorFlow module

Examples

```
## Not run:
library(tensorflow)

hello <- tf$constant('Hello, TensorFlow!')
zeros <- tf$Variable(tf$zeros(shape(1L)))

sess <- tf$Session()
sess$run(tf$global_variables_initializer())

sess$run(hello)
```



```
sess$run(zeros)

## End(Not run)
```

```
tfe_enable_eager_execution
```

Enables, for the rest of the lifetime of this program, eager execution.

Description

If not called immediately on startup risks creating breakage and bugs.

Usage

```
tfe_enable_eager_execution(config = NULL, device_policy = c("explicit",
  "warn", "silent"))
```

Arguments

config	(Optional) A <code>tf\$ConfigProto()</code> protocol buffer with configuration options for the Context. Note that a lot of these options may be currently unimplemented or irrelevant when eager execution is enabled.
device_policy	(Optional) What policy to use when trying to run an operation on a device with inputs which are not on that device. Valid values: "explicit": raises an error if the placement is not correct. "warn": copies the tensors which are not on the right device but raises a warning. "silent": silently copies the tensors. This might hide performance problems.

Details

After eager execution is enabled, operations are executed as they are defined and tensors hold concrete values, and can be accessed as R matrices or arrays with `as.matrix()`, `as.array()`, `as.double()`, etc.

Examples

```
## Not run:

# load tensorflow and enable eager execution
library(tensorflow)
tfe_enable_eager_execution()

# create a random 10x10 matrix
x <- tf$random_normal(shape(10, 10))

# use it in R via as.matrix()
heatmap(as.matrix(x))

## End(Not run)
```

train	<i>Train a Model</i>
-------	----------------------

Description

Train a model object. See implementation in the [tfestimators](#) package.

Usage

```
train(object, ...)
```

Arguments

object	A trainable R object.
...	Optional arguments passed on to implementing methods.

Implementations

- [tfestimators](#)

train_and_evaluate	<i>Simultaneously Train and Evaluate a Model</i>
--------------------	--

Description

Train and evaluate a model object. See implementation in the [tfestimators](#) package.

Usage

```
train_and_evaluate(object, ...)
```

Arguments

object	An R object.
...	Optional arguments passed on to implementing methods.

Implementations

- [tfestimators](#)

use_session_with_seed *Use a session with a random seed*

Description

Set various random seeds required to ensure reproducible results. The provided seed value will establish a new random seed for R, Python, NumPy, and TensorFlow. GPU computations and CPU parallelism will also be disabled by default.

Usage

```
use_session_with_seed(seed, disable_gpu = TRUE, disable_parallel_cpu = TRUE,  
  quiet = FALSE)
```

Arguments

seed	A single value, interpreted as an integer
disable_gpu	TRUE to disable GPU execution (see <i>Parallelism</i> below).
disable_parallel_cpu	TRUE to disable CPU parallelism (see <i>Parallelism</i> below).
quiet	TRUE to suppress printing of messages.

Details

This function must be called at the very top of your script (i.e. immediately after `library(tensorflow)`, `library(keras)`, etc.). Any existing TensorFlow session is torn down via `tf$reset_default_graph()`.

This function takes all measures known to promote reproducible results from TensorFlow sessions, however it's possible that various individual TensorFlow features or dependent libraries escape its effects. If you encounter non-reproducible results please investigate the possible sources of the problem, contributions via pull request are very welcome!

Packages which need to be notified before and after the seed is set can register for the "tensorflow.on_before_use_session" and "tensorflow.on_use_session" hooks (see [setHook\(\)](#) for additional details on hooks).

Value

TensorFlow session object, invisibly

Parallelism

By default the `use_session_with_seed()` function disables GPU and CPU parallelism, since both can result in non-deterministic execution patterns (see <https://stackoverflow.com/questions/42022950/>). You can optionally enable GPU or CPU parallelism by setting the `disable_gpu` and/or `disable_parallel_cpu` parameters to `FALSE`.

Examples

```
## Not run:  
library(tensorflow)  
use_session_with_seed(42)  
  
## End(Not run)
```

view_savedmodel	<i>View a Saved Model</i>
-----------------	---------------------------

Description

View a serialized model from disk.

Usage

```
view_savedmodel(model_dir)
```

Arguments

model_dir The path to the exported model, as a string.

Value

URL for browsing TensorBoard (invisibly).

Index

*Topic **datasets**

tf, 8

as.array(), 9

as.double(), 9

as.matrix(), 9

evaluate, 2

export_savedmodel, 3

install_tensorflow, 3

install_tensorflow(), 4

install_tensorflow_extras, 4

keras, 2, 3

parse_arguments, 5

parse_flags, 5

setHook(), 11

shape, 6

tensorboard, 7

tensorflow, 8

tensorflow-package (tensorflow), 8

tf, 8

tfe_enable_eager_execution, 9

tfestimators, 2, 3, 10

train, 10

train_and_evaluate, 10

use_session_with_seed, 11

utils::browseURL(), 7

view_savedmodel, 12

yaml.load(), 5