

Package ‘tabby’

October 15, 2026

Title Tidy Interfaces for Tabular Deep Learning

Version 0.0.1

Description Model definitions for fitting, tuning, and predicting with deep learning models for tabular data using the 'parsnip' and 'tidymodels' frameworks. Supported models include the 'TabPFN' foundation model of Hollmann et al. (2025) [<doi:10.1038/s41586-024-08328-6>](https://doi.org/10.1038/s41586-024-08328-6), 'TabICL' by Qu et al. (2025) [<doi:10.48550/arXiv.2502.05564>](https://doi.org/10.48550/arXiv.2502.05564), 'SAINT' by Somepalli et al. (2021) [<doi:10.48550/arXiv.2106.01342>](https://doi.org/10.48550/arXiv.2106.01342), and the tabular 'ResNet' of Gorishniy et al. (2021) [<doi:10.48550/arXiv.2106.11959>](https://doi.org/10.48550/arXiv.2106.11959), among others. Helper functions are also included to create space-filling grids for tuning parameters that are defined on a per-layer basis. These functions define the model interfaces; the computations are carried out by separate engine packages.

License MIT + file LICENSE

URL <https://github.com/tidymodels/tabby>, <https://tabby.tidymodels.org/>

BugReports <https://github.com/tidymodels/tabby/issues>

Depends parsnip (>= 1.6.1), R (>= 4.1.0)

Imports cli, dials (>= 1.4.4), dplyr, generics, hardhat, purrr, rlang (>= 1.1.0), tibble

Suggests brulee (>= 1.2.0), modeldata, spelling, testthat (>= 3.0.0), torch, withr

Encoding UTF-8

Language en-US

Config/Needs/website tidyverse/tidytemplate, brulee

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Max Kuhn [aut, cre] (ORCID: [<https://orcid.org/0000-0003-2402-136X>](https://orcid.org/0000-0003-2402-136X)),
Edgar Ruiz [aut],
Posit Software, PBC [cph, fnd] (ROR: [<https://ror.org/03wc8by49>](https://ror.org/03wc8by49))

Maintainer Max Kuhn <max@posit.co>
Repository CRAN
Date/Publication 2026-10-15 17:00:02 UTC

Contents

multi_predict._brulee_auto_int	2
neural_net_grid_space_filling	3
Index	5

multi_predict._brulee_auto_int
<i>Model predictions across many sub-models</i>

Description

Model predictions across many sub-models

Usage

```
## S3 method for class '`_brulee_auto_int`'  
multi_predict(object, new_data, type = NULL, epochs = NULL, ...)  
  
## S3 method for class '`_brulee_resnet`'  
multi_predict(object, new_data, type = NULL, epochs = NULL, ...)  
  
## S3 method for class '`_brulee_rln`'  
multi_predict(object, new_data, type = NULL, epochs = NULL, ...)
```

Arguments

object	A model fit .
new_data	A rectangular data object, such as a data frame.
type	A single character value or NULL. Possible values are: • regression: "numeric" • classification: "class", "prob" • censored regression: "survival", "time", "hazard", "linear_pred" • quantile regression: "quantile" • interval estimates: "conf_int", "pred_int" • other: "raw" When NULL, predict() will choose an appropriate value based on the model's mode.
epochs	An integer vector for the number of training epochs.
...	Optional arguments to pass to predict.model_fit(type = "raw") such as type.

Value

A tibble with the same number of rows as `new_data`. Its `.pred` column is a list of tibbles, each containing the predictions for the corresponding row of `new_data` across the requested values of epochs.

`neural_net_grid_space_filling`

Working with Tuning Parameters that are Lists

Description

`neural_net_grid_space_filling()` takes the number of layers in a neural network as an input, creates a space-filling design with the multidimensional layer parameter(s), and collapses the grid into list parameters that the model function needs. `expand_list_parameters()` is a convenience function that converts the list columns to a wide structure for printing, visualization, etc.

Usage

```
neural_net_grid_space_filling(
  wflow,
  num_layers = 3,
  size = 10,
  param_info = NULL,
  range = c(2, 50),
  collapse = TRUE
)

expand_list_parameters(x, pattern = "*")
```

Arguments

<code>wflow</code>	A model specification or <code>workflows::workflow()</code> object.
<code>num_layers</code>	A single integer for the number of layers containing hidden units.
<code>size</code>	The <i>minimum</i> grid size. It may be adjusted upwards to find a feasible design.
<code>param_info</code>	A <code>dials::parameters()</code> object or <code>NULL</code> . If none is given, a parameters set is derived from other arguments. Passing this argument can be useful when parameter ranges need to be customized.
<code>range</code>	A range for the number of hidden units in each layer.
<code>collapse</code>	A single logical for whether to collapse the parameters into list columns.
<code>x</code>	A data frame of grid points, some of which are list columns.
<code>pattern</code>	A regular expression pattern to select which list columns should be expanded to a wide format.

Value

`neural_net_grid_space_filling()` returns a tibble with grid points, some of which are list-columns containing integer vectors. `expand_list_parameters()` returns a tibble where the selected list columns are replaced by one integer column per layer.

Examples

```
rn_spec <-  
  tabular_resnet(hidden_units = tune(),  
                 bottleneck_units = tune(),  
                 penalty = tune())  
  
rn_grid <- neural_net_grid_space_filling(rn_spec)  
rn_grid  
  
rn_grid |> expand_list_parameters()
```

Index

dials::parameters(), [3](#)

expand_list_parameters
 (neural_net_grid_space_filling),
 [3](#)

model fit, [2](#)

multi_predict._brulee_auto_int, [2](#)

multi_predict._brulee_resnet
 (multi_predict._brulee_auto_int),
 [2](#)

multi_predict._brulee_rln
 (multi_predict._brulee_auto_int),
 [2](#)

neural_net_grid_space_filling, [3](#)

workflows::workflow(), [3](#)