

Package ‘churon’

September 27, 2026

Title 'ONNX Runtime' Integration

Version 0.1.13

Description Provides high-performance R bindings for 'ONNX Runtime'

[<https://onnxruntime.](https://onnxruntime.ai/)

[ai/>](https://onnxruntime.ai/), enabling efficient machine learning model inference. Written in 'Rust' for memory safety and speed, the package supports cross-platform model execution with multiple execution providers. Includes comprehensive error handling and validation, with bundled MNIST example model for immediate testing and prototyping.

Designed for production use with support for macOS (arm64), Linux (x64/arm64), and Windows (x64). Runtime libraries are downloaded on explicit request from

[<https://github.com/microsoft/onnxruntime/releases>](https://github.com/microsoft/onnxruntime/releases).

License MIT + file LICENSE

URL <https://github.com/mrchypark/churon>

BugReports <https://github.com/mrchypark/churon/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/rextendr/version 0.5.0

SystemRequirements Cargo (>= 1.88.0), rustc (>= 1.88.0); ONNX Runtime (>= 1.28.0, optional at runtime)

Depends R (>= 4.0.0)

Imports digest

Suggests testthat (>= 3.2.0)

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Chanyub Park [aut, cre] (ORCID:

[<https://orcid.org/0000-0001-6474-2570>](https://orcid.org/0000-0001-6474-2570)),

The authors of the dependency Rust crates [ctb] (See inst/AUTHORS for details),

ONNX Project contributors [ctb, cph] (Bundled MNIST model, Apache License 2.0; see inst/COPYRIGHTS),

Microsoft Corporation [cph] (ONNX Runtime, downloaded only on explicit user request)

Maintainer Chanyub Park <mrchypark@gmail.com>

Archs churon.so.dSYM

Contents

batch_process_data	2
check_onnx_runtime_available	3
get_onnx_runtime_info	3
install_onnx_runtime	3
onnx_example_models	4
onnx_example_session	4
onnx_input_info	5
onnx_model_path	5
onnx_output_info	6
onnx_providers	6
onnx_run	7
onnx_runtime_is_installed	7
onnx_runtime_lib_path	8
onnx_session	8
safe_onnx_run	9
safe_onnx_session	9
setup_onnx_runtime	10
\$.TensorInfo	10
Index	11

batch_process_data	<i>Optimize Session Performance Batch Process Data</i>
--------------------	--

Description

Process data in batches for memory efficiency with large datasets.

Usage

```
batch_process_data(session, data_list, batch_size = 32)
```

Arguments

session	An RSession object created by onnx_session()
data_list	A list of input data to process
batch_size	Number of items to process in each batch

Value

A list of results from batch processing

`check_onnx_runtime_available`*Check if ONNX Runtime is Available*

Description

This function checks if ONNX Runtime is properly configured and available.

Usage

```
check_onnx_runtime_available()
```

Value

Logical indicating whether ONNX Runtime is available

`get_onnx_runtime_info` *Get ONNX Runtime Information*

Description

This function returns information about the current ONNX Runtime configuration.

Usage

```
get_onnx_runtime_info()
```

Value

A list containing ONNX Runtime configuration information

`install_onnx_runtime` *Install ONNX Runtime*

Description

Download and install ONNX Runtime library for your platform. This is required before using the churon package if ONNX Runtime is not already installed on your system.

Usage

```
install_onnx_runtime(version = "1.29.0", quiet = FALSE, destdir, ...)
```

Arguments

<code>version</code>	Character string specifying the ONNX Runtime version to install. Defaults to "1.29.0". Supported versions are "1.28.0" and "1.29.0"; "latest" selects "1.29.0".
<code>quiet</code>	Logical. If TRUE, suppress download progress messages.
<code>destdir</code>	Required character string naming the installation directory.
<code>...</code>	Additional arguments passed to <code>download.file()</code>

Details

This function installs external software only when explicitly called. Supply `destdir` yourself; no installation directory is selected by default. The library is placed in `destdir/lib` and configured for this R session. In later sessions, set `ORT_DYLIB_PATH` to the installed library file before loading `churon`. Restart R before switching an already loaded runtime.

Value

Invisible TRUE on success, stops with error on failure.

<code>onnx_example_models</code>	<i>ONNX Example Models</i>
----------------------------------	----------------------------

Description

List available example models bundled with the package.

Usage

```
onnx_example_models()
```

Value

A named character vector with model names as names and paths as values

<code>onnx_example_session</code>	<i>ONNX Example Session</i>
-----------------------------------	-----------------------------

Description

Create a session with an example model.

Usage

```
onnx_example_session(model_name = "mnist", providers = NULL)
```

Arguments

<code>model_name</code>	Character string specifying the model name (default: "mnist")
<code>providers</code>	Optional execution providers

Value

An `RSession` object

onnx_input_info	<i>Get Input Information</i>
-----------------	------------------------------

Description

Retrieve information about model input tensors.

Usage

```
onnx_input_info(session)
```

Arguments

session An RSession object created by onnx_session()

Value

A list of TensorInfo objects containing input tensor metadata

Examples

```
if (check_onnx_runtime_available()) {  
  session <- onnx_example_session()  
  input_info <- onnx_input_info(session)  
  print(input_info)  
}
```

onnx_model_path	<i>Get Model Path</i>
-----------------	-----------------------

Description

Get the model path from a session.

Usage

```
onnx_model_path(session)
```

Arguments

session An RSession object created by onnx_session()

Value

Character string with the model path

Examples

```
if (check_onnx_runtime_available()) {  
  session <- onnx_example_session()  
  model_path <- onnx_model_path(session)  
  cat("Model path:", model_path, "\n")  
}
```

onnx_output_info	<i>Get Output Information</i>
------------------	-------------------------------

Description

Retrieve information about model output tensors.

Usage

```
onnx_output_info(session)
```

Arguments

session An RSession object created by onnx_session()

Value

A list of TensorInfo objects containing output tensor metadata

Examples

```
if (check_onnx_runtime_available()) {  
  session <- onnx_example_session()  
  output_info <- onnx_output_info(session)  
  print(output_info)  
}
```

onnx_providers	<i>Get Execution Providers</i>
----------------	--------------------------------

Description

Get the execution providers available for the session.

Usage

```
onnx_providers(session)
```

Arguments

session An RSession object created by onnx_session()

Value

A character vector of available execution providers

Examples

```
if (check_onnx_runtime_available()) {  
  session <- onnx_example_session()  
  providers <- onnx_providers(session)  
  cat("Available execution providers:", paste(providers, collapse = ", "), "\n")  
}
```

`onnx_run`*Run ONNX Inference*

Description

Execute inference on an ONNX model with input data.

Usage

```
onnx_run(session, inputs)
```

Arguments

<code>session</code>	An RSession object created by <code>onnx_session()</code>
<code>inputs</code>	A named list of input tensors. Names should match model input names.

Value

A named list of output tensors

Examples

```
if (check_onnx_runtime_available()) {  
  session <- onnx_example_session()  
  inputs <- list(Input3 = array(0, dim = c(1, 1, 28, 28)))  
  outputs <- onnx_run(session, inputs)  
}
```

`onnx_runtime_is_installed`*Check if ONNX Runtime is Installed*

Description

Check if the ONNX Runtime library is installed and available.

Usage

```
onnx_runtime_is_installed()
```

Value

Logical indicating whether ONNX Runtime is installed.

onnx_runtime_lib_path	<i>Get ONNX Runtime Library Path</i>
-----------------------	--------------------------------------

Description

Get the path to the ONNX Runtime library for the current platform.

Usage

```
onnx_runtime_lib_path()
```

Value

Character string with the library path.

onnx_session	<i>Create ONNX Session</i>
--------------	----------------------------

Description

Create a new ONNX Runtime session from a model file.

Usage

```
onnx_session(model_path, providers = NULL)
```

Arguments

model_path	Character string specifying the path to the ONNX model file
providers	Optional character vector specifying execution providers to use. Available providers: "cuda", "tensorrt", "directml", "onednn", "coreml", "cpu". If NULL, uses default provider priority.

Value

An RSession object for running inference

Examples

```
if (check_onnx_runtime_available()) {
  # Create session with default providers
  session <- onnx_example_session()

  # Create session with specific providers
  session <- onnx_example_session(providers = "cpu")
}
```

`safe_onnx_run`*Safe ONNX Run*

Description

Run inference with automatic error handling and monitoring.

Usage

```
safe_onnx_run(session, inputs, monitor_performance = FALSE)
```

Arguments

<code>session</code>	An RSession object created by <code>onnx_session()</code>
<code>inputs</code>	A named list of input tensors
<code>monitor_performance</code>	Logical indicating whether to monitor performance

Value

Result of inference or NULL if failed

`safe_onnx_session`*Safe ONNX Session Creation*

Description

Safe ONNX Session Creation

Usage

```
safe_onnx_session(model_path, providers = NULL)
```

Arguments

<code>model_path</code>	Character string specifying the path to the ONNX model file
<code>providers</code>	Optional character vector specifying execution providers

Details

Create an ONNX session with automatic error handling.

Value

An RSession object or NULL if creation fails

setup_onnx_runtime	<i>Setup ONNX Runtime Library Path</i>
--------------------	--

Description

This function sets up the ONNX Runtime library path for the current session. It looks for the ONNX Runtime library in the package installation directory.

Usage

```
setup_onnx_runtime()
```

Value

Logical indicating whether setup was successful

\$.TensorInfo	<i>S3 Methods for TensorInfo and RSession</i>
---------------	---

Description

These methods provide convenient access to object properties.

Usage

```
## S3 method for class 'TensorInfo'
x$name
```

Arguments

x	A TensorInfo or RSession object
name	Property name to access

Value

For name and data_type, a character scalar identifying the tensor or its element type. For shape, an integer vector of tensor dimensions (negative dimensions indicate dynamic sizes). For a method name, a function bound to the object; for an unknown name, NULL.

Index

- * **internal**
 - onnx_runtime_is_installed, [7](#)
 - onnx_runtime_lib_path, [8](#)
 - setup_onnx_runtime, [10](#)
- \$.TensorInfo, [10](#)
- batch_process_data, [2](#)
- check_onnx_runtime_available, [3](#)
- get_onnx_runtime_info, [3](#)
- install_onnx_runtime, [3](#)
- onnx_example_models, [4](#)
- onnx_example_session, [4](#)
- onnx_input_info, [5](#)
- onnx_model_path, [5](#)
- onnx_output_info, [6](#)
- onnx_providers, [6](#)
- onnx_run, [7](#)
- onnx_runtime_is_installed, [7](#)
- onnx_runtime_lib_path, [8](#)
- onnx_session, [8](#)
- safe_onnx_run, [9](#)
- safe_onnx_session, [9](#)
- setup_onnx_runtime, [10](#)