VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Mathematical and Computing Sciences
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 471 5341, Fax: +64 4 495 5045
Email: Tech.Reports@mcs.vuw.ac.nz
Http://www.mcs.vuw.ac.nz/comp/Publications

Visualizing the function computed by a feedforward neural network

Tony Plate (VUW)
Joel Bert (UBC)
John Grace (UBC)
Pierre Band (Health Canada)

Technical Report CS-TR-98-5
July 29, 1998

Abstract

A method for visualizing the function computed by a feedforward neural network is presented. It is most suitable for models with continuous inputs and a small number of outputs, where the output function is reasonably smooth, as in regression or probabilistic classification tasks. The visualization makes readily apparent the effects of each input and the way in which the functions deviates from a linear function. The visualization can also assist in identifying interactions in the fitted model. The method uses only the input-output relationship and thus can be applied to any predictive statistical model, including bagged and committee models, which are otherwise difficult to interpret. The visualization method is demonstrated on a neural-network model of how the risk of lung cancer is affected by smoking and drinking.

Author Information

Tony Plate
School of Mathematical and Computing Sciences
Victoria University of Wellington, New Zealand
tap@mcs.vuw.ac.nz

Joel Bert
John Grace
Dept of Chemical Engineering
University of British Columbia
2216 Main Mall, Vancouver, BC, Canada V6T 1Z4

Pierre Band
Environmental Health Centre, Health Canada
Ottawa, Ontario, Canada, K1A OL2

1 Introduction

In application settings, neural networks are commonly regarded as "black-boxes", because of the difficulty in understanding both the function computed, and the way in which it was computed. Even if neural networks perform better than more interpretable models, their "black-box" nature can make them less valuable. For example, in a recent series on medical applications of neural network in the Lancet, many authors and correspondents expressed their fear in trusting important decisions to systems they do not understand (Wyatt 1995; Sharp 1995; Dodds 1995). Thus, the development of good methods for interpreting neural networks could make them far more useful.

Researchers have proposed a number of methods for understanding feed-forward neural networks. These fall at various points along a "how-what" spectrum. "How" methods aim to provide an understanding of how the network computes a function, usually by interpreting weights and hidden unit activations. "What" methods ignore the internal workings and just aim to describe the function computed by the network. Methods such as contribution analysis (Shultz, Oshima-Takane, and Takane 1995; Shultz and Elman 1994; Sanger 1989), which are designed to provide an interpretable view of the internal representational space, fall firmly at the "how" end of the spectrum. Rule extraction methods that translate hidden unit connectivities into rules, e.g., Towell and Shavlik (1993), Blasig (1994), fall somewhere in the middle of the spectrum. Other rule extraction methods that treat the network as a black box and derive a description of its function in the form of rules, e.g., Saito and Nakano (1988) fall at the "what" end of the spectrum. In general, rule extraction methods appear to be more suited to understanding networks for clear-cut classification tasks. For continuous-output tasks, including classification tasks in which classes overlap so much that outputs are best described probabilistically, it seems more appropriate to use "what" methods that provide some quantitative or graphical indication of the effect of input variables on the output. Baxt (1992), Baxt and White (1995), and Moseholm, Taudorf, and Frosig (1993) describe some such methods. The method proposed in this paper is similar to these latter methods, but is intended to overcome some of their limitations by adapting the graphical plots of generalized additive models (Hastie and Tibshirani 1990) to the display of functions computed by neural networks and other flexible statistical models.

2 Background

Before proceeding further we review generalized additive models and previously proposed methods for graphically displaying the effects of inputs on neural networks.

2.1 Plots of generalized additive models

Part of the attraction of Hastie and Tibshirani's (1990) generalized additive models (GAMs) is that, although they are non-linear, they are easily interpreted from simple graphical displays. A generalized additive model can be expressed as

$$g(\mathbf{x}) = h(\theta_1(x_1) + \theta_2(x_2) + \cdots + \theta_k(x_k))$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ is the vector of input values and $g(\mathbf{x})$ is the predicted output value. The functions θ_1, θ_2 etc can be arbitrary non-linear functions. The function h relates the sum of the θ_i to the output variable. The choice of h depends on the distribution of the target variable and the distribution of errors in the target values. For example, for continuous target variables with Gaussian errors of constant variance, the identity function is appropriate for h. For binary valued target variables, the logistic function $h(a) = 1/(1+e^{-a})$

is usually appropriate: it gives the probability of the target being 1. The possibility of different functions for h makes GAMs "generalized"; appropriate functions exist for a wide range of error distributions. The way the functions of the inputs combine makes GAMs "additive". Although the effect of a particular input can be non-linear, there are no interactions between inputs — the contribution to the sum of θ_i of a particular input does not depend on the values of other inputs. (However, one can use functions of more than one variable in a GAM, which allows for interactions between the inputs in the same function).

A GAM is easily interpreted from plots of the effects of inputs on the additive scale of the model, i.e., plots of θ_i. The θ_i are usually functions of one variable, and hence are simple to display. Note that the effects of the inputs on the response scale of the model, i.e., the range of h, may not be additive and cannot be easily displayed. For the remainder of this paper, I refer to the domain of h, i.e., the range of the θ_i, as the *natural additive scale* of the model.

2.2 Plots and quantitative summaries of the effects of inputs in neural networks

One straightforward way to graphically illustrate the function computed by a neural network is to plot two-dimensional surfaces showing the response of the network to varying values for two particular inputs, while the rest of the inputs are held at some constant value. Moseholm, Taudorf, and Frosig (1993) show pairwise surface plots for their model of how various conditions affect lung function. Each plot shows the effect of different values of a pair of variables while the others are held at their mean values. This type of display can illustrate main effects and pairwise interactions. However, it fails to illustrate whether or not other variables are interacting with the pair displayed: it is possible that the shape of the surface varies greatly with the values of the other inputs. Thus, this type of display could possibly give misleading impressions of the effects of inputs. An additional drawback is that the number of pairwise plots becomes unwieldy when there are more than a few inputs.

Baxt (1992) introduced a technique for quantitatively summarizing, in single number, the effect of a particular input on the output of a neural network. He defines the *average effect* of a particular input as the average change in the output when that input is perturbed, for the examples in a sample. The average effect for input i is defined as

$$\bar{\Delta}_i \hat{f} = \frac{1}{n} \sum_{p=1}^{n} \mathrm{sign}(\delta_i(x_i^{(p)}) - x_i^{(p)}) \left[\hat{f}(x_1^{(p)}, \ldots, x_{i-1}^{(p)}, \delta_i(x_i^{(p)}), x_{i+1}^{(p)}, \ldots, x_k^{(p)}) - \hat{f}(\mathbf{x}^{(p)}) \right]$$

where $\hat{f}$ is the function computed by the network, δ_i is a function that perturbs values of ith input variable X_i, n is the number of training cases, and $\mathbf{x}^{(p)}$ is the vector of k input values $(x_1^{(p)}, \ldots, x_k^{(p)})$ for the pth training case. The input-perturbation function δ_i must be chosen appropriately. When X_i is a binary variable the choice is unproblematic: δ_i should swap the value. However, for continuous X_i the best definition of $\delta(x_i^{(p)})$ is not clear. Baxt uses a randomly chosen value of X_i.

Baxt's technique is well suited to summarizing the effects of binary variables. However, like some other variants of contribution and sensitivity analysis, this technique is not so well suited to describing the effects of continuous variables, because it does not relate the magnitude of the effect to the magnitude of the variable — it calculates an average effect. Furthermore, the average effect can be zero where there are strong effects of opposite sign in different parts of the input space.

3 Adapting GAM-style plots to function with interactions

The additive-scale GAM-style plots can be adapted to non-additive functions (i.e., ones with interactions) by plotting the effect on the output of a particular input variable for some selection of points in the input space. The plot for input variable i is no longer a function as with GAMs, but is a set of 2-d points (effects). Each point records the effect of changing x_i from some baseline value b_i, in the context of fixed values for the other inputs. The set of points is $\{(x_i, \Delta_i(\mathbf{x})) | \mathbf{x} \in \mathcal{X}\}$, where x_i is the ith element of $\mathbf{x}$, and $\mathcal{X}$ is some suitable set of points in the input space. $\Delta_i(\mathbf{x})$ is the effect on $\hat{f}$ of changing X_i from b_i to x_i in the context of $\mathbf{x}$, where b_i is some baseline value for X_i. As with GAMs, effects are best plotted on the natural additive scale of the model, and it will be assumed that this is $\hat{f}(\cdot)$. For a neural network this will usually be the total input of the output unit. Δ_i is defined as

$$\Delta_i(\mathbf{x}) = \hat{f}(\mathbf{x}) - \hat{f}(x_1, \ldots, x_{i-1}, b_i, x_{i+1}, \ldots, x_k).$$

The plot of these points can be enhanced by drawing each point as a short line segment, where the slope of the segment represents the partial gradient $\partial \hat{f} / \partial x_i$ evaluated at $\mathbf{x}$. As with GAMs, the plots should be made so that the vertical scale on each plot covers the same range, in order to allow comparison of the effects of different variables.

Figure 1 shows plots of the effects of variables for four 2-d functions from Hwang et al (1994). These functions are used here because they illustrate the different effects of non-linearities and interactions: $g^{(0)}$ is linear, $g^{(4)}$ and $g^{(5)}$ have non-linearities, and $g^{(1)}$ and $g^{(5)}$ have interactions. The definitions of the functions are as follows.

$$\begin{array}{ll}
\text{Linear additive:} & g^{(0)}(x_1, x_2) = x_1 + 2x_2 \\
\text{Simple interaction:} & g^{(1)}(x_1, x_2) = 10.391((x_1 - 0.4)(x_2 - 0.6) + 0.36) \\
\text{Additive function:} & g^{(4)}(x_1, x_2) = 1.3356(1.5(1 - x_1) + \exp(2x_1 - 1)\sin(3\pi(x_1 - 0.6)^2) \\
& \qquad + 1.3356\exp(3(x_2 - 0.5))\sin(4\pi(x_2 - 0.9)^2)) \\
\text{Complex interaction:} & g^{(5)}(x_1, x_2) = 1.9(1.35 + \exp(x_1)\sin(13(x_1 - 0.6)^2)\exp(-x_2)\sin(7x_2))
\end{array}$$

The effects are plotted at random locations in $X_1 \in [0:1], X_2 \in [0:1]$, and the "additive scale" of each function is the function value. The vertical dotted lines show the baseline values. The non-linearities in these functions show up as a curves in the pattern of effects, and the interactions show up as vertical spread at a particular horizontal location.

These plots have a number of informative features:

trend: An overall trend in a plot indicates a trend in the effect of the variable. For example, the plots for $g^{(0)}$ show that the value of the function increases with increasing x_1 and x_2.

overall vertical range: The overall vertical range reflects the importance of the variable, i.e., the degree to which the variable can affect the output. Plots for variables with no effect will be a horizontal line. The overall vertical range can be affected by the choice of baseline value for the variable, but the degree of this effect will be small except in pathological cases. For the function $g^{(0)}$, the greater range in effects of x_2 reflects its greater importance: $g^{(0)}$ is defined as $g^{(0)} = x_1 + 2x_2$.

local vertical spread: Vertical spread of values at a particular horizontal position indicates interaction between the variable plotted and one or more other variables. Care must be taken to not read too much into the horizontal location of points of maximum or minimum spread in these plots, e.g., $x_1 = 0$ for $g^{(1)}$, or $x_2 = 0.45$ for $g^{(5)}$. These locations are entirely determined by the choice of baseline values: spread is necessarily zero at the baseline value.

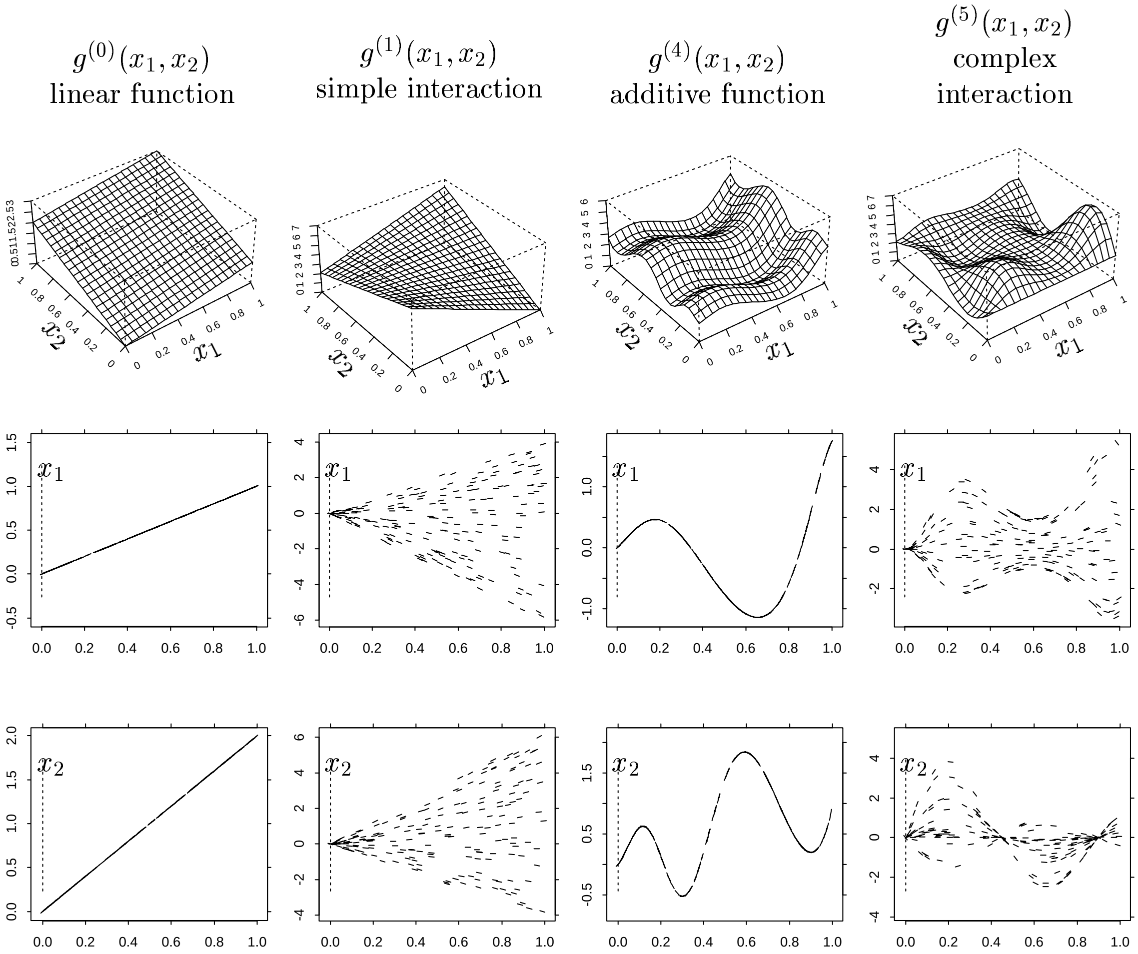


Figure 1: The effects of x_1 and x_2 on the four example functions.

non-linearity: In the absence of spread, non-linearity is easily detected as curves in the patterns of effects, as with $g^{(4)}$. However, in the presence of spread, non-linearity can be masked. However, the angles of segments can provide clues as to the absence and presence of non-linearities, such as with $g^{(5)}$.

These plots of effects differ from a projection onto one axis in that the additive effects of the other variables are removed. This results in a single curve for additive functions, as with $g^{(0)}$ and $g^{(4)}$, whereas projections of the same functions would have vertical spread.

Once it is clear that a certain variable, say x_1, is involved in interactions, it is possible to infer whether the interaction involves another variable, say x_2, by colouring (or some other means of identification such as separate panels) the effects in the plot for x_1 according to their value of x_2. If the effects of x_1 differ markedly for different values of x_2, it indicates an interaction between the two variables. An example of this on real data is given in the next section.

3.1 Choosing the points for visualization

In order to not overload the additive effects plots with too much information, one must plot the additive effects at only a limited number of points (the set $\mathcal{X}$). This is especially true when the input space is high-dimensional. In such cases a reasonable number of points can populate the input space only very sparely, and consequently the points must be chosen carefully. The set of of examples (training or testing or both) is an obvious choice for the points at which to plot effects. The advantage of this over using random points is that the relative density of points in different parts of the plot is apparent. When plotting at points corresponding to examples, isolated points may indicate either outliers in the input space, or regions of the function in which there is very little support for the value the function is producing. However, one disadvantage of plotting only at points corresponding to examples is that one may fail to spot undesirable or anomalous behavior of the fitted model in regions on the input space not populated by any examples. One possibly informative approach would be to make a color-coded plot at training examples, test examples, and random points.

3.2 Scatter plots of partial gradients

The partial gradients at particular points, which are shown as slopes on line segments in the additive effects plots, can be illustrated in a scatter plot. Such plots contain useful information, though they are not as straightforward to interpret as additive effects plots. Scatter plots of partial gradients for the four test functions are shown in Figure 2. As with additive effects plots, the points for additive functions form single curves in the plot. The points for linear functions form horizontal lines. Interactions show up as spread at a horizontal position. These gradient scatterplots have one advantage over additive effects plots: they do not depend on any "baseline" value. However, the vertical axis of these plot is more difficult to interpret, as it is the gradient of the effect, rather than the effect. Furthermore, the vertical scales for different inputs are somewhat arbitrary, as the absolute value of a partial gradient with respect to input x_i depends on the range of values of x_i, and can thus be changed by scaling the input. Consequently, all the gradient scatterplots in this paper have been scaled in terms of "screen gradients" – a line running from the bottom left to the top right of the corresponding additive effects plot is considered to have a gradient of 1. This is not entirely satisfactory, as outliers can have a large effect on the range of a particular input, and thus omitting or including them can affect the relative values of plotted partial gradients for different inputs.

3.3 Measuring importance of variables and degrees of interaction

Although the importance of variables and the extent to which they are involved in interactions (i.e., degree of non-additivity) can be estimated from plots of additive effects, it is also useful to have objective measures. There are two reasonable measures of the importance of variable of a particular variable.

1. The variance of additive effects for x_i.
2. The variance of partial gradients for x_i.

The advantage of the first is that it relates directly to the vertical range in additive effects plots, and is not affected by scaling of the input variables. However, it is affected by the choice of baseline value. The second does not depend on any choice of baseline value, but does suffer from the disadvantage that relative importances for different variables are affected by scaling the input values.

One possible measure of the degree to which x_i interacts with other variables is the average squared error in a smooth fit to the partial gradients of f with respect to x_i. This corresponds to the error in smooth fits to the points in the various plots in Figure 2. For smooth additive functions, such as $g^{(0)}$ and $g^{(4)}$, the error in the fit will be very small. The error will be

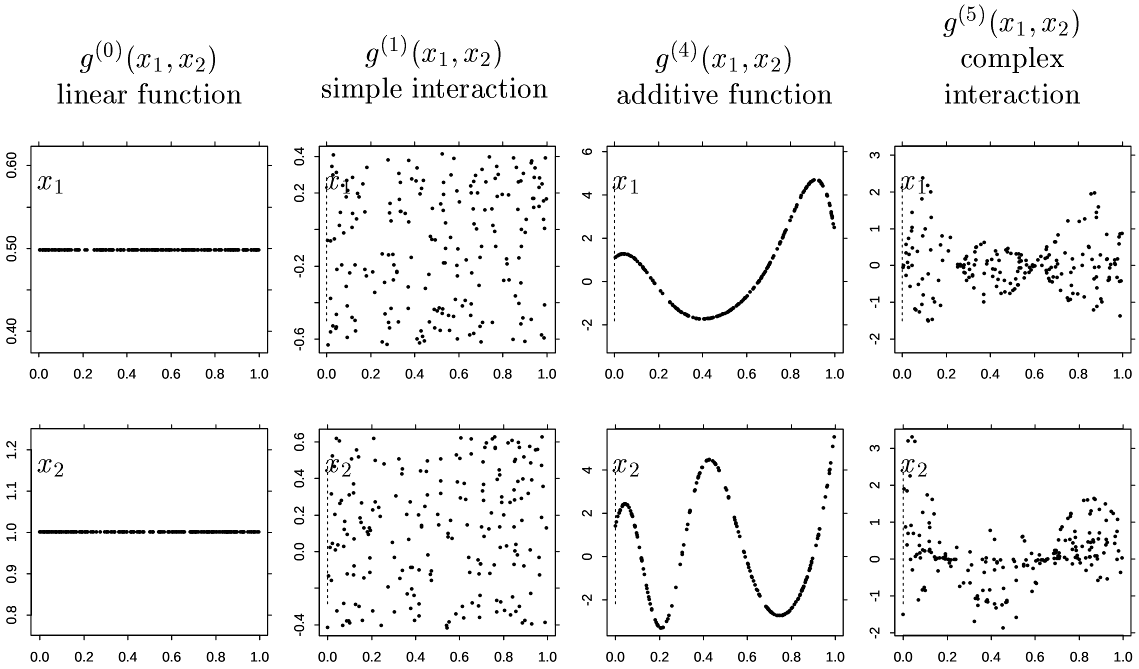


Figure 2: Scatter plots of partial gradients for the four example functions.

large when there is a large vertical spread at a particular value of the input variable. A smooth fit to the gradient points can be calculated with any 1-d smoother – the degrees of interaction reported in this paper were based on the smooth fit computed by the BRUTO curve fitter described in Hastie and Tibshirani (1990). This measure of degree of interaction is independent of the baseline value, but is affected by scaling of the inputs variables.

Confidence bounds on these measure for the importance of a feature, or the degree of interaction, could be calculated using a bootstrap procedure in the same manner as Baxt and White (1995) construct confidence intervals on the average effects of variables.

4 Visualizing a neural network model for the risk of cancer

We have developed a neural network model for the risk of developing squamous-cell lung cancer versus developing other non-smoking related cancers, based on knowledge of age and smoking and drinking habits (see Plate, Bert, Grace, and Band (1997)). The network has 19 inputs, 2 tanh hidden units, and 1 logistic output unit (see Bishop (1995) for details of this type of network). The inputs are age and various measures of tobacco and alcohol use. For each of six products, cigarettes (C), cigars (G), pipes (P), beer (B), wine (W) and spirits (S), there are three consumption measures. For cigarettes these are number of years cigarette smoking (CYR), number of cigarettes smoked per day (CDAY), and number of years since the subject quit smoking cigarettes (CYQUIT). There are analogous variables for the other products, though alcohol consumption rate is measured per week (e.g., BWK – number of standard beers per week.) The network was trained using MacKay's (1992) evidence framework to choose optimal values for weight penalities. The effects of variables in a trained network are shown in Figure 3. The vertical scale in these plots is the total input of the output unit. This value is the natural log of the predicted relative odds of developing squamous-cell lung cancer, thus a movement of one unit on the vertical axis corresponds to a change in relative odds by a factor of 2.718. The baseline for variables in these plots is the minimum value of the variable (20 for age, 0 for all others).

Tests of the predictive accuracy of this network indicate that its predictions are slightly better than those of a stepwise logistic-regression model of the data (i.e., an additive linear model). One such model selected variables AGE, CDAY, CYR, CYQUIT, and WYR. The plots of the additive effects for this fitted model would all consist of straight lines. However, Figure 3 shows that the neural network deviates strongly from an additive linear model. The neural network identifies more variables as important (the variables whose effects have a large vertical range), and also involves CYR, GDAY, and PYR in strong interactions. The neural network does agree with logistic regression model in the sign of effects: in both, AGE, CYR, and CDAY are positively associated with risk, CYQUIT and WYR are negatively associated with risk.

We can begin to identify interactions by partioning examples into sets according to one input variable and looking at whether effects of another variable are distinct for each partition. The examples in each partition can be shown in separate plots, or identified by color. Figure 4 shows separate plots of the effect of CYR for different ranges of GDAY. These plots strongly suggest that there is an interaction between years of cigarette smoking and number of cigars smoked per day: for people who smoke more cigars, the additional risk due to cigarette smoking is less. Indeed, this interaction tested as highly significant when added to a logistic regression model.

Color plots are a more economical way to present this information. Figure 5 shows the effects of eight of the more important variables color-coded by ranges of GDAY, CYR, and CYQUIT. The bands of color in the effects of CYR in Figure 5(a) reproduce Figure 4 and show that CYR interacts with GDAY: for higher values of GDAY, CYR has a lesser effect. In fact, Figure 5(a) shows that GDAY is involved in interactions with all of the plotted variables. Figure 5(b) shows that CYR is involved in interactions with the plotted variables, but not as strongly as CDAY — the bands of colors in (b) are not as clear as in (a). As expected CYR has clear interactions with GDAY (effects for GDAY in (b)), and PYR, but not with other variables. In contrast, no interactions for CYQUIT stand out in Figure 5(c). This does not mean that CYQUIT is not involved in interactions — obviously it is — but rather that the interactions are of a complex nature or involve variables not plotted here.

The color-coded scatter plots of partial gradients (Figure 6) are also informative. Again, the interactions between GDAY and CYR and between GDAY and CDAY stand out clearly. The protective effects of GYQUIT, CYQUIT, and WYR can also been seen.

Finally, Figure 7 shows where the variables fall into a two-dimensional space defined by overall importance (vertical axis) and degree of interaction (horizontal axis). Overall importance is measured as the variance of the additive effects, that is the variance in the vertical direction in the plots in Figures 3 or 5. Degree of interaction is measured as the variance of the (vertical) error in a smooth fit to the partial gradients, i.e., a smooth fit to the points in Figure 6. Variables which lie towards the bottom left of this plot have little overall effect and do not have strong degrees of interaction. Variables which lie towards the top right have a large effect and are strongly interactive. Variables which lie towards the top left have a large additive effect. Interestingly, the variables for the neural network lie near the diagonal — none of the variables in the neural network has a large effect that is mainly additive. Bearing in mind that an additive linear function provides a model of this data that is very nearly as good as the the neural network, one must be suspicious that the function computed by the trained neural network has gratuitous interactions — ones which are not supported by the data. This raises the question of whether one could construct flexible statistical models in which there is some control over the degree of interactions in the data. This possibility is pursued in (Plate to appear).

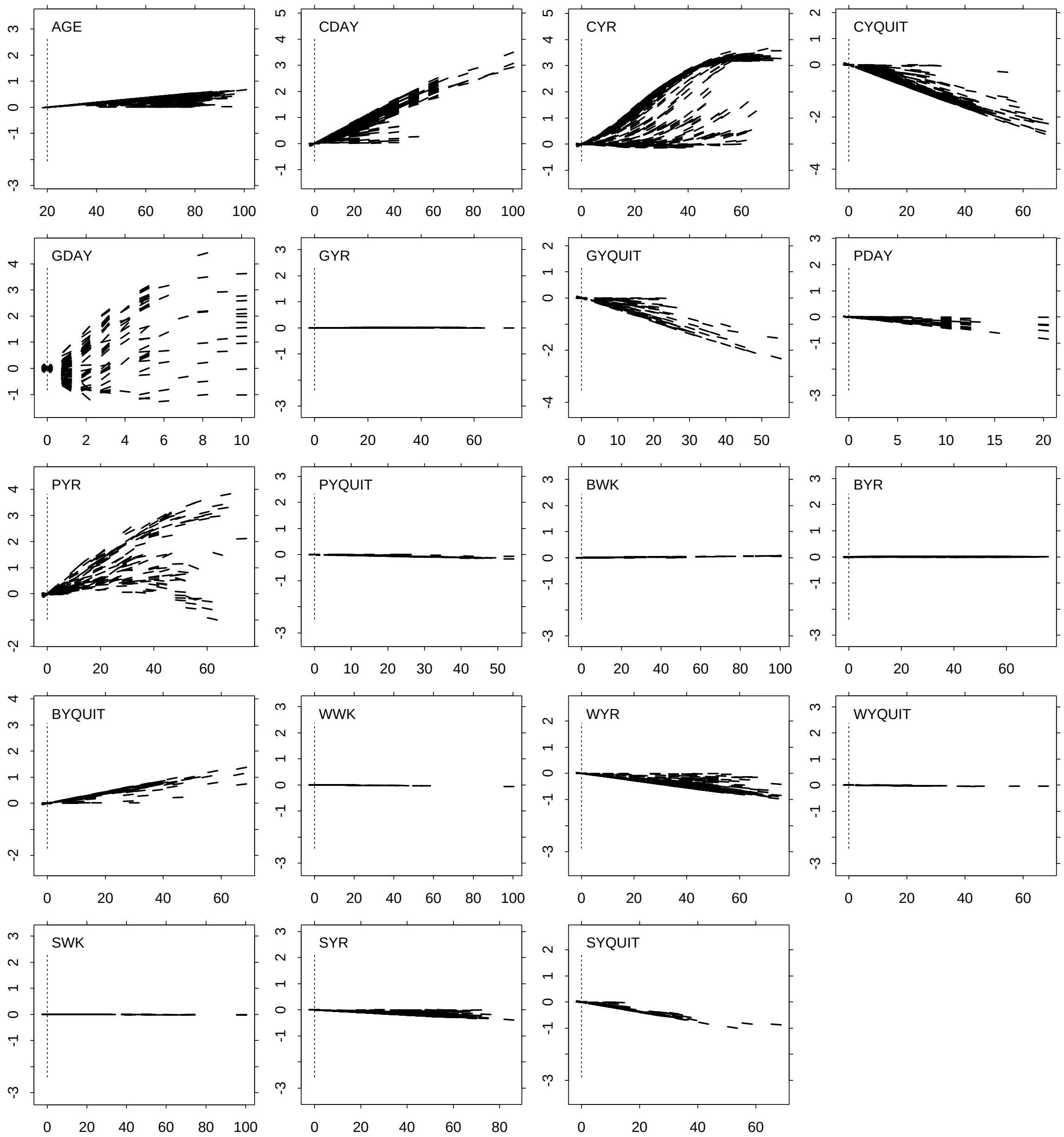


Figure 3: The effects of variables on a fitted neural network model for lung cancer.

One final point to note is that the proposed method for calculating additive effects can require the model to predict an output for an “absurd” input. For example, in calculating the additive effect of age for a 50 year old who has been smoking for 30 years, we ask the network to calculate an output for a 20 year old who has been smoking 30 years. While this may seem suspicious, a possible defense is that we are not really interested in the actual output for the absurd case, but only need it to estimate how much 50 years of age contributes in the real case. This type of situation likely to be more of a problem for nearest-neighbor style models in which an “absurd” input may not have any close neighbors.

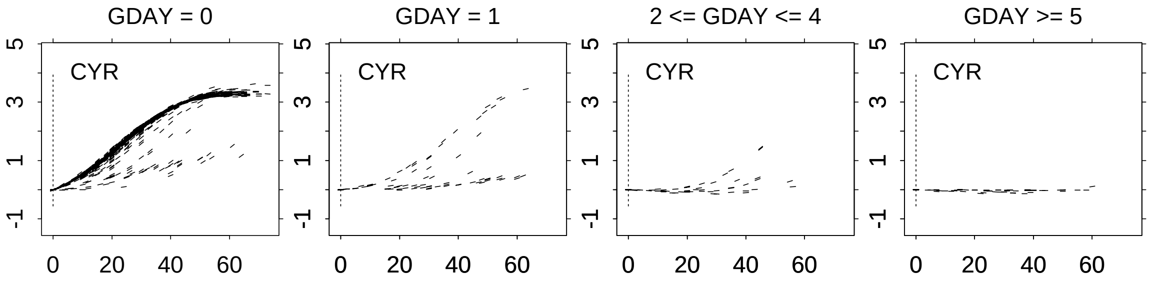


Figure 4: The effects of years of cigarette smoking partitioned by number of cigars smoked per day.

5 Discussion

Plots of additive effects provide rapid insight into the function computed by a feedforward neural network. The plots show the overall importance and direction of effects of individual inputs, and whether or not inputs are involved in interactions. They also make apparent the way in which a fitted model differs from a linear or generalized linear model. These differences can be of two kinds: (a) non-linearities in the response to single variables, and (b) interactions between variables. It is both interesting and valuable to have some idea of how a particular neural network differs from a linear model, especially as linear or logistic regression models are surprisingly effective on many real-world applications.

The method for constructing these plots does not depend on the internal structure of the network. Consequently, it can be applied to understanding any statistical model which can be used in a predictive mode, including committee and bagged models, which are difficult to interpret in any other way.

Acknowledgments

Some of this research was funded by grants from the Workers Compensation Board of British Columbia, NSERC, and IRIS (Canada), and conducted while the first author was a postdoctoral fellow at the University of British Columbia and the BC Cancer Agency.

References

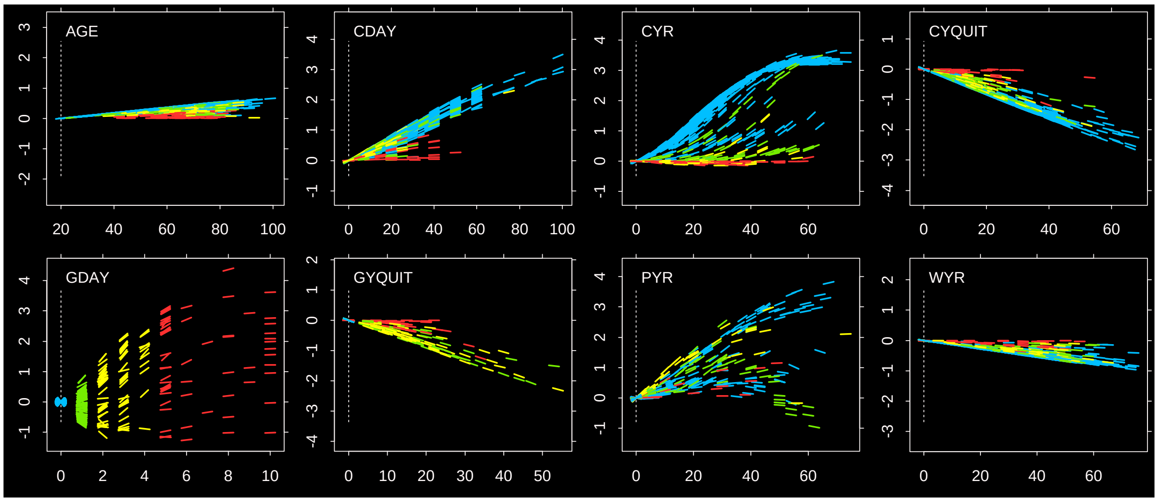
Baxt, W. G. (1992). Analysis of the clinical variables driving decision in an artificial neural network trained to identify the presence of myocardial infarcation. *Annals of Emergency Medicine 21*(12), 1439–1444.

Baxt, W. G. and H. White (1995). Bootstrapping confidence intervals for clinical input variable effects in a network trained to identify the presence of acture myocardial infarction. *Neural Computation 7*, 624–638.

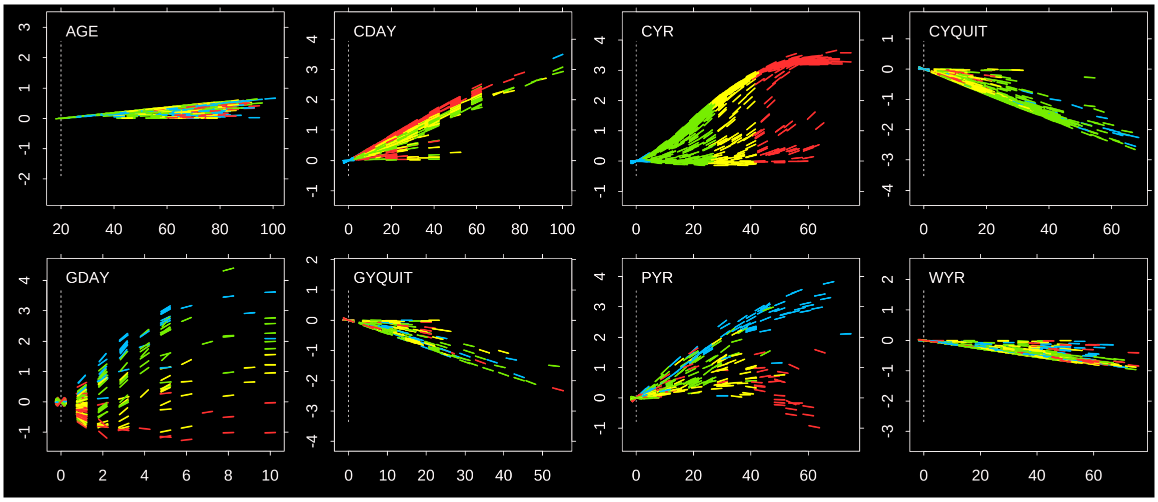
Bishop, C. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press.

Blasig, R. (1994). GDS: Gradient descent generation of symbolic classification rules. In J. D. Cowan, G. Tesauro, and J. Alspector (Eds.), *Advances in Neural Information Processing Systems 6 (NIPS*93)*, San Mateo, CA, pp. 1093–1100. Morgan Kaufmann.

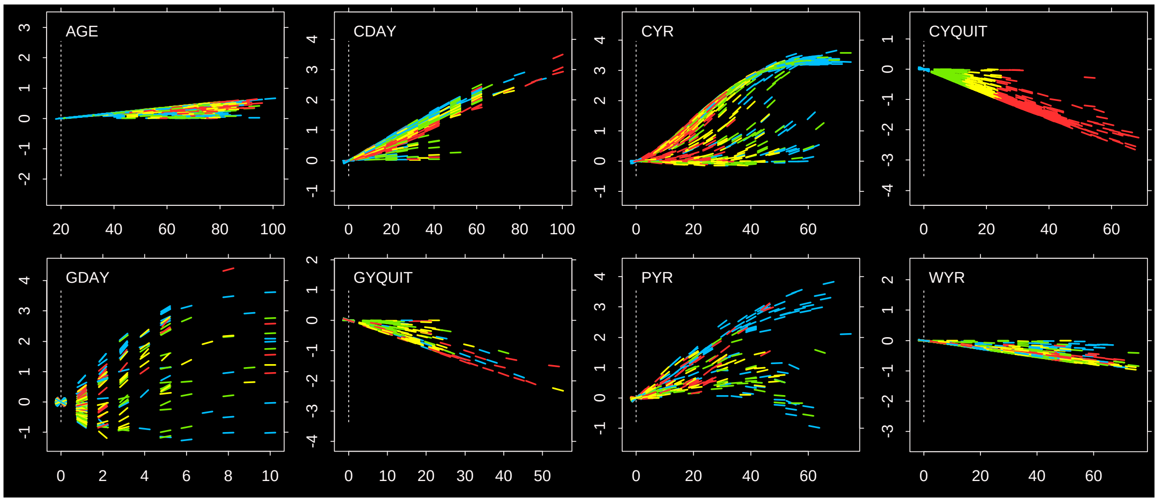
Dodds, S. R. (1995, Dec 2). Neural networks (letter). *The Lancet 346*, 1500–1501.



(a) Color-coded by ranges of GDAY.



(b) Color-coded by ranges of CYR.



(c) Color-coded by ranges of CYQUIT.

Figure 5: Effects of variables color-coded by ranges of particular variables.

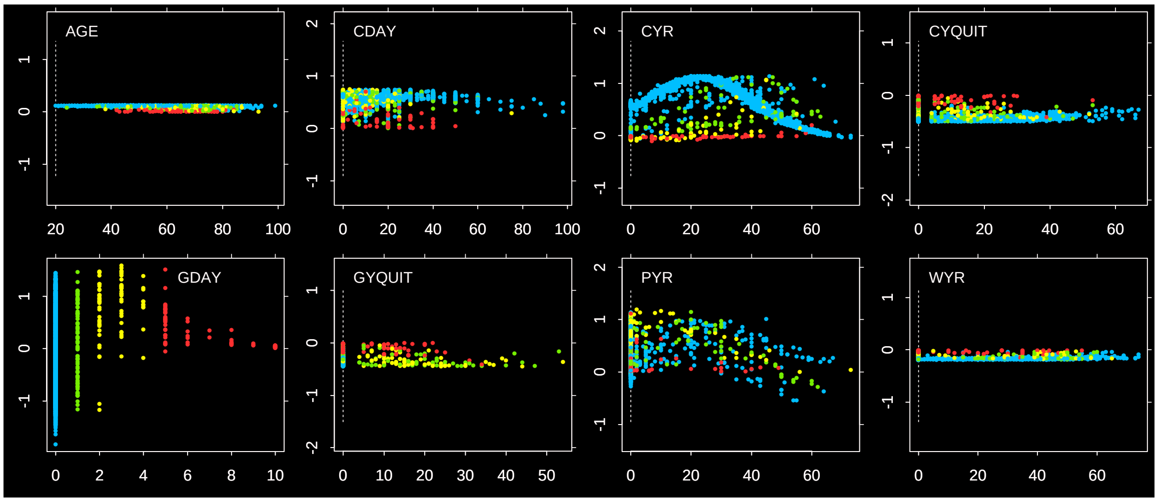


Figure 6: Scatter plots of partial gradients, colored by GDAY.

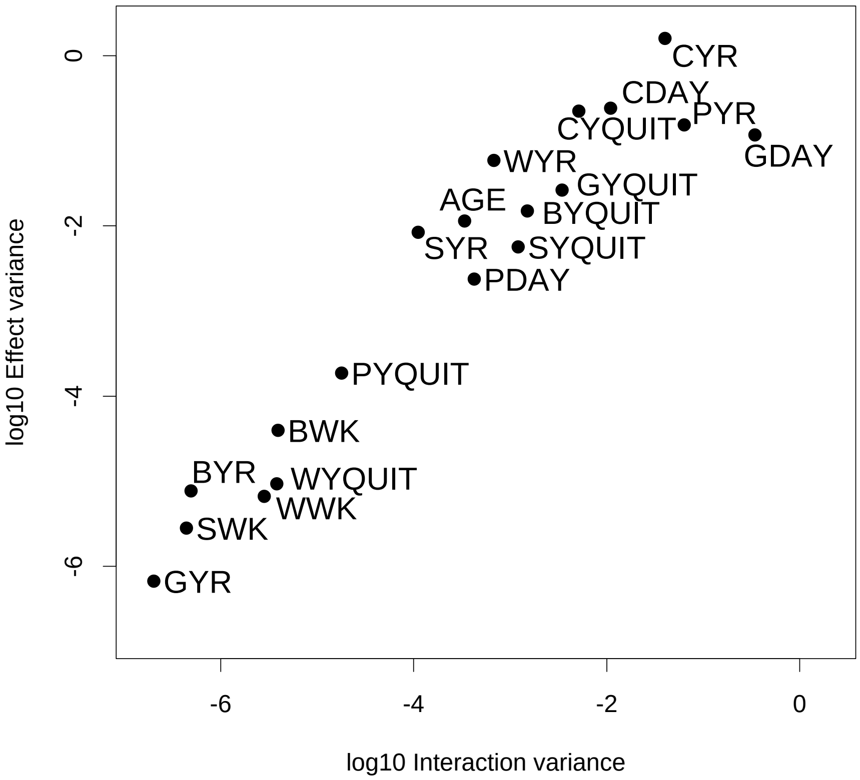


Figure 7: Importance of effects versus strength of interaction.

Hastie, T. J. and R. J. Tibshirani (1990). *Generalized additive models*. London: Chapman and Hall.

Hwang, J.-N., S.-R. Lay, M. Maechler, R. D. Martin, and J. Schimert (1994). Regression modeling in back-propagation and projection pursuit learning. *IEEE Transactions on Neural Networks 5*(3), 342–353.

MacKay, D. J. C. (1992). The evidence framework applied to classification networks. *Neural Computation 4*(5), 698–714.

Moseholm, L., E. Taudorf, and A. Frosig (1993). Pulmonary function changes in asthmatics associated with low-level SO_2 and NO_2, air pollution, weather, and medicine intake. *Allergy 48*, 334–344.

Plate, T., J. Bert, J. Grace, and P. Band (1997). A comparison between neural networks and other statistical techniques for modeling the relationship between tobacco and alcohol and cancer. In M. Mozer, M. Jordan, and T. Petsche (Eds.), *Advances in Neural Information Processing 9 (NIPS*96)*. MIT Press.

Plate, T. A. (to appear). Accuracy versus interpretability in flexible modeling: implementing a tradeoff using gaussian process models. *Behaviourmetrika*. Special issue on interpreting neural network models.

Saito, K. and R. Nakano (1988). Medical diagnostic expert system based on pdp model. In *IEEE International Conference on Neural Networks*, San Diego CA, pp. 255–262.

Sanger, D. (1989). Contribution analysis: A technique for assigning responsibilities to hidden units in connectionist networks. *Connection Science 1*, 115–138.

Sharp, D. (1995, Oct 21). From "black box" to bedside, one day? (commentary). *The Lancet 346*, 1050.

Shultz, T., Y. Oshima-Takane, and Y. Takane (1995). Analysis of unstandardized contributions in cross connected networks. In G. Tesauro, D. S. Touretzky, and T. K. Leen (Eds.), *Advances in Neural Information Processing Systems 7*, Cambridge, MA, pp. 601–608. MIT Press.

Shultz, T. R. and J. L. Elman (1994). Analyzing cross connected networks. In J. D. Cowan, G. Tesauro, and J. Alspector (Eds.), *Advances in Neural Information Processing Systems 6 (NIPS*93)*, San Mateo, CA, pp. 1117–1124. Morgan Kaufmann.

Towell, G. and J. Shavlik (1993). Extracting refined rules from knowledge-based neural networks. *Machine Learning 13*(1), 71–101.

Wyatt, J. (1995, Nov 4). Nervous about artificial neural networks? (commentary). *The Lancet 346*, 1175–1177.