# A Flexible Generalized Link Function for Credit Scoring

D. A. de Waal[a] J. V. du Toit[b] T. de la Rey[c]

[a]Centre for Business Mathematics and Informatics, North-West University, Potchefstroom Campus, Private Bag X6001, Potchefstroom, 2520, South Africa, +27 18 2992535, Andre.DeWaal@nwu.ac.za
[b]School of Computer, Statistical and Mathematical Sciences, North-West University, Potchefstroom Campus, Private Bag X6001, Potchefstroom, 2520, South Africa, +27 18 2992548, Tiny.DuToit@nwu.ac.za
[c]Centre for Business Mathematics and Informatics, North-West University, Potchefstroom Campus, Private Bag X6001, Potchefstroom, 2520, South Africa, +27 18 2992566, Tanja.DelaRey@nwu.ac.za

**Abstract**

Logistic regression is the most popular tool used to develop scorecards in the financial industry. It provides, in combination with the weights of evidence (WOE) measure and the traditional scorecard format, an easy to interpret development methodology that is widely understood and accepted by most regulatory authorities because of its transparency. The current financial crisis has however highlighted the need for better risk management and one building block may be more accurate scorecards. The current methodology provides very little leeway for improvement, as it is clearly defined and standard statistical software packages are usually used during the process. The most obvious areas where a company can distinguish itself is in better characteristic selection and the grouping of attributes. But, only concentrating on these areas may not provide the scope of improvement demanded by risk managers. One area that has received little attention is the selection of the optimal link function: usually the logit link function is assumend to be the *de facto* standard for building a good scorecard. This however should not always be the case. Non-symmetrical link functions may be more appropriate for data where one response is much more frequent than the other (as it happens in the credit scoring context). Various authors have also proposed different link functions with varying degrees of success. The ideal solution would be a data driven approach where the link function is chosen to suit the data. As neural networks have been used before to compute the optimal independent variable transformations for a generalized additive model (given some data set), the question now arises whether this approach can be generalized to also compute an optimal link function (the link function is also a univariate function and just a transformation of the dependent variable). This paper investigates the proposed approach of computing an optimal link function using a neural network. The approach extends the previous work on generalized additive neural networks to also include the output activation function (the inverse of the link function). Some of the consequences of this approach are also discussed (e.g. what effect does the use of a neural network to compute the link function have on transparency) and illustrated with an example from the credit scoring domain.

## 1 Introduction

Logistic regression originated in the early nineteenth century (Cramer, 2002) and the logistic regression model based on the logit link function is currently the most popular tool used for constructing scorecards (Hand, 2004). Other frequently used functions include the probit and complementary log-log link functions. However, these popular link functions do not always provide the best option given some dataset. Chen, Dey and Shao (1999) used the rates at which the probability of a given binary response approaches 0 and 1 to describe the link function. The logit and probit link functions are symmetric link functions, whereas the complementary log-log link function is positively skewed (the response curve approach 0 slowly, and 1 sharply). Symmetric link functions do not provide good fits for data where one response is much more frequent than the other (as it happens in the credit scoring context).

Various authors have proposed different link functions. Stukel (1988) suggested a class of generalized logistic models for modelling binary data. These models are very general, and several

important and commonly used symmetric and asymmetric link models can be approximated by members of this family. Wang and Dey (2008) built an appropriate and extremely flexible model for binary data, with the link function based on the generalized extreme value distribution. Kim, Chen and Dey (2008) introduced a new link function based on the generalized t-distribution. To the authors' knowledge, these different types of link functions have not yet been applied to the credit scoring field.

In this paper a data driven approach is followed where the link function is chosen to suit the data. Du Toit (2006) utilized neural networks to compute the optimal independent variable transformations for a generalized additive model given some data set. The question now arises whether this approach can be generalized to also compute an optimal link function? The latter is also a univariate function and just a transformation of the dependent variable. The approach extends the previous work on generalized additive neural networks (De Waal, Du Toit and De La Rey, 2005) to also include the output activation function (the inverse of the link function).

In Section 2, background information on generalized additive neural networks are given. A flexible link function is introduced in Section 3. Section 4 contains a credit scoring example and Section 5 generalises the approach to include multiple scorecards. The article ends with conclusions and ideas for future research.

## 2 Generalized Additive Neural Network Architecture

A logistic regression model (Hosmer and Lemeshow, 1989); (Kleinbaum, 1994) is defined as

$$g_0^{-1}(E(y)) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k,$$

where the expected dependent variable (on the link scale) is expressed as the sum of the products of unknown parameters and the independent variables. The output activation function (the inverse of the logit link function) is defined as

$$g_0(z) = \frac{1}{1 + e^{-z}}.$$

Figure 1 contains a graphical representation of the model implemented as a neural network.

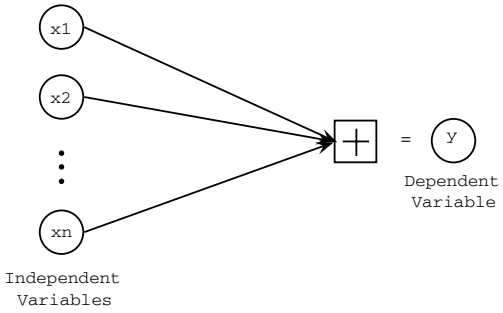


Figure 1: Logistic regression model

The direct connections between the independent variables and the target variable represent the parameters that must be estimated: $\beta_1, \beta_2, \ldots, \beta_n$. Note also that a bias, $\beta_0$, is connected to the dependent variable, Y. This neural network architecture represents the "classical" way of implementing logistic regression as a neural network.

Hastie and Tibshirani (1990) define a generalized additive model as

$$g_0^{-1}(E(y)) = \beta_0 + f_1(x_1) + f_2(x_2) + \ldots + f_k(x_k),$$

where the expected target (also on the link scale) is expressed as the sum of individual unspecified univariate functions. Sarle (1994) proposed the neural network representation in Figure 2 for a generalized additive model.

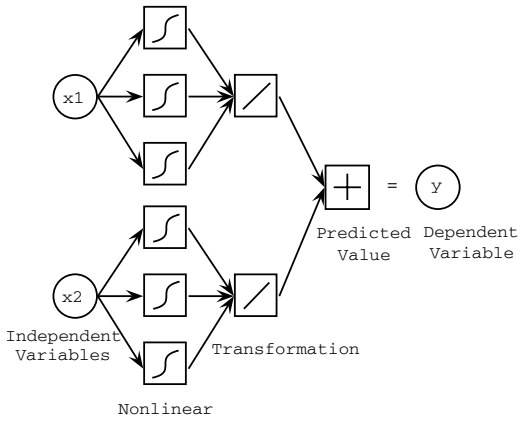


Figure 2: Generalized Additive Neural Network

As MLPs are universal approximators that can model any continuous function (Ripley, 1996), Potts (1999) suggested an MLP that has a single layer with $h$ hidden neurons for each input. Furthermore, the basic architecture of Sarle was enhanced with an additional parameter for a direct connection (skip layer) so that the linear model is a special case. The enhanced architecture has the form

$$f_j(x_j) = w_{0j}x_j + w_{1j}\tanh(w_{01j} + w_{11j}x_j) + \ldots + w_{hj}\tanh(w_{0hj} + w_{1hj}x_j).$$

The updated generalized additive neural network (GANN) is shown in Figure 3.

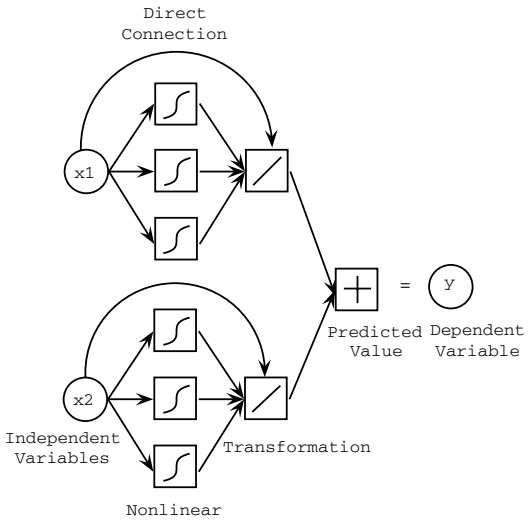


Figure 3: Enhanced Generalized Additive Neural Network

This generalized additive neural network architecture can now be further enhanced with an additional hidden and output layer that computes the output activation function. This new architecture is described in more detail in the following section.

## 3 A Flexible Link Function

In the architecture proposed by Potts, the output activation function is specified by the inverse of the link function. Both these functions are also univariate functions and consequently the output activation function can be approximated with a neural network in a similar manner as the univariate functions of the independent variables. The enhanced architecture of Potts is now further generalized with additional nodes implementing the output activation function. This generalized architecture is shown in Figure 4 with the inverse of the link function (output activation function) accentuated by dotted lines.

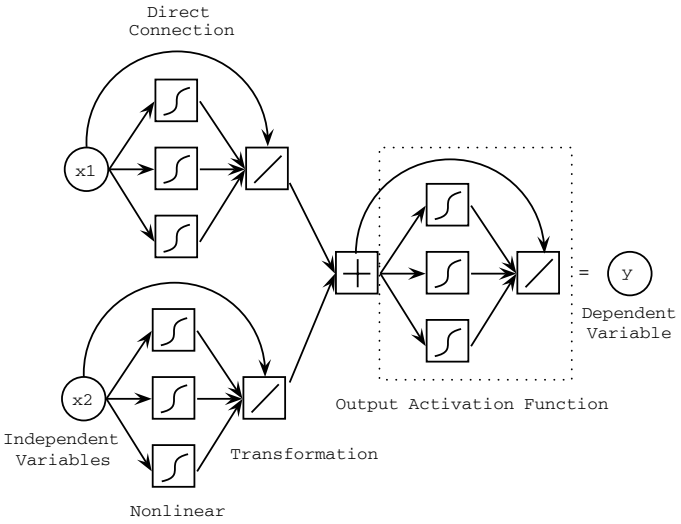


Figure 4: Flexible Link Generalized Additive Neural Network

Note that the previous architecture of Potts is a special case of this architecture: the part of the network computing the output activation function only has a direct connection with the parameter set equal to 1 and the final transformation implements the known output activation function. However, in the rest of the paper the direct connection is ignored to simplify the presentation.

The success of this proposed architecture is critically dependent on a suitable choice of activation functions for the hidden nodes as well as the final node in the architecture. As the result will be interpreted as a probability, it will be prudent to select the activation functions with this in mind.

Selecting the logistic activation function

$$l(n) = \frac{1}{1 + e^{-n}} \tag{1}$$

for the hidden nodes restricts the hidden nodes' output to $(0, 1)$. A suitably chosen linear combination of the outputs of the hidden nodes leads to the required result. Assuming that the links connecting the hidden nodes to the output node are represented by parameters $c_1, c_2, \ldots, c_p$, the following constraints

$$c_1 \geq 0, c_2 \geq 0, \ldots, c_p \geq 0 \tag{2}$$

$$c_1 + c_2 + \ldots + c_p = 1 \tag{3}$$

scales the output to $(0, 1)$.

The output activation function given by the described architecture is therefore

$$g_0(z) = \frac{c_1}{1 + e^{-\alpha z}} + \frac{c_2}{1 + e^{-\beta z}} + \ldots + \frac{c_p}{1 + e^{-\rho z}} \tag{4}$$

where $c_i$'s and $\alpha, \ldots, \kappa$ are weights chosen by the neural network with constraints (2) and (3) and

$$z = \beta_0 + f_1(x_1) + f_2(x_2) + \ldots + f_k(x_k). \tag{5}$$

Due to restrictions on the combination of error function and activation function that may be specified in the last node of the proposed architecture, and the inability to specify constraints on parameters in the system used to implement the neural network, the following approximation to the output activation function described above was actually implemented. The linear combination function was replaced by the logistic activation function. If the logistic activation function is broken down into two steps, namely the computation of a linear combination of the outputs from the hidden nodes and the scaling of the result to $(0, 1)$, the reasons for choosing this function is clear. The first step implements the linear combination function as required and the second step scales the result to $(0, 1)$, which allows for the removal of the constraints stated above as the output is now

guaranteed to lie in the interval $(0, 1)$. So in effect the choice of logistic activation function for the final node drastically simplifies the implementation. The implemented function is therefore

$$g_0(z) = \frac{1}{1 + e^{-(\beta_0 + \frac{c_1}{1+e^{-z}} + \frac{c_2}{1+e^{-\beta z}} + ... + \frac{c_p}{1+e^{-\rho z}})}}. \tag{6}$$

The $\beta_0$ is required in the formula to allow for the correct scaling of the linear combination of outputs from the hidden nodes and the link to the first hidden node was set equal to 1 to guarantee that the result is a smooth sigmoid function and not a step function. Although the developed output activation function looks very complex, its components are easy to understand and it provides a flexible function that may be applied to any classification problem. The inverse of this output activation function (6) is the final link function. Although the transparent interpretation of the link function as the log of the odds does not hold any more, the proposed method may provide a more accurate rank ordering of the cases. For many applications this is sufficient and the odds can be inferred from the estimated probabilities. In the next section the developed output activation function is applied to an example data set from the credit scoring domain.

## 4 Credit Scoring Example

With the Home Equity data set analyzed by Wielenga, Lucas and Georges (1999), the aim is to predict whether an applicant will eventually default or be seriously delinquent on a loan that allows owners to borrow against the equity in their homes. The data set contains actual loan performance information for 5 960 recent home equity loans. Information that could identify the borrowers was removed. The binary dependent variable ($DEFAULT$) indicates whether an applicant eventually defaulted or was seriously delinquent and occurred in approximately 20% (1 189 cases). There are 12 independent variables: $REASON$, home improvement or debt consolidation, $JOB$, six occupational categories, $LOAN$, loan amount requested, $MORTDUE$, amount due to existing mortgage, $VALUE$, value of current property, $DEBTINC$, debt to income ratio, $YOJ$, years at present job, $DEROG$, number of derogatory reports, $CLNO$, number of trade lines, $DELINQ$, number of delinquent trade lines, $CLAGE$, age of oldest trade line in months, and $NINQ$, number of recent credit enquiries.

De Waal et al. (2005) partitioned the data into training (67%) and validation (33%) sets with the missing interval values imputed with their mean and the missing class values imputed with their most frequently occuring values. A search for the best GANN model was then conducted. The best model found was used to build a scorecard on the data set which is compared to one created using only standard logistic regression.

To determine the effect of different link functions on a model, the data set is partitioned and missing values are imputated as in De Waal et al. (2005). A baseline GANN model with a logit link function is then created in a preprocessing step. Table 1 shows the variables included in the constructed model, each variable's GANN architecture and the relationships between the selected variables and the target. From the table it can be seen that three (independent) variables were removed, three variables have linear relationships with the target and six variables have nonlinear relationships with the target.

| Variable | GANN architecture | Relationship with target |
|---|---|---|
| *LOAN* | MLP with a skip layer and 1 hidden node | Nonlinear |
| *JOB* | No MLP (variable removed from the model) | None |
| *REASON* | No MLP (variable removed from the model) | None |
| *CLAGE* | MLP with a skip layer and 1 hidden node | Nonlinear |
| *CLNO* | MLP with a skip layer | Linear |
| *DEBTINC* | MLP with a skip layer and 1 hidden node | Nonlinear |
| *DELINQ* | MLP with a skip layer | Linear |
| *DEROG* | MLP with a skip layer and 1 hidden node | Nonlinear |
| *NINQ* | MLP with a skip layer and 1 hidden node | Nonlinear |
| *MORTDUE* | MLP with a skip layer | Linear |
| *VALUE* | MLP with a skip layer and 1 hidden node | Nonlinear |
| *YOJ* | No MLP (variable removed from the model) | None |

Table 1: Baseline GANN architecture

The baseline model has a misclassification rate of 15.95%, an average squared error (ASE) of 0.1219 and a Gini coefficient of 0.5771 on the validation set. Figure 5 shows the corresponding output activation function where *Transform1* denotes $\beta_0 + f_1(x_1) + f_2(x_2) + \ldots + f_k(x_k)$, and $p\_1$ the predicted probabilities. As expected the output activation function produces a value of 0.5 for an input of 0.

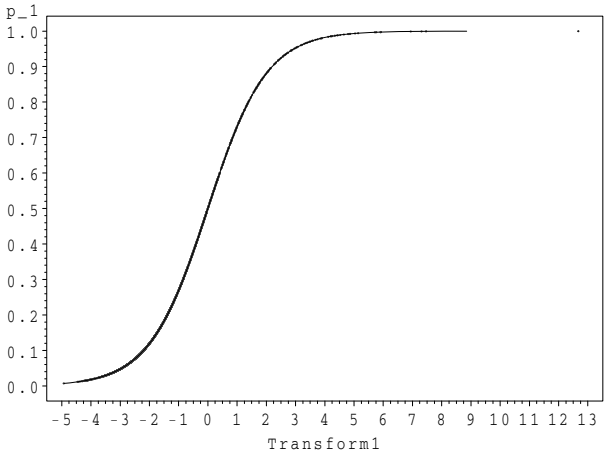


Figure 5: Baseline GANN output activation function

Next, the baseline model is adjusted so that the output activation function has two hidden nodes (Figure 6).

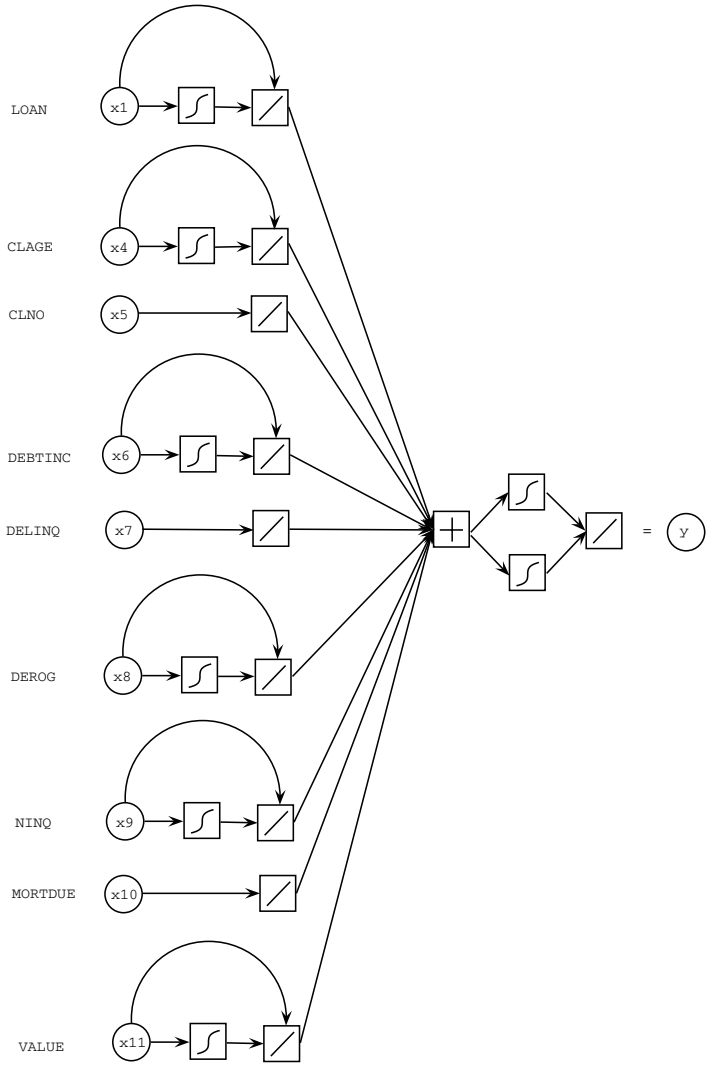


Figure 6: Adjusted baseline GANN architecture

Figure 7 shows the output activation function. This GANN model has a misclassification rate of 12.50%, an ASE of 0.0926 and a Gini coefficient of 0.7371 on the validation set. Figure 8 shows the two logistic basis functions that are combined to form the output activation function.

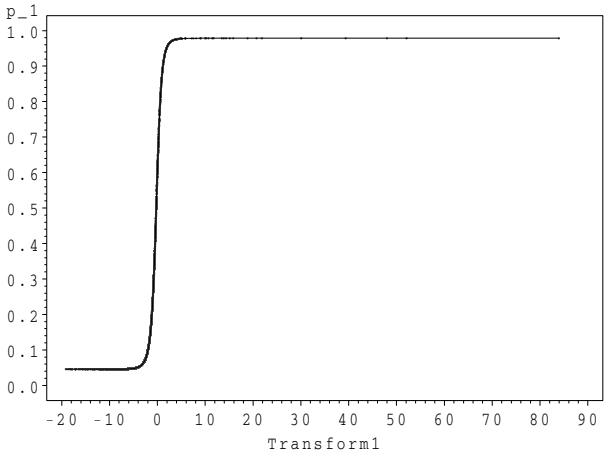


Figure 7: Output activation function for 2 hidden nodes

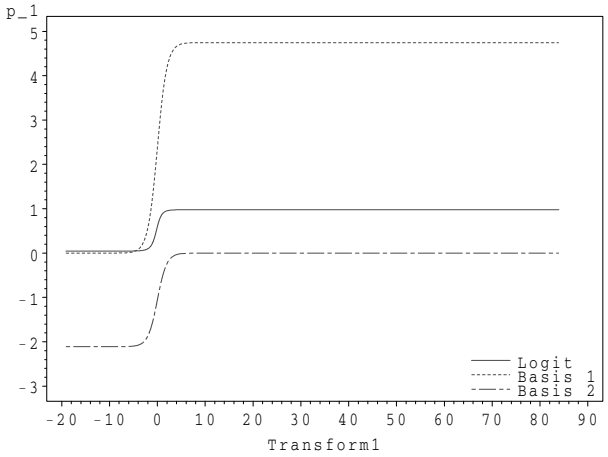


Figure 8: Basis functions for link function with 2 hidden nodes

Finally, the baseline model is adapted to three hidden nodes in the output activation function. The latter is shown in Figure 9. This model has a misclassification rate of 13.26%, an ASE of 0.0966 and a Gini coefficient of 0.7251 on the validation set. The three basis functions that are combined to form the output activation function is given in Figure 10.

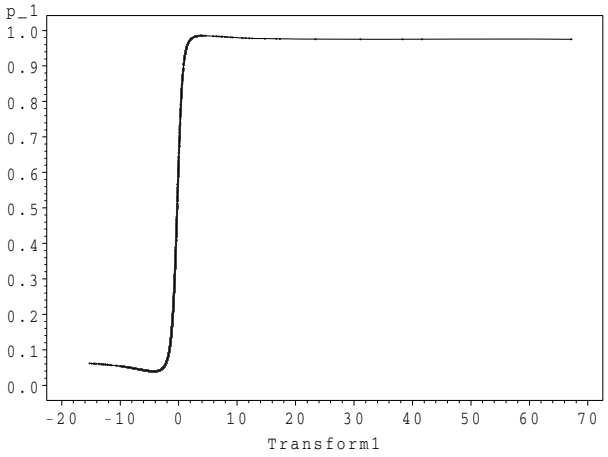


Figure 9: Output activation function for 3 hidden nodes

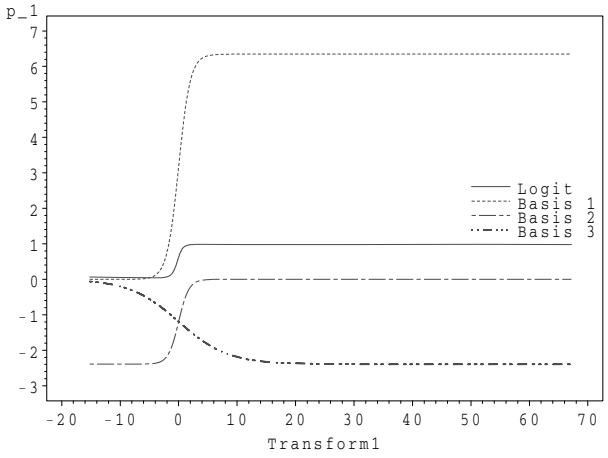


Figure 10: Basis functions for link function with 3 hidden nodes

Figure 11 denotes the three output activation functions (Figures 5, 7 and 9) superimposed on the interval [-4, 4].

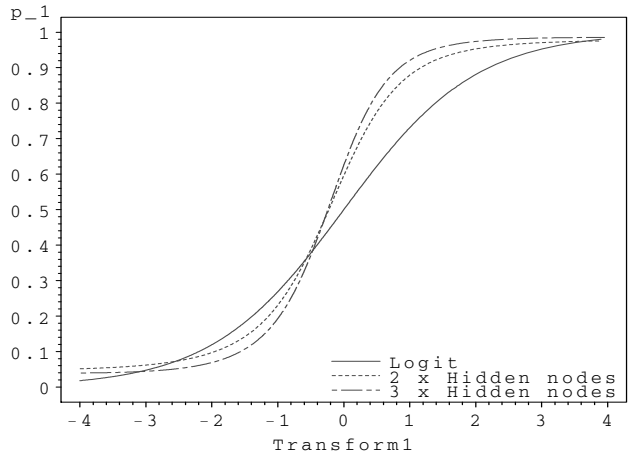


Figure 11: Output activation functions superimposed on interval [-4, 4]

Finally, Figure 12 denotes the three models' ROC curves superimposed on one graph. A summary of the results can be found in Table 2.

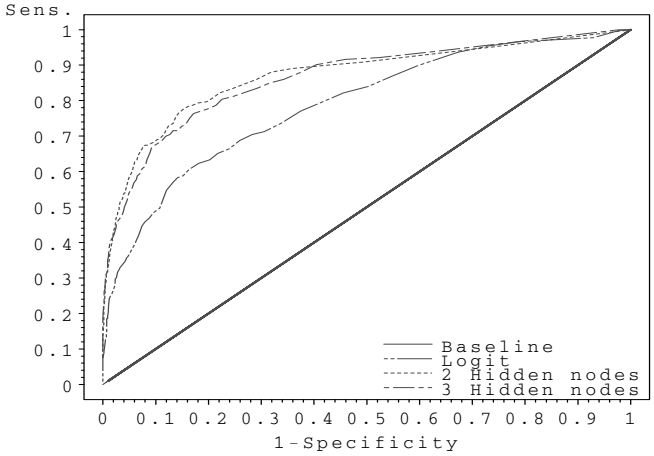


Figure 12: ROC curves superimposed

| Model | ASE | Misclassification rate | Gini coefficient |
|---|---|---|---|
| Baseline GANN | 0.1210 | 15.95% | 0.5771 |
| Baseline with 2 hidden nodes | 0.0926 | 12.50% | 0.7371 |
| Baseline with 3 hidden nodes | 0.0966 | 13.26% | 0.7251 |

Table 2: Summary of results

The three conducted experiments demonstrate that it is possible to create a flexible link function by means of a neural network. Furthermore, the 28% increase in Gini coefficient obtained with the more flexible link function shows that a substantial increase in accuracy may be obtained with relatively few extra parameters.

# 5 Two Scorecard Architecture

The generalized additive neural network architecture with the flexible link function given in Figure 4 can be further generalized to include more than one model. One of the first decisions that must be made during the development of a scorecard is whether the data set must be segmented and a model or scorecard developed for each segment. The architecture in Figure 13 implements this idea. Two generalized additive models are estimated in parallel and the results (compared to the results of using the architecture in Figure 4) will indicate whether it is necessary to segment the data set and develop two models or scorecards.

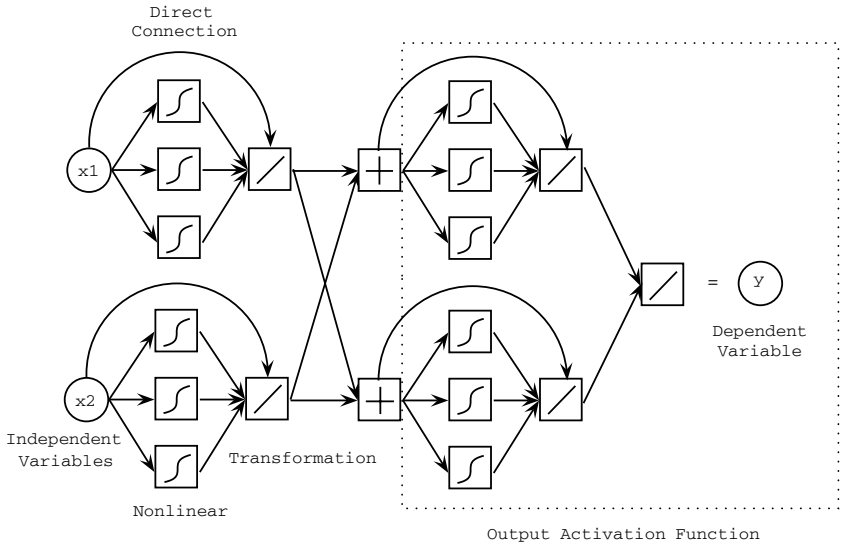


Figure 13: Two Scorecard Architecture

# 6 Conclusions

The flexible generalized link function proposed in this paper was implemented as an extension to the AutoGANN modelling node previously developed by the authors. It allows for easy comparison of different neural network architectures and link functions and ranks models according to Gini coefficient. The plots given in this paper are part of the standard output of the node and provides visual information that assists in evaluating the computed models.

Future work include a detailed evaluation of the proposed generalization of the logit link function on more complex data sets. Also, the consequences of replacing the logit link function by a more flexible link function in the standard credit scoring methodology of using weights of evidence will be investigated. Although the interpretation of the results might become more complex, the increased accuracy promised by a more flexible link function could have financial benefits that outway the drawbacks.

# Acknowledgements

The authors wish to thank SAS® Institute for providing them with Base SAS® and SAS® Enterprise Miner™ software used in computing all the results presented in this paper. This work forms part of the research done at the North-West University within the TELKOM CoE research programme, funded by TELKOM, GRINTEK TELECOM and THRIP.

## References


Chen, M.-H., Dey, D. K. and Shao, Q.-M. (1999), 'A new skewed link model for dichotomous quantal response data', *Journal of the American Statistical Association* **94**(448), 1172–1186.

Cramer, J. S. (2002), The origins of logistic regression, Discussion Paper TI 2002-119/4, Tinbergen Institute, Faculty of Economics and Econometrics, University of Amsterdam, Tinbergen Institute.

De Waal, D. A., Du Toit, J. V. and De La Rey, T. (2005), An investigation into the use of generalized additive neural networks in credit scoring, *in* 'Proceedings of Credit Scoring & Credit Control IX, Edinburgh, Scotland', Pollock Halls, University of Edinburgh, Scotland.

Du Toit, J. V. (2006), Automated Construction of Generalized Additive Neural Networks for Predictive Data Mining, PhD thesis, School for Computer, Statistical and Mathematical Sciences, North-West University, South Africa.

Hand, D. J. (2004), Credit scoring, *in* J. Teugels and B. Sundt, eds, 'Encyclopedia of Actuarial Science', Vol. 1, Wiley, Chichester, pp. 410–441.

Hastie, T. J. and Tibshirani, R. J. (1990), *Generalized Additive Models*, Vol. 43 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, London.

Hosmer, D. W. and Lemeshow, S. (1989), *Applied logistic regression*, Wiley series in probability and mathematical statistics, Wiley, New York.

Kim, S., Chen, M.-H. and Dey, D. K. (2008), 'Flexible generalized t-link models for binary response data', *Biometrika* **95**(1), 93–106.

Kleinbaum, D. G. (1994), *Logistic regression: a self-learning text*, Springer series in statistics, Springer, New York.

Potts, W. J. E. (1999), Generalized additive neural networks, *in* 'Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 194–200.

Ripley, B. D. (1996), *Pattern Recoqnition and Neural Networks*, Cambridge University Press, Cambridge, United Kingdom.

Sarle, W. S. (1994), Neural networks and statistical models, *in* 'Proceedings of the Ninteenth Annual SAS® Users Group International Conference'.

Stukel, T. A. (1988), 'Generalized logistic models', *Journal of the American Statistical Association* **83**(402), 426–431.

Wang, X. and Dey, D. K. (2008), A flexible skewed link function for binary response data, Technical Report 2008-5, Statistical and Applied Mathematical Sciences Institute, Research Triangle Park, NC.

Wielenga, D., Lucas, B. and Georges, J. (1999), *Enterprise Miner$^{TM}$: Applying Data Mining Techniques Course Notes*, SAS Institute Inc., Cary, NC.