Package 'densemlp'

August 8, 2026

Title Dense Neural Networks for Tabular Classification and Regression

Version 0.5.0

Author Imad EL BADISY [aut, cre]

Maintainer Imad EL BADISY <elbadisyimad@gmail.com>

Description Provides dense feed-forward neural network models for tabular regression and classification using 'torch'. The package supports modern extensions around dense neural network blocks, including dropout, batch normalization, residual connections, gated blocks, and optional input projection.

URL https://github.com/ielbadisy/densemlp

BugReports https://github.com/ielbadisy/densemlp/issues

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports ggplot2, stats, torch

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2026-08-08 13:50:11 UTC

Contents

`autoplot.densemlp_fit` *Plot training history*

Description

Plot training history

Usage

```
## S3 method for class 'densemlp_fit'
autoplot(object, ...)
```

Arguments

`object`	A fitted `densemlp_fit` object.
`...`	Unused.

Value

A ggplot object.

`densemlp` *Fit a tabular dense multilayer perceptron*

Description

Fit a tabular dense multilayer perceptron

Usage

```
densemlp(
  formula = NULL,
  data = NULL,
  x = NULL,
  y = NULL,
  task = c("auto", "classification", "regression"),
  hidden_units = c(64, 32),
  activation = c("relu", "tanh", "gelu"),
  dropout = 0,
  batch_norm = TRUE,
```

```
  residual = FALSE,
  gated = FALSE,
  input_projection = NULL,
  epochs = 100,
  batch_size = 32,
  lr = 0.001,
  optimizer = c("adam", "sgd"),
  lr_schedule = c("none", "cosine", "step"),
  weight_decay = 0,
  validation = 0.2,
  early_stopping = TRUE,
  patience = 10,
  min_delta = 0,
  min_epochs = max(10L, floor(epochs * 0.2)),
  loss = NULL,
  label_smoothing = 0,
  focal_gamma = 2,
  metrics = NULL,
  seed = 1,
  verbose = TRUE,
  log_every = 1,
  device = c("auto", "cpu", "cuda")
)
```

Arguments

formula	A formula specification.
data	A data frame used with formula.
x	Predictor data frame or matrix. Retained for backward compatibility with the x/y interface.
y	Outcome vector. Retained for backward compatibility with the x/y interface.
task	Optional task override. "auto" infers the task from the outcome.
hidden_units	Hidden layer sizes.
activation	Activation function.
dropout	Dropout probability.
batch_norm	Use batch normalization in hidden layers.
residual	Use residual skip connections between hidden blocks.
gated	Use learned gating inside hidden blocks.
input_projection	Optional input projection dimension before hidden blocks.
epochs	Number of epochs.
batch_size	Mini-batch size.
lr	Learning rate.
optimizer	Optimizer name.

lr_schedule	Learning-rate schedule.
weight_decay	Weight decay.
validation	Validation fraction.
early_stopping	Enable early stopping.
patience	Early stopping patience.
min_delta	Minimum validation loss improvement.
min_epochs	Minimum number of epochs before early stopping can trigger.
loss	Loss function. "focal" is available for binary classification.
label_smoothing	Label smoothing for classification losses.
focal_gamma	Focal-loss focusing parameter.
metrics	Reserved for future custom metrics.
seed	Random seed.
verbose	Verbosity level. FALSE or 0 silences output, TRUE or 1 prints a standard log, and 2 prints a detailed log.
log_every	Epoch logging frequency.
device	Device to use.

Value

A densemlp_fit object.

---

densemlp_metrics *Compute densemlp metrics*

Description

Compute densemlp metrics

Usage

```
densemlp_metrics(truth, estimate, task = NULL, prob = NULL)
```

Arguments

truth	Ground truth values.
estimate	Predicted classes for classification or numeric predictions for regression.
task	Optional task override. When NULL, the task is inferred from truth and estimate.
prob	Optional class probabilities.

Value

A named list of metrics.

perm_importance *Permutation variable importance*

Description

Permutation variable importance

Usage

```
perm_importance(object, new_data, truth, metric = NULL, seed = object$seed)
```

Arguments

object	A fitted densemlp_fit object.
new_data	Evaluation data frame.
truth	Ground truth outcome values.
metric	Metric name. Defaults to accuracy for classification and RMSE for regression.
seed	Random seed used for shuffling.

Value

A densemlp_importance object.

plot.densemlp_importance *Plot permutation importance*

Description

Plot permutation importance

Usage

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

Arguments

x	A densemlp_importance object.
...	Unused.

Value

A ggplot object.

plot_history *Plot training history*

Description

Plot training history

Usage

```
plot_history(object)
```

Arguments

object	A fitted densemlp_fit object.

Value

A ggplot object.

predict.densemlp_fit *Predict from a fitted dense multilayer perceptron*

Description

Predict from a fitted dense multilayer perceptron

Usage

```
## S3 method for class 'densemlp_fit'
predict(object, new_data, type = NULL, ...)
```

Arguments

object	A fitted densemlp_fit object.
new_data	New predictor data.
type	Prediction type.
...	Unused.

Value

Predictions in a task-appropriate format.

print.densemlp_fit *Print a fitted dense multilayer perceptron*

Description

Print a fitted dense multilayer perceptron

Usage

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

Arguments

x	A fitted densemlp_fit object.
...	Unused.

Value

x, invisibly.

tune_densemlp *Tune a dense multilayer perceptron over a task-aware hyperparameter grid*

Description

Tune a dense multilayer perceptron over a task-aware hyperparameter grid

Usage

```
tune_densemlp(
  formula = NULL,
  data = NULL,
  x = NULL,
  y = NULL,
  task = c("auto", "classification", "regression"),
  grid = NULL,
  metric = NULL,
  validation = 0.2,
  early_stopping = TRUE,
  patience = 10,
  min_delta = 0,
  min_epochs = NULL,
  seed = 1,
```

```
  repeats = 3,
  verbose = FALSE,
  device = c("auto", "cpu", "cuda"),
  refit = TRUE
)
```

Arguments

formula	A formula specification.
data	A data frame used with formula.
x	Predictor data frame or matrix.
y	Outcome vector.
task	Optional task override. "auto" infers the task from the outcome.
grid	A named list of candidate values.
metric	Optional ranking metric. Use "accuracy" for classification and "rmse" or "valid_loss" for regression.
validation	Validation fraction for each fit.
early_stopping	Enable early stopping during tuning.
patience	Early stopping patience.
min_delta	Minimum validation loss improvement.
min_epochs	Minimum number of epochs before early stopping can trigger.
seed	Base random seed.
repeats	Number of repeated seeds per candidate.
verbose	Print per-candidate progress.
device	Device to use.
refit	Refit the best configuration on the supplied data.

Value

A list with ranked tuning results and, when refit = TRUE, the best fitted model.

Index