Package: rvflnet (via r-universe)

May 23, 2026

Title Random Vector Functional Link Networks with glmnet Backend

Version 0.1.0

Description Implements Random Vector Functional Link (RVFL) networks using glmnet for elastic-net regularized output layer training. Supports Gaussian, uniform, and Sobol random projections with various activation functions.

Depends R (>= 3.5.0)

Imports glmnet, randtoolbox

Suggests testthat, knitr, rmarkdown, MASS, survival

License GPL-2

LazyData true

RoxygenNote 7.3.2

Encoding UTF-8

VignetteBuilder knitr

Repository https://techtonique.r-universe.dev

Date/Publication 2026-04-22 21:40:27 UTC

RemoteUrl https://github.com/thierrymoudiki/rvflnet

RemoteRef HEAD

RemoteSha e353a0e6cba00e935bfabe9a1524084d186d985e

Contents

`.apply_scaling` *Apply scaling to new data*

Description

Apply scaling to new data

Usage

```
.apply_scaling(x, center, scale_vec)
```

Arguments

`x`	input matrix
`center`	centering vector
`scale_vec`	scaling vector

Value

scaled matrix

.compute_scaling *Compute scaling parameters from data*

Description

Compute scaling parameters from data

Usage

```
.compute_scaling(x, scale_input = TRUE)
```

Arguments

x	input matrix
scale_input	logical, whether to scale

Value

list with center and scale vectors

.generate_random_weights *Generate random weights for RVFL*

Description

Internal helper for generating random projection matrix

Usage

```
.generate_random_weights(p, n_hidden, W_type, seed)
```

Arguments

p	number of input features
n_hidden	number of hidden units
W_type	type of random weights ("gaussian", "sobol", "uniform")
seed	random seed for reproducibility

Value

random weight matrix of dimension p x n_hidden

.prepare_rvfl_data *Prepare RVFL design matrix*

Description

Internal helper for consistent preprocessing

Usage

```
.prepare_rvfl_data(
  x,
  W,
  center = NULL,
  scale_vec = NULL,
  scale_input = TRUE,
  activation,
  include_original = TRUE
)
```

Arguments

x	input matrix
W	random weight matrix
center	centering vector (if NULL, compute from data)
scale_vec	scaling vector (if NULL, compute from data)
scale_input	logical, whether to scale inputs
activation	activation function
include_original	logical, whether to include original features

Value

list with Z (design matrix) and preprocessing parameters

.rvfl_features *Generate RVFL random features*

Description

Internal helper to build the RVFL design matrix.

Usage

```
.rvfl_features(x, W, center, scale_vec, activation)
```

Arguments

x	numeric matrix of predictors
W	random projection matrix
center	centering vector
scale_vec	scaling vector
activation	activation function name or custom function

Value

augmented design matrix cbind(x, hidden features)

---

best_lambda *Extract best lambda from CV object*

Description

Extract best lambda from CV object

Usage

```
best_lambda(model, which = c("min", "1se"))
```

Arguments

model	cv.rvflnet object
which	which lambda to return ("min" or "1se")

Value

numeric lambda value

---

coef.cv.rvflnet *Extract coefficients from cross-validated RVFLNet model*

Description

Extract coefficients from cross-validated RVFLNet model

Usage

```
## S3 method for class 'cv.rvflnet'
coef(object, s = "lambda.min", ...)
```

Arguments

object	cv.rvflnet object
s	lambda value ("lambda.min", "lambda.1se", or numeric)
...	additional arguments passed to coef.glmnet

Value

coefficient matrix. See coef.rvflnet for interpretation notes.

Examples

```
## Not run:
coef(cv_model, s = "lambda.min")

## End(Not run)
```

---

coef.rvflnet *Extract coefficients from RVFLNet model*

Description

Extract coefficients from RVFLNet model

Usage

```
## S3 method for class 'rvflnet'
coef(object, s = NULL, ...)
```

Arguments

object	rvflnet object
s	lambda value (passed to glmnet)
...	additional arguments passed to coef.glmnet

Value

coefficient matrix. NOTE: Coefficients apply to the augmented feature space [original features | random hidden features]. The first p coefficients correspond to original inputs, remaining n_hidden to random features. These are NOT directly interpretable as "feature importance" in the original input space due to the nonlinear random projection.

Examples

```
## Not run:
coef(model, s = "lambda.min")

## End(Not run)
```

`cv.rvflnet` *Cross-validated RVFLNet model*

Description

Fits RVFL features then applies cv.glmnet.

Usage

```
cv.rvflnet(
  x,
  y,
  n_hidden = 200L,
  activation = c("sigmoid", "tanh", "relu", "identity"),
  W_type = c("gaussian", "uniform", "sobol"),
  seed = 1,
  scale = TRUE,
  include_original = TRUE,
  store_y = FALSE,
 family = c("gaussian", "binomial", "poisson", "multinomial", "cox", "mgaussian"),
  ...
)
```

Arguments

`x`	design matrix
`y`	response vector
`n_hidden`	number of hidden units
`activation`	activation function
`W_type`	random feature type ("gaussian", "uniform", "sobol")
`seed`	random seed
`scale`	logical scaling
`include_original`	logical, whether to include original features
`store_y`	logical, whether to store y in model
`family`	response type
`...`	additional arguments passed to glmnet::cv.glmnet

Value

object of class "cv.rvflnet"

Examples

```
## Not run:
x <- matrix(rnorm(100*5), 100, 5)
y <- x[,1] + sin(x[,2]) + rnorm(100, 0, 0.1)
cv_model <- cv.rvflnet(x, y, n_hidden = 50, nfolds = 5)
plot(cv_model)
predict(cv_model, newx = x[1:10,])

## End(Not run)
```

`fitted.cv.rvflnet` *Extract fitted values from cross-validated RVFLNet model*

Description

Extract fitted values from cross-validated RVFLNet model

Usage

```
## S3 method for class 'cv.rvflnet'
fitted(object, s = "lambda.min", ...)
```

Arguments

`object`	cv.rvflnet object
`s`	lambda value
`...`	additional arguments

Value

fitted values

`fitted.rvflnet` *Extract fitted values from RVFLNet model*

Description

Extract fitted values from RVFLNet model

Usage

```
## S3 method for class 'rvflnet'
fitted(object, s = NULL, ...)
```

Arguments

object	rvflnet object
s	lambda value
...	additional arguments

Value

fitted values

---

get_activation *Extract activation function*

Description

Extract activation function

Usage

```
get_activation(model)
```

Arguments

model	rvflnet or cv.rvflnet object

Value

character

---

get_weight_type *Extract weight type*

Description

Extract weight type

Usage

```
get_weight_type(model)
```

Arguments

model	rvflnet or cv.rvflnet object

Value

character

is.cv.rvflnet *Check if object is a CV-RVFLNet model*

Description

Check if object is a CV-RVFLNet model

Usage

```
is.cv.rvflnet(x)
```

Arguments

x	object to check

Value

logical

is.rvflnet *Check if object is an RVFLNet model*

Description

Check if object is an RVFLNet model

Usage

```
is.rvflnet(x)
```

Arguments

x	object to check

Value

logical

n_hidden *Extract number of hidden units*

Description

Extract number of hidden units

Usage

```
n_hidden(model)
```

Arguments

model	rvflnet or cv.rvflnet object

Value

integer

plot.cv.rvflnet *Plot CV curve for RVFLNet*

Description

Plot CV curve for RVFLNet

Usage

```
## S3 method for class 'cv.rvflnet'
plot(x, ...)
```

Arguments

x	cv.rvflnet object
...	passed to plot.glmnet

Examples

```
## Not run:
plot(cv_model)

## End(Not run)
```

plot.rvflnet *Plot RVFLNet model*

Description

Generic plot method for rvflnet objects

Usage

```
## S3 method for class 'rvflnet'
plot(x, type = c("coef"), ...)
```

Arguments

x	rvflnet object
type	type of plot ("coef" for coefficient path)
...	additional arguments passed to plot functions

predict.cv.rvflnet *Predict method for cross-validated RVFLNet*

Description

Predict method for cross-validated RVFLNet

Usage

```
## S3 method for class 'cv.rvflnet'
predict(object, newx = NULL, s = "lambda.min", type = "response", ...)
```

Arguments

object	cv.rvflnet model
newx	new data matrix (if NULL, returns fitted values if available)
s	lambda value ("lambda.min", "lambda.1se", or numeric)
type	type of prediction ("response", "coefficients", etc.)
...	additional arguments passed to predict.glmnet

Value

predictions

Examples

```
## Not run:
predictions <- predict(cv_model, newx = x_test, s = "lambda.min")

## End(Not run)
```

`predict.rvflnet` *Predict method for RVFLNet*

Description

Predict method for RVFLNet

Usage

```
## S3 method for class 'rvflnet'
predict(object, newx = NULL, s = NULL, type = "response", ...)
```

Arguments

`object`	rvflnet model
`newx`	new data matrix (if NULL, returns fitted values if available)
`s`	lambda value (passed to glmnet)
`type`	type of prediction ("response", "coefficients", etc.)
`...`	additional arguments passed to predict.glmnet

Value

predictions

Examples

```
## Not run:
predictions <- predict(model, newx = x_test, s = "lambda.min")

## End(Not run)
```

print.cv.rvflnet *Print cross-validated RVFLNet model*

Description

Print cross-validated RVFLNet model

Usage

```
## S3 method for class 'cv.rvflnet'
print(x, ...)
```

Arguments

x	cv.rvflnet object
...	ignored

print.rvflnet *Print RVFLNet model*

Description

Print RVFLNet model

Usage

```
## S3 method for class 'rvflnet'
print(x, ...)
```

Arguments

x	rvflnet object
...	ignored

residuals.cv.rvflnet *Residuals method for cross-validated RVFLNet*

Description

Residuals method for cross-validated RVFLNet

Usage

```
## S3 method for class 'cv.rvflnet'
residuals(object, s = "lambda.min", type = "response", ...)
```

Arguments

object	cv.rvflnet object
s	lambda value ("lambda.min", "lambda.1se", or numeric)
type	type of residuals
...	additional arguments

Value

residuals vector

residuals.rvflnet *Residuals method for RVFLNet*

Description

Residuals method for RVFLNet

Usage

```
## S3 method for class 'rvflnet'
residuals(object, s = NULL, type = "response", ...)
```

Arguments

object	rvflnet object
s	lambda value
type	type of residuals ("response", "deviance", etc.)
...	additional arguments

Value

residuals vector

Examples

```
## Not run:
model <- rvflnet(x, y, store_y = TRUE)
residuals(model, s = "lambda.min")

## End(Not run)
```

`rvflnet` *Random Vector Functional Link Network (glmnet backend)*

Description

Fits an RVFL model using glmnet on augmented random feature space.

Usage

```
rvflnet(
  x,
  y,
  n_hidden = 200L,
  activation = c("sigmoid", "tanh", "relu", "identity"),
  W_type = c("gaussian", "uniform", "sobol"),
  seed = 1,
  scale = TRUE,
  include_original = TRUE,
  store_y = FALSE,
 family = c("gaussian", "binomial", "poisson", "multinomial", "cox", "mgaussian"),
  ...
)
```

Arguments

`x`	design matrix
`y`	response vector
`n_hidden`	number of random hidden units (must be > 0)
`activation`	activation function ("tanh", "relu", "sigmoid", "identity", or custom function)
`W_type`	type of random weights ("gaussian", "uniform", "sobol")
`seed`	random seed for reproducibility
`scale`	logical, whether to standardize inputs
`include_original`	logical, whether to include original features (default TRUE)
`store_y`	logical, whether to store y in model (for residuals, default FALSE)
`family`	response type ("gaussian", "binomial", "poisson", "multinomial", "cox", "mgaussian"). For family = "cox", y must be a Surv object from the survival package.
`...`	additional arguments passed to glmnet::glmnet

Value

object of class "rvflnet"

Examples

```
## Not run:
x <- matrix(rnorm(100*5), 100, 5)
y <- x[,1] + sin(x[,2]) + rnorm(100, 0, 0.1)
model <- rvflnet(x, y, n_hidden = 50, activation = "tanh")
predict(model, newx = x[1:10,])
coef(model)  # Note: coefficients apply to [X | random features] space

## End(Not run)
```

---

summary.cv.rvflnet *Summary method for cross-validated RVFLNet*

Description

Summary method for cross-validated RVFLNet

Usage

```
## S3 method for class 'cv.rvflnet'
summary(object, s = "lambda.min", ...)
```

Arguments

object	cv.rvflnet object
s	lambda value ("lambda.min" or "lambda.1se")
...	additional arguments

Value

invisible summary object

`summary.rvflnet` *Summary method for RVFLNet*

Description

Summary method for RVFLNet

Usage

```
## S3 method for class 'rvflnet'
summary(object, s = NULL, ...)
```

Arguments

`object`	rvflnet object
`s`	lambda value
`...`	additional arguments

Value

invisible summary object

Index