Package: nnetLM (via r-universe)

May 27, 2026

Type Package

Title Neural Network with Levenberg-Marquardt Optimization

Version 1.0.1.9000

Description An implementation of a Neural Network using the Levenberg-Marquardt optimization from 'minpack.lm', ideal for small datasets. For more details see Moré (1978) <doi:10.1007/BFb0067700>.

License MIT + file LICENSE

BugReports https://github.com/umbe1987/nnetLM/issues

Imports minpack.lm, stats

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

URL https://github.com/umbe1987/nnetLM

Repository https://umbe1987.r-universe.dev

Date/Publication 2026-02-25 16:46:40 UTC

RemoteUrl https://github.com/umbe1987/nnetlm

RemoteRef HEAD

RemoteSha 0ebb18f12a1cd532696aeb95146cefb2f0bfb5cf

Contents

`flatten_params` *Flattens the network parameters so they can be passed to [minipack.lm::nls.lm] 'par' argument*

Description

Flattens the network parameters so they can be passed to [minipack.lm::nls.lm] 'par' argument

Usage

```
flatten_params(object)
```

Arguments

`object`	an object of class "nnetLM"

Value

flattened vector with network parameters (weights and biases)

`nnetLM` *Initalize the neural network object*

Description

Initalize the neural network object

Usage

```
nnetLM(X, y, hidden, actFn)
```

Arguments

`X`	Matrix of independent variables
`y`	Vector of dependent variables
`hidden`	Vector of number of nodes in each hidden layer
`actFn`	List of activation functions (must be length(hidden)+1 for the output node)

Details

The activation functions within [actFn] list can be any existing or user-defined function. They must have a single numeric argument (e.g. [x]), and must return a numeric value of the same length as [x].

Value

An object with S3 class "nnetLM"

Examples

```
set.seed(123)
x <- seq(-10, 10, by = 0.1)
y <- sin(x) + rnorm(length(x), mean = 0, sd = 0.1)
X <- matrix(x, nrow = length(x), ncol = 1)
hidden <- c(10)
linear <- function(x) x
actFn <- list(tanh, linear)
nnet.obj <- nnetLM(X, y, hidden, actFn)
```

---

predict.nnetLM *Performs a forward pass*

Description

Performs a forward pass

Usage

```
## S3 method for class 'nnetLM'
predict(object, newdata)
```

Arguments

object	a trained network object of class "nnetLM"
newdata	Matrix of predictors

Value

a numeric vector with predicted values

Examples

```
set.seed(123)
x <- seq(-10, 10, by = 0.1)
y <- sin(x) + rnorm(length(x), mean = 0, sd = 0.1)
X <- matrix(x, nrow = length(x), ncol = 1)
hidden <- c(10)
linear <- function(x) x
actFn <- list(tanh, linear)
nnet.obj <- nnetLM(X, y, hidden, actFn)
nnet.obj <- train.nnetLM(nnet.obj,50)
pred.nnetLM <- predict(nnet.obj, X)
```

residFun *Residual function needed by [minipack.lm::nls.lm]*

Description

Residual function needed by [minipack.lm::nls.lm]

Usage

```
residFun(params, observed, object, xx)
```

Arguments

params	vector of flattened network parameters (weights and biases)
observed	an object of class "nnetLM"
object	an object of class "nnetLM"
xx	an object of class "nnetLM"

Value

unflattened list with network parameters (weights and biases)

train.nnetLM *Train the neural network with Levenberg-Marquardt optimization using [minipack.lm::nls.lm]*

Description

Train the neural network with Levenberg-Marquardt optimization using [minipack.lm::nls.lm]

Usage

```
train.nnetLM(object, epochs, progress = FALSE)
```

Arguments

object	an object of class "nnetLM"
epochs	maximum number of iteration
progress	flag for printing network progress. Default is FALSE

Value

the trained network object

See Also

[minipack.lm::nls.lm()]

Examples

```
x <- seq(-10, 10, by = 0.1)
y <- sin(x) + rnorm(length(x), mean = 0, sd = 0.1)
X <- matrix(x, nrow = length(x), ncol = 1)
hidden <- c(10)
linear <- function(x) x
actFn <- list(tanh, linear)
nnet.obj <- nnetLM(X, y, hidden, actFn)
nnet.obj <- train.nnetLM(nnet.obj,1)
```

`unflatten_params` *Unflattens the network parameters after they have been used by [minipack.lm::nls.lm]*

Description

Unflattens the network parameters after they have been used by [minipack.lm::nls.lm]

Usage

```
unflatten_params(params, object)
```

Arguments

`params`	vector of flattened network parameters (weights and biases)
`object`	an object of class "nnetLM"

Value

unflattened list with network parameters (weights and biases)

Index