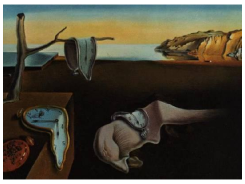

Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

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

The increasing quantity of information and computational capacity to address relevant research questions has contributed to the growth of data science as a legitimate field of study. Data science is a relatively new paradigm of analysis that focuses on the synthesis of unstructured information from multiple sources to identify patterns or trends 'born from the data' (Kelling, Hochachka, Fink, Riedewald, Caruana, Ballard, and Hooker 2009). A central theme is the focus on data exploration and prediction as compared to specific hypothesis-testing of traditional, domain-specific methods of scientific exploration (Kell and Oliver 2003). Demand for quantitative toolsets to address challenges in data-rich environments has increased drastically with the continuing advancement of techniques for rapid acquisition of data. Fields of research characterized by high-throughput data have a strong foundation in computationally-intensive methods of analysis (e.g., Saeys, Inza, and naga (2007)). By contrast, disciplines that have historically been limited by data quantity, such as ecological studies across broad temporal and spatial scales, have shown a need for data intensive approaches given the improved ability to acquire information (e.g., Swanson, Kosmala, Lintott, Simpson, Smith, and Packer (2015)). Regardless of the discipline, quantitative approaches that explicitly focus on inductive reasoning can serve a complementary role to conventional, hypothesis-driven approaches to scientific discovery (Kell and Oliver 2003).

Statistical methods that have been used to support data exploration for deductive analysis are numerous. A common theme among many of these methods is the use of machine-learning algorithms where the primary objective is to identify emergent patterns in the data with minimal human intervention. Neural networks, in particular, are designed to mimic the neuronal structure of the human brain by 'learning' inherent data structures through adaptive algorithms. Although the conceptual model was introduced several decades ago (Maier and Dandy 2000), neural networks continue to maintain a central role in data intensive science. The most popular form of neural network is the multilayer perceptron (mlp) trained using the backpropagation algorithm (Rumelhart, Hinton, and Williams 1986). This model is typically used to predict the response of one or more variables given co-occurrence of any of a number of explanatory variables. The hallmark feature of the mlp is the characterization of relationships between variabiles using an arbitrary number of parameters (i.e., the hidden layer) that are chosen through an iterative training process with the backpropation algorithm. Conceptually, the mlp is nothing more than a hyper-parameterized non-linear model that can a fit smooth function with almost non-existent residual error for any dataset.

An arbitrarily large number of parameters to fit a neural network provides obvious predictive advantages, but conversely complicates the extraction of critical model information. Information such as variable importance or model sensitivity are necessary aspects of exploraty data analysis that are not easily obtained from a neural network. As such, a common criticism is that neural networks are 'black-boxes' that offer minimal insight into relationships among variables. Olden and Jackson (2002) provides a rebuttal to this concern by providing methods to extract information from neural networks, most of which were previously available but not commonly used to evaluate neural networks. For example, Olden and Jackson (2002) describes neural interpretation diagrams for plotting (Özesmi and Özesmi 1999), the Garson algorithm for variable importance (Garson 1991), and the Profile method for sensitivity analysis (Lek, Delacoste, Baran, Dimopoulos, Lauga, and Aulagnier 1996). These quantitative tools 'illuminate the black box' by disaggregating the network parameters as a means to characterize relationships between variables. In essence, mlp neural networks were developed for prediction but methods in (Olden and Jackson 2002) leverage these models to describe data signals. Increasing the accesibility of these diagnostic tools will have value for exploratory analysis in data science.

This article describes the **NeuralNetTools** package for R that was developed to improve the breadth and quality of information obtained from the mlp neural network. Functions provided by the package are those previously described in (Olden and Jackson 2002) but have not been available in an open-source proramming environment. The reach of the package is all-inclusive such that generic functions were developed using S3 methods for all neural network object classes available in R. The objecives of this article are to 1) provide an overview of the statistical foundation the mlp network, 2) briefly describe similarities and differences between existing neural network packages in R, and 3) describe the theory and application of the primary functions in the **NeuralNetTools** package. The package is currently available on CRAN, whereas the development version is maintained as a GitHub repository.

2. Theoretical foundation and existing R packages

Packages available in R to create neural networks (similarities, differences) hyper-dimensional

Formulaically (Ripley 1996) $y_k = f_k \left(\sum_{j=1}^{k} w_{jk} f_j \left(\sum_{i=1}^{j} w_{ij} x_i \right) \right)$

3. Package structure

3.1. 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 Özesmi 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.

3.2. 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...

3.3. Sensitivity analysis

The Lek profile method is described briefly in Lek *et al.* (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 expalanatory variables at different quantiles (e.g., minimum, 20th percentile, maximum). This is implemented in 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 unevalutaed 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.

4. Future development

5. Conclusions

A cautionary note about the 'optimal network' and reproducibility of results.

6. 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.

Garson GD (1991). "Interpreting neural network connection weights." *Artificial Intelligence Expert*, **6**(4), 46–51.

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.

Kell DB, Oliver SG (2003). "Here is the evidence, now what is the hypothesis? The complementary roles of inductive and hypothesis-driven science in the post-genomic era." *BioEssays*, **26**(1), 99–105.

Kelling S, Hochachka WM, Fink D, Riedewald M, Caruana R, Ballard G, Hooker G (2009). "Data-intensive science: A new paradigm for biodiversity studies." *BioScience*, **59**(7), 613–620.

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.

Maier HR, Dandy GC (2000). "Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications." *Environmental Modelling and Software*, **15**(1), 101–124.

Olden JD, Jackson DA (2002). "Illuminating the "black box": A randomization approach for understanding variable contributions in artifical neural networks." *Ecological Modelling*, **154**(1-2), 135–150.

Olden JD, Joy MK, Death RG (2004). "An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data." *Ecological Modelling*, **178**(3-4), 389–397.

Özesmi SL, Özesmi U (1999). "An artificial neural network approach to spatial habitat modelling with interspecific interaction." *Ecological Modelling*, **116**(1), 15–31.

Ripley BD (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, United Kingdom.

Rumelhart DE, Hinton GE, Williams RJ (1986). "Learning representations by back-propagating errors." *Nature*, **323**(6088), 533–536.

Saeys Y, Inza I, naga PL (2007). "A review of feature selection techniques in bioinformatics." *Bioinformatics*, **23**(19), 2507–2517.

Swanson A, Kosmala M, Lintott C, Simpson R, Smith A, Packer C (2015). "Snapshot Serengeti: High-frequency annotated camera trap images of 40 mammalian species in African savanna." *Scientific Data*, **2**, 150026.

Affiliation:

Marcus W. Beck
Oak Ridge Institude 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

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/

Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd