Error bars for linear and nonlinear neural network regression models

William D. Penny and Stephen J. Roberts

Neural Systems Research Group,
Department of Electrical and Electronic Engineering,
Imperial College of Science, Technology and Medicine,
London SW7 2BT., U.K.
w.penny@ic.ac.uk, s.j.roberts@ic.ac.uk

November 16, 1998

Abstract

We derive error bars for the cases of linear and nonlinear neural network regression using the so-called 'delta method'. Neural network *classifiers* are not considered. None of this material is new to statisticians but it may well be new to neural network modellers. For all cases considered, the error bars are shown to consist of two additive terms; the first term being intrinsic target noise and the second term being due to the uncertainty in the inferred network parameters. For all cases the first term is considered constant over the input space (homoskedasticity). The second term is shown to be input dependent. For the case of univariate linear regression the second term is shown to increase quadratically with the distance of the test point to the mean training point and to decrease linearly with the number of training examples. For multivariate linear regression the second term is shown to increase linearly with the Mahalanobis distance between the test vector and the mean training vector. This holds too for the case of generalized linear regression, but now in the space of hidden units instead of input units. For the nonlinear regression models we use a first order Taylor series expansion of the network output about the maximum likelihood model parameters. The error bars for multivariate nonlinear regression are equivalent to those derived under the Bayesian evidence framework as this also assumes a first order Taylor series expansion.

1 Introduction

We derive error bars for four types of 'neural network' regression models; univariate (eg. one input) linear models, univariate nonlinear models, multivariate linear models and multivariate nonlinear models. The last case is what we'd usually call a neural network (eg. a Multi-Layer Perceptron).

Out of all the derivations, the one for univariate linear regression is probably the most long-winded and we include it despite the fact that it is a special case of multivariate linear regression. Its inclusion caters for the unlikely reader who only has one input and one output to model. Nevertheless, it is useful as a tutorial example.

In all of the derivations the models are trained to minimise a squared error cost function and, for the nonlinear models, the error bars are estimated using a first order Taylor series expansion.

2 Univariate linear regression

Consider data generated from the following univariate linear function

$$y_i = ax_i + b + e_i \tag{1}$$

where e_i is Independent, Identically Distributed (IID) random noise drawn from a distribution having standard deviation σ. For the moment, we assume σ to be uniform over the input space x. This data can be modelled with the linear regression equation

$$\hat{y}_i = \hat{a}x_i + \hat{b} \tag{2}$$

The goodness of fit of the model to the data may be measured by the least squares cost function

$$E = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3}$$

The values of $\hat{a}$ and $\hat{b}$ that minimize the above cost function can be calculated by setting the first derivatives of the cost function to zero and solving the resulting simultaneous equations. The result is derived in Appendix B, Part 1. The solutions are

$$\hat{a} = \frac{\sigma_{xy}}{\sigma_x^2} \tag{4}$$

and

$$\hat{b} = \bar{y} - \hat{a}\bar{x} \tag{5}$$

where $\bar{x}$ and $\bar{y}$ are the mean observed values of the data and σ_x^2 and σ_{xy} are the input variance and input-output covariance. This enables least squares fitting of a regression line to a data set as shown in Figure 1.

If there were no noise on the y_i's (see equation 1) then $\hat{a}$ and $\hat{b}$ could be calculated exactly, provided we had at least two data points. They would be equal to a and b. But because there is target noise then different instantions of this noise will result in some variation in the calculated values of $\hat{a}$ and $\hat{b}$. If we were to repeat the regression calculation on a number of data sets drawn from equation 1 the mean values of $\hat{a}$ and $\hat{b}$ would be equal to a and b. The variances of $\hat{a}$ and $\hat{b}$ can be derived as shown in Appendix B, part 2. It is important to note, however, that the values of $\hat{a}$ and $\hat{b}$ are correlated. That is, $Var(\hat{a}, \hat{b}) \neq 0$. Because of this correlation the calculation of the prediction variance of a new test point becomes more involved. To simplify this calculation we consider a re-parameterised model involving a new parameter $\hat{c}$.

$$\hat{y}_i = \hat{a}(x_i - \bar{x}) + \hat{c} \tag{6}$$

where $\hat{a}$ is exactly the same as before and $\hat{c}$ is given by

$$\hat{c} = \hat{a}\bar{x} + \hat{b} = \bar{y} \tag{7}$$

The variances of these estimates are derived in Appendix B, parts 2 and 3

$$Var(\hat{a}) = \frac{\sigma^2}{N\sigma_x^2} \tag{8}$$

and

$$Var(\hat{c}) = \frac{\sigma^2}{N} \tag{9}$$

The predicted value, $\tilde{y}$, on a new input data point, $\tilde{x}$ is

$$\tilde{y} = \hat{a}(\tilde{x} - \bar{x}) + \hat{c} + \tilde{e} \tag{10}$$

The inclusion of the error term, $\tilde{e}$, may seem incorrect as it is not included in our definition of the prediction model (see equation 2 or equation 6). For the purposes of calculating error bars, however, its inclusion is essential. This is because, even if the model prediction were perfect (i.e. $\hat{a} = a$ and $\hat{b} = b$ or $\hat{c} = c$) the prediction would still be in error by an amount $\tilde{e}$ - see equation (1). The variance of the prediction is

$$Var(\tilde{y}) = Var(\hat{a}(\tilde{x} - \bar{x}) + \hat{c} + \tilde{e}) \tag{11}$$

The three random variables $\hat{a}$, $\hat{c}$ and $\tilde{e}$ are all independent. Hence, from Appendix A

$$Var(\tilde{y}) = Var(\tilde{e}) + Var(\hat{c}) + (\tilde{x} - \bar{x})^2 Var(\hat{a}) \tag{12}$$

Substituting in previous results gives

$$Var(\tilde{y}) = \sigma^2 + \frac{\sigma^2}{N}\left(1 + \frac{(\tilde{x} - \bar{x})^2}{\sigma_x^2}\right) \tag{13}$$

where $\bar{x}$ and σ_x^2 are the mean and variance of x as measured on the training set. The variance of predictions on new data points has two parts; the part due to the intrinsic noise on the new target, σ, and the part due to the uncertainty in estimating the weights. The second factor increases the further away the new data point is from $\bar{x}$. This can be seen in Figure 1 which shows linear prediction with error bars at $\pm 2\sigma_{\tilde{y}}$. Note that data points further away from the training set are assigned quadratically larger error bars. Equation 13 also shows that the second factor decreases linearly with the number of examples in the training set.

3 Univariate nonlinear regression

We now consider the case where we have a univariate *nonlinear* model. This is a model where the output is a nonlinear function of the inputs and of the parameters (if the output remains a linear function of the parameters then we have generalized linear regression). For simplicity the nonlinear function is considered to be a function of a single parameter, a. The data is generated by

$$y_i = f(ax_i) + e_i \tag{14}$$

The model is

$$\hat{y}_i = f(\hat{a}x_i) \tag{15}$$

The goodness of fit of the model to the data may be measured by the least squares cost function

$$E = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{16}$$

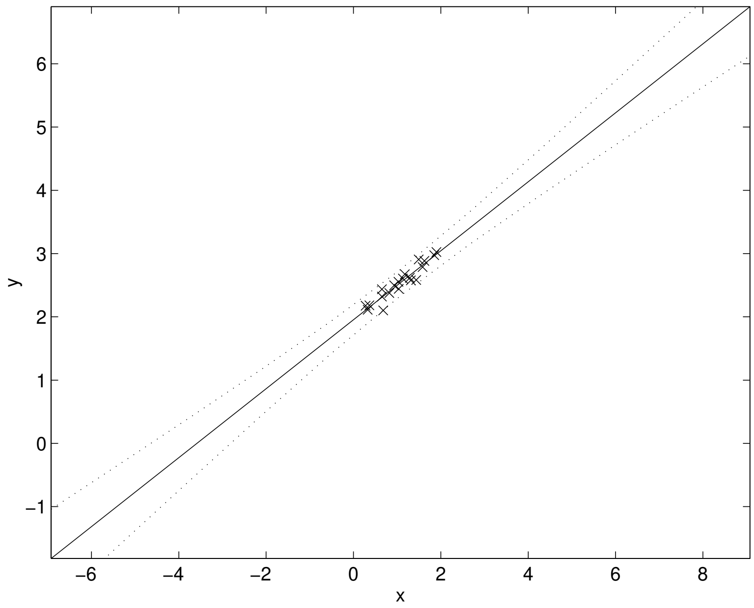

Figure 1: Linear regression (-) with error bars at $\pm 2\sigma_{\tilde{y}}$ (..). These error bars grow quadratically with the distance of the new data point from the mean of the training data.

The value of $\hat{a}$ that minimizes the above cost function can be found by a nonlinear optimization process such as conjugate gradients. For our simple one parameter case, however, a one-dimensional line search algorithm will suffice.

To find the variance of the estimated parameter, $\hat{a}$, we assume that the nonlinear function can be locally approximated by a linear function. From a first order Taylor series we obtain

$$\hat{y}_i = f(\hat{a}x_i) = f(ax_i) + \Delta\hat{a}g_i \tag{17}$$

where $\Delta\hat{a} = \hat{a} - a$ and $g_i = df(ax_i)/da$. Substituting this into our cost function gives

$$\begin{aligned} E &= \sum_{i=1}^{N}(y_i - f(ax_i) - \Delta\hat{a}g_i)^2 \\ &= \sum_{i=1}^{N}(e_i - \Delta\hat{a}g_i)^2 \end{aligned} \tag{18}$$

This is the same form as the first equation in Appendix B with e_i taking the place of y_i, $\Delta\hat{a}$ taking the place of $\hat{a}$ and g_i taking the place of x_i ($\hat{b}$ is zero). Hence, by analogy with equation 8 (the equation for $Var(\hat{a})$ in the linear regression case)

$$Var(\Delta\hat{a}) = \frac{\sigma^2}{N\sigma_g^2} \tag{19}$$

where σ_g^2 is the variance of g measured over the data set. The predicted value, $\tilde{y}$, on a new input data point, $\tilde{x}$ is

$$\tilde{y} = f(\hat{a}\tilde{x}) + \tilde{e} \tag{20}$$

See section 2 for the justification for including the $\tilde{e}$ term. Because $\hat{a}$ and $\tilde{e}$ are independent, the variance of the predicted value is

$$Var(\tilde{y}) = Var(f(\hat{a}\tilde{x})) + Var(\tilde{e}) \tag{21}$$

The first term is estimated by looking at the change in predicted value resulting from a change in the estimated parameter. A local linear approximation (equation 17) gives

$$\Delta f(\hat{a}\tilde{x}) = \Delta\hat{a}\tilde{g} \tag{22}$$

where $\tilde{g} = df(a\tilde{x})/da$. Hence,

$$Var(f(\hat{a}\tilde{x})) = Var(\Delta f(\hat{a}\tilde{x})) = \tilde{g}^2 Var(\Delta\hat{a}) \tag{23}$$

Therefore,

$$Var(\tilde{y}) = \sigma^2 + \frac{\sigma^2}{N}\frac{\tilde{g}^2}{\sigma_g^2} \tag{24}$$

In Figure 2 we show univariate nonlinear regression where the nonlinear function is

$$f(x) = \frac{1}{1+e^{-ax}} \tag{25}$$

where the only unknown parameter is a, the slope of the sigmoid. There is no offset (y-intercept) parameter, therefore, it is known beforehand that the regression curve must pass through the points $(-\infty, 0)$, $(0, 0.5)$ and $(\infty, 1)$. These provide constraints in addition to those imposed by the data points. It is not surprising therefore that the largest error bars are at $x \sim \pm 3$.

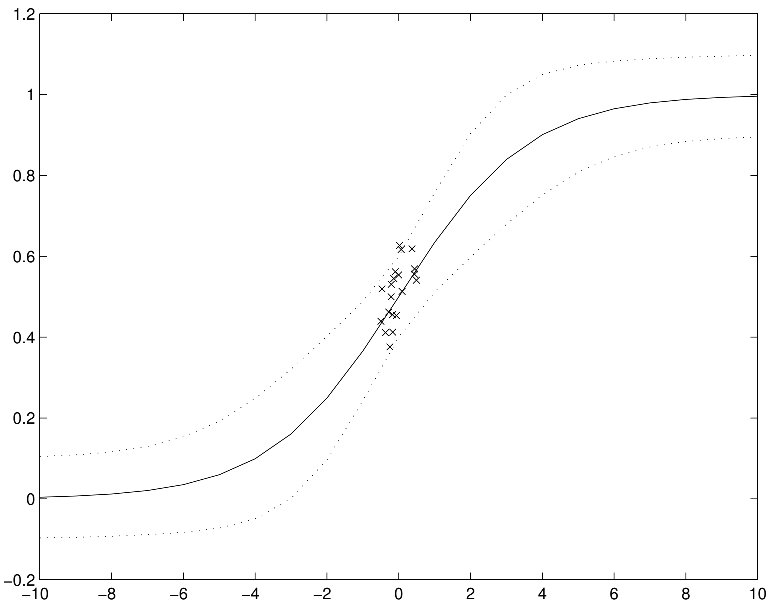

Figure 2: Nonlinear regression (-) with error bars at $\pm 2\sigma_{\tilde{y}}$ (..). The nonlinearity, in this example, is a sigmoid.

4 Multivariate linear regression

For a multivariate linear data set, the dependent variable y_i depends on the p dimensional input vector, $\boldsymbol{x}_i$. The data is hence generated as

$$y_i = \boldsymbol{x}_i^T \boldsymbol{w} + e_i \tag{26}$$

where $\boldsymbol{w}$ is the true weight vector. This may be modelled by the regression equation

$$\hat{y}_i = \boldsymbol{x}_i^T \hat{\boldsymbol{w}} \tag{27}$$

where $\hat{\boldsymbol{w}}$ is the estimated weight vector. The goodness of fit can be assessed by the least squares cost function

$$\begin{aligned} E &= \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \qquad (28) \\ &= \sum_{i=1}^{N} (y_i - \hat{\boldsymbol{x}}_i^T \hat{\boldsymbol{w}})^2 \end{aligned}$$

We can re-write this in matrix notation as

$$E = (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}})^T (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}}) \tag{29}$$

where $\boldsymbol{X}$ is an N-by-p matrix whose rows are made up of different training vectors. The weight vector that minimises this cost function can be calculated by setting the first derivative of the cost function to zero and solving the resulting equation. From Appendix C, part 1 we get

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y} \tag{30}$$

Different instantiations of target noise will generate different estimated weight vectors according to the last equation. The corresponding weight covariance matrix is derived in Appendix C, part 2

$$Var(\hat{\boldsymbol{w}}) = \sigma^2 (\boldsymbol{X}^T \boldsymbol{X})^{-1} \tag{31}$$

These different weight vectors will give different predictions. The prediction, $\tilde{y}$, on a new data point $\tilde{\mathbf{x}}$ is

$$\tilde{y} = \tilde{\boldsymbol{x}}^T \hat{\boldsymbol{w}} + \tilde{e} \tag{32}$$

See section 2 for a justification for including the $\tilde{e}$ term. The variance of this prediction is, by Appendix A

$$\begin{aligned} Var(\tilde{y}) &= \sigma^2 + \tilde{\boldsymbol{x}}^T [Var(\hat{\boldsymbol{w}})] \tilde{\boldsymbol{x}} \qquad (33) \\ &= \sigma^2 + \sigma^2 \tilde{\boldsymbol{x}}^T (\boldsymbol{X}^T \boldsymbol{X})^{-1} \tilde{\boldsymbol{x}} \end{aligned}$$

The error bars may be written in another form by re-writing the regression equation as

$$\hat{y}_i = (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \hat{\boldsymbol{w}} + \hat{b} + \tilde{e} \tag{34}$$

where $\hat{b}$ is a bias parameter given by

$$
\begin{aligned}
\hat{b} &= \bar{\boldsymbol{x}}^T \hat{\boldsymbol{w}} \\
&= \bar{\boldsymbol{y}}
\end{aligned} \tag{35}
$$

The variance of this parameter is

$$
\begin{aligned}
Var(\hat{b}) &= Var(\bar{\boldsymbol{y}}) \\
&= \frac{\sigma^2}{N}
\end{aligned} \tag{36}
$$

The variance of the weight vector is, by analogy with equation 30

$$
\begin{aligned}
Var(\hat{\boldsymbol{w}}) &= \sigma^2((\boldsymbol{X} - \bar{\boldsymbol{X}})^T(\boldsymbol{X} - \bar{\boldsymbol{X}}))^{-1} \\
&= \sigma^2 \boldsymbol{C}^{-1}
\end{aligned} \tag{37}
$$

where $\boldsymbol{C}$ is the covariance matrix of input data in the training set

$$
\boldsymbol{C} = (\boldsymbol{X} - \bar{\boldsymbol{X}})^T(\boldsymbol{X} - \bar{\boldsymbol{X}}) \tag{38}
$$

A new prediction is given by

$$
\tilde{y} = (\tilde{\boldsymbol{x}} - \bar{\boldsymbol{x}})^T \hat{\boldsymbol{w}} + \hat{b} + \tilde{e} \tag{39}
$$

The variance of this prediction is

$$
Var(\tilde{y}) = Var(\tilde{e}) + Var(b) + (\tilde{\boldsymbol{x}} - \bar{\boldsymbol{x}})^T [Var(\hat{\boldsymbol{w}})](\tilde{\boldsymbol{x}} - \bar{\boldsymbol{x}}) \tag{40}
$$

Substituting in earlier results gives

$$
Var(\tilde{y}) = \sigma^2 \left(1 + \frac{1}{N} + d(\tilde{\boldsymbol{x}})\right) \tag{41}
$$

where $d(\tilde{\boldsymbol{x}})$ is the Mahalanobis distance between $\tilde{\boldsymbol{x}}$ and $\bar{\boldsymbol{x}}$. This measures the distance of the new input vector from the centre of the training data and is given by

$$
d(\tilde{\boldsymbol{x}}) = (\tilde{\boldsymbol{x}} - \bar{\boldsymbol{x}})^T \boldsymbol{C}^{-1} (\tilde{\boldsymbol{x}} - \bar{\boldsymbol{x}}) \tag{42}
$$

The above result also holds for the case of generalized linear regression (where the inputs are first transformed by a fixed nonlinearity into hidden unit activations eg. a Radial Basis Function network with fixed Gaussians - only the output layer is trained), but now in the space of hidden units instead of input units.

5 Multivariate nonlinear regression

Data is generated from

$$
y_i = f(\boldsymbol{x}_i, \boldsymbol{w}) + e_i \tag{43}
$$

The data is modelled with the regression equation

$$
\hat{y}_i = f(\boldsymbol{x}_i, \hat{\boldsymbol{w}}) \tag{44}
$$

where f is for example a Multi-Layer Perceptron neural network and $\hat{\boldsymbol{w}}$ are the weights in that network. The goodness of fit can be assessed by the least squares cost function

$$E = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{45}$$

The choice of $\hat{\boldsymbol{w}}$ that minimises the above function can be found by an iterative optimization procedure such as steepest descent, conjugate gradients or pseudo-Newton methods. To find the variance of the estimated parameter, $\hat{\boldsymbol{w}}$, we assume that the nonlinear function can be locally approximated by a linear function. From a first order Taylor series we obtain

$$\hat{y}_i = f(\boldsymbol{x}_i, \hat{\boldsymbol{w}}) = f(\boldsymbol{x}_i, \boldsymbol{w}) + \boldsymbol{g}_i^T \Delta\hat{\boldsymbol{w}} \tag{46}$$

where $\Delta\hat{\boldsymbol{w}} = \hat{\boldsymbol{w}} - \boldsymbol{w}$ and $\boldsymbol{g}_i = df(\boldsymbol{x}_i, \boldsymbol{w})/d\boldsymbol{w}$. Substituting this into our cost function gives

$$\begin{aligned} E &= \sum_{i=1}^{N} (y_i - f(\boldsymbol{x}_i, \boldsymbol{w}) - \boldsymbol{g}_i^T \Delta\hat{\boldsymbol{w}})^2 \\ &= \sum_{i=1}^{N} (e_i - \boldsymbol{g}_i^T \Delta\hat{\boldsymbol{w}})^2 \end{aligned} \tag{47}$$

This can be written in matrix form as

$$E = (\boldsymbol{e} - \boldsymbol{G}\Delta\hat{\boldsymbol{w}})^T (\boldsymbol{e} - \boldsymbol{G}\Delta\hat{\boldsymbol{w}}) \tag{48}$$

where $\boldsymbol{G}$ is an N-by-p matrix whose ith row consists of the vector g_i and $\boldsymbol{e}$ is the corresponding vector of errors. This is the same form as equation 28 in the previous section where $\boldsymbol{e}$ takes the place of $\boldsymbol{y}$, $\Delta\hat{\boldsymbol{w}}$ takes the place of $\hat{\boldsymbol{w}}$ and $\boldsymbol{G}$ takes the place of $\boldsymbol{X}$. Hence, by analogy with equation 29

$$\Delta\hat{\boldsymbol{w}} = (\boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{G}^T\boldsymbol{e} \tag{49}$$

and by analogy with equation 30

$$\begin{aligned} Var(\Delta\hat{\boldsymbol{w}}) &= \sigma^2 (\boldsymbol{G}^T\boldsymbol{G})^{-1} \\ &= \boldsymbol{H}^{-1} \end{aligned} \tag{50}$$

where $\boldsymbol{H}$ is the outer product approximation to the Hessian matrix

$$\boldsymbol{H} = \frac{1}{\sigma^2}(\boldsymbol{G}^T\boldsymbol{G}) \tag{51}$$

The prediction, $\tilde{y}$, on a new data point $\tilde{\mathbf{x}}$ is

$$\tilde{y} = f(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{w}}) + \tilde{e} \tag{52}$$

See section 2 for a justification for including the $\tilde{e}$ term. Because $\hat{\boldsymbol{w}}$ and $\tilde{e}$ are independent the variance of the predicted value is

$$Var(\tilde{y}) = Var(f(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{w}})) + Var(\tilde{e}) \tag{53}$$

The first term is estimated by looking at the change in predicted value due to the change in the model's estimated parameters. From a local linear approximation

$$\Delta f(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{w}}) = \tilde{\boldsymbol{g}}^T \Delta\hat{\boldsymbol{w}} \tag{54}$$

Hence,

$$\begin{aligned} Var(f(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{w}})) &= Var(\Delta f(\tilde{\boldsymbol{x}}, \hat{\boldsymbol{w}})) \\ &= \tilde{\boldsymbol{g}}^T Var(\Delta \hat{\boldsymbol{w}}) \tilde{\boldsymbol{g}} \\ &= \tilde{\boldsymbol{g}}^T \boldsymbol{H}^{-1} \tilde{\boldsymbol{g}} \end{aligned} \tag{55}$$

Therefore,

$$Var(\tilde{y}) = \sigma^2 + \tilde{\boldsymbol{g}}^T \boldsymbol{H}^{-1} \tilde{\boldsymbol{g}} \tag{56}$$

These error bars are equivalent to those obtained under the Bayesian evidence framework as this also uses a first order Taylor series expansion (Mackay, 1992).

6 Summary

The error bars for univariate linear and nonlinear regression and multivariate linear and nonlinear regression are given by equations 13, 24, 32 or 40 and 55, respectively. We have shown that for all cases considered, the error bars are shown to consist of two additive terms; the first term being intrinsic target noise and the second term being due to the uncertainty in the inferred network parameters. For all cases the first term was considered constant over the input space (homoskedasticity). The second term was shown to be input dependent. For the case of univariate linear regression the second term was shown to increase quadratically with the distance of the test point to the mean of the training data and to decrease linearly with the number of examples in the training set. For multivariate linear regression the second term was shown to increase linearly with the Mahalanobis distance between the test point and the mean of the training data. This holds too for the case of generalized linear regression, but now in the space of hidden units instead of input units. For the nonlinear regression models we used a first order Taylor series expansion of the network output about the maximum likelihood model parameters. The error bars for multivariate nonlinear regression are equivalent to those derived under the Bayesian evidence framework as this also assumes a first order Taylor series expansion.

7 Acknowledgements

William Penny is a research fellow funded by the UK EPSRC (GR/K79062), whose support we gratefully acknowledge.

References

Bevington, P.R. and Robinson,K. (1992) *Data reduction and error analysis for the physical sciences.* Mc-Graw Hill, pp 41-43.

MacKay, D.J.C. (1992) A practical Bayesian framework for backprop networks. *Neural Computation* 4(3):448-472.

Seber, G.A.F. and Wild, C.J. (1989). *Nonlinear regression.* New York: John Wiley.

Weisberg, S. (1980). *Applied linear regression.* John Wiley.

APPENDIX A. Statistics of linear functions of random variables

These results are taken from Weisberg (1980) and are approximations from a first order Taylor series. For a more detailed discussion see Bevington and Robinson (1992). Given random variables, u_i and constant values a_i, the variance of a sum of products is given by

$$Var\left(a_0 + \sum a_i u_i\right) = \sum_i a_i^2 Var(u_i) + 2 \sum_i \sum_j a_i a_j Var u_i, u_j) \tag{57}$$

where $Var(x, y)$ denotes the covariance of the random variables x and y. The covariance of a linear function of a pair of random variables is

$$Var\,(a_0 + a_1 u_1, a_3 + a_2 u_2) = a_1 a_2 Var(u_1, u_2) \tag{58}$$

For a vector of random variables, $\boldsymbol{z}$, and a matrix of constants, $\boldsymbol{C}$, and a vect or of constants, $\boldsymbol{d}$, we have

$$Var(\boldsymbol{Cz} + \boldsymbol{d}) = \boldsymbol{C}[Var(\boldsymbol{z})]\boldsymbol{C}^T \tag{59}$$

where, here, Var() denotes a covariance matrix. The covariance between a pair of random vectors is given by

$$Var(\boldsymbol{C}_1\boldsymbol{z}, \boldsymbol{C}_2\boldsymbol{z}) = \boldsymbol{C}_1[Var(\boldsymbol{z})]\boldsymbol{C}_2^T \tag{60}$$

APPENDIX B. Univariate Linear Regression

1. Calculating the weights

Substituting (2) into (3) gives

$$E = \sum_{i=1}^{N} (y_i - \hat{a}x_i - \hat{b})^2 \tag{61}$$

Differentiating with respect to $\hat{a}$ gives

$$\frac{\partial E}{\partial \hat{a}} = -2 \sum_{i=1}^{N} x_i(y_i - \hat{a}x_i - \hat{b}) \tag{62}$$

Differentiating with respect to $\hat{b}$ gives

$$\frac{\partial E}{\partial \hat{b}} = -2 \sum_{i=1}^{N} (y_i - \hat{a}x_i - \hat{b}) \tag{63}$$

By setting the above derivatives to zero we obtain the 'normal equations' of the regression. Re-arranging the normal equations gives

$$\hat{a} \sum_{i=1}^{N} x_i^2 + \hat{b} \sum_{i=1}^{N} x_i = \sum_{i=1}^{N} x_i y_i \tag{64}$$

and

$$\hat{a} \sum_{i=1}^{N} x_i + \hat{b}N = \sum_{i=1}^{N} y_i \tag{65}$$

By substituting the mean observed values $\bar{x}$ and $\bar{y}$ into the last equation we get

$$\hat{b} = \bar{y} - \hat{a}\bar{x} \tag{66}$$

Now let

$$\begin{aligned} S_{xx} &= \sum_{i=1}^{N} (x_i - \bar{x})^2 && (67) \\ &= \sum_{i=1}^{N} x_i^2 - N\bar{x}^2 && (68) \end{aligned}$$

and

$$\begin{aligned} S_{xy} &= \sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y}) && (69) \\ &= \sum_{i=1}^{N} x_i y_i - N\bar{x}\bar{y} && (70) \end{aligned}$$

Substiting for $\hat{b}$ into the first normal equation gives

$$\hat{a}\sum_{i=1}^{N} x_i^2 + (\bar{y} - \hat{a}\bar{x})\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} x_i y_i \tag{71}$$

Re-arranging gives

$$\begin{aligned} \hat{a} &= \frac{\sum_{i=1}^{N} x_i y_i - \bar{y}\sum_{i=1}^{N} x_i}{\sum_{i=1}^{N} x_i^2 + \bar{x}\sum_{i=1}^{N} x_i} && (72) \\ &= \frac{\sum_{i=1}^{N} x_i y_i - N\bar{x}\bar{y}}{\sum_{i=1}^{N} x_i^2 + N\bar{x}^2} \\ &= \frac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N} (x_i - \bar{x})^2} && (73) \end{aligned}$$

We now define the data variance and covariances

$$\sigma_x^2 = \frac{\sum_{i=1}^{N} (x_i - \bar{x})^2}{N} \tag{74}$$

and

$$\sigma_{xy} = \frac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{N} \tag{75}$$

Hence,

$$\hat{a} = \frac{\sigma_{xy}}{\sigma_x^2} \tag{76}$$

2. Calculating the weight variances

The data are generated from

$$y_i = ax_i + b + e_i \tag{77}$$

where the noise, e_i has mean zero and variance σ^2. The mean and variance of each data point are

$$E(y_i) = ax_i + b \tag{78}$$

and

$$Var(y_i) = Var(e_i) = \sigma^2 \tag{79}$$

We now calculate the variance of the estimate $\hat{a}$. From section 2 we see that

$$\hat{a} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{80}$$

Let

$$c_i = \frac{(x_i - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{81}$$

We also note that $\sum_{i=1}^{N} c_i = 0$ and $\sum_{i=1}^{N} c_i x_i = 1$.
Hence,

$$\begin{aligned} \hat{a} &= \sum_{i=1}^{N} c_i(y_i - \bar{y}) && (82) \\ &= \sum_{i=1}^{N} c_i y_i - \bar{y}\sum_{i=1}^{N} c_i \end{aligned}$$

$$\tag{83}$$

The mean estimate is therefore

$$\begin{aligned} E(\hat{a}) &= \sum_{i=1}^{N} c_i E(y_i) - \bar{y}\sum_{i=1}^{N} c_i && (84) \\ &= a\sum_{i=1}^{N} c_i x_i + b\sum_{i=1}^{N} c_i - \bar{y}\sum_{i=1}^{N} c_i \\ &= a \end{aligned}$$

$$\tag{85}$$

The variance is

$$Var(\hat{a}) = Var(\sum_{i=1}^{N} c_i y_i - \bar{y}\sum_{i=1}^{N} c_i) \tag{86}$$

The second term contains two fixed quantities so acts like a constant. From Appendix A we see that

$$\begin{aligned} Var(\hat{a}) &= Var(\sum_{i=1}^{N} c_i y_i) &(87) \\ &= \sum_{i=1}^{N} c_i^2 Var(y_i) \\ &= \sigma^2 \sum_{i=1}^{N} c_i^2 \\ &= \frac{\sigma^2}{\sum_{i=1}^{N}(x_i - \bar{x})^2)} \end{aligned}$$

(88)

The variance of $\hat{b}$ is derived in a similar manner. It is, however, correlated with $\hat{a}$. For this reason we do not go into the derivation as the result is not subsequently used. Instead, we look at a re-parameterised model.

3. Calculating the weight variances of the re-parameterised model

We now re-write equation (1) by adding $a\bar{x} - a\bar{x}$ to the right hand side.

$$y_i = ax_i - a\bar{x} + a\bar{x} + b + e_i \tag{89}$$

We define a third parameter

$$c = a\bar{x} + b \tag{90}$$

Hence,

$$y_i = a(x_i - \bar{x}) + c + e_i \tag{91}$$

This data is modelled with

$$\hat{y}_i = \hat{a}(x_i - \bar{x}) + \hat{c} + e_i \tag{92}$$

The variable $\hat{a}$ is evaluated exactly as before. It's mean and variance are also the same as before. For $\hat{c}$, we get

$$E(\hat{c}) = \bar{y} \tag{93}$$

and

$$Var(\hat{c}) = Var(\bar{y}) = \frac{\sigma^2}{N} \tag{94}$$

Now, because the slope of the line in no way depends on the average target value we have

$$Var(\hat{a}, \hat{c}) = 0 \tag{95}$$

APPENDIX C. Multivariate linear regression

1. Estimating the weights

We wish to find the weight vector that minimises the cost function

$$E = (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}})^T(\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}}) \tag{96}$$

where $\boldsymbol{X}$ is an N-by-p matrix whose rows are made up of different training vectors, an d $\boldsymbol{y}$ is the corresponding vector of targets. By expanding the brackets and collecting terms (using the matrix identity $(\boldsymbol{AB})^T = \boldsymbol{B}^T\boldsymbol{A}^T$) we get

$$E = \boldsymbol{y}^T\boldsymbol{y} - 2\hat{\boldsymbol{w}}\boldsymbol{X}^T\boldsymbol{y} - \hat{\boldsymbol{w}}^T\boldsymbol{X}^T\boldsymbol{X}\hat{\boldsymbol{w}} \tag{97}$$

Equating the first derivative to zero gives

$$(\boldsymbol{X}^T\boldsymbol{X})\hat{\boldsymbol{w}} = \boldsymbol{X}^T\boldsymbol{y} \tag{98}$$

Hence,

$$\hat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{99}$$

2. Estimating the weight covariance matrix

Different instantiations of target noise will generate different estimated weight vectors according to the last equation. The corresponding weight covariance matrix is given by

$$Var(\hat{\boldsymbol{w}}) = Var((\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}) \tag{100}$$

Substituting $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{e}$ gives

$$Var(\hat{\boldsymbol{w}}) = Var((\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} + (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{e}) \tag{101}$$

This is in the form of equation 59 in Appendix A with $\boldsymbol{d}$ being given by the first term which is constant, $\boldsymbol{C}$ being given by $(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ and $\boldsymbol{z}$ being given by $\boldsymbol{e}$. Hence,

$$\begin{aligned} Var(\hat{\boldsymbol{w}}) &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T[Var(\boldsymbol{e})][(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T]^T \\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T(\sigma^2\boldsymbol{I})[(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T]^T \\ &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T(\sigma^2\boldsymbol{I})\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1} \end{aligned} \tag{102}$$

Re-arranging further gives

$$Var(\hat{\boldsymbol{w}}) = \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1} \tag{103}$$