

# Package: survnet (via r-universe)

November 1, 2024

**Type** Package

**Title** Artificial neural networks for survival analysis

**Version** 0.0.5

**Date** 2018-11-12

**Author** Marvin N. Wright

**Maintainer** Marvin N. Wright <cran@wrig.de>

**Description** Artificial neural networks for survival analysis

**License** MIT + file LICENSE

**Depends** survival, keras

**Imports** magrittr

**Suggests** testthat

**RoxygenNote** 6.1.0

**Repository** <https://bips-hb.r-universe.dev>

**RemoteUrl** <https://github.com/bips-hb/survnet>

**RemoteRef** HEAD

**RemoteSha** 99ac8a5a0071dd770195b0c8dceb7eea01fc080f

## Contents

convert_surv_cens . . . . .	2
loss_cif_loglik . . . . .	2
predict.survnet . . . . .	3
survnet . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

convert\_surv\_cens      *Create binary response matrix for survival data*

---

**Description**

Creates matrix with at-risk and event information. Format: (S\_1, ..., S\_K, E\_1, ..., E\_K). Dimensions: obs X 2\*causes\*time.

**Usage**

```
convert_surv_cens(time, status, breaks, num_causes)
```

**Arguments**

time	Survival time
status	Censoring indicator: 0 for censored observations, positive values for events.
breaks	Right interval limits for discrete survival time.
num_causes	Number of competing risks.

**Value**

Binary response matrix.

---

loss\_cif\_loglik      *Cumulative incidence log-likelihood*

---

**Description**

Likelihood of parametric inference for the cumulative incidence functions as defined by Jeong & Fine 2006. Also used by Lee et al. 2018.

**Usage**

```
loss_cif_loglik(num_intervals, num_causes = 1)
```

**Arguments**

num_intervals	Number of time intervals
num_causes	Number of causes for competing risks

**Details**

Data structure:

`y_true` True survival: Matrix with at-risk and event information. Format: (S\_1, ..., S\_K, E\_1, ..., E\_K). Dimensions: obs X 2\*causes\*time.

`y_pred` Network output: One probability for each time and cause. Format: (y\_11, ..., y\_T1, ..., y\_TK). Dimensions: obs X causes\*time.

**Value**

Negative log-likelihood

**References**

- Jeong, J. & Fine, J. (2006). Direct parametric inference for the cumulative incidence function. *J R Stat Soc Ser C Appl Stat* 55:187-200. <https://doi.org/10.1111/j.1467-9876.2006.00532.x>.
- Lee, C., Zame, W.R., Yoon, J. & van der Shaar, M. (2018). DeepHit: A deep learning approach to survival analysis with competing risks. AAAI 2018. [http://medianetlab.ee.ucla.edu/papers/AAAI\\_2018\\_DeepHit](http://medianetlab.ee.ucla.edu/papers/AAAI_2018_DeepHit).

---

predict.survnet	<i>survnet prediction</i>
-----------------	---------------------------

---

**Description**

survnet prediction

**Usage**

```
## S3 method for class 'survnet'
predict(object, newdata, cause = NULL, ...)
```

**Arguments**

object	survnet object
newdata	New data predictors: matrix, array or data.frame.
cause	Select cause for competing risks, NULL returns list of all causes.
...	Further arguments passed to or from other methods.

**Value**

Cumulative incidence function of selected or all causes.

**Description**

Artificial neural networks for survival analysis

**Usage**

```
survnet(y, x, breaks, units = c(3, 5), units_rnn = c(4, 6),
  units_causes = c(3, 2), epochs = 100, batch_size = 16,
  validation_split = 0.2, loss = loss_cif_loglik,
  activation = "tanh", rnn_type = "LSTM", skip = TRUE,
  dropout = rep(0, length(units)), dropout_rnn = rep(0,
  length(units_rnn)), dropout_causes = rep(0, length(units_causes)),
  l2 = rep(0, length(units)), l2_rnn = rep(0, length(units_rnn)),
  l2_causes = rep(0, length(units_causes)),
  optimizer = optimizer_rmsprop(lr = 0.001), verbose = 2)
```

**Arguments**

y	Survival outcome: matrix, data.frame or Surv() object.
x	Predictors: matrix, data.frame or array (time-series). Also accepts a list of matrix/data.frame and array for both time-constant and time-series predictors.
breaks	Right interval limits for discrete survival time.
units	Vector of units, each specifying the number of units in one hidden layer.
units_rnn	Vector of units for recurrent layers.
units_causes	Vector of units for cause-specific layers (competing risks only). Either a vector (will be repeated for each cause) or a list of vectors with layers for each cause.
epochs	Number of epochs to train the model.
batch_size	Number of samples per gradient update.
validation_split	Fraction in [0,1] of the training data to be used as validation data.
loss	Loss function.
activation	Activation function.
rnn_type	Type of RNN layers. Either "LSTM" (default), "GRU", "CUDA_LSTM" or "CUDA_GRU".
skip	Add skip connection from input and RNN layers to cause-specific layers.
dropout	Vector of dropout rates after each hidden layer. Use 0 for no dropout (default).
dropout_rnn	Vector of dropout rates after each recurrent layer. Use 0 for no dropout (default).
dropout_causes	Vector of dropout rates after each cause-specific layer. Use 0 for no dropout (default).

l2	Vector of L2 regularization factors for each hidden layer. Use 0 for no regularization (default).
l2_rnn	Vector of L2 regularization factors for each recurrent layer. Use 0 for no regularization (default).
l2_causes	Vector of L2 regularization factors for each cause-specific layer. Use 0 for no regularization (default).
optimizer	Name of optimizer or optimizer instance.
verbose	Verbosity mode (0 = silent, 1 = progress bar, 2 = one line per epoch).

**Value**

Fitted model.

**Examples**

```
library(survival)
library(survnet)

# Survival data
y <- veteran[, c(3, 4)]
x <- veteran[, c(-2, -3, -4)]
x <- data.frame(lapply(x, scale))
breaks <- c(1, 50, 100, 200, 500, 1000)

# Fit simple model
fit <- survnet(y = y, x = x, breaks = breaks)
plot(fit$history)
```

# Index

`convert_surv_cens`, [2](#)

`loss_cif_loglik`, [2](#)

`predict.survnet`, [3](#)

`survnet`, [4](#)