

Package ‘TSdeeplearning’

May 7, 2026

Type Package

Title Deep Learning Model for Time Series Forecasting

Version 1.0.1

Maintainer Ronit Jaiswal <ronitjaiswal2912@gmail.com>

Description Provides deep learning models for time series forecasting using Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). These models capture temporal dependencies and address vanishing gradient issues in sequential data. The package enables efficient forecasting for univariate time series. For methodological details see Jaiswal and co-authors (2022). <doi:10.1007/s00521-021-06621-3>.

Language en-US

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports tensorflow, keras, reticulate, tsutils, BiocGenerics, utils, graphics, magrittr

Depends R (>= 2.10)

NeedsCompilation no

Author Ronit Jaiswal [aut, cre],
Girish Kumar Jha [aut, ths, ctb],
Rajeev Ranjan Kumar [aut, ctb],
Kapil Choudhary [aut, ctb]

Repository CRAN

Date/Publication 2026-04-13 07:20:15 UTC

Contents

Data_Maize	2
GRU_ts	3

LST_ts	4
RNN_ts	5

Index	8
--------------	----------

Data_Maize	<i>Monthly International Maize Price Data</i>
------------	---

Description

Monthly international Maize price (Dollor per million ton) from January 2010 to June 2020.

Usage

```
data("Data_Maize")
```

Format

A time series data with 126 observations.

```
price a time series
```

Details

Dataset contains 126 observations of monthly international Maize price (Dollor per million ton). It is obtained from World Bank "Pink sheet".

Source

<https://www.worldbank.org/en/research/commodity-markets>

References

<https://www.worldbank.org/en/research/commodity-markets>

Examples

```
## Not run:
data("Data_Maize")
GRU_ts(Data_Maize)

## End(Not run)
```

GRU_ts

*Gated Recurrent Unit Model***Description**

The GRU function computes forecasted value with different forecasting evaluation criteria for gated recurrent unit model.

Usage

```
GRU_ts(xt, xtlag = 4, uGRU = 2, Drate = 0, nEpochs = 10,
Loss = "mse", AccMetrics = "mae", ActFn = "tanh",
Split = 0.8, Valid = 0.1)
```

Arguments

xt	Input univariate time series (ts) data.
xtlag	Lag of time series data.
uGRU	Number of unit in GRU layer.
Drate	Dropout rate.
nEpochs	Number of epochs.
Loss	Loss function.
AccMetrics	Metrics.
ActFn	Activation function.
Split	Index of the split point and separates the data into the training and testing datasets.
Valid	Validation set.

Details

The gated recurrent unit (GRU) was introduced by Cho et al.(2014). A GRU is part of a specific model of recurrent neural network that intends to use connections through a sequence of nodes to perform machine learning tasks associated with memory and clustering. Its internal structure is simpler and, therefore, it is also easier to train, as less calculation is required to upgrade the internal states. The update port controls the extent to which the state information from the previous moment is retained in the current state, while the reset port determines whether the current state should be combined with the previous information. Gated recurrent units help to adjust neural network input weights to solve the vanishing gradient problem that is a common issue with recurrent neural networks.

Value

TrainFittedValue	Training Fitted value for given time series data.
TestPredictedValue	Final forecasted value of the GRU model.
fcast_criteria	Different Forecasting evaluation criteria for GRU model.

References

Cho, K., Van Merriënboer, B., Bahdanau, D. and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

See Also

LSTM, RNN

Examples

```
## Not run:
data("Data_Maize")
GRU_ts(Data_Maize)

## End(Not run)
```

LST_ts

Long- Short Term Memory Model

Description

The LSTM function computes forecasted value with different forecasting evaluation criteria for long- short term memory model.

Usage

```
LSTM_ts(xt, xtlag = 4, uLSTM = 2, Drate = 0, nEpochs = 10,
Loss = "mse", AccMetrics = "mae", ActFn = "tanh",
Split = 0.8, Valid = 0.1)
```

Arguments

xt	Input univariate time series (ts) data.
xtlag	Lag of time series data.
uLSTM	Number of unit in LSTM layer.
Drate	Dropout rate.
nEpochs	Number of epochs.
Loss	Loss function.
AccMetrics	Metrics.
ActFn	Activation function.
Split	Index of the split point and separates the data into the training and testing datasets.
Valid	Validation set.

Details

Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) based RNN is designed to overcome the vanishing gradients problem while dealing with long term dependencies. In contrast to standard RNN, LSTM has this peculiar and unique inbuilt ability by maintaining a memory cell to determine which unimportant features should be forgotten and which important features should be remembered during the learning process (Jaiswal et al., 2022). An LSTM model analyses and captures both short-term and long-term temporal dependencies of a complex time series effectively due to its architecture of recurrent neural network and the memory function used in the hidden nodes.

Value

TrainFittedValue

Training Fitted value for given time series data.

TestPredictedValue

Final forecasted value of the LSTM model.

fcast_criteria Different Forecasting evaluation criteria for LSTM model.

References

Cho, K., Van Merriënboer, B., Bahdanau, D. and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

See Also

GRU, RNN

Examples

```
## Not run:
data("Data_Maize")
LSTM_ts(Data_Maize)

## End(Not run)
```

RNN_ts

Recurrent neural network Model

Description

The RNN function computes forecasted value with different forecasting evaluation criteria for recurrent neural network model.

Usage

```
RNN_ts(xt, xtlag = 4, uRNN = 2, Drate = 0, nEpochs = 10,
Loss = "mse", AccMetrics = "mae", ActFn = "tanh",
Split = 0.8, Valid = 0.1)
```

Arguments

<code>xt</code>	Input univariate time series (ts) data.
<code>xtlag</code>	Lag of time series data.
<code>uRNN</code>	Number of unit in RNN layer.
<code>Drate</code>	Dropout rate.
<code>nEpochs</code>	Number of epochs.
<code>Loss</code>	Loss function.
<code>AccMetrics</code>	Metrics.
<code>ActFn</code>	Activation function.
<code>Split</code>	Index of the split point and separates the data into the training and testing datasets.
<code>Valid</code>	Validation set.

Details

Recurrent neural networks (RNNs) (Rumelhart 1986) add the explicit handling of order between observations when learning a mapping function from inputs to outputs. RNNs actually process single elements of any input sequence at a particular time, and maintain a ‘state vector’ in their hidden units. Nevertheless, when the interval of data dependencies increases, the standard RNNs tend to suffer increasingly heavily from the problem of either vanishing gradient or exploding gradient (Bengio et al. 1994; Lin et al. 1996).

Value

<code>TrainFittedValue</code>	Training Fitted value for given time series data.
<code>TestPredictedValue</code>	Final forecasted value of the RNN model.
<code>fcast_criteria</code>	Different Forecasting evaluation criteria for RNN model.

References

- Bengio et al. 1994; Lin Sagheer A, Kotb M (2019) Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing* 323: 203–213.
- Rumelhart DE (1986) Learning internal representations by error propagation. In: *Parallel distributed processing: Explorations in the microstructure of cognition*. pp 318–362.
- Jha, G. K. and Sinha, K. (2014). Time-delay neural networks for time series prediction: An application to the monthly wholesale price of oilseeds in India. *Neural Computing and Applications*, 24(3–4), 563–571.
- Jaiswal, R., Jha, G. K., Kumar, R. R. and Choudhary, K. (2022). Deep long short-term memory based model for agricultural price forecasting. *Neural Computing and Applications*, 34(6), 4661–4676.

See Also

LSTM, GRU

Examples

```
## Not run:  
data("Data_Maize")  
RNN_ts(Data_Maize)  
  
## End(Not run)
```

Index

- * **GRU**

- GRU_ts, 3

- * **LSTM**

- LST_ts, 4

- * **RNN**

- RNN_ts, 5

- * **datasets**

- Data_Maize, 2

Data_Maize, 2

GRU_ts, 3

LST_ts, 4

LSTM_ts (LST_ts), 4

RNN_ts, 5