



NeuralNetTools: Visualization and Analysis Tools for Neural Networks

Marcus W. Beck

Oak Ridge Institute for Science and Education
US Environmental Protection Agency

Abstract

Functions within this package can be used for the interpretation of neural network models created in R, including functions to plot a neural network interpretation diagram, evaluation of variable importance, and a sensitivity analysis of input variables.

Keywords: neural networks, plotnet, sensitivity, variable importance, R.

1. Introduction

(Goh 1995)

2. Package structure

Packages in R that can create neural networks and relevant methods in NeuralNetTools
Cautionary note about finding the ‘optimal’ model.

2.1. Visualizing neural networks

The number of existing functions in **R** to view neural networks is minimal. Such tools have practical use for visualizing network architecture and connections between layers that mediate variable importance. To our knowledge, only the **neuralnet** package provides a method for plotting neural networks created with the **neuralnet** function. Although useful for viewing the basic structure, the output is visually minimal and does not include options for customization (verify).

This function plots a neural network as a neural interpretation diagram as in Özkesmi and

Özesmi (1999). Options to plot without color-coding or shading of weights are also provided. The default settings plot positive weights between layers as black lines and negative weights as grey lines. Line thickness is in proportion to relative magnitude of each weight. The first layer includes only input variables with nodes labelled arbitrarily as I1 through In for n input variables. One through many hidden layers are plotted with each node in each layer labelled as H1 through Hn. The output layer is plotted last with nodes labeled as O1 through On. Bias nodes connected to the hidden and output layers are also shown. Neural networks created using `mlp` do not show bias layers.

A primary network and a skip layer network can be plotted for `nnet` models with a skip layer connection. The default is to plot the primary network, whereas the skip layer network can be viewed with `skip = TRUE`. If `nid = TRUE`, the line widths for both the primary and skip layer plots are relative to all weights. Viewing both plots is recommended to see which network has larger relative weights. Plotting a network with only a skip layer (i.e., no hidden layer, `size = 0`) will include bias connections to the output layer, whereas these are not included in the plot of the skip layer if `size` is greater than zero.

Pruned networks in `RSNNS`.

2.2. Evaluating variable importance

The `garson` function uses Garson's algorithm to evaluate relative variable importance. This function identifies the relative importance of explanatory variables for a single response variable by deconstructing the model weights. The importance of each variable can be determined by identifying all weighted connections between the layers in the network. That is, all weights connecting the specific input node that pass through the hidden layer to the response variable are identified. This is repeated for all other explanatory variables until a list of all weights that are specific to each input variable is obtained. The connections are tallied for each input node and scaled relative to all other inputs. A single value is obtained for each explanatory variable that describes the relationship with the response variable in the model. The results indicate relative importance as the absolute magnitude from zero to one. The function cannot be used to evaluate the direction of the response. Only neural networks with one hidden layer and one output node can be evaluated.

The `olden` function is an alternative and more flexible approach to evaluate variable importance. The function calculates importance as the product of the raw input-hidden and hidden-output connection weights between each input and output neuron and sums the product across all hidden neurons. An advantage of this approach is the relative contributions of each connection weight are maintained in terms of both magnitude and sign as compared to Garson's algorithm which only considers the absolute magnitude. For example, connection weights that change sign (e.g., positive to negative) between the input-hidden to hidden-output layers would have a cancelling effect whereas Garson's algorithm may provide misleading results based on the absolute magnitude. An additional advantage is that Olden's algorithm is capable of evaluating neural networks with multiple hidden layers and response variables. The importance values assigned to each variable are in units that are based directly on the summed product of the connection weights. The actual values should only be interpreted based on relative sign and magnitude between explanatory variables. Comparisons between different models should not be made.

Issues with different indications of variable importance as a model is refit...

2.3. Sensitivity analysis

The Lek profile method is described briefly in [Lek, Delacoste, Baran, Dimopoulos, Lauga, and Aulagnier \(1996\)](#) and in more detail in [Gevrey, Dimopoulos, and Lek \(2003\)](#). The profile method is fairly generic and can be extended to any statistical model in R with a predict method. However, it is one of few methods used to evaluate sensitivity in neural networks.

The profile method can be used to evaluate the effect of explanatory variables by returning a plot of the predicted response across the range of values for each separate variable. The original profile method evaluated the effects of each variable while holding the remaining explanatory variables at different quantiles (e.g., minimum, 20th percentile, maximum). This is implemented in the function by creating a matrix of values for explanatory variables where the number of rows is the number of observations and the number of columns is the number of explanatory variables. All explanatory variables are held at their mean (or other constant value) while the variable of interest is sequenced from its minimum to maximum value across the range of observations. This matrix (or data frame) is then used to predict values of the response variable from a fitted model object. This is repeated for each explanatory variable to obtain all response curves. Values passed to `split_vals` must range from zero to one to define the quantiles for holding unevaluated explanatory variables.

An alternative implementation of the profile method is to group the unevaluated explanatory variables using groupings defined by the statistical properties of the data. Covariance among predictors may present unlikely scenarios if holding all unevaluated variables at the same level. To address this issue, the function provides an option to hold unevaluated variable at mean values defined by natural clusters in the data. kmeans clustering is used on the input data.frame of explanatory variables if the argument passed to `split_vals` is an integer value greater than one. The centers of the clusters are then used as constant values for the unevaluated variables. An arbitrary grouping scheme can also be passed to `split_vals` as a data.frame where the user can specify exact values for holding each value constant (see the examples). Examples in [Beck, Wilson, Vondracek, and Hatch \(2014\)](#) show this...

For all plots, the legend with the 'splits' label indicates the colors that correspond to each group. The groups describe the values at which unevaluated explanatory variables were held constant, either as specific quantiles, group assignments based on clustering, or in the arbitrary grouping defined by the user. The constant values of each explanatory variable for each split can be viewed as a barplot by using `split_show = TRUE`.

Note that there is no predict method for neuralnet objects from the nn package. The lekprofile method for nn objects uses the nnet package to recreate the input model, which is then used for the sensitivity predictions. This approach only works for networks with one hidden layer.

3. Future development

4. Acknowledgments

References

- Beck MW, Wilson BN, Vondracek B, Hatch LK (2014). “Application of neural networks to quantify the utility of indices of biotic integrity for biological monitoring.” *Ecological Indicators*, **45**, 195–208.
- Gevrey M, Dimopoulos I, Lek S (2003). “Review and comparison of methods to study the contribution of variables in artificial neural network models.” *Ecological Modelling*, **160**(3), 249–264.
- Goh ATC (1995). “Back-propagation neural networks for modeling complex systems.” *Artificial Intelligence in Engineering*, **9**(3), 143–151.
- Lek S, Delacoste M, Baran P, Dimopoulos I, Lauga J, Aulagnier S (1996). “Application of neural networks to modelling nonlinear relationships in ecology.” *Ecological Modelling*, **90**(1), 39–52.
- Özesmi SL, Özesmi U (1999). “An artificial neural network approach to spatial habitat modelling with interspecific interaction.” *Ecological Modelling*, **116**(1), 15–31.

Affiliation:

Marcus W. Beck
Oak Ridge Institute for Science and Education
US Environmental Protection Agency
National Health and Environmental Effects Research Laboratory
Gulf Ecology Division, 1 Sabine Island Drive
Gulf Breeze, Florida, 32561, USA
E-mail: beck.marcus@epa.gov