SESUG Paper 102-2019

Glass Box Neural Networks

Ross Bettinger, Silver Spring, MD

Abstract

Neural network models are typically described as “black boxes” because their inner workings are not easy to understand. We propose that, since a neural network model that accurately predicts its target variable is a good representation of the training data, the output of the model may be recast as a target variable and subjected to standard regression algorithms to “explain” it as a response variable. Thus, the “black box” of the internal mechanism is transformed into a “glass box” that facilitates understanding of the underlying model. Deriving a regression model from a set of training data analogous to a neural network is an effective means to understand a neural network model because regression algorithms are commonly-used tools and the interpretation of a regression model is straight-forward and well-understood.

Keywords

Ordinary least squares regression, logistic regression, multilayer perceptron, neural networks, target variable, dependent variable, continuous variable, categorical variable, variable selection, black box, glass box, SAS® Enterprise Miner®

Introduction

Neural networks are machine learning algorithms that are noted for their ability to learn descriptive features of a set of training data. The model represented by a neural network can be applied to new data for the purpose of predicting the value of an unknown target variable. Neural networks use supervised learning to create a model based on features of a continuous target variable or of a categorical target variable, and thus can be used for building regression or classification models. Unlike ordinary least squares regression or logistic regression models, neural networks do not produce a set of parameter estimates. Such parameters can be used to signify the unit change in a continuous target variable given a unit change in a regressor, or the change in the odds ratio of a categorical target variable for a specified value of a categorical predictor. Hence, neural networks are often called “black box” models because their inner workings are opaque and are hard to interpret.

We demonstrate a technique by which the output of a neural network can be analyzed by regression to transform the neural network output into the context of a regression problem. The parameter estimates of the regression model may be used as surrogates for the neural network variable weights and biases to reveal the inner workings of the neural network.

Discussion

If a neural network accurately fits its set of training data, we may conclude that it has successfully abstracted from the data the relevant relationships between the target variable and the independent variables associated with the target variable. We will assume that this is the case, so that the output of the neural network, which represents the prediction of the model, may be reinterpreted to be a target variable for a subsequent model.

We may then build a second model using the neural network output and all of the original variables used in the first model. While we understand that a model is an approximation to and an abstraction

from the relationships in the data used to build it, we base our thesis on the concept that "… all models are wrong; the practical question is how wrong do they have to be to not be useful."[1]

We restrict our exposition to multilayer perceptron neural networks that produce a single output value which becomes the dependent variable for a regression algorithm.

Methodology

We describe the methodology of converting a "black box" neural network into a "glass box" model briefly and demonstrate the technique with an example.

There are three phases to the technique:

1. Build a single-output neural network model
   1.1. Build a classification model if the neural network target variable[2] is categorical in nature, e.g., the target variable is to be assigned a label from a (typically small) finite set of labels. The output is then a label stored in the predicted target variable that is assigned to an observation.
   1.2. Build a prediction model if the target variable is numeric and continuous in nature, e.g., the target variable may represent a potentially infinite number of values. The output is then a numeric value assigned to the predicted target variable.
2. Build a regression model using the output of the neural network model as the dependent variable based on all of the original variables used to build the NN model. All of the original variables must be used because the information contained in the modeling data is related to the output of the NN, e.g., the label assigned to the target variable or the value computed for it, and the regression algorithm, must use the same information to *interpret* the NN output as was used to *create* the NN model.[3]
   2.1. If a classification NN model was built, use logistic regression for a binary-valued target variable or multinomial logistic regression for a nominal or ordinal-valued target variable.[4]
   2.2. If a continuous NN model was built, use ordinary least-squares regression.
3. Assuming that the regression model is a close approximation to the neural network model, use the parameter estimates of the regression model to explain the effect of the predictor variables on the value of the target variable.

By assumption, since the regression model output closely approximates the NN model output, the regression parameter estimates are useful proxies for the NN model predictor variable weights, and we may describe the opaque workings of the NN model in terms of the transparent regression equation.

Example of Categorical Target Variable

A categorical target variable can be binary, nominal, or ordinal in its measurement scale. It has a finite set of values, typically a very small number. For the purpose of this discussion, we use sample data from

[1] George Box, https://en.wikipedia.org/wiki/All_models_are_wrong

[2] In the machine learning literature, a "target variable" is the variable whose values are to be predicted by a machine learning algorithm. The ML "target variable" is the same as the statistician's "dependent variable". It is not clear to us why there is a difference in terminology, but there are some things which are not given us to know.

[3] More complex algorithms may be used, e.g., generalized linear models, but a simple, well-understood algorithm admits of readily-understood interpretations.

[4] A generalized linear model may be used if the relationship between the linkage of the odds ratio and the dependent variables is not linear, but increasing sophistication may beget increasing subtlety of interpretation.

the 1994 Census database [1]. The target variable is a binary variable which contains 1 if a person's income is over $50,000/year and 0 if the person's income is less than or equal to $50,000/year. In addition to the binary target variable, there were four interval-scale and eight nominal-scale input variables. Table 1 contains a brief description of the variables used in the model.

Table 1: Categorical Target Modeling Variables

Variable	Measurement Scale	Description
Class	Binary	Target variable
Age	Continuous	Person's age
Cap_Gain	Continuous	Income from investments, apart from wages/ salary
Cap_Loss	Continuous	Losses from investments, apart from wages/salary
Country	Nominal	Country of origin
Educ	Nominal	Highest educational level achieved
Hourweek	Continuous	Hours worked per week
Marital	Nominal	Marital status
Occupatn	Nominal	Occupational category
Race	Nominal	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
Relation	Nominal	Wife, Own-child, Husband, Not-in family, Other-relative, Unmarried
Sex	Nominal	Male/Female
Workclass	Nominal	Private sector, public sector, &c.

Exploratory Data Analysis

Exploratory data analysis revealed that the variables most strongly correlated with the target variable were Age, Educ, Hourweek, Occupatn, and Relation. The other variables were omitted from the analysis because they were not strongly associated with the target variable. The Relation variable was later discarded because it created an error condition called "quasi-complete separation". This topic is discussed below.

Stacked histograms of the predictor variables show the distribution of the target variable by grouping interval:

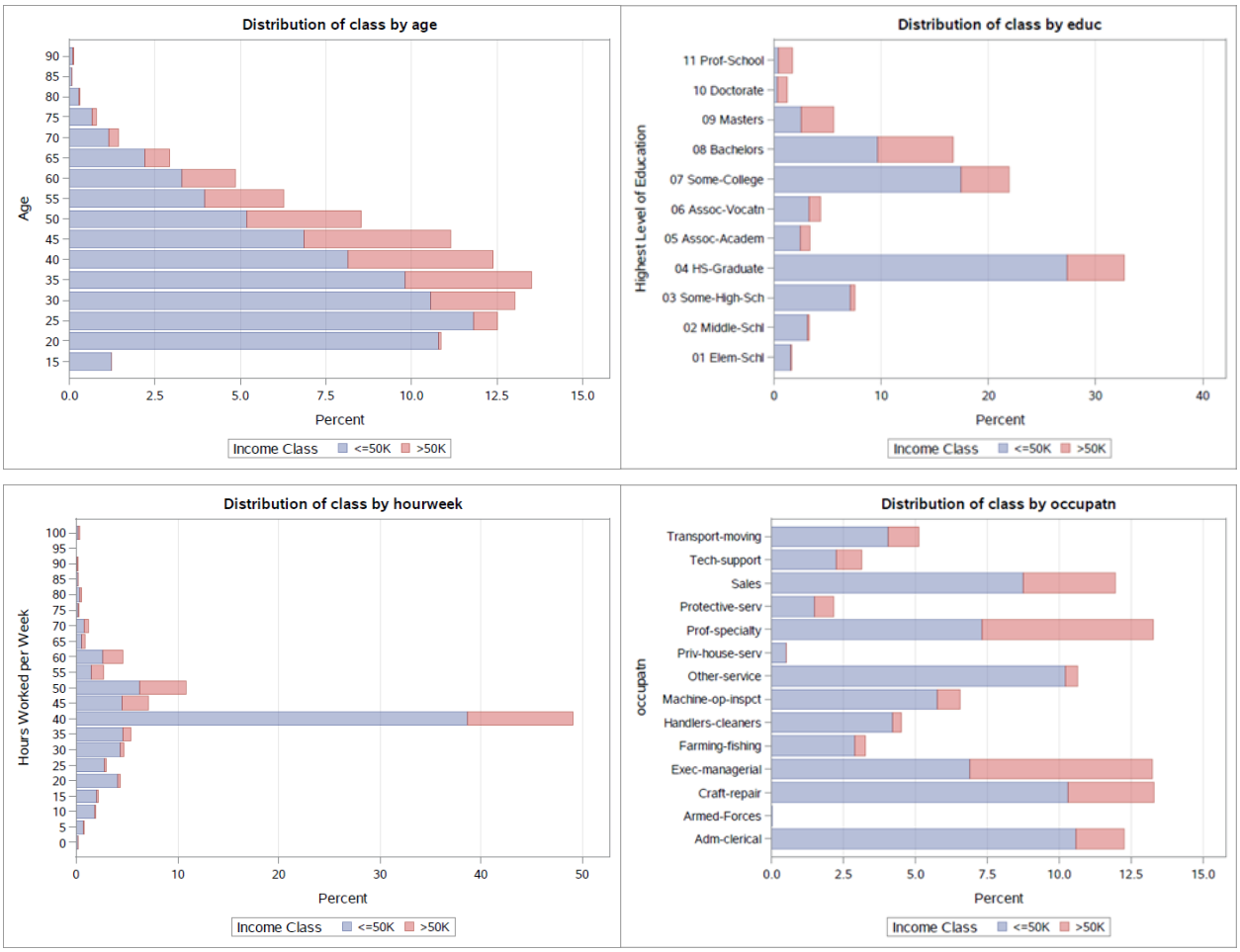

Figure 1: Stacked histograms of raw data

We see that the peak earning years for the ages of persons included in the data sample extend from the mid-30's to the mid-50's. College graduates with bachelor's degrees and higher education are much more likely to have incomes greater than $50,000 than those who did not complete four years of higher education. Those who worked more than 40 hours/week are represented proportionately more in the greater than $50,000 class than other workers. Salaried employees in sales, professional, and executive occupations are also high-income individuals compared to blue collar workers or support occupations.

Quasi-Complete Separation

After we built preliminary NN models, we noticed that the Relation variable contained categories that did not contain any values for target variable = 1, e.g., for the case where the income was greater than $50,000. The NN models classified all cases of 'Own-child' and 'Unmarried' into tgt_class = 0, thus creating a condition called "Quasi-complete separation". The logistic regression algorithm is designed to produce a rule that separates the set of input data into two subsets that have minimal overlap, and if the

data contain disjoint subsets, the parameter estimation algorithm fails to converge. One remedy in this case is to exclude the variable causing the separation from the modeling process.[5]

Where appropriate and meaningful, we grouped the predictors into fewer discrete categories than are present in the data so as to ensure that there would be an adequate representation of the target variable in each category to avoid quasi-complete separation. For example, in Table 2, we see that the rule "If Relation = 'Own-child' or Relation = 'Unmarried' then Into: tgt_class = 0" would completely separate the dataset into two disjoint sets.[6] No other variables would be required, and in this case, the logistic regression algorithm would fail. We did not include the variable Relation in subsequent analysis for this reason.

Table 2: Example of Quasi-Complete Separation

	Into: tgt_class	
	0	1
	N	N
relation		
Husband	7367	5680
Not-in-family	8023	73
Other-relative	898	1
Own-child	3935	.
Unmarried	3356	.
Wife	738	690

Modifying the Data

Histograms of the grouped variables are shown in Figure 2. We arranged the groups based on visual inspection for variables Age, Educ, and Hourweek, and used the SAS Enterprise Miner® Decision Tree node to group the Occupatn variable.

[5] SAS Usage Node 22599: "Understanding and correcting complete or quasi-complete separation problems", addresses this situation (http://support.sas.com/kb/22/599.html)

[6] The variable "Into: tgt_class" is created by the NN model and contains the decision made by the NN model to assign an observation into the class "<=50K" or ">50K". This variable is critical to the glass box process in that it links the NN model to the logistic regression model. "Into: tgt_class" is the output of the NN model and it is used as the dependent variable of the logistic regression modeling procedure.

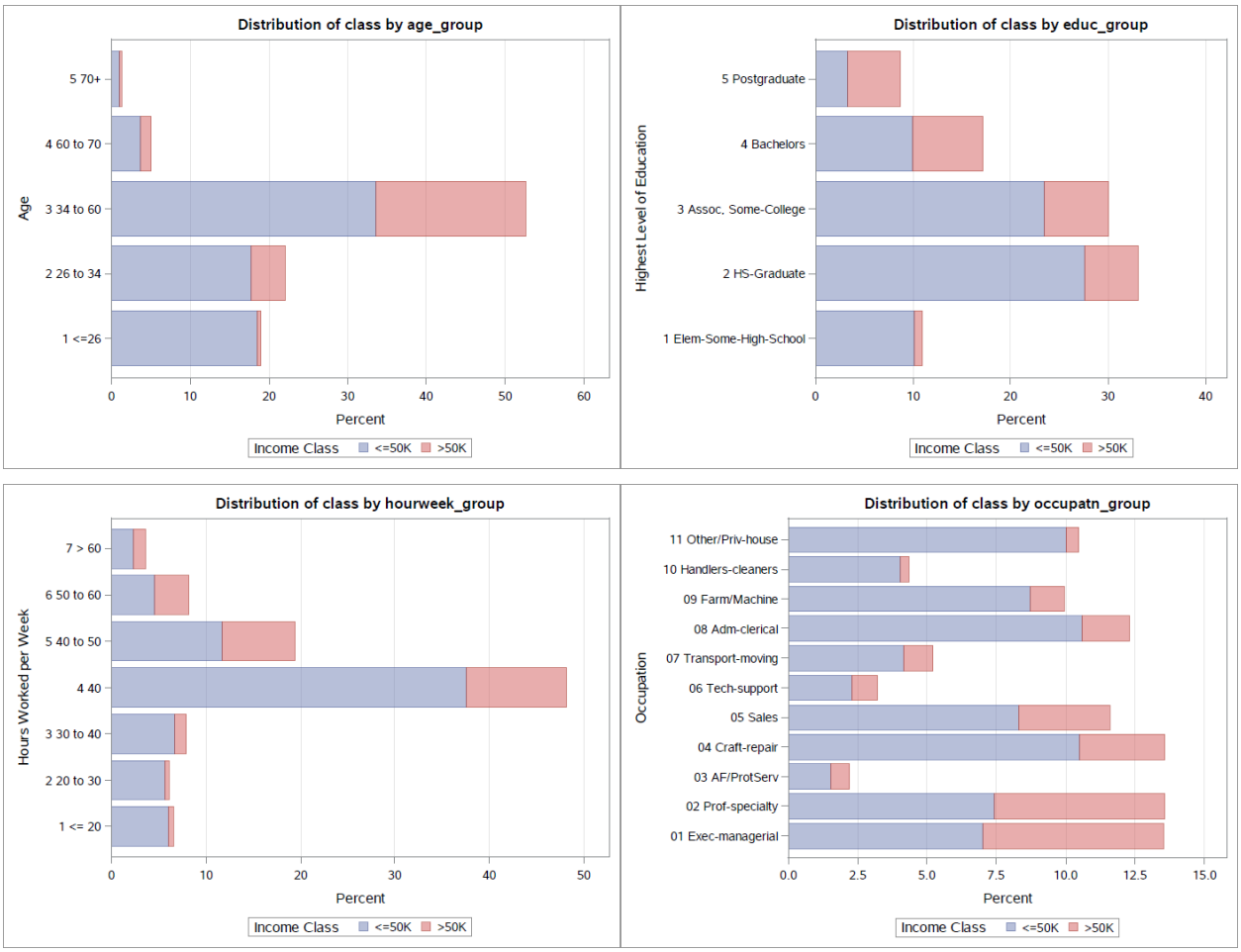

Figure 2: Stacked Histograms of Grouped Data

Neural Network Modeling

We used Enterprise Miner to build three NN models to explore the effect of complexity on classification accuracy. The multilayer perceptron neural networks had one, two, and three combination functions in the hidden layer. We tested the hypothesis that complex relationships between the target variable and the predictor variables would be better represented by more complex NN models. Figure 3 shows the ROC plot for the three models across the training, validation, and test datasets used to train and evaluate the models.

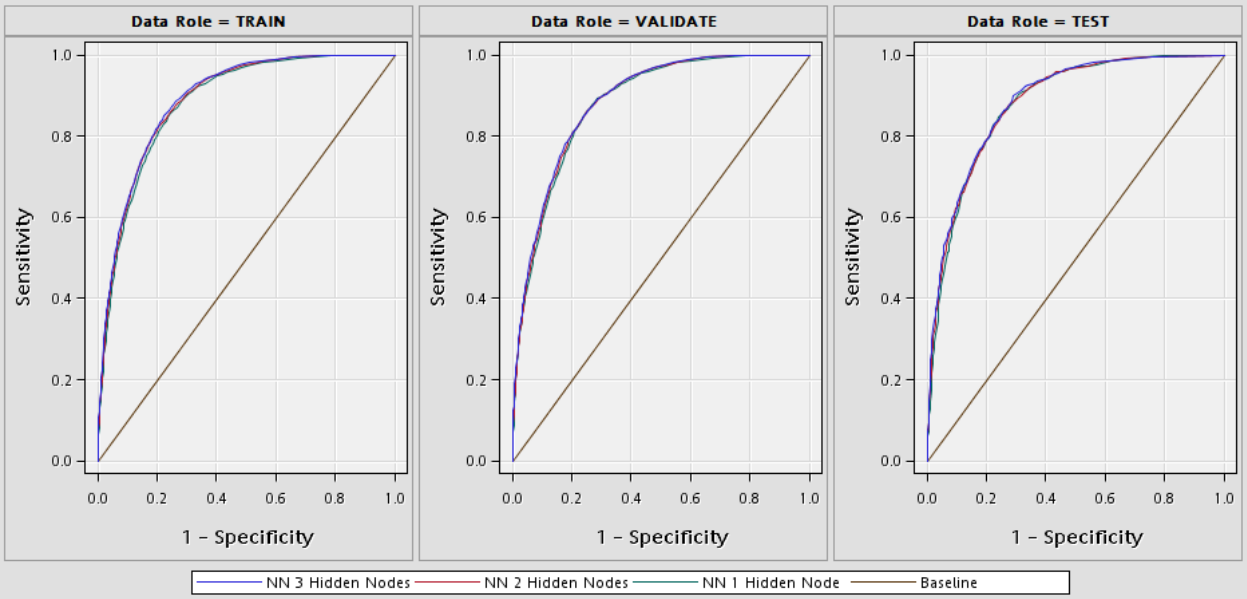

Figure 3: ROC Plots for Neural Network Models

Table 2 indicates the improvement in performance due to increased NN complexity, which is implemented by increasing the number of hidden nodes. The AUROC statistic is computed from the Test data, which represents the holdout sample and is assumed to be similar to data that would be used in scoring for a deployed model.

Table 3: Number of Hidden Nodes

Number of Hidden Nodes	Area Under ROC Plot
1	0.877
2	0.878
3	0.876

We see that there is very little improvement in classification accuracy attributable to increasing complexity, so we used Ockham's Razor[7] and invoked the principle of parsimony to select the simplest model, e.g., the NN model with one hidden node.[8]

Logistic Regression Modeling

We used the same data (training and validation datasets) and predictors (Age_Group, Educ_Group, Hourweek_Group, Occupatn_Group) that served as inputs to the NN model as input to the logistic regression model. The dependent variable for the logistic regression model was the output of the NN model, "Into: tgt_class". The resulting logistic regression model is

$$log\left(\frac{p_i}{1-p_i}\right)=\beta_0 + \beta_1 Age\_Group_i + \beta_2 Educ\_Group_i + \beta_3 Hourweek\_Group_i + \beta_4 Occupatn\_Group_i \quad [1]$$

where p_i represents the probability that observation *i* belongs to the income > $50,000 class, e.g., Into_class = 1. If we define the log of the odds ratio as the logit(p_i), then we can say that

[7] The principal of parsimony states that "Entities are not to be multiplied without necessity". See, e.g., https://en.wikipedia.org/wiki/Occam%27s_razor for historical background.

[8] It can be shown that a three-layer NN with one hidden node is equivalent to a logistic regression algorithm. See Appendix A.

$logit(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{\beta' x_i}$ so that $p_i = \frac{e^{\boldsymbol{\beta' x_i}}}{1+e^{\boldsymbol{\beta' x_i}}}$ is the probability that observation *i* is in class 1 [2].

We note that the model diagnostics indicated satisfactory performance and that the logistic regression model based on the output of the NN model represented the NN model's performance to a high degree of accuracy. The area under the ROC curve (AUROC) was 0.9324, indicating that the logistic regression model performed very well under a variety of event definitions where the probability of Into: tgt_class ranged from 0 to 1. Perfect separation of the Into:tgt_class dependent variable into disjoint subsets would produce an AUROC of 1. Figure 4 shows this performance.

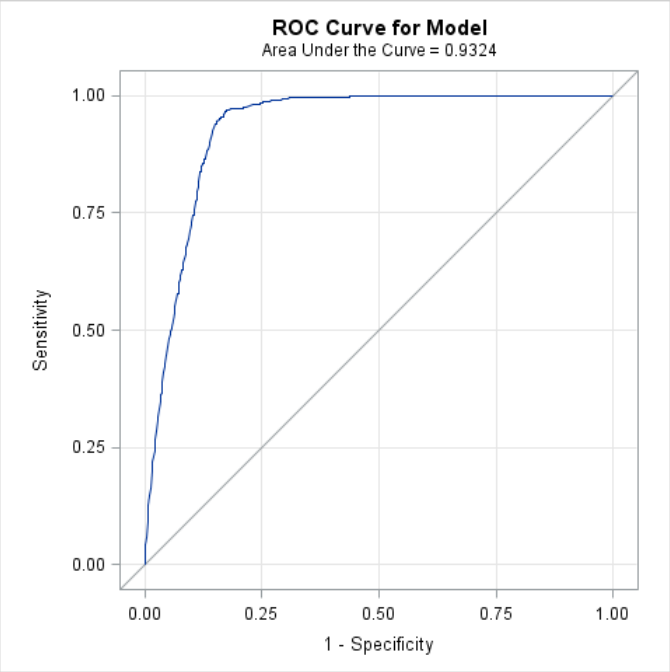

Figure 4: LR Model Based on NN Model

Interpretation of Logistic Regression Results

To simplify the interpretation of the LR results, we built a model using educational attainment alone. Table 4 shows the distribution of the dependent variable, Into:tgt_class, by category. We see that every category is populated, although "Elem-Some-High-School" is sparse for high-income observations.

Table 4: Education Group

	Into: tgt_class	
	0	1
	N	N
educ_group		
1 Elem-Some-High-School	3394	3
2 HS-Graduate	9533	622
3 Assoc, Some-College	7452	1781
4 Bachelors	2947	2367
5 Postgraduate	991	1671

The SAS code used to build the model shown in Equation 2 is

```
proc logistic data=train_validate
    plots( only )=( oddsratio( group ) roc ) ;

    class I_tgt_class educ_group( ref='4 Bachelors' ) ;

    model I_tgt_class( event = '1' ) = educ_group / rsquare ;

    oddsratio educ_group / diff=ref ;
run ;
```

The model equation is

$$\begin{aligned} logit(p_i) = \beta_0 &+\beta_1(Educ\_Group_i = '1 Elem - Some - High - School') \\ &+\beta_2(Educ\_Group_i = '2\ HS - Graduate') \\ &+\beta_3(Educ\_Group_i = '3\ Assoc, Some - College') \\ &+0\ \ (Educ\_Group_i = '4\ Bachelors') \\ &+\beta_5(Educ\_Group_i = '5\ Postgraduate') \end{aligned} \quad [2]$$

Since we used the '4 Bachelors' category as the reference value to which other categories are compared, it is not represented in the equation. Table 5 contains the parameter estimates obtained by the maximum likelihood process.

Table 5: Logistic Regression Maximum Likelihood Estimates

Analysis of Maximum Likelihood Estimates						
Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	-2.1777	0.1163	350.4081	<.0001
educ_group	1 Elem-Some-High-School	1	-4.8533	0.4623	110.2325	<.0001
educ_group	2 HS-Graduate	1	-0.5519	0.1207	20.9145	<.0001
educ_group	3 Assoc, Some-College	1	0.7464	0.1181	39.9330	<.0001
educ_group	5 Postgraduate	1	2.7002	0.1204	502.8745	<.0001

The parameter estimates from the maximum likelihood estimation process have been substituted into Eq. 2 to produce the model that represents the effect of educational attainment on achieving high income.

$$\begin{aligned} logit(p_i) = -2.1777 &-4.8533 \cdot (Educ\_Group_i = '1 Elem - Some - High - School') \\ &-0.5519 \cdot (Educ\_Group_i = '2\ HS - Graduate') \\ &+0.7464 \cdot (Educ\_Group_i = '3\ Assoc, Some - College') \\ &+0 \quad\ \cdot (Educ\_Group_i = '4\ Bachelors') \\ &+2.7002 \cdot (Educ\_Group_i = '5\ Postgraduate') \end{aligned} \quad [3]$$

Table 6 displays the odds ratio estimates computed by PROC LOGISTIC. Each category is compared to '4 Bachelors'. We recall that the odds ratio of a particular category is $OR = e^{\beta' x}$.

Table 6: Odds Ratio Estimates

Odds Ratio Estimates and Wald Confidence Intervals			
Odds Ratio	Estimate	95% Confidence Limits	
educ_group 1 Elem-Some-High-School vs 4 Bachelors	0.001	<0.001	0.003
educ_group 2 HS-Graduate vs 4 Bachelors	0.081	0.074	0.090
educ_group 3 Assoc, Some-College vs 4 Bachelors	0.298	0.276	0.321
educ_group 5 Postgraduate vs 4 Bachelors	2.099	1.908	2.309

Figure 5 graphically represents the impact of education on income.

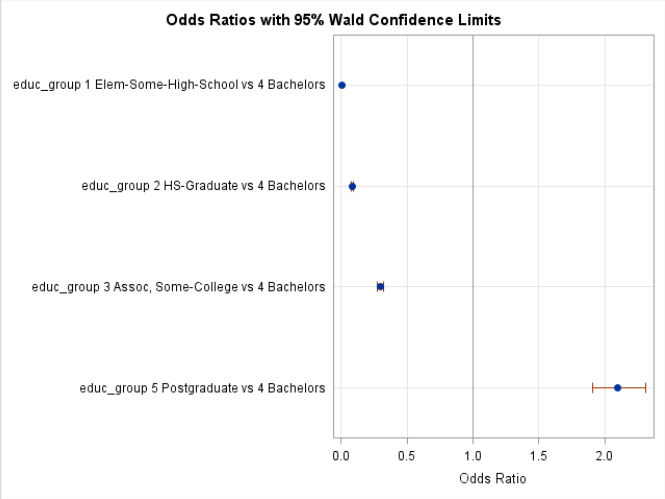

Figure 5: Odds Ratios of Education Referred to Baccalaureate

The odds ratio estimates in Table 5 can be converted into probabilities by using the relationship $p = \frac{OR}{1+OR}$.

We computed the probabilities of being in the high-income class based on educational achievement compared to a Bachelors degree and include them in Table 7:

Table 7: Probability of High Income Based on Educational Achievement

Category	**Probability**
Elem-Some-High-School vs Bachelors	0.000999
HS Graduate vs Bachelors	.0749
Assoc, Some College vs Bachelors	.2296
Postgraduate vs Bachelors	.6773

Clearly, the importance of educational achievement on earning power cannot be disregarded. The ROC plot for the logistic regression model is shown in Figure 6. The single variable Educ_Group has strong predictive power!

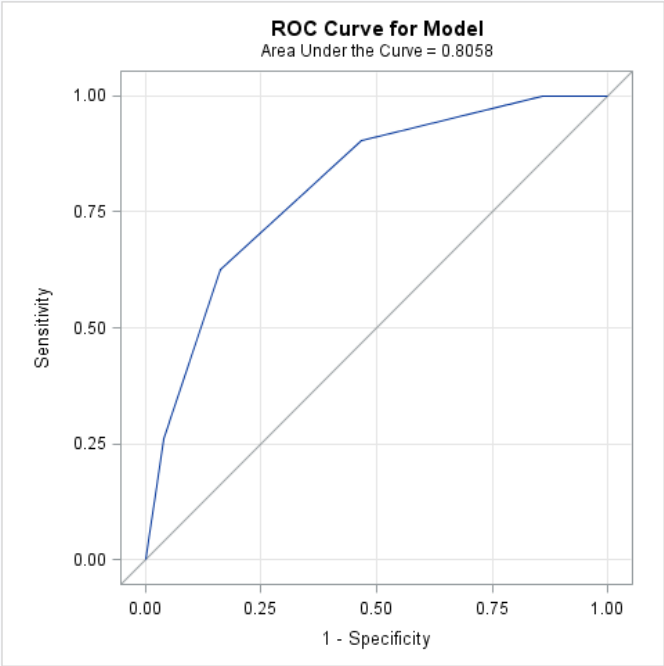

Figure 6: ROC Curve for Educ_Group Model

Summary

We propose a three-phase model building approach in the context of a classification problem. It may equally well be applied to a prediction problem for a continuous target variable.

We demonstrated the feasibility of using logistic regression to illuminate the inner workings of a simple feed-forward neural network model with a categorical target variable. By mapping NN methodology into a regression context, we converted the black box NN model into a “glass box” logistic regression model.

We believe that this technique is applicable to various modeling problems and is highly useful in understanding NN models. Converting the “black box” of the hidden layer of NN modeling into a “glass box” of regression removes the mystery of the NN model and may reduce the natural tendency to avoid what cannot be understood. We hope that this work may encourage application of NN modeling technology to a wider audience of decision makers.

Appendix

The single hidden layer perceptron shown in Figure A1 computes the signum function (Figure A2). The combination function f combines the bias and weighted inputs and produces an input to the signum function. The signum function applies thresholding to the input and creates a discrete value in the interval [-1,1].

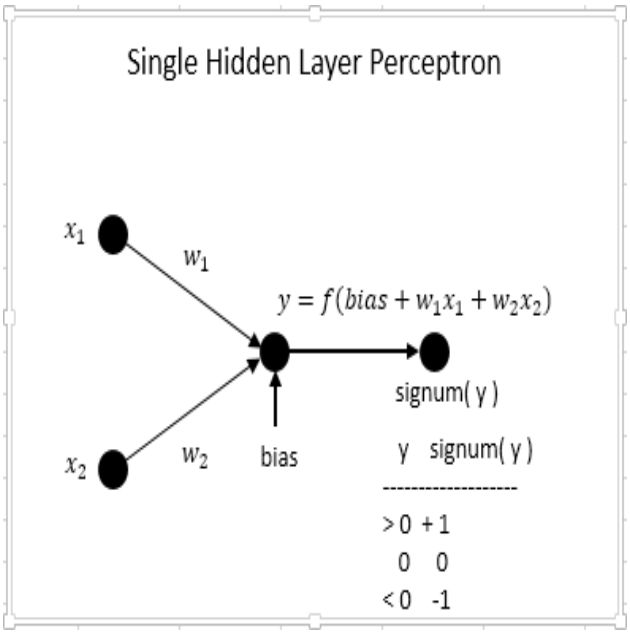

Figure A1: Single Hidden Layer Perceptron

The signum function applies hard limiting to its input. If x < 0 then signum(x) = -1, if x = 0 then signum(x) = 0, and if x > 0 then signum(x) = +1.

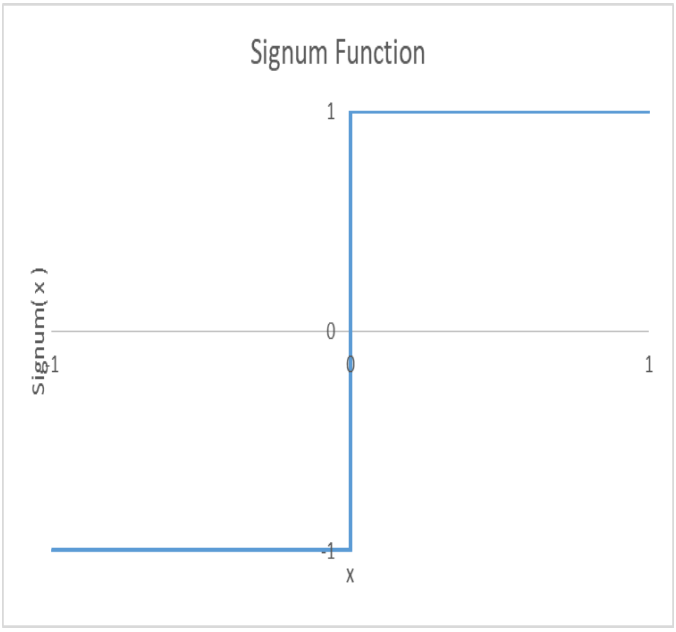

Figure A2: Signum Function

The logistic function is defined to be

$$logistic(\,x\,) = \frac{1}{1 + e^{-x}}$$

If the logistic function is used instead of the signum function, the discriminatory power of the neural network is increased because the output of logistic(x) is a continuous value in the interval $(0, 1)$. This output may be defined to be the probability of an event, which represents the outcome of some process under observation. Then we may say that, if the probability of an event is, e.g., 0.75, and if the output of the neural network is 0.80, the label applied to the event is "Occurred". Otherwise, it did not occur if the threshold of 0.75 is not exceeded, and the label is "Nonoccurrence". In this case, the event is binary-valued. Other definitions are possible, based on the number of states (labels) that an event can represent.

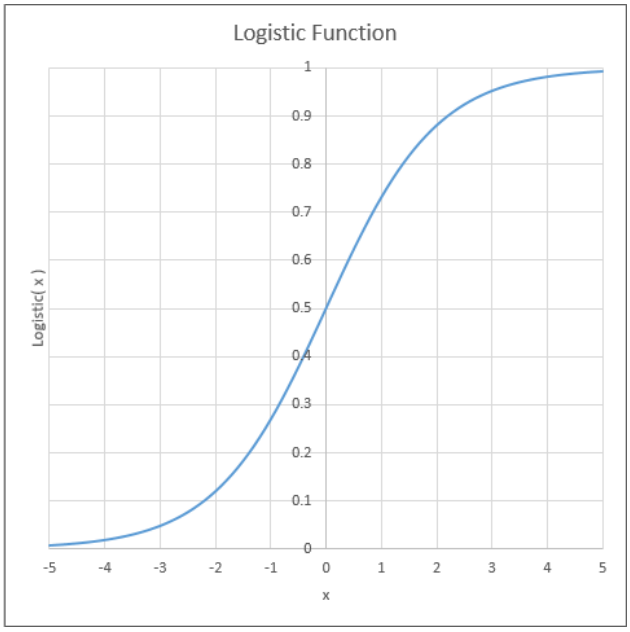

Figure A3: Logistic Function

Then the neural network equation

$$y = f(bias + w_1x_1 + w_2x_2)$$

has the same structure as the logistic regression equation

$$logit(\frac{p}{1-p}) = \beta_0 + \beta_1x_1 + \beta_2x_2$$

where $f(x) = logistic(x)$ and the equivalence between a neural network model and a logistic regression model is apparent.

References

[1] Kohavi, Ronny and Becker, Barry (1996). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml/datasets/Adult]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Allison, Paul D. (1999). *Logistic Regression Using the SAS® System: Theory and Application.* Cary, NC: SAS Institute Inc.

Acknowledgements

We thank Garrett Frere and Mark Leventhal for their generosity in taking the time to review this document.

Contact Information

Your comments and questions are valued and encouraged. Contact the author at:

Name: Ross Bettinger

Enterprise: Consultant

E-mail: rsbettinger@gmail.com