Package 'nndiagram'

May 9, 2026

Title Generator of 'LaTeX' Code for Drawing Neural Network Diagrams with 'TikZ'

Version 1.0.0

Maintainer Chencheng Fang <ccfang@uni-bonn.de>

Description Generates 'LaTeX' code for drawing well-formatted neural network diagrams with 'TikZ'. Users have to define number of neurons on each layer, and optionally define neuron connections they would like to keep or omit, layers they consider to be oversized and neurons they would like to draw with lighter color. They can also specify the title of diagram, color, opacity of figure, labels of layers, input and output neurons. In addition, this package helps to produce 'LaTeX' code for drawing activation functions which are crucial in neural network analysis. To make the code work in a 'LaTeX' editor, users need to install and import some 'TeX' packages including 'TikZ' in the setting of 'TeX' file.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.1

Imports dplyr, utils

Suggests testthat (>= 3.0.0)

URL https://github.com/ccfang2/nndiagram

BugReports https://github.com/ccfang2/nndiagram/issues

Config/testthat/edition 3

NeedsCompilation no

Author Chencheng Fang [aut, cre]

Repository CRAN

Date/Publication 2023-04-18 14:40:02 UTC

Contents

`activation_curve` *Producing 'LaTeX' Code for Drawing Activation Functions*

Description

The `activation_curve` command is used to produce 'LaTeX' code for drawing well-formatted activation functions, which are crucial in neural network analysis. To make the code work in a 'LaTeX' editor, users need to install and import two 'TeX' packages TikZ and pgfplots in the setting of 'TeX' file. Syntax of importing these packages is included in the output of function.

Usage

```
activation_curve(
  expr = "ReLU",
  title = NULL,
  xlabel = NULL,
  ylabel = NULL,
  xmin = -10,
  xmax = 10,
  ymin = NULL,
  ymax = NULL,
  suppress = FALSE
)
```

Arguments

`expr`	an optional character that specifies the activation function. It could be `"ReLU"`, `"sigmoid"` or `"step"`, which are commonly used activation functions. Or, it can be a call or an expression written as a function of `x` that will evaluate to an object of the same length as `x`. Default is `"ReLU"`.
`title`	an optional character that specifies the main title of diagram. Default is `NULL`.
`xlabel`	an optional character that specifies the label of x-axis. Default is `NULL`.
`ylabel`	an optional character that specifies the label of y-axis. Default is `NULL`.
`xmin`	an optional numeric value that specifies the minimum value of x-axis. Default is `-10`.
`xmax`	an optional numeric value that specifies the maximum value of x-axis. Default is `10`.
`ymin`	an optional numeric value that specifies the minimum value of y-axis. Default is `NULL`.
`ymax`	an optional numeric value that specifies the maximum value of y-axis. Default is `NULL`.
`suppress`	an optional logical value that specifies whether `activation_curve` should suppress the output of 'LaTeX' code to be directly printed on console. Default is `FALSE`.

Value

activation_curve uses cat() to print out 'LaTeX' code on console, if not suppressed. Also, activation_curve saves the same output as a character vector invisibly, so users could use cat() to print it out later at their demand, as shown in Examples. The activation_curve 'LaTeX' output can be directly copied and pasted to produce activation curve in any 'LaTeX' editor.

Author(s)

Chencheng Fang, Bonn Graduate School of Economics, University of Bonn. Email: <ccfang@uni-bonn.de>

See Also

nndiagram; nndiagram_oversize; nndiagram_nodeCoverup.

Examples

```
# Rectified Linear Unit Function with all arguments default
activation_curve()

# Sigmoid function with domain from -5 to 5.
activation_curve(expr="sigmoid", title="Sigmoid Function", xmin=-5, xmax=5)

# Define a parametric ReLU in the argument of \code{expr}.
activation_curve(expr="(x>=0)*x+0.2*x*(x<0)", title="Parametric Rectified Linear Unit Function")

# Suppress the output of 'LaTeX' code to be directly printed on the console and save the output
# to an object, which can be printed later at demand.
nnd_activation <- activation_curve(suppress=TRUE)
cat(paste(nnd_activation,"\n"))
```

---

nndiagram *Producing 'LaTeX' Code for Drawing Neural Network Diagrams*

Description

The nndiagram command is used to produce 'LaTeX' code for drawing well-formatted neural network diagrams. To make the code work in a 'LaTeX' editor, users need to install and import two 'TeX' packages, i.e., TikZ and ifthen in the setting of 'TeX' file. Syntax of importing these packages is included in the output of function.

Usage

```
nndiagram(
  input,
  hidden,
  keep = NULL,
  omit = NULL,
  title = NULL,
```

```
  color = "black",
  alpha = 1,
  layer.sep = 2.5,
  layer.label = NULL,
  input.label = NULL,
  output.label = NULL,
  suppress = FALSE
)
```

Arguments

input	a positive integer that specifies the number of input neurons.
hidden	a positive integer vector that specifies the number of neurons on each hidden layer. For example, c(4,4,4) specifies that there are 3 hidden layers and 4 neurons on each hidden layer. Non-positive and non-integer numbers are not allowed.
keep	an optional character vector that specifies the connections of neurons to be kept. For example, c("1->4","5->8") specifies the connections from neuron 1 to 4 and from neuron 5 to 8. Neurons are counted from top to bottom and from left to right. As special cases, "->4" specifies all connections with neuron 4 as destination, and "5->" specifies all connections with neuron 5 as source. No space in the string is allowed. Default is NULL.
omit	an optional character vector that specifies the connections of neurons to be omitted. The specification of connections is the same as that in 'keep'. However, users are not allowed to assign values to 'keep' and 'omit' simultaneously. Default is NULL.
title	an optional character that specifies the main title of diagram. Default is NULL.
color	an optional character that specifies the color of lines. Default is "black".
alpha	an optional numeric value between 0 and 1 that specifies the opacity of lines. 1 indicates lines to be opaque, and 0 indicates lines to be transparent. Default is 1.
layer.sep	an optional positive numeric value that specifies the distance between layers of a neural network. Default is 2.5.
layer.label	an optional character vector that specifies label for each layer, including input, hidden and output layers.
input.label	an optional character vector that specifies label for each input neuron.
output.label	an optional character that specifies label for output neuron.
suppress	an optional logical value that specifies whether nndiagram should suppress the output of 'LaTeX' code to be directly printed on console. Default is FALSE.

Value

nndiagram uses cat() to print out 'LaTeX' code on console, if not suppressed. Also, nndiagram saves the same output as a character vector invisibly, so users could use cat() to print it out later at their demand, as shown in Examples. The nndiagram 'LaTeX' output can be directly copied and pasted to produce neural network diagram in any 'LaTeX' editor.

Note

This package is an ongoing project, and more functions will be added in the future, such as those to produce pdf version of diagrams or convert handdrawing neural network diagrams to computerized ones. Collaborations are sincerely welcome. Comments and suggestions are always highly appreciated.

Author(s)

Chencheng Fang, Bonn Graduate School of Economics, University of Bonn. Email: <ccfang@uni-bonn.de>

See Also

nndiagram_oversize; nndiagram_nodeCoverup; activation_curve.

Examples

```
# A neural network with 3 neurons on input layer, 4 neurons on each of 3 hidden layers,
# and 1 neuron on output layer. No connection is omitted and all other arguments are default.
nndiagram(input=3, hidden=c(4,4,4))

# Same as the first example but connections from neuron 1 to 4 and from neuron 5 to 8 are omitted.
nndiagram(input=3, hidden=c(4,4,4), omit=c("1->4","5->8"))

# Same as the first example but connections with neuron 4 as destination are omitted.
nndiagram(input=3, hidden=c(4,4,4), omit=c("->4"))

# Same as the first example but connections with neuron 5 as source are omitted.
nndiagram(input=3, hidden=c(4,4,4), omit=c("5->"))

# Suppress the output of 'LaTeX' code to be directly printed on the console and save the output
# to an object, which can be printed later at demand.
nnd <- nndiagram(input=3, hidden=c(4,4,4), suppress=TRUE)
cat(paste(nnd,"\n"))
```

---

nndiagram_nodeCoverup *Producing 'LaTeX' Code for Drawing Neural Network Diagrams with Some Neurons being Covered-up*

Description

The nndiagram_nodeCoverup command is used to produce 'LaTeX' code for drawing well-formatted neural network diagrams, some neurons of which users hope to cover up with lighter color. To make the code work in a 'LaTeX' editor, users need to install and import the 'TeX' package TikZ in the setting of 'TeX' file. Syntax of importing this package is included in the output of function.

Usage

```
nndiagram_nodeCoverup(
  input,
  hidden,
  node.coverup = NULL,
  title = NULL,
  color = "black",
  alpha = 1,
  layer.sep = 2.5,
  layer.label = NULL,
  input.label = NULL,
  output.label = NULL,
  suppress = FALSE
)
```

Arguments

input	a positive integer that specifies the number of input neurons.
hidden	a positive integer vector that specifies the number of neurons on each hidden layer. For example, c(6,6,6) specifies that there are 3 hidden layers and 6 neurons on each hidden layer. Non-positive and non-integer numbers are not allowed.
node.coverup	an optional positive integer vector that specifies the index of neurons users hope to cover up with lighter color. Neurons are counted from top to bottom and from left to right. For example, in a neural network with `input=3` and `hidden=c(6,6,6)`, `node.coverup=c(4,10)` means that the user needs to cover up the first neuron on both the first and second hidden layer. Default is `NULL`.
title	an optional character that specifies the main title of diagram. Default is `NULL`.
color	an optional character that specifies the color of lines. Default is `"black"`.
alpha	an optional numeric value between 0 and 1 that specifies the opacity of lines. `1` indicates lines to be opaque, and `0` indicates lines to be transparent. Default is `1`.
layer.sep	an optional positive numeric value that specifies the distance between layers of a neural network. Default is `2.5`.
layer.label	an optional character vector that specifies label for each layer, including input, hidden and output layers.
input.label	an optional character vector that specifies label for each input neuron.
output.label	an optional character that specifies label for output neuron.
suppress	an optional logical value that specifies whether `nndiagram_nodeCoverup` should suppress the output of 'LaTeX' code to be directly printed on console. Default is `FALSE`.

Value

`nndiagram_nodeCoverup` uses `cat()` to print out 'LaTeX' code on console, if not suppressed. Also, `nndiagram_nodeCoverup` saves the same output as a character vector invisibly, so users could use `cat()` to print it out later at their demand, as shown in Examples. The `nndiagram_nodeCoverup` 'LaTeX' output can be directly copied and pasted to produce neural network diagram in any 'LaTeX' editor.

Note

This package is an ongoing project, and more functions will be added in the future, such as those to produce pdf version of diagrams or convert handdrawing neural network diagrams to computerized ones. Collaborations are sincerely welcome. Comments and suggestions are always highly appreciated.

Author(s)

Chencheng Fang, Bonn Graduate School of Economics, University of Bonn. Email: <ccfang@uni-bonn.de>

See Also

nndiagram; nndiagram_oversize; activation_curve.

Examples

```
# A neural network with 3 neurons on input layer, 6 neurons on each of 3 hidden layers,
# and 1 neuron on output layer. All other arguments are default, so no neuron is
# drawn with lighter color.
nndiagram_nodeCoverup(input=3, hidden=c(6,6,6))

# Same as the first example but neurons indexed with 4 and 11 are designed to be drawn
# with lighter color.
nndiagram_nodeCoverup(input=3, hidden=c(6,6,6), node.coverup=c(4,10))

# Same as the first example but distance between layers is defined to be smaller.
nndiagram_nodeCoverup(input=3, hidden=c(6,6,6), layer.sep=1.5)

# Suppress the output of 'LaTeX' code to be directly printed on the console and save the output
# to an object, which can be printed later at demand.
nnd_nodeCoverup <- nndiagram_nodeCoverup(input=3, hidden=c(6,6,6), suppress=TRUE)
cat(paste(nnd_nodeCoverup,"\n"))
```

`nndiagram_oversize` *Producing 'LaTeX' Code for Drawing Over-sized Neural Network Diagrams*

Description

The `nndiagram_oversize` command is used to produce 'LaTeX' code for drawing well-formatted neural network diagrams, some layers of which have excess neurons that users hope to leave out. To make the code work in a 'LaTeX' editor, users need to install and import the 'TeX' package TikZ in the setting of 'TeX' file. Syntax of importing this package is included in the output of function.

Usage

```
nndiagram_oversize(
  input,
  hidden,
  size.cutoff = 7,
  title = NULL,
  color = "black",
  alpha = 1,
  layer.sep = 2.5,
  layer.label = NULL,
  input.label = NULL,
  output.label = NULL,
  suppress = FALSE
)
```

Arguments

input	a positive integer that specifies the number of input neurons.
hidden	a positive integer vector that specifies the number of neurons on each hidden layer. For example, c(6,6,6) specifies that there are 3 hidden layers and 6 neurons on each hidden layer. Non-positive and non-integer numbers are not allowed.
size.cutoff	an optional numeric value that specifies the cutoff number of neurons. If the number of neurons on a certain layer is greater than this cutoff value, it is considered as an over-sized layer, which will be drawn with some neurons left out. Otherwise, all neurons on the layer will be drawn. In nndiagram_oversize, the cutoff number is defined to be no less than 5. Default is 7.
title	an optional character that specifies the main title of diagram. Default is NULL.
color	an optional character that specifies the color of lines. Default is "black".
alpha	an optional numeric value between 0 and 1 that specifies the opacity of lines. 1 indicates lines to be opaque, and 0 indicates lines to be transparent. Default is 1.
layer.sep	an optional positive numeric value that specifies the distance between layers of a neural network. Default is 2.5.
layer.label	an optional character vector that specifies label for each layer, including input, hidden and output layers.
input.label	an optional character vector that specifies label for each input neuron.
output.label	an optional character that specifies label for output neuron.
suppress	an optional logical value that specifies whether nndiagram_oversize should suppress the output of 'LaTeX' code to be directly printed on console. Default is FALSE.

Value

nndiagram_oversize uses cat() to print out 'LaTeX' code on console, if not suppressed. Also, nndiagram_oversize saves the same output as a character vector invisibly, so users could use cat() to print it out later at their demand, as shown in Examples. The nndiagram_oversize 'LaTeX' output can be directly copied and pasted to produce neural network diagram in any 'LaTeX' editor.

Note

This package is an ongoing project, and more functions will be added in the future, such as those to produce pdf version of diagrams or convert handdrawing neural network diagrams to computerized ones. Collaborations are sincerely welcome. Comments and suggestions are always highly appreciated.

Author(s)

Chencheng Fang, Bonn Graduate School of Economics, University of Bonn. Email: <ccfang@uni-bonn.de>

See Also

nndiagram; nndiagram_nodeCoverup; activation_curve.

Examples

```
# A neural network with 3 neurons on input layer, 6 neurons on each of 3 hidden layers, and 1 neuron
# on output layer. All other arguments are default, so no layer is considered to be over-sized.
nndiagram_oversize(input=3, hidden=c(6,6,6))

# Same as the first example but cutoff value of size is designed to be 5, so all hidden layers are
# considered to be over-sized.
nndiagram_oversize(input=3, hidden=c(6,6,6), size.cutoff=5)

# Same as the second example but labels of input neurons are designed to be letters from a to c.
nndiagram_oversize(input=3, hidden=c(6,6,6), size.cutoff=5, input.label=letters[1:3])

# Suppress the output of 'LaTeX' code to be directly printed on the console and save the output
# to an object, which can be printed later at demand.
nnd_oversize <- nndiagram_oversize(input=3, hidden=c(6,6,6), size.cutoff=5, suppress=TRUE)
cat(paste(nnd_oversize,"\n"))
```

Index