Package: onnxr (via r-universe)

June 8, 2026

Type Package

Title Bindings to 'ONNX' Runtime

Version 0.1.2

Description Provides native access to the 'Open Neural Network Exchange' (ONNX) Runtime <https://onnxruntime.ai/>, which is a performant engine for running machine learning models that are saved to a standardized format. Rather than interfacing with 'ONNX' via 'Python', as in the official 'onnx' package, 'onnxr' directly interfaces with the runtime's 'C++' API via 'cpp11'. Models saved to '.onnx' files can be loaded and run on various backends, including CPUs and Apple's 'CoreML' library.

License MIT + file LICENSE

LinkingTo cpp11

Suggests jpeg, knitr, rmarkdown, testthat (>= 3.0.0)

URL https://corymccartan.com/onnxr/, https://github.com/CoryMcCartan/onnxr

BugReports https://github.com/CoryMcCartan/onnxr/issues

Language en-US

Encoding UTF-8

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Config/roxygen2/markdown TRUE

Config/testthat/edition 3

Config/build/compilation-database true

NeedsCompilation yes

Author Cory McCartan [aut, cre, cph], Caleb Carr [cph] (Author of 'nativeORT' package that is the basis for this package), Microsoft Corporation [cph] (Copyright holder of src/onnxruntime headers)

Maintainer Cory McCartan <mccartan@psu.edu>

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

Date/Publication 2026-06-08 20:00:10 UTC

RemoteUrl https://github.com/cran/onnxr

RemoteRef HEAD

RemoteSha 79b8dbf12f36c9ce3e5f677c809067f0c58affc9

Contents

`onnx_find_lib` *Find the ONNX Runtime shared library*

Description

Searches for the ONNX Runtime shared library in standard locations: the `ORT_ROOT` environment variable, common system library paths, the per-user install from `onnx_install()`, the Python `onnxruntime` package, and `pkg-config`.

Usage

```
onnx_find_lib()
```

Value

Full path to the shared library, or `NULL` if not found.

Examples

```
onnx_find_lib()
```

onnx_install *Install ONNX Runtime*

Description

Downloads pre-built ONNX Runtime binaries for the current platform and installs them to a per-user data directory. The library is loaded immediately after installation, so there is no need to restart R.

Usage

```
onnx_install(cuda = NULL)
```

Arguments

cuda	Whether to install the CUDA-enabled build for GPU acceleration. • NULL (default): auto-detect by checking for nvidia-smi on the system PATH. Installs the CUDA build if a GPU is found and the platform is supported (Linux x64 or Windows x64), otherwise falls back to the CPU build. • TRUE: force the CUDA build. Errors if the platform is not Linux x64 or Windows x64. • FALSE: always install the CPU-only build.

Value

Invisibly, the path to the installation directory.

onnx_is_installed *Check whether ONNX Runtime is available*

Description

Check whether ONNX Runtime is available

Usage

```
onnx_is_installed()
```

Value

TRUE if the ONNX Runtime shared library can be found in any of the standard search locations, FALSE otherwise.

Examples

```
onnx_is_installed()
```

onnx_is_loaded *Check whether ONNX Runtime is loaded*

Description

Check whether ONNX Runtime is loaded

Value

`TRUE` if the ONNX Runtime shared library has been loaded in the current R session, `FALSE` otherwise.

Examples

```
onnx_is_loaded()
```

onnx_model *Load an ONNX model*

Description

Loads an `.onnx` model file and creates a model object.

Usage

```
onnx_model(
  path,
  backend = c("cpu", "coreml", "cuda", "xnnpack", "openvino"),
  cache_dir = tools::R_user_dir("onnxr", "cache"),
  threads = 1L,
  opt_level = 99L
)
```

Arguments

`path`	Path to an `.onnx` model file.
`backend`	Execution backend. Available options depend on the platform and ORT build: • `"cpu"` — Default, available everywhere. • `"coreml"` — Apple Neural Engine + CPU (macOS/iOS only). • `"cuda"` — NVIDIA GPU (Linux x64 and Windows x64 only). Requires CUDA toolkit and the CUDA-enabled ORT build from `onnx_install(cuda = TRUE)`. • `"xnnpack"` — Optimized CPU kernels (mobile/embedded). Requires an ORT build with XNNPACK support (not provided by `onnx_install()`).

- "openvino" — Intel hardware acceleration. Requires OpenVINO installation and ORT build with OpenVINO EP (not provided by onnx_install()).

cache_dir	Optional directory for CoreML model cache. Set to NULL to disable caching.
threads	Number of threads. 0 uses all available; a positive integer sets a fixed thread count.
opt_level	Graph optimization level. 99 (default) enables all optimizations; 1 for basic only; 0 to disable.

Value

An "onnx_model" object (a named list) with model metadata and internal pointers used by onnx_run().

Examples

```
model_path <- system.file("extdata", "lm_iris.onnx", package = "onnxr")
if (onnx_is_loaded() && nzchar(model_path)) {
    sess <- onnx_model(model_path)
    sess
}
```

---

onnx_run *Run or predict from an ONNX model*

Description

Passes input through the model and returns the results. For models with a single output, returns the output array directly. For models with multiple outputs, returns a named list of arrays.

Usage

```
onnx_run(model, ..., simplify = FALSE)
```

Arguments

model	An "onnx_model" object created by onnx_model().
...	Input arrays, either as unnamed arguments (matched to model inputs by position) or as named arguments (matched by name). Each input must be a numeric or integer matrix/array with dimensions matching the model's expected input shape. For single-input models, a single array can be passed directly.
simplify	If TRUE, return the output array directly for single-output models instead of a length-1 named list.

Details

Handles conversion between R's column-major arrays and ONNX's row-major tensors, and between R's numeric types and the model's declared element types (float, double, int32, int64, bool). Note that int64 outputs are cast to doubles, which may lose precision for large integers.

Value

A named list of output arrays, or (if `simplify = TRUE` and the model has a single output) the output array directly.

Examples

```
model_path <- system.file("extdata", "lm_iris.onnx", package = "onnxr")
if (onnx_is_loaded() && nzchar(model_path)) {
    sess <- onnx_model(model_path)
    input <- as.matrix(iris[1:5, c("Sepal.Length", "Sepal.Width", "Petal.Length")])
    onnx_run(sess, input)
}
```

Index