

# Package ‘tsfgrnn’

May 30, 2022

**Type** Package

**Title** Time Series Forecasting Using GRNN

**Version** 1.0.2

**Maintainer** Francisco Martinez <fmartin@ujaen.es>

**Description** A general regression neural network (GRNN) is a variant of a Radial Basis Function Network characterized by a fast single-pass learning. 'tsfgrnn' allows you to forecast time series using a GRNN model Francisco Martinez et al. (2019) <[doi:10.1007/978-3-030-20521-8\\_17](https://doi.org/10.1007/978-3-030-20521-8_17)> and Francisco Martinez et al. (2022) <[doi:10.1016/j.neucom.2021.12.028](https://doi.org/10.1016/j.neucom.2021.12.028)>. When the forecasting horizon is higher than 1, two multi-step ahead forecasting strategies can be used. The model built is autoregressive, that is, it is only based on the observations of the time series. You can consult and plot how the prediction was done. It is also possible to assess the forecasting accuracy of the model using rolling origin evaluation.

**License** GPL-2

**Encoding** UTF-8

**RoxygenNote** 7.2.0

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

**Imports** ggplot2, Rcpp

**VignetteBuilder** knitr

**URL** <https://github.com/franciscomartinezdelrio/tsfgrnn>

**BugReports** <https://github.com/franciscomartinezdelrio/tsfgrnn>

**LinkingTo** Rcpp

**Config/testthat/edition** 3

**NeedsCompilation** yes

**Author** Maria Pilar Frias-Bustamante [aut],  
Ana Maria Martinez-Rodriguez [aut],  
Antonio Conde-Sanchez [aut],  
Francisco Martinez [aut, cre]

**Repository** CRAN

**Date/Publication** 2022-05-30 12:10:02 UTC

## R topics documented:

autoplot.grnnForecast . . . . .	2
grnn_examples . . . . .	3
grnn_forecasting . . . . .	3
grnn_weights . . . . .	5
plot.grnnForecastRO . . . . .	5
plot_example . . . . .	6
predict.grnnForecast . . . . .	7
rolling_origin . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

autoplot.grnnForecast *Create a ggplot object from a grnnForecast object*

---

### Description

It uses a grnnForecast object to create a ggplot object that plots a time series and its forecast using GRNN regression.

### Usage

```
## S3 method for class 'grnnForecast'
autoplot(forecast, highlight = c("none", "points"))
```

### Arguments

forecast	The grnnForecast object.
highlight	A string value indicating what elements should be highlighted. Possible values are "none" and "points".

### Value

The ggplot object representing a plotting with the forecast.

### Examples

```
pred <- grnn_forecasting(USAccDeaths, h = 12, lags = 1:12, sigma = 50)
library(ggplot2)
autoplot(pred)
```

---

`grnn_examples`*Examples of a GRNN model*

---

**Description**

It shows the examples of the model associated to a `grnnForecast` object.

**Usage**

```
grnn_examples(forecast)
```

**Arguments**

`forecast`      A `grnnForecast` object.

**Value**

A matrix including the features and targets of the examples associated with the model of a `grnnForecast` object.

**Examples**

```
pred <- grnn_forecasting(ts(1:8), h = 1, lags = 1:2)
grnn_examples(pred)
```

---

`grnn_forecasting`*Time series forecasting using GRNN regression*

---

**Description**

It applies GRNN regression to forecast the future values of a time series. The lags used as autoregressive variables are set with the `lags` parameter. If the user does not set the lags, these values are selected automatically.

**Usage**

```
grnn_forecasting(  
  timeS,  
  h,  
  lags = NULL,  
  sigma = "ROLLING",  
  msas = c("recursive", "MIMO"),  
  transform = c("additive", "multiplicative", "none")  
)
```

**Arguments**

times	A numeric vector or time series of class <code>ts</code> .
h	A positive integer. Number of periods for forecasting.
lags	An integer vector in increasing order expressing the lags used as autoregressive variables. If <code>NULL</code> (the default) the lags are selected in a fast, heuristic way. It is also possible to use the values "FS" and "BE", in which case, the lags are selected using forward selection or backward elimination respectively. These techniques are feature selection approaches.
sigma	A positive real value or a character value. The smoothing parameter in GRNN regression. Two character values are possible, "ROLLING" (the default) and "FIXED", in which case the parameter is chosen using an optimization tool with rolling origin evaluation or fixed origin evaluation.
msas	A string indicating the Multiple-Step Ahead Strategy used when more than one value is predicted. It can be "MIMO" or "recursive" (the default).
transform	A character value indicating whether the training samples are transformed. If the time series has a trend it is recommended. By default is "additive" (additive transformation). It is also possible a multiplicative transformation or no transformation.

**Value**

An object of class "grnnForecast". The function `summary` can be used to obtain or print a summary of the results. An object of class "gnnForecast" is a list containing at least the following components:

call	the matched call.
msas	the Multi-Step Ahead Strategy.
prediction	a time series with the forecast.
model	an object of class "grnnModel" with the GRNN model

**References**

F. Martinez et al. (2022). "Strategies for time series forecasting with generalized regression neural networks", *Neurocomputing*, 491, pp. 509–521.

**Examples**

```
pred <- grnn_forecasting(USAccDeaths, h = 12, lags = 1:12)
plot(pred)
```

---

grnn_weights	<i>Training examples and their corresponding weights used in a prediction</i>
--------------	---

---

**Description**

It shows the input vector and the weights of the training examples used in a prediction associated with a "grnnForecast" object.

**Usage**

```
grnn_weights(forecast)
```

**Arguments**

forecast      A grnnForecast object.

**Value**

A list including the input vectors used in GRNN regression and the training examples, with their weights, used in the prediction.

**Examples**

```
pred <- grnn_forecasting(UKgas, h = 4, lags = 1:4, msas = "MIMO")
grnn_weights(pred)
```

---

plot.grnnForecastR0	<i>Plot the prediction for a test set</i>
---------------------	---

---

**Description**

It plots the forecast associated with a test set generated with the function [rolling\\_origin](#).

**Usage**

```
## S3 method for class 'grnnForecastR0'
plot(x, h = NULL, ...)
```

**Arguments**

x              the object obtained from a call to [rolling\\_origin](#).  
h              an integer. The forecasting horizon. If NULL (the default), the maximum forecasting horizon of all the test sets is used.  
...            Other plotting parameters to affect the plot.

**Value**

None

---

plot\_example

*Plot an example used in a prediction of a grnnForecast object*

---

**Description**

This function is useful to see how the forecast has been computed. An ordinal specifying the order of the weight has to be supplied and the function plots the training pattern associated with that ordinal.

**Usage**

```
plot_example(forecast, position, h = 1)
```

**Arguments**

forecast	The grnnForecast object.
position	An integer. It is an ordinal number indicating what training pattern to plot. For instance, if position is 1 it means that the training pattern with the greatest weight should be plotted. If position is 2 the training pattern with the second greatest weight is plotted and so on.
h	An integer. This value is only useful when the recursive strategy is being used. It indicates the forecasting horizon

**Value**

A ggplot object representing an example used in the prediction.

**Examples**

```
pred <- grnn_forecasting(USAccDeaths, h = 12, lags = 1:12, sigma = 50)
library(ggplot2)
plot_example(pred, 1)
```

---

predict.grnnForecast *Predict method for GRNN models for time series forecasting.*

---

### Description

Predicted values based on a GRNN model for time series forecasting.

### Usage

```
## S3 method for class 'grnnForecast'  
predict(object, h, ...)
```

### Arguments

object	a grnnForecast object obtained by a call to the <a href="#">grnn_forecasting</a> function.
h	an integer. The forecasting horizon.
...	further arguments passed to or from other methods.

### Details

If the models uses the MIMO strategy for multiple-step ahead prediction, the forecasting horizon is fixed to the model forecasting horizon.

### Value

a grnnForecast object with the prediction and information about the GRNN model, see the documentation of [grnn\\_forecasting](#) for the structure of grnnForecast objects.

### Examples

```
pred <- grnn_forecasting(UKgas, h = 4, msas = "MIMO")  
new_pred <- predict(pred, h = 4)  
print(new_pred$prediction)  
plot(new_pred) # To see a plot with the forecast
```

---

rolling\_origin *Assessing forecasting accuracy with rolling origin*

---

### Description

It uses the model and the time series associated with a grnnForecast object to assess the forecasting accuracy of the model using the last h values of the time series to build test sets applying a rolling origin evaluation.

**Usage**

```
rolling_origin(grnnf, h = NULL, rolling = TRUE)
```

**Arguments**

grnnf	A grnnForecast object.
h	A positive integer. The forecast horizon. If NULL (the default) the prediction horizon of the grnnForecast object is used.
rolling	A logical. If TRUE (the default), forecasting horizons from 1 to h are used. Otherwise, only horizon h is used.

**Details**

This function assesses the forecast accuracy of the model used by the grnnForecast object. It uses  $h$  different test and training sets. The first test set consists of the last  $h$  values of the time series (the training set is formed by the previous values). The next test set consists of the last  $h - 1$  values of the time series and so on (the last test set is formed by the last value of the time series).

**Value**

A list containing at least the following fields:

test_sets	a matrix containing the test sets used in the evaluation. Every row contains a different test set.
predictions	The predictions for the test sets.
errors	The errors for the test sets.
global_accu	Different measures of accuracy applied to all the errors.
h_accu	Different measures of accuracy applied to all the errors for every forecasting horizon.

**Examples**

```
pred <- grnn_forecasting(UKgas, h = 4, lags = 1:4)
ro <- rolling_origin(pred)
print(ro$global_accu)
```



# Index

`autoplot.grnnForecast`, 2

`grnn_examples`, 3

`grnn_forecasting`, 3, 7

`grnn_weights`, 5

`plot.grnnForecastRO`, 5

`plot_example`, 6

`predict.grnnForecast`, 7

`rolling_origin`, 5, 7

`summary`, 4