Package: cloudml (via r-universe)

May 17, 2026

Title Interface to the Google Cloud Machine Learning Platform

Version 0.7.1.9000

Description Interface to the Google Cloud Machine Learning Platform <https://cloud.google.com/vertex-ai>, which provides cloud tools for training machine learning models.

Depends R (>= 3.3.0), tfruns (>= 1.3)

Imports config, jsonlite, packrat, processx, rprojroot, rstudioapi, tools, utils, withr, yaml

Suggests tensorflow (>= 1.4.2), keras (>= 2.1.2), knitr, rmarkdown, testthat

License Apache License 2.0

SystemRequirements Python (>= 2.7.0)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

URL https://github.com/rstudio/cloudml

BugReports https://github.com/rstudio/cloudml/issues

Config/pak/sysreqs libpng-dev python3

Repository https://rstudio.r-universe.dev

Date/Publication 2025-12-03 16:26:10 UTC

RemoteUrl https://github.com/rstudio/cloudml

RemoteRef HEAD

RemoteSha 73cc8891df0bb2a67608d223ae521cfc95c59071

Contents

`cloudml_deploy` *Deploy SavedModel to CloudML*

Description

Deploys a SavedModel to CloudML model for online predictions.

Usage

```
cloudml_deploy(
  export_dir_base,
  name,
  version = paste0(name, "_1"),
  region = NULL,
  config = NULL
)
```

Arguments

`export_dir_base`	A string containing a directory containing an exported SavedModels. Consider using `tensorflow::export_savedmodel()` to export this SavedModel.
`name`	The name for this model (required)
`version`	The version for this model. Versions start with a letter and contain only letters, numbers and underscores. Defaults to name_1

region	The region to be used to deploy this model.
config	A list, YAML or JSON configuration file as described https://cloud.google.com/vertex-ai.

See Also

cloudml_predict()

Other CloudML functions: cloudml_predict(), cloudml_train()

---

cloudml_predict *Perform Prediction over a CloudML Model.*

Description

Perform online prediction over a CloudML model, usually, created using cloudml_deploy()

Usage

```
cloudml_predict(instances, name, version = paste0(name, "_1"), verbose = FALSE)
```

Arguments

instances	A list of instances to be predicted. While predicting a single instance, list wrapping this single instance is still expected.
name	The name for this model (required)
version	The version for this model. Versions start with a letter and contain only letters, numbers and underscores. Defaults to name_1
verbose	Should additional information be reported?

See Also

cloudml_deploy()

Other CloudML functions: cloudml_deploy(), cloudml_train()

`cloudml_train` *Train a model using Cloud ML*

Description

Upload a TensorFlow application to Google Cloud, and use that application to train a model.

Usage

```
cloudml_train(
  file = "train.R",
  master_type = NULL,
  flags = NULL,
  region = NULL,
  config = NULL,
  collect = "ask",
  dry_run = FALSE
)
```

Arguments

`file`	File to be used as entrypoint for training.
`master_type`	Training master node machine type. "standard" provides a basic machine configuration suitable for training simple models with small to moderate datasets. See the documentation at https://cloud.google.com/vertex-ai/docs/reference/rest/v1/MachineSpec for details on available machine types.
`flags`	Named list with flag values (see `flags()`) or path to YAML file containing flag values.
`region`	The region to be used for training.
`config`	A list, `YAML` or `JSON` configuration file as described https://cloud.google.com/vertex-ai.
`collect`	Logical. If TRUE, collect job when training is completed (blocks waiting for the job to complete). The default (`"ask"`) will interactively prompt the user whether to collect the results or not.
`dry_run`	Triggers a local dry run over the deployment phase to validate packages and packing work as expected.

See Also

`job_status()`, `job_collect()`, `job_cancel()`

Other CloudML functions: `cloudml_deploy()`, `cloudml_predict()`

Examples

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

gcloud_install()
job <- cloudml_train("train.R")

## End(Not run)
```

`gcloud_init` *Initialize the Google Cloud SDK*

Description

Initialize the Google Cloud SDK

Usage

```
gcloud_init()
```

See Also

Other Google Cloud SDK functions: `gcloud_install()`, `gcloud_terminal()`

`gcloud_install` *Install the Google Cloud SDK*

Description

Installs the Google Cloud SDK which enables CloudML operations.

Usage

```
gcloud_install(update = TRUE)
```

Arguments

`update`	Attempt to update an existing installation.

See Also

Other Google Cloud SDK functions: `gcloud_init()`, `gcloud_terminal()`

Examples

```
## Not run:
library(cloudml)
gcloud_install()

## End(Not run)
```

`gcloud_terminal` *Create an RStudio terminal with access to the Google Cloud SDK*

Description

Create an RStudio terminal with access to the Google Cloud SDK

Usage

```
gcloud_terminal(command = NULL, clear = FALSE)
```

Arguments

`command`	Command to send to terminal
`clear`	Clear terminal buffer

Value

Terminal id (invisibly)

See Also

Other Google Cloud SDK functions: `gcloud_init()`, `gcloud_install()`

`gcloud_version` *Gcloud version*

Description

Get version of Google Cloud SDK components.

Usage

```
gcloud_version()
```

Value

a list with the version of each component.

gs_copy *Copy files to / from Google Storage*

Description

Use the `gsutil cp` command to copy data between your local file system and the cloud, copy data within the cloud, and copy data between cloud storage providers.

Usage

```
gs_copy(source, destination, recursive = FALSE, echo = TRUE)
```

Arguments

`source`	The file to be copied. This can be either a path on the local filesystem, or a Google Storage URI (e.g. `gs://[BUCKET_NAME]/[FILENAME.CSV]`).
`destination`	The location where the `source` file should be copied to. This can be either a path on the local filesystem, or a Google Storage URI (e.g. `gs://[BUCKET_NAME]/[FILENAME.CSV]`).
`recursive`	Boolean; perform a recursive copy? This must be specified if you intend on copying directories.
`echo`	Echo command output to console.

gs_data_dir *Google storage bucket path that syncs to local storage when not running on CloudML.*

Description

Refer to data within a Google Storage bucket. When running on CloudML the bucket will be read from directly. Otherwise, the bucket will be automatically synchronized to a local directory.

Usage

```
gs_data_dir(url, local_dir = "gs", force_sync = FALSE, echo = TRUE)
```

Arguments

`url`	Google Storage bucket URL (e.g. `gs://<your-bucket>`).
`local_dir`	Local directory to synchonize Google Storage bucket(s) to.
`force_sync`	Force local synchonization even if the data directory already exists.
`echo`	Echo command output to console.

Details

This function is suitable for use in TensorFlow APIs that accept gs:// URLs (e.g. TensorFlow datasets). However, many package functions accept only local filesystem paths as input (rather than gs:// URLs). For these cases you can the `gs_data_dir_local()` function, which will always synchronize gs:// buckets to the local filesystem and provide a local path interface to their contents.

Value

Path to contents of data directory.

See Also

`gs_data_dir_local()`

---

gs_data_dir_local *Get a local path to the contents of Google Storage bucket*

Description

Provides a local filesystem interface to Google Storage buckets. Many package functions accept only local filesystem paths as input (rather than gs:// URLs). For these cases the `gcloud_path()` function will synchronize gs:// buckets to the local filesystem and provide a local path interface to their contents.

Usage

```
gs_data_dir_local(url, local_dir = "gs", echo = FALSE)
```

Arguments

`url`	Google Storage bucket URL (e.g. `gs://<your-bucket>`).
`local_dir`	Local directory to synchonize Google Storage bucket(s) to.
`echo`	Echo command output to console.

Details

If you pass a local path as the `url` it will be returned unmodified. This allows you to for example use a training flag for the location of data which points to a local directory during development and a Google Cloud bucket during cloud training.

Value

Local path to contents of bucket.

Note

For APIs that accept gs:// URLs directly (e.g. TensorFlow datasets) you should use the `gs_data_dir()` function.

See Also

gs_data_dir()

---

gs_rsync *Synchronize content of two buckets/directories*

Description

The gs_rsync function makes the contents under destination the same as the contents under source, by copying any missing files/objects (or those whose data has changed), and (if the delete option is specified) deleting any extra files/objects. source must specify a directory, bucket, or bucket subdirectory.

Usage

```
gs_rsync(
  source,
  destination,
  delete = FALSE,
  recursive = FALSE,
  parallel = TRUE,
  dry_run = FALSE,
  options = NULL,
  echo = TRUE
)
```

Arguments

source	The file to be copied. This can be either a path on the local filesystem, or a Google Storage URI (e.g. gs://[BUCKET_NAME]/[FILENAME.CSV]).
destination	The location where the source file should be copied to. This can be either a path on the local filesystem, or a Google Storage URI (e.g. gs://[BUCKET_NAME]/[FILENAME.CSV]).
delete	Delete extra files under destination not found under source By default extra files are not deleted.
recursive	Causes directories, buckets, and bucket subdirectories to be synchronized recursively. If you neglect to use this option gs_rsync() will make only the top-level directory in the source and destination URLs match, skipping any subdirectories.
parallel	Causes synchronization to run in parallel. This can significantly improve performance if you are performing operations on a large number of files over a reasonably fast network connection.
dry_run	Causes rsync to run in "dry run" mode, i.e., just outputting what would be copied or deleted without actually doing any copying/deleting.

options	Character vector of additional command line options to the gsutil rsync command (as specified at https://cloud.google.com/storage/docs/gsutil/commands/rsync).
echo	Echo command output to console.

---

job_cancel *Cancel a job*

Description

Cancel a job.

Usage

```
job_cancel(job = "latest")
```

Arguments

job	Job name or job object. Pass "latest" to indicate the most recently submitted job.

See Also

Other job management functions: job_collect(), job_list(), job_status(), job_stream_logs(), job_trials()

---

job_collect *Collect job output*

Description

Collect the job outputs (e.g. fitted model) from a job. If the job has not yet finished running, job_collect() will block and wait until the job has finished.

Usage

```
job_collect(
  job = "latest",
  trials = "best",
  destination = "runs",
  timeout = NULL,
  view = interactive()
)
```

Arguments

job	Job name or job object. Pass "latest" to indicate the most recently submitted job.
trials	Under hyperparameter tuning, specifies which trials to download. Use "best" to download best trial, "all" to download all, or a vector of trials c(1,2) or 1.
destination	The destination directory in which model outputs should be downloaded. Defaults to runs.
timeout	Give up collecting job after the specified minutes.
view	View the job results after collecting it. You can also pass "save" to save a copy of the run report at tfruns.d/view.html

See Also

Other job management functions: job_cancel(), job_list(), job_status(), job_stream_logs(), job_trials()

---

job_list *List all jobs*

Description

List existing Google Cloud ML jobs.

Usage

```
job_list(
  filter = NULL,
  limit = NULL,
  page_size = NULL,
  sort_by = NULL,
  uri = FALSE
)
```

Arguments

filter	Filter the set of jobs to be returned.
limit	The maximum number of resources to list. By default, all jobs will be listed.
page_size	Some services group resource list output into pages. This flag specifies the maximum number of resources per page. The default is determined by the service if it supports paging, otherwise it is unlimited (no paging).
sort_by	A comma-separated list of resource field key names to sort by. The default order is ascending. Prefix a field with ~ for descending order on that field.
uri	Print a list of resource URIs instead of the default output.

See Also

Other job management functions: job_cancel(), job_collect(), job_status(), job_stream_logs(), job_trials()

job_status *Current status of a job*

Description

Get the status of a job, as an R list.

Usage

```
job_status(job = "latest")
```

Arguments

job	Job name or job object. Pass "latest" to indicate the most recently submitted job.

See Also

Other job management functions: job_cancel(), job_collect(), job_list(), job_stream_logs(), job_trials()

job_stream_logs *Show job log stream*

Description

Show logs from a running Cloud ML Engine job.

Usage

```
job_stream_logs(
  job = "latest",
  polling_interval = getOption("cloudml.stream_logs.polling", 5),
  task_name = NULL,
  allow_multiline_logs = FALSE
)
```

Arguments

job	Job name or job object. Pass "latest" to indicate the most recently submitted job.
polling_interval	Number of seconds to wait between efforts to fetch the latest log messages.
task_name	If set, display only the logs for this particular task.
allow_multiline_logs	Output multiline log messages as single records.

See Also

Other job management functions: job_cancel(), job_collect(), job_list(), job_status(), job_trials()

`job_trials` *Current trials of a job*

Description

Get the hyperparameter trials for job, as an R data frame

Usage

```
job_trials(x)
```

Arguments

x	Job name or job object.

See Also

Other job management functions: job_cancel(), job_collect(), job_list(), job_status(), job_stream_logs()

Index