Using Neural Networks to Model Conditional Multivariate Densities

Peter M Williams
School of Cognitive and Computing Sciences
University of Sussex
Falmer, Brighton BN1 9QH
email: peterw@cogs.susx.ac.uk

CSRP 371

February 15, 1995

Abstract

Neural network outputs are interpreted as parameters of statistical distributions. This allows us to fit conditional distributions in which the parameters depend on the inputs to the network. We exploit this in modelling multivariate data, including the univariate case, in which there may be input-dependent (e.g. time-dependent) correlations between output components. This provides a novel way of modelling conditional correlation as well as providing input-dependent (local) error bars.

1 Introduction

Neural networks provide a way of modelling the statistical relationship between a dependent variable Y and an independent variable X. For example, X could be financial data up to a certain time and Y could be a future stock index, exchange rate, option price etc. Alternatively X could represent geophysical features of a prospect and Y could represent mineralization at a certain depth. In general X and Y can be vectors of continuous or discrete quantities.

Suppose that the conditional distribution of Y belongs to a family of distributions characterised by a finite set of parameters which are functions of conditioning values of X. These functions, which in general will be non-linear, can then be modelled by a neural network. For discrete distributions this approach has been known for some time in the form of the softmax rule (Bridle, 1990). Bishop (1994) extends this framework to absolutely continuous distributions, in particular to the case of finite Gaussian mixtures. The case of a single kernel is treated independently by Nix and Weigend (1995). Bishop uses radial kernels though it is straightforward to extend the approach to Gaussians with diagonal covariance matrices. The purpose

of this paper is to consider the case of multivariate data in which the conditional covariance matrix may be non-diagonal.

2 Multivariate data

The conditional distribution of the n-dimensional quantity Y given $X = x$ is assumed to be described by the multivariate Gaussian density

$$P(y \mid x) \;=\; (2\pi)^{-n/2}\,|\Sigma|^{-1/2}\exp\left\{-\tfrac{1}{2}(y-\mu)^{\mathrm{T}}\Sigma^{-1}(y-\mu)\right\} \tag{1}$$

where $\mu(x)$ is the vector of conditional means and $\Sigma(x)$ is the conditional covariance matrix. Both μ and Σ are understood to be functions of x in a way that depends on the outputs of a neural network when the conditioning vector x is given as input.

It is assumed that the network has linear output units and that μ and Σ are determined by the activations of these units. We now discuss the link between network outputs and the components of μ and Σ. The **mean** presents no problem. The network will be required to have n output units whose activations, $\{z_i^{\mu}\}$ say, are related to the n components of μ by

$$\mu_i \;=\; z_i^{\mu} \qquad\qquad i = 1, \ldots, n. \tag{2}$$

These units compute the components of the mean directly. It is less obvious how to represent the **covariance matrix**. Being symmetric Σ has at most $n(n+1)/2$ independent entries but it must also be positive definite.[1] The problem is to parameterise the class of symmetric positive definite matrices in such a way that (i) the parameters can take any values independently in R^n (ii) the determinant is a simple expression of the parameters and (iii) the correspondence is bijective.

To solve this problem we recall the Cholesky factorisation of a symmetric positive definite matrix as $A^{\mathrm{T}}A$ where A is upper triangular with strictly positive diagonal elements. The square root of the determinant of $A^{\mathrm{T}}A$ is the product of the diagonal elements of A. Conversely if A is any upper triangular matrix with strictly positive diagonal entries, $A^{\mathrm{T}}A$ is symmetric positive definite and the correspondence is unique.[2] Applying this factorisation to the **inverse** covariance matrix when $n = 4$, for example, gives

$$\Sigma^{-1} \;=\; A^{\mathrm{T}}A \;=\; \begin{pmatrix} \alpha_{11} & 0 & 0 & 0 \\ \alpha_{12} & \alpha_{22} & 0 & 0 \\ \alpha_{13} & \alpha_{23} & \alpha_{33} & 0 \\ \alpha_{14} & \alpha_{24} & \alpha_{34} & \alpha_{44} \end{pmatrix} \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ 0 & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ 0 & 0 & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & \alpha_{44} \end{pmatrix} \tag{3}$$

with

$$|\Sigma|^{-1/2} \;=\; \alpha_{11}\alpha_{22}\alpha_{33}\alpha_{44}.$$

[1] We restrict to the proper case where Σ is invertible.

[2] The diagonal entries of A are the square roots of the pivots under Gaussian elimination (Strang, 1988; Horn & Johnson, 1985; Golub & Loan, 1989). Note that every positive definite matrix is invertible, the inverse of a positive definite matrix is positive definite and every symmetric positive definite matrix is the covariance matrix of some multivariate Gaussian.

To represent the matrix A we stipulate that the network is provided with an additional set of *dispersion* output units whose activations $\{z_i^\pi\}$ and $\{z_{ij}^\alpha\}$ are related to the elements of A by

$$\alpha_{ii} = \exp z_i^\pi \qquad i = 1, \ldots, n \tag{4}$$

$$\alpha_{ij} = z_{ij}^\alpha \qquad i = 1, \ldots, n-1, \quad j = 2, \ldots, n, \quad i < j. \tag{5}$$

In this way n network outputs (2) are needed for the mean, another n for the positive diagonal entries (4) and $n(n-1)/2$ for the off-diagonal entries (5) making $n(n+3)/2$ in all.[3] Every possible assignment of real values to these outputs corresponds to one and only one multivariate Gaussian.

Note that Σ can be recovered by inverting Σ^{-1}. This is easy to compute now that Σ^{-1} is known as the product (3) of lower and upper triangular matrices (Press et al., 1992, Ch.2).

3 Likelihood

Suppose N pairs of corresponding observations $\{(x_p, y_p) : p = 1, \ldots, N\}$ have been made on X and Y. The negative conditional log likelihood of the data is assumed to factorise as

$$E = \sum_{p=1}^{N} E_p \tag{6}$$

where from (1) the negative log likelihood of an individual pattern is

$$E_p = \tfrac{1}{2} \log |\Sigma_p| + \tfrac{1}{2} (y_p - \mu_p)^{\mathrm{T}} \Sigma_p^{-1} (y_p - \mu_p) \tag{7}$$

apart from a constant.[4] Maximum likelihood estimation would seek network weights w that minimise E. Whatever form of estimation is used, with or without some form of regularisation, the gradient of (6) with respect to network weights is of interest.

Concentrating on (7) and omitting the suffix p we define

$$\eta_i = \mu_i - y_i \qquad i = 1, \ldots, n$$

$$\xi_i = \sum_{j=i}^{n} \alpha_{ij} \eta_j \qquad i = 1, \ldots, n.$$

The negative log likelihood for an individual pattern is then

$$E = \sum_{i=1}^{n} \left\{ \tfrac{1}{2} \xi_i^2 - z_i^\pi \right\}$$

[3] Network output activations are likely to be stored in a one-dimensional structure for most implementations. It is left to the reader how to manage the two-dimensional indexing.

[4] It will not be investigated under what assumptions this factorisation over patterns is justified. It is sufficient, but not necessary in the case of equispaced time series data for example, that the observation pairs are jointly independent.

and partial derivatives with respect to network outputs are easily seen to be

$$\begin{aligned}
\frac{\partial E}{\partial z_i^\mu} &= \sum_{j=1}^{i} \xi_j \alpha_{ji} \qquad i = 1, \ldots, n \\
\frac{\partial E}{\partial z_i^\pi} &= \xi_i \eta_i \alpha_{ii} - 1 \qquad i = 1, \ldots, n \\
\frac{\partial E}{\partial z_{ij}^\alpha} &= \xi_i \eta_j \qquad i = 1, \ldots, n-1, \quad j = 2, \ldots, n, \quad i < j.
\end{aligned}$$

These expressions can be used with backpropagation to calculate ∇E with respect to network weights.

3.1 Constant dispersion

It is interesting to consider the special case in which the network weights attached to the dispersion output units vanish. This would be appropriate if the noise distribution were constant over the whole training set. However this case may arise, the activations $\{z_i^\pi\}$ and $\{z_{ij}^\alpha\}$ are then independent of network inputs and determined just by the biases on the corresponding output units. It can then be shown that, at any local minimum of E as a function of weights and biases, the dispersion output biases must assume values such that the inverse of $A^{\mathrm{T}}A$ is the sample covariance matrix

$$S = \frac{1}{N} \sum_{p=1}^{N} (y_p - \mu_p)(y_p - \mu_p)^{\mathrm{T}}$$

where μ_p is the conditional mean for input x_p as computed by the network at this local minimum.[5] Substituting S for each Σ_p in (6) and (7) leads to

$$E = \tfrac{1}{2} N \log |S| + \text{constant} \tag{8}$$

as the expression for the negative log likelihood, permitting dispersion output units to be dispensed with. In the case of univariate data, or more generally of uncorrelated multivariate data, (8) can also be obtained by integrating out the diagonal elements of the covariance matrix using an uninformative prior (Buntine & Weigend, 1991; Williams, 1995). The present approach, however, is more flexible in allowing dispersion to vary over the input domain and, even in the case of constant dispersion for multivariate data, more efficient than tackling (8) directly.

4 Examples

We consider simulated data for which the generating distribution is known.

[5] The proof follows the lines of the usual treatment of maximum likelihood estimators of parameters of multivariate normal distributions, together with their invariance under invertible reparameterisations (Anderson, 1958; Rao, 1973).

4.1 Univariate data

Weigend and Nix (1994) discuss univariate data ($n = 1$) drawn from normal distributions $N(\mu, \sigma)$ with means

$$\mu(x) = \sin(2.5\,x)\sin(1.5\,x)$$

and variances

$$\sigma^2(x) = 0.01 + 0.25\,[1 - \sin(2.5\,x)]^2\,.$$

1000 training examples were generated using this example with x drawn randomly from a uniform distribution on $[0, \pi]$. The training set is shown in Figure 1. Results are shown in Figure 2. These were obtained using a simple fully connected 3-layer network with 1 input unit, 10 hidden units and 2 output units. Networks were trained using the optimisation and regularisation algorithms of Williams (1991, 1995) which pruned the network to 6 hidden units with 23 remaining non-zero weights and biases. Weigend and Nix in fact propose a considerably more complex architecture and training regime. This seems not to be needed by present methods which fit both first and second moments together and appear to give significantly improved results.[6]

4.2 Bivariate data

Continuing this example we consider data drawn from the bivariate normal distribution ($n = 2$) with mean (μ_1, μ_2) and covariance matrix

$$\begin{pmatrix} \sigma_1^2 & \rho\,\sigma_1\sigma_2 \\ \rho\,\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

where the means are given by

$$\begin{aligned} \mu_1(x) &= \sin(2.5\,x)\sin(1.5\,x) \\ \mu_2(x) &= \cos(3.5\,x)\cos(0.5\,x) \end{aligned}$$

the variances by

$$\begin{aligned} \sigma_1^2(x) &= 0.01 + 0.25\,[1 - \sin(2.5\,x)]^2 \\ \sigma_2^2(x) &= 0.01 + 0.25\,[1 - \cos(3.5\,x)]^2 \end{aligned}$$

and the correlation coefficient by

$$\rho(x) = \sin(2.5\,x)\cos(0.5\,x).$$

[6] To investigate variability between local minima, 20 similar networks were trained and the results averaged. For the mean this gives $\mu = \langle \mu_k \rangle$ and for the variance $\sigma^2 = \langle \sigma_k^2 \rangle + \{ \langle \mu_k^2 \rangle - \langle \mu_k \rangle^2 \}$ where $\mu_k(x)$ and $\sigma_k^2(x)$ are the mean and variance for the kth network, $k = 1, \ldots, 20$, and $\langle \mu_k \rangle$ is the average of the means etc. The results for $\mu(x)$ and $\sigma(x)$ for the mixture are indistinguishable at this scale from those shown in Figure 2. Note that this form of averaging corresponds to rudimentary integration of the predictive distribution over weight space (Buntine & Weigend, 1991; Neal, 1992, 1995).

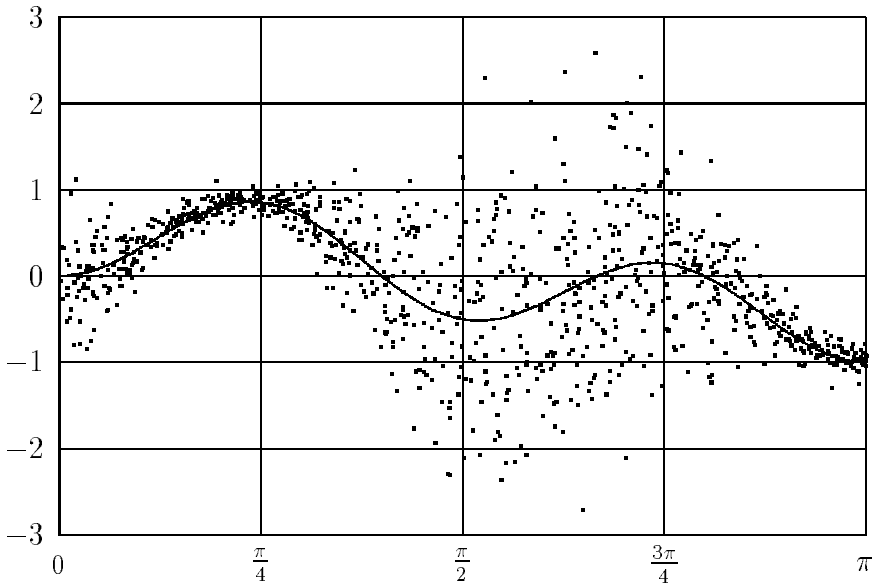

Figure 1: Training set for the univariate case showing the random distribution of training data around the mean $\mu(x) = \sin(2.5\,x)\sin(1.5\,x)$ for $0 < x < \pi$ with variance $\sigma^2(x) = 0.01 + 0.25\,[1 - \sin(2.5\,x)]^2$.

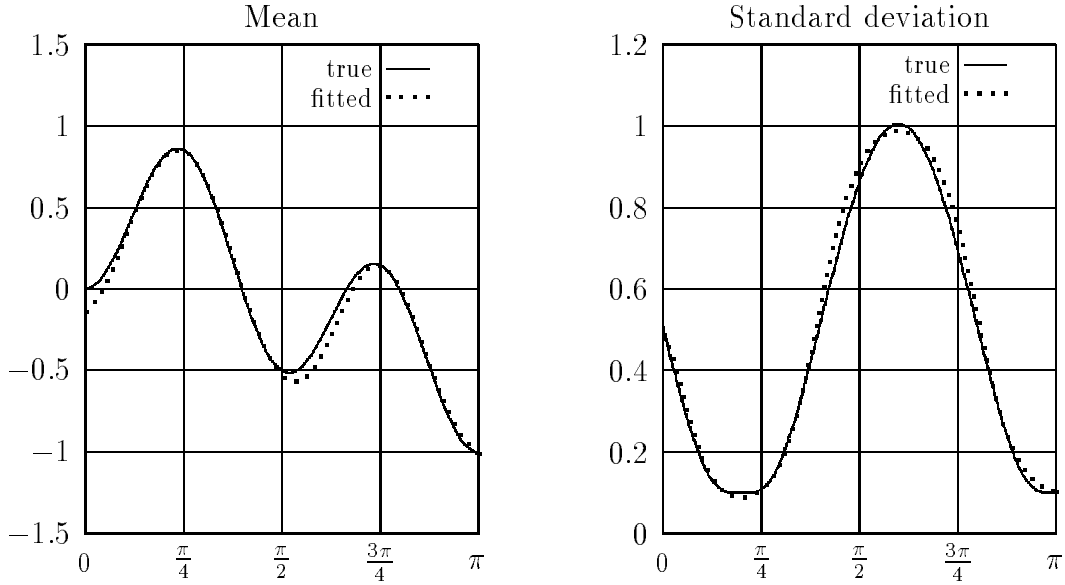

Figure 2: Neural network fit for univariate data using a 3-layer network.

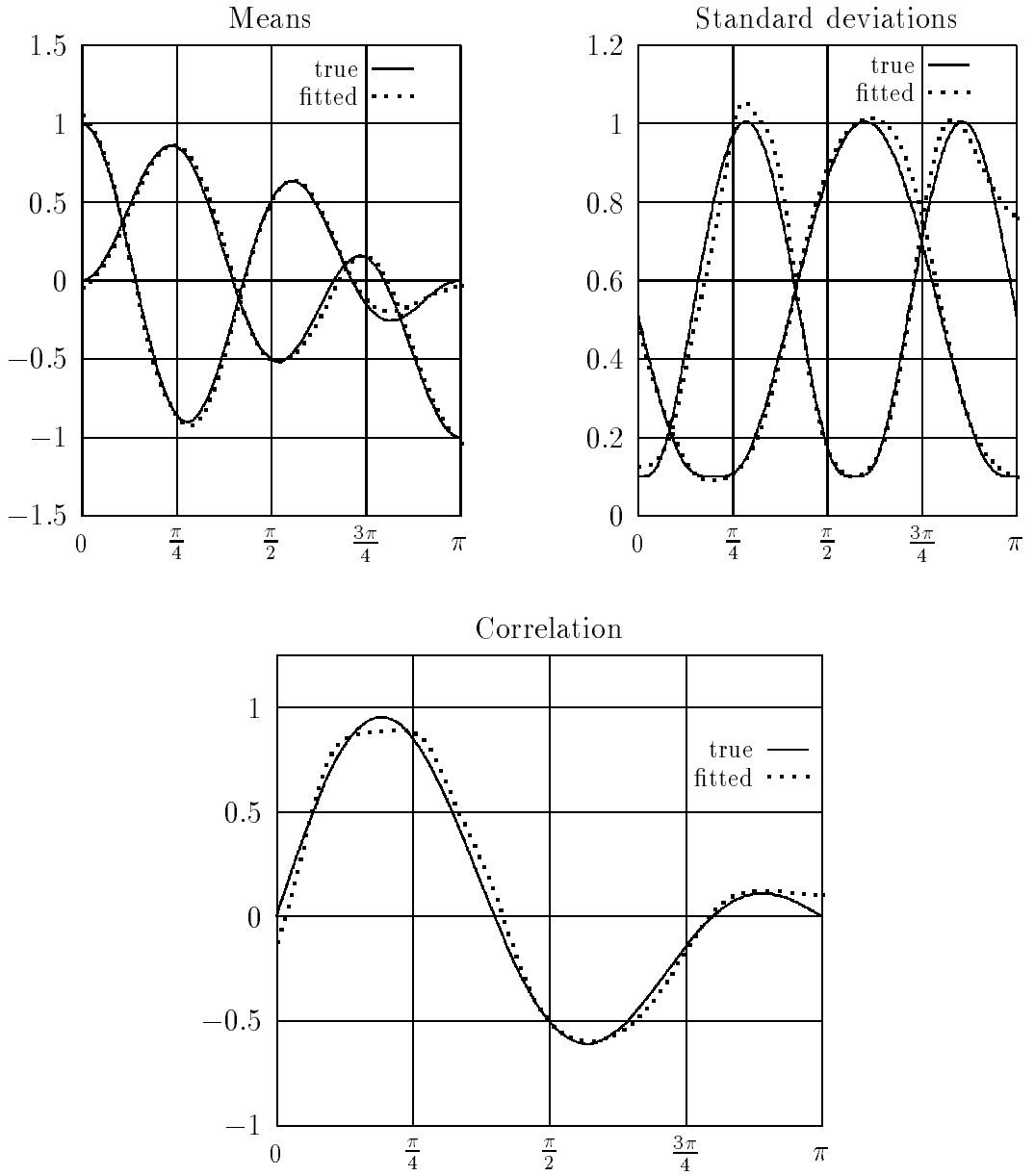

Figure 3: Neural network fit for bivariate data.

3000 training examples were generated with x randomly distributed over $[0, \pi]$.[7] These were modelled using a fully connected 3-layer network with 1 input unit, 20 hidden units and 5 output units (2 for the means and 3 for the inverse covariance matrix). As an effect of Laplace regularisation these were pruned to 12 hidden units with 62 non-zero weights and biases. Results are shown in Figure 3. These show a reasonable fit for most of the interval.

[7]Specifically y_1, y_2 were generated as $y_1 = \mu_1 + \sigma_1(\alpha\xi_1 + \beta\xi_2)$ and $y_2 = \mu_2 + \sigma_2(\alpha\xi_1 - \beta\xi_2)$ where $\alpha^2 = \frac{1}{2}(1+\rho)$ and $\beta^2 = \frac{1}{2}(1-\rho)$ with ξ_1, ξ_2 being independent standard normal deviates.

5 Conclusion

Modelling correlation inevitably requires larger samples. For the specific bivariate example considered, the fit to the covariance matrix is significantly poorer for samples of less than around 2000. More data points are needed in the neighbourhood of a given input x to obtain a reliable estimate of the local pairwise correlations than to estimate just the means and variances. The extent of this need depends on the smoothness of the functional dependence on x. Note that there is no difficulty in finding a neural network model for the five functions in Figure 3 if direct examples of each are given. The problem is to extract the local values $\mu(x)$ and $\Sigma(x)$ from input-dependent statistical properties of the sample when only the noisy data pairs $(x, (y_1, y_2))$ are given.

For smaller samples a better estimate of the variance of individual components might be obtained by modelling each component separately. This approach, however, would normally make the assumption, which in practice it is hard to avoid, that the likelihood factorises over patterns as in (6). In order to achieve this factorisation it may be necessary, in the case of time series data for example, to train on possibly correlated multivariate targets rather than on single items. This will be the subject of a separate paper. The present paper shows how this can be achieved in an efficient way. Modelling conditional correlation is, in any case, a subject of interest in its own right and the present methods provide a new and effective approach.

References

Anderson, T. W. 1958. *An Introduction to Multivariate Statistical Analysis*. Wiley.

Bishop, C. M. 1994. Mixture density networks. Neural computing research group report NCRG/4288, Aston University.

Bridle, J. S. 1990. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Fogelman Soulié, F., and Hérault, J., eds., *Neurocomputing: Algorithms, Architectures and Applications*. Springer-Verlag.

Buntine, W. L., and Weigend, A. S. 1991. Bayesian back-propagation. *Complex Systems*, *5*, 603–643.

Golub, G. H., and Loan, C. F. V. 1989. *Matrix Computations* (2nd edition). The Johns Hopkins University Press.

Horn, R. A., and Johnson, C. R. 1985. *Matrix Analysis*. Cambridge University Press.

Neal, R. M. 1992. Bayesian training of backpropagation networks by the hybrid Monte Carlo method. Technical report CRG-TR-92-1, Department of Computer Science, University of Toronto.

Neal, R. M. 1995. *Bayesian Learning for Neural Networks.* Ph.D. thesis, Graduate Department of Computer Science, University of Toronto.

Nix, D. A., and Weigend, A. S. 1995. Local error bars for nonlinear regression and time series prediction. In Tesauro, G., Touretzky, D. S., and Leen, T. K., eds., *Advances in Neural Information Processing Systems 7.* MIT Press.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. 1992. *Numerical Recipes in C* (2nd edition). Cambridge University Press.

Rao, C. R. 1973. *Linear Statistical Inference and Its Applications* (2nd edition). Wiley.

Strang, G. 1988. *Linear Algebra and its Applications* (3rd edition). Harcourt Brace Jovanovich.

Weigend, A. S., and Nix, D. A. 1994. Predictions with confidence intervals (local error bars). In *Proceedings of the International Conference on Neural Information Processing*, pp. 847–852 Seoul, Korea.

Williams, P. M. 1991. A Marquardt algorithm for choosing the step-size in backpropagation learning with conjugate gradients. Cognitive Science Research Paper CSRP 229, University of Sussex.

Williams, P. M. 1995. Bayesian regularization and pruning using a Laplace prior. *Neural Computation*, 7, 117–143.