Package: janus (via r-universe)

October 12, 2024

Type Package

Title Optimized Recommending System Based on 'tensorflow'

Version 1.0.0

Maintainer Giancarlo Vercellino <giancarlo.vercellino@gmail.com>

Description Proposes a coarse-to-fine optimization of a recommending system based on deep-neural networks using 'tensorflow'.

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.1

Imports keras (>= 2.9.0), tensorflow (>= 2.9.0), dplyr (>= 1.0.10), purrr (>= 0.3.4), forcats (>= 0.5.1), tictoc (>= 1.0.1), readr (>= 2.1.2), ggplot2 (>= 3.3.6), narray (>= 0.4.1.1), lubridate (>= 1.7.10), RcppAlgos (>= 2.6.0), Rmpfr (>= 0.8-7), Metrics (>= 0.1.4), StatRank (>= 0.0.6), hash (>= 2.2.6.2), reticulate (>= 1.26)

URL https://rpubs.com/giancarlo_vercellino/janus

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Giancarlo Vercellino [aut, cre, cph]

Repository CRAN

Date/Publication 2022-12-16 10:00:02 UTC

Contents

janus *janus*

Description

Coarse-to-fine optimization of a recommending system based on deep neural networks with Tensorflow/Keras back-end

Usage

```
janus(
  data,
  rating_label,
  rater_label,
  rated_label,
  task,
  skip_shortcut = FALSE,
  rater_embedding_size = c(8, 32),
  rated_embedding_size = c(8, 32),
  layers = c(1, 5),
 activations = c("elu", "selu", "relu", "sigmoid", "softmax", "softplus", "softsign",
  "tanh", "linear", "leaky_relu", "parametric_relu", "thresholded_relu", "swish",
    "gelu", "mish", "bent"),
  nodes = c(8, 512),
  regularization_L1 = c(0, 100),
  regularization_L2 = c(0, 100),
  dropout = c(0, 1),
  batch_size = 64,
  epochs = 10,
 optimizer = c("adam", "sgd", "adamax", "adadelta", "adagrad", "nadam", "rmsprop"),
  opt_metric = "bac",
  folds = 3,
  reps = 1,
  holdout = 0.1,
  n_steps = 3,
  n_samp = 10,
  offset = 0,
  n_top = 3,
  seed = 999,
  verbose = TRUE
)
```

Arguments

data	A data frame including at least three features: rating actor, rated item and rating value.
rating_label	String. Single label for the feature containing the rating values.

`rater_label` String. Single label for the feature containing the rating actors.

`rated_label` String. Single label for the feature containing the rated items.

`task` String. Available options are: "regr", for regression (when the rating value is numeric); "classif", for classification (when the rating value is a class or a factor).

`skip_shortcut` Logical. Option to add a skip shortcut to improve network performance in case of many layers. Default: FALSE.

`rater_embedding_size`
Integer. Output dimension for embedding the rating actors. Default: coarse-to-fine search (8 to 32).

`rated_embedding_size`
Integer. Output dimension for embedding the rated items. Default: coarse-to-fine search (8 to 32).

`layers` Positive integer. Number of layers for DNN. Default: coarse-to-fine search (1 to 5).

`activations` String. String vector with the activation functions for each layer. Default: coarse-to-fine search ("elu", "selu", "relu", "sigmoid", "softmax", "softplus", "softsign", "tanh", "linear", "leaky_relu", "parametric_relu", "thresholded_relu", "swish", "gelu", "mish", "bent").

`nodes` Positive integer. Integer vector with nodes for each layer. Default: coarse-to-fine search (8 to 512).

`regularization_L1`
Positive numeric. Value for L1 regularization of loss function. Default: coarse-to-fine search (0 to 100).

`regularization_L2`
Positive numeric. Value for L2 regularization of loss function. Default: coarse-to-fine search (0 to 100).

`dropout` Positive numeric. Value for dropout parameter at each layer (bounded between 0 and 1). Default: coarse-to-fine search (0 to 1).

`batch_size` Positive integer. Maximum batch size for training. Default: 64.

`epochs` Positive integer. Maximum number of forward and backward propagation. Default: 10.

`optimizer` String. Standard Tensorflow/Keras Optimization methods are available. Default: coarse-to-fine search ("adam", "sgd", "adamax", "adadelta", "adagrad", "nadam", "rmsprop").

`opt_metric` String. Error metric to track for the coarse-to-fine optimization. Different options: for regression, "rmse", "mae", "mdae", "mape", "smape", "rae", "rrse"; for classification, "bac", "avs", "avp", "avf", "kend", "ndcg".

`folds` Positive integer. Number of folds for repeated cross-validation. Default: 3.

`reps` Positive integer. Number of repetitions for repeated cross-validation. Default: 1.

`holdout` Positive numeric. Percentage of cases for holdout validation. Default: 0.1.

`n_steps` Positive integer. Number of phases for the coarse-to-fine optimization process (minimum 2). Default: 3.

n_samp	Positive integer. Number of sampled models per coarse-to-fine phase. Default: 10.
offset	Positive numeric. Percentage of expansion of numeric boundaries during the coarse-to-fine optimization. Default: 0.
n_top	Positive integer. Number of candidates selected during the coarse-to-fine phase. Default: 3.
seed	Positive integer. Seed value to control random processes. Default: 42.
verbose	Printing specific messages. Default: TRUE.

Value

This function returns a list including:

- pipeline:
- model:
  - configuration: DNN hyper-parameters (layers, activations, regularization_L1, regularization_L2, nodes, dropout)
  - model: Keras standard model description
  - recommend: function to use to recommend on rating actors
  - plot: Keras standard history plot
  - training_metrics: tracking of opt_metric across folds and repetitions
  - test_frame: testing set with the related predictions, including
  - testing_metrics: summary statistics for testing
- time_log

Author(s)

Maintainer: Giancarlo Vercellino <giancarlo.vercellino@gmail.com> [copyright holder]

Giancarlo Vercellino <giancarlo.vercellino@gmail.com>

See Also

Useful links:

- https://rpubs.com/giancarlo_vercellino/janus

Index