Matrix Logarithm Parametrizations for Regularized Neural Network Regression Models

Peter M Williams

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

November 5, 1997

CSRP 470

Abstract

Neural networks are commonly used to model conditional probability distributions. The idea is to represent distributional parameters as functions of conditioning events, where the function is determined by the architecture and weights of the network. An issue to be resolved is the link between distributional parameters and network outputs. The latter are unconstrained real numbers whereas distributional parameters may be required to lie in proper subsets, or be mutually constrained, e.g. by the positive definiteness requirement for a covariance matrix. The paper explores the matrix-logarithm parametrization of covariance matrices for multivariate normal distributions. From a Bayesian point of view the choice of parametrization is linked to the choice of prior. This is treated by investigating the invariance of predictive distributions, for the chosen parametrization, with respect to an important class of priors.

1 Introduction

Neural networks are now commonly used to model conditional probability distributions (Ghahramani & Jordan, 1994; Nix & Weigend, 1995; Bishop & Legleye, 1995; Williams, 1996; Baldi & Chauvin, 1996; Williams, 1998). The idea is for the neural network to output distributional parameters of the conditional distribution. These parameters are taken to be functions of conditioning events, where the function is determined by weights in the network, as well as by the underlying architecture.

An issue to be resolved is the link between the distributional parameters and network outputs. The latter are primarily unconstrained real numbers, whereas distributional parameters may have to lie in a restricted subset. More problematically, there may be mutual constraints between distributional parameters. The case considered here is the positive definiteness requirement for a covariance matrix, which arises when the conditional distribution is multivariate normal. This was previously treated in Williams (1996). Here we consider an alternative unconstrained parametrization.

From a Bayesian point of view the choice of parametrization is also closely linked with the choice of prior. We explore this issue to the extent of investigating the invariance of predictive distributions, for the chosen parametrization, with respect to a class of priors of the 'weight decay' type. Although this is a limited class, priors of this type are often used in practical applications.

2 Multivariate distributions

The conditional distribution of the n-dimensional quantity $\mathbf{Y}$, given $\mathbf{X} = \mathbf{x}$, is assumed to be given by the multivariate Gaussian density

$$p(\mathbf{y}|\mathbf{x}) = (2\pi)^{-n/2} (\det \mathbf{\Sigma})^{-1/2} \exp \left\{ -\tfrac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^{\mathrm{T}} \mathbf{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu}) \right\} \tag{1}$$

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

It is assumed that the network has linear output units and that $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ are determined by the activations of these units. We now discuss the link between network outputs and the components of $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$. The **mean** presents no problem. The network will be required to have n output units whose activations correspond directly to the n components μ_i $(i = 1, \ldots, n)$ of the mean. It is less obvious how to represent the **covariance matrix**. Being symmetric $\mathbf{\Sigma}$ has at most $\frac{1}{2}n(n+1)$ independent entries but it must also be positive definite, assuming we restrict to the proper case where $\mathbf{\Sigma}$ is invertible. Ideally we should choose to parameterize the class of symmetric positive definite matrices in such a way that the parameters can freely assume any real values, and the correspondence is bijective.

Pinheiro and Bates (1996) discuss a number of such unconstrained parametrizations, including the log-Cholesky parametrization used with neural networks in Williams (1996). We explore here the **matrix-logarithm** parametrization.[1] Although this has additional computational overheads, it has considerable advantages; in particular it is invariant under permutations of variables (see §5.2.1 below). We therefore parametrize $\mathbf{\Sigma}$ by the components of $\mathbf{A} = \log \mathbf{\Sigma}$, which is well-defined for any real symmetric positive definite matrix $\mathbf{\Sigma}$. For example if

$$\mathbf{\Sigma} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

[1] This parametrization was used by Leonard and Hsu (1992) to determine a class of priors for a covariance matrix. It also forms the basis of the generalized linear models of Chiu et al. (1996).

with $|\rho| < 1$ then

$$\log \mathbf{\Sigma} \;=\; \frac{1}{2}\begin{bmatrix} \log(1-\rho^2) & \log\dfrac{1+\rho}{1-\rho} \\ \log\dfrac{1+\rho}{1-\rho} & \log(1-\rho^2) \end{bmatrix}.$$

Conversely if $\mathbf{A}$ is any real symmetric matrix, then $\mathbf{\Sigma} = \exp \mathbf{A}$ is symmetric positive definite and the correspondence between $\mathbf{A}$ and $\mathbf{\Sigma}$ is bijective. We therefore stipulate that the network is provided with an additional set of *dispersion* output units whose activations correspond directly to the diagonal and above-diagonal elements α_{ij} $(i \le j)$ of $\mathbf{A} = \log \mathbf{\Sigma}$. In this way n network outputs are needed for the mean and another $\frac{1}{2}n(n+1)$ for the log covariance matrix.

3 Likelihood

Suppose N pairs of corresponding observations $\{(\mathbf{x}_k, \mathbf{y}_k) : k = 1, \ldots, N\}$ have been made on $\mathbf{X}$ and $\mathbf{Y}$. The negative conditional log likelihood of the data is assumed to factorize as $\sum_{k=1}^{N} E_k$ where, from (1), the negative log likelihood of an individual observation is

$$E_k = \tfrac{1}{2}\log(\det \mathbf{\Sigma}_k) + \tfrac{1}{2}(\mathbf{y}_k - \boldsymbol{\mu}_k)^{\mathrm{T}}\mathbf{\Sigma}_k^{-1}(\mathbf{y}_k - \boldsymbol{\mu}_k) + \text{constant}. \tag{2}$$

Recall that $\boldsymbol{\mu}_k$ and $\mathbf{\Sigma}_k$ are the conditional mean and covariance matrix, as determined by network outputs when $\boldsymbol{x}_k$ is given as input. Assuming this factorization of the likelihood function, we can concentrate on the log likelihood of a single observation.[2] Both the log likelihood of the full data, and any of its derivatives, can then be obtained by summation.

Omitting the subscript k in equation (2) and replacing $\mathbf{\Sigma}$ by $\exp \mathbf{A}$, the negative log likelihood of an individual observation can be written as

$$E = \tfrac{1}{2}\,\text{trace}\,\mathbf{A} \;+\; \tfrac{1}{2}(\mathbf{y} - \boldsymbol{\mu})^{\mathrm{T}} \exp(-\mathbf{A})\,(\mathbf{y} - \boldsymbol{\mu}) \;+\; \text{constant} \tag{3}$$

where we have used the fact that

$$\det(\exp \mathbf{A}) = \exp(\text{trace}\,\mathbf{A}).$$

Evaluation of the trace in (3) is immediate. Efficient algorithms for calculating the matrix exponential directly, without prior knowledge of eigenvalues or eigenvectors, can be found in Golub and Van Loan (1989) and Najfeld and Havel (1995). For simplicity in calculating partial derivatives, however, we shall assume the spectral representation

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} \tag{4}$$

[2] For this factorization to be justified, it is sufficient for the observation to be independent, but not necessary (Williams, 1996).

where $\mathbf{U}$ is orthogonal and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ is the matrix of eigenvalues. It follows that

$$\mathbf{\Sigma^{-1}} = \exp(-\mathbf{A}) = \mathbf{U}\exp(-\mathbf{\Lambda})\,\mathbf{U^{-1}} \tag{5}$$

where $\exp(-\mathbf{\Lambda}) = \mathrm{diag}(e^{-\lambda_1}, \ldots, e^{-\lambda_n})$. The elements σ^{ij} of $\mathbf{\Sigma^{-1}}$ are therefore given by

$$\sigma^{ij} = \sum_k u_{ik}\, u_{jk}\, e^{-\lambda_k}$$

and hence, writing

$$\eta_i = y_i - \mu_i,$$

(3) can be calculated directly as

$$E = \tfrac{1}{2}\sum_i \alpha_{ii} \;+\; \tfrac{1}{2}\sum_{i,j} \eta_i \eta_j \sigma^{ij}$$

where we have discarded the constant. For efficiency note that σ^{ij}, and hence the double summation, is symmetric in i and j.

3.1 Partial derivatives

Whatever form of model fitting is used, the gradient of (3) with respect to network weights is of interest. This is straightforward to calculate using backpropagation (Bishop, 1995; Ripley, 1996) if we know the gradient of (3) with respect to network outputs, i.e. with respect to each μ_i and α_{ij} $(i \leq j)$. Partial derivatives with respect to μ_i are simple to calculate and given by

$$\frac{\partial E}{\partial \mu_i} \;=\; -\sum_j \eta_j \sigma^{ij}. \tag{6}$$

To compute the partial derivatives with respect to α_{ij}, we need to know effect on the entire matrix exponential of perturbing a single entry in the matrix $\mathbf{A}$. The results are as follows; a proof may be found in Appendix A.

Let the $n \times n$ array elements ϕ_{ij} be defined by

$$\phi_{ij} \;=\; \begin{cases} \dfrac{e^{-\lambda_i} - e^{-\lambda_j}}{\lambda_i - \lambda_j} & \text{if } \lambda_i \neq \lambda_j \\ -e^{-\lambda_i} & \text{if } \lambda_i = \lambda_j \end{cases}$$

and write

$$\xi_i \;=\; \sum_j \eta_j u_{ji}$$

and

$$\psi_{ij} \;=\; \xi_i\, \xi_j\, \phi_{ij}.$$

Partial derivatives with respect to α_{ij} $(i \leq j)$ are then given by

$$\frac{\partial E}{\partial \alpha_{ij}} = \begin{cases} \sum\limits_{k,l} u_{ik} u_{jl} \psi_{kl} & \text{if } i < j \\ \frac{1}{2}\Big(1 + \sum\limits_{k,l} u_{ik} u_{jl} \psi_{kl}\Big) & \text{if } i = j. \end{cases} \tag{7}$$

Expressions (6) and (7) can now be used with backpropagation to calculate ∇E with respect to network weights.

3.2 Complexity

The expression of highest complexity in the formulae for E and its derivatives is (7). This is $\mathcal{O}(n^4)$ since it requires a double summation for each of the $\mathcal{O}(n^2)$ parameters α_{ij}. Corresponding complexity for the log-Cholesky parametrization (Williams, 1996) is $\mathcal{O}(n^2)$. It should be noted, however, that calculation of network output activations alone, for this type of network, is typically already $\mathcal{O}(n^4)$. This is because there are $\mathcal{O}(n^2)$ output units and each output unit requires $\mathcal{O}(n^2)$ multiplications and additions if we assume that the number of hidden units is of the same order as the number of outputs units. Thus the inherent $\mathcal{O}(n^4)$ complexity of this type of network already arises from the decision to model the full conditional covariance matrix, rather than from the form of parametrization.

Spectral decomposition (4) is $\mathcal{O}(n^3)$. In cases where n is small, however, this may nonetheless dominate. For example, the straightforward Jacobi method for real symmetric matrices (Press et al., 1992) may require up to $20n^3$ operations, so that more efficient methods would be advantageous for smaller n.

4 Model invariance

We now examine the extent to which the modelling process, using the above parametrization, is invariant under transformations of the target variables $\mathbf{Y}$. We are particularly interested in this question when the model is fitted using some form of regularization to avoid overfitting. Regularization is interpreted as the use of a Bayesian prior over model parameters, and we proceed by reviewing the modelling process from a Bayesian point of view.

4.1 Bayesian model

The conditional distribution of $\mathbf{Y}$, given $\mathbf{X} = \mathbf{x}$, is assumed to belong to a family of distributions parametrized by quantities $\boldsymbol{\theta}$, say. Presently we are considering the multivariate normal family, parametrized by mean and log covariance matrix. These distributional parameters $\boldsymbol{\theta} = \boldsymbol{\theta}(\mathbf{x}; \mathbf{w})$ are understood to be functions of $\mathbf{x}$ in a way that depends on further parameters $\mathbf{w}$ which specify the nature of the function.

Specifically we are considering $\mathbf{w}$ to be the adjustable weights and biases of a neural network. The aim is to determine the density of the predictive distribution

$$p(\mathbf{y}|\mathbf{x}, D) = \int p(\mathbf{y}|\boldsymbol{\theta}(\mathbf{x};\mathbf{w}))\, p(\mathbf{w}|D)\, d\mathbf{w} \tag{8}$$

where D are the observed data and

$$p(\mathbf{w}|D) \propto p(D|\mathbf{w})\, p(\mathbf{w}) \tag{9}$$

is the posterior density for $\mathbf{w}$. Since both the conditional density $p(\mathbf{y}|\boldsymbol{\theta}(\mathbf{x};\mathbf{w}))$ and the likelihood $p(D|\mathbf{w})$ are given by the model, the remaining problems are, first, the conceptual problem of determining the prior $p(\mathbf{w})$ in (9) and, secondly, the computational problem of evaluating the integral in (8). Our considerations depend on the first of these issues, namely the prior, but not on the second.

4.2 Invariance

Suppose that the target variables $\mathbf{Y}$ are transformed to new variables $\mathbf{Y}'$ by an invertible and differentiable transform $\mathbf{Y}' = \boldsymbol{\phi}(\mathbf{Y})$. Assuming that a conditional density $p(\mathbf{y}|\mathbf{x}, D)$ for $\mathbf{Y}$ has already been determined by (8), the density of $\mathbf{Y}'$ is given by

$$p'(\mathbf{y}|\mathbf{x}, D) = \mathbf{J}(\mathbf{y})\, p(\boldsymbol{\phi}^{-1}(\mathbf{y})|\mathbf{x}, D) \tag{10}$$

where $\mathbf{J}$ is the Jacobian of $\boldsymbol{\phi}^{-1}$. Note that p' is obtained by first fitting a model to the original data $D = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\}$ and then transforming the result. But a model could also be fitted directly to the transformed data $D' = \{(\mathbf{x}_1, \mathbf{y}'_1), \ldots, (\mathbf{x}_N, \mathbf{y}'_N)\}$ which would lead, via (8), to the solution $p(\mathbf{y}|\mathbf{x}, D')$ with D' replacing D. We shall say that the model is *invariant under* $\boldsymbol{\phi}$ if

$$p(\mathbf{y}|\mathbf{x}, D') \;\equiv\; p'(\mathbf{y}|\mathbf{x}, D) \tag{11}$$

identically in $\mathbf{y}$. In that case, essentially the same results are obtained whether we work in terms of $\mathbf{Y}$ or $\mathbf{Y}'$ (compare Bishop, 1995, §9.2.2).[3]

Substituting (10) into (11) and writing $\mathbf{y}' = \boldsymbol{\phi}(\mathbf{y})$, we obtain the equivalent condition

$$p(\mathbf{y}'|\mathbf{x}, D') \;\equiv\; \mathbf{J}(\mathbf{y})\, p(\mathbf{y}|\mathbf{x}, D). \tag{12}$$

For linear transformations $\mathbf{Y}' = \mathbf{B}\mathbf{Y} + \mathbf{c}$, where $\mathbf{B}$ is an invertible matrix and $\mathbf{c}$ is a vector offset, (12) simplifies to

$$p(\mathbf{y}'|\mathbf{x}, D') \;\propto\; p(\mathbf{y}|\mathbf{x}, D) \tag{13}$$

since $\mathbf{J}(\mathbf{y}) = |\det \mathbf{B}|^{-1}$ is then independent of $\mathbf{y}$. These are the cases we consider below.

[3] It would be more correct to speak of *covariance*, rather than *invariance*, since the density changes, but covariantly with the transformation of variables. The present terminology, however, may seem more natural to some readers.

4.3 Weight priors

To proceed further, we have to be more specific about the prior. We shall restrict attention to prior densities essentially of the form

$$p(\mathbf{w}) \propto \Big(\|\mathbf{w}\|_p\Big)^{-\gamma} \tag{14}$$

for $p = 1, 2$ where

$$\|\mathbf{w}\|_p = \Big(\sum_i |w_i|^p\Big)^{1/p}$$

and γ is a positive constant. The choice of (14) is discussed in Appendix B. The case $p = 1$ will be referred to as the Laplacian prior, and the case $p = 2$ as the Gaussian prior.

In fact we shall deal with a more general class of priors defined as follows. Let $\mathcal{W}_1, \ldots, \mathcal{W}_C$ be a disjoint collection of non-empty subsets of the weights and biases in the network, and let $\mathbf{w}_1, \ldots, \mathbf{w}_C$ be their corresponding weight vectors. Then we consider priors taking the form of the product

$$p(\mathbf{w}) \propto \Big(\|\mathbf{w}_1\|_p\Big)^{-\gamma_1} \cdots \Big(\|\mathbf{w}_C\|_p\Big)^{-\gamma_C} \tag{15}$$

where $\gamma_1, \ldots, \gamma_C$ are positive numbers associated with each class.[4] The classes need not be exhaustive. Weights or biases not belonging to any of these classes are effectively governed by a uniform prior. $\mathcal{W}_1, \ldots, \mathcal{W}_C$ will be referred to as *regularization classes*, and parameters not belonging to any of these classes will be said to be *unregularized.* For definiteness, we assume that the regularization classes are composed as follows:

(C1) all input weights to location output units comprise a single class, $\mathcal{W}_1$ say;

(C2) all input weights to dispersion output units comprise a single class, $\mathcal{W}_2$ say;

(C3) output biases are not included in any of the regularization classes $\mathcal{W}_1, \ldots, \mathcal{W}_C$.

Other weights and biases in the network may be classified freely, for present concerns, provided they are not included in either $\mathcal{W}_1$ or $\mathcal{W}_2$.

4.4 Network architecture

It is assumed that the network has location output units corresponding to the various components of the mean $\boldsymbol{\mu}$, and dispersion outputs units corresponding to the independent elements of the log covariance matrix $\mathbf{A} = \log \boldsymbol{\Sigma}$. Each output unit is understood to be connected to $H > 0$ hidden units. For notational simplicity we assume that the same set of hidden units serves for both location and dispersion output units, but this is not essential.

[4] Compare MacKay (1992). The norm could also differ between classes, but we ignore this for simplicity.

The biases on location output units form a vector which we refer to as $\mathbf{m}_0$. Similarly the weights on connections from a given hidden unit h, to the various location output units, form a vector which we shall refer to as $\mathbf{m}_h$ $(h > 0)$. Writing $z_1, \ldots, z_H$ for the activations of hidden units, the conditional mean is given by the activations of location output units as

$$\boldsymbol{\mu} = \mathbf{m}_0 + \sum_h z_h \mathbf{m}_h \tag{16}$$

where each z_h is a function $z_h(\mathbf{x})$ of the conditioning value $\mathbf{x}$, as well as of remaining weights in the network. Thus the conditional mean is represented as a variable combination of fixed components. Similarly the conditional log covariance matrix is given by the activations of dispersion output units as

$$\log \boldsymbol{\Sigma} = \mathbf{A}_0 + \sum_h z_h \mathbf{A}_h \tag{17}$$

where $\mathbf{A}_0$ is the symmetric array corresponding to the biases on dispersion output units, and $\mathbf{A}_h$ $(h > 0)$ is the symmetric array corresponding to weights on connections from the hth hidden unit to the various dispersion output units.

The norm of $\mathbf{w}_1$, the vector of input weights to location output units, can now be written in terms of the vector norms of $\mathbf{m}_1, \ldots, \mathbf{m}_H$ as

$$\|\mathbf{w}_1\|_p^p \;=\; \|\mathbf{m}_1\|_p^p + \cdots + \|\mathbf{m}_H\|_p^p \tag{18}$$

where the superscript p indicates exponentiation. Similarly the norm of $\mathbf{w}_2$, the vector of input weights to dispersion output units, can be written as

$$\|\mathbf{w}_2\|_p^p \;=\; \|\mathbf{A}_1\|_p^p + \cdots + \|\mathbf{A}_H\|_p^p \tag{19}$$

where $\|\mathbf{A}\|_p$ is the matrix norm[5] of $\mathbf{A}$ defined by

$$\|\mathbf{A}\|_p = \left(\sum_{i,j} |a_{ij}|^p \right)^{1/p} .$$

Note that dispersion output weights are counted in (19) according to their multiplicities in the symmetric matrices $\mathbf{A}_h$, rather than in the network itself where off-diagonal elements occur only once. In terms of Appendix B, this means that diagonal elements are expected to be twice as large as off-diagonal elements. The reason for this assumption is that the resulting prior then has an important invariance property in the Gaussian case. For consistency the same form will be used with the Laplacian prior. In summary, the prior for $\mathbf{w}$ is assumed to be of the form

$$p(\mathbf{w}) \;\propto\; \left(\|\mathbf{m}_1\|_p^p + \cdots + \|\mathbf{m}_H\|_p^p \right)^{-\gamma_1/p} \left(\|\mathbf{A}_1\|_p^p + \cdots + \|\mathbf{A}_H\|_p^p \right)^{-\gamma_2/p} \cdots \tag{20}$$

where reference to further classes has been omitted, since we are only considering changes in variables belonging to $\mathcal{W}_1$ or $\mathcal{W}_2$.

[5] This is indeed a matrix norm for $p = 1, 2$ in that $\|AB\| \leq \|A\| \, \|B\|$ (Horn & Johnson, 1985).

5 Invariance under linear transformations

Restricting attention to linear transforms of the type $\mathbf{Y}' = \mathbf{B}\mathbf{Y} + \mathbf{c}$, we can now discuss invariance in the sense of (13). We have to consider whether $p(\mathbf{y}'|\mathbf{x}, D')$ is proportional to $p(\mathbf{y}|\mathbf{x}, D)$ when both are defined by (8) and (9). A rigorous treatment follows from the rules for change of variables in multiple integrals. Our treatment will be more sketchy, leaving the interested reader to fill in the details. The approach is to determine the changes of variables necessary to preserve the likelihood function, and then to consider the consequences for the prior. Before beginning we recall the following.

1. If the random vector $\mathbf{Y}$ has mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, the random vector $\mathbf{Y}' = \mathbf{B}\mathbf{Y}+\mathbf{c}$ has mean $\boldsymbol{\mu}' = \mathbf{B}\boldsymbol{\mu}+\mathbf{c}$ and covariance matrix $\boldsymbol{\Sigma}' = \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^{\mathrm{T}}$.

2. If $\mathbf{A}$ is a square matrix, f is an analytic matrix function and $\mathbf{B}$ is an invertible matrix of the same size as $\mathbf{A}$, then $f(\mathbf{B}\mathbf{A}\mathbf{B}^{-1}) = \mathbf{B}f(\mathbf{A})\mathbf{B}^{-1}$. In particular, if $\mathbf{B}$ is orthogonal, then $\log(\mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^{\mathrm{T}}) = \mathbf{B}(\log\boldsymbol{\Sigma})\mathbf{B}^{\mathrm{T}}$.

5.1 Common change of scale

Consider first the case $\mathbf{B} = b\mathbf{I}$, where b is a non-zero scalar and $\mathbf{I}$ is the identity matrix. This means that $\mathbf{Y}$ transforms to

$$\mathbf{Y}' = b\mathbf{Y} + \mathbf{c} \tag{21}$$

which amounts to a common rescaling of all components of $\mathbf{Y}$ followed by a displacement. The transformed mean is $\boldsymbol{\mu}' = b\boldsymbol{\mu} + \mathbf{c}$ and the transformed covariance matrix is $\boldsymbol{\Sigma}' = b^2\boldsymbol{\Sigma}$ so that $\log\boldsymbol{\Sigma}' = \log\boldsymbol{\Sigma} + \beta\mathbf{I}$, where $\beta = \log b^2$. The network will now output the transformed conditional mean and log covariance matrix, identically in $z_1, \ldots, z_H$, if and only if weights and biases in the output layer are transformed by

$$\begin{aligned} \mathbf{m}_0' &= b\mathbf{m}_0 + \mathbf{c} && (22)\\ \mathbf{m}_h' &= b\mathbf{m}_h \qquad (h = 1, \ldots, H) && (23) \end{aligned}$$

and

$$\begin{aligned} \mathbf{A}_0' &= \mathbf{A}_0 + \beta\mathbf{I} && (24)\\ \mathbf{A}_h' &= \mathbf{A}_h \qquad (h = 1, \ldots, H). && (25) \end{aligned}$$

It is then easy to verify that

$$\begin{aligned} p(\mathbf{y}'|\boldsymbol{\theta}(\mathbf{x};\mathbf{w}')) &\propto p(\mathbf{y}|\boldsymbol{\theta}(\mathbf{x};\mathbf{w})) && (26)\\ p(D'|\mathbf{w}') &\propto p(D|\mathbf{w}) && (27) \end{aligned}$$

for the transformation from $\mathbf{w}$ to $\mathbf{w}'$ corresponding to (22)–(25). It only remains to consider the effect on $p(\mathbf{w})$. Since biases are excluded from $\mathcal{W}_1$ and $\mathcal{W}_2$, transformations of $\mathbf{m}_0$ and $\mathbf{A}_0$ leave $p(\mathbf{w})$ unchanged. Remaining $\mathbf{A}_h$ are unaffected, hence the only change to be considered is $\mathbf{m}'_h = b\mathbf{m}_h$ for $h = 1, \ldots, H$. Since $\|b\mathbf{m}\| = |b|\,\|\mathbf{m}\|$, the first term on the right of (20) is changed only by the constant multiplicative factor $|b|^{-\gamma_1}$ and hence $p(\mathbf{w}') \propto p(\mathbf{w})$. The condition for invariance (13) now follows from the fact that the Jacobian of the transformation of weights corresponding to (22)–(25) is constant.

Note that this invariance requires that biases on location outputs should be unregularized. It also requires that input weights to location outputs should belong to different regularization classes from input weights to dispersion outputs, or indeed from any other regularization class. Furthermore, biases on dispersion outputs corresponding to diagonal elements of $\log \boldsymbol{\Sigma}$ should be unregularized. However, this invariance is independent of whether or not the biases for off-diagonal elements of $\log \boldsymbol{\Sigma}$ are regularized (see the end of §5.3 below for further discussion).

5.2 Orthogonal transformations

Suppose now that $\mathbf{Y}$ is transformed to

$$\mathbf{Y}' = \mathbf{B}\mathbf{Y} + \mathbf{c} \tag{28}$$

where $\mathbf{B}$ is orthogonal.[6] The transformed mean is $\boldsymbol{\mu}' = \mathbf{B}\boldsymbol{\mu} + \mathbf{c}$ and the transformed log covariance matrix is $\log \boldsymbol{\Sigma}' = \mathbf{B}(\log \boldsymbol{\Sigma})\mathbf{B}^{\mathrm{T}}$. Then (16) and (17) imply that the network will output the transformed mean and log covariance matrix, identically in $z_1, \ldots, z_H$, if and only if weights and biases in the output layer are transformed by

$$\begin{aligned} \mathbf{m}'_0 &= \mathbf{B}\mathbf{m}_0 + \mathbf{c} && (29)\\ \mathbf{m}'_h &= \mathbf{B}\mathbf{m}_h \qquad (h = 1, \ldots, H) && (30) \end{aligned}$$

and

$$\begin{aligned} \mathbf{A}'_0 &= \mathbf{B}\mathbf{A}_0\mathbf{B}^{\mathrm{T}} && (31)\\ \mathbf{A}'_h &= \mathbf{B}\mathbf{A}_h\mathbf{B}^{\mathrm{T}} \qquad (h = 1, \ldots, H). && (32) \end{aligned}$$

Since (26) and (27) again hold by construction, we need only consider the effect on $p(\mathbf{w})$. If biases are excluded from $\mathcal{W}_1$ and $\mathcal{W}_2$, transformations of $\mathbf{m}_0$ and $\mathbf{A}_0$ again leave $p(\mathbf{w})$ unchanged. For the input weights $\mathbf{m}_h$ to location outputs and $\mathbf{A}_h$ to dispersion outputs, we must now distinguish the cases $p = 1, 2$. For $p = 2$ we have the following unitary invariances:

$$\|\mathbf{B}\mathbf{m}\|_2 = \|\mathbf{m}\|_2$$

for any orthogonal $\mathbf{B}$, and

$$\|\mathbf{B}\mathbf{A}\mathbf{C}\|_2 = \|\mathbf{A}\|_2$$

for any orthogonal $\mathbf{B}$ and $\mathbf{C}$ (Horn & Johnson, 1985, §5.6). These invariances generally fail to hold for $p = 1$. Putting $\mathbf{C} = \mathbf{B}^{\mathrm{T}}$, which is orthogonal whenever $\mathbf{B}$ is, it follows that $p(\mathbf{w}') = p(\mathbf{w})$ for $p = 2$. Invariance for the Gaussian prior then follows

[6] Note that (21) is a special case of (28) only if $b = \pm 1$.

from the fact that the Jacobian of the transformation of weights corresponding to (29)–(32) is constant. Note that, in this case, it is essential that the biases for diagonal elements of $\log \mathbf{\Sigma}$ should be treated in the same way as for off-diagonal elements. Because of (31), they must all belong to the same regularization class, or else all be unregularized. Since invariance under (21) requires that diagonal elements should be unregularized, we conclude that none should be regularized when using the Gaussian prior.

5.2.1 Permutations

An important special case of (28) occurs when $\mathbf{B}$ is a *permutation* matrix, $\mathbf{P}$ say. A permutation matrix has exactly one entry in each row and column equal to 1, with all other entries equal to 0. Multiplication of $\mathbf{Y}$ by $\mathbf{P}$ in (28) simply renumbers the variables. Since $\mathbf{P}$ is orthogonal, $\log(\mathbf{P\Sigma P}^{\mathrm{T}}) = \mathbf{P}(\log \mathbf{\Sigma})\mathbf{P}^{\mathrm{T}}$, so that the components of $\log \mathbf{\Sigma}$ are permuted in the same way. It follows that all solutions will be invariant under such permutations, provided only that the prior is. This is certainly the case for (20), using either the Gaussian or Laplacian priors, since the norms are invariant under permutations. Invariance does not normally hold for the Cholesky parametrization. If $\mathbf{\Sigma} = \mathbf{AA}^{\mathrm{T}}$ is the Cholesky factorization of a symmetric positive definite matrix $\mathbf{\Sigma}$, with $\mathbf{A}$ lower triangular, then $\mathbf{P\Sigma P}^{\mathrm{T}} = \mathbf{PA}(\mathbf{PA})^{\mathrm{T}}$. But this is not generally a Cholesky factorization, since $\mathbf{PA}$ need not be triangular.

5.3 General linear transformations

Invariance does not hold, for either the Gaussian or Laplacian priors, for a general invertible linear transformation. In particular, consider $\mathbf{B} = \mathrm{diag}(b_1, \ldots, b_n)$, which corresponds to an independent rescaling of each of the variables. Unless the b_is are all the same, the eigenvalues of the new covariance matrix $\mathbf{\Sigma}'$ will be different, and $\log \mathbf{\Sigma}'$ will be non-linearly related to $\log \mathbf{\Sigma}$. The solution, for either the Gaussian or Laplacian priors, will depend on the choice of scales for the variables.[7] It therefore seems reasonable to use the unconditional sample variance to standardise all the variables from the outset to have unit variance. If the Gaussian prior is used, it makes no difference whether the resulting standardised variables are decorrelated or not, in view of the invariance under (28). For the Laplace prior, it seems reasonable to decorrelate the variables, and then to apply a "whitening" transform (Fukunaga, 1990). There is then no reason for excluding the biases on off-diagonal elements of $\log \mathbf{\Sigma}$ from regularization. These should therefore be included in class (C2) of §4.3, for the Laplace prior, on the grounds that regularization classes should be maximal subject to enforceable invariances.

[7] Compare Kendall (1980), for example, for a discussion of similar problems with conventional principal component analysis.

6 Conclusion

A basic requirement of consistency for a statistical model is that it should be independent of the arbitrary labelling of variables. The matrix-logarithm parametrization discussed in Section 3 satisfies this condition for any prior which is similarly invariant. Others, including the log-Cholesky parametrization, do not guarantee this invariance even if, in practice, the dependency on the ordering may be small. The matrix-logarithm approach is relatively expensive in computation, but its complexity is no greater than is already inherent in models of this type.

For regularizers of the 'weight decay' type, it has been shown that the matrix-logarithm parametrization leads to solutions that are invariant under translations and common changes of scale. Solutions are also invariant under orthogonal transformations, e.g. rotations, for the Gaussian regularizer. In general, however, solutions may depend on the choice of units for the separate elements of the multivariate random vector, unless suitable normalization is enforced as a pre-requisite of the modelling process.

A Directional derivatives of analytic matrix functions

If $f(z) = \sum_{n=0}^{\infty} c_n z^n$ is an analytic function and $\mathbf{A}$ is a square matrix, let

$$f(\mathbf{A}) = \sum_{n=0}^{\infty} c_n \mathbf{A}^n$$

denote the corresponding function of a matrix argument. In particular the matrix exponential is given by

$$\exp \mathbf{A} = \sum_{n=0}^{\infty} \frac{\mathbf{A}^n}{n!}.$$

Now let $\mathbf{V}$ be a square matrix of the same size as $\mathbf{A}$. The effect on $f(\mathbf{A})$ of perturbing $\mathbf{A}$ in the direction $\mathbf{V}$ is given by

$$\mathbf{D}_{\mathbf{V}}(f(\mathbf{A})) = \lim_{h\to 0} \frac{1}{h}\Big(f(\mathbf{A} + h\mathbf{V}) - f(\mathbf{A})\Big) \tag{33}$$

which defines the first directional derivative of f evaluated at $\mathbf{A}$ in the direction $\mathbf{V}$.

Najfeld and Havel (1995) derive a formula for evaluating (33) in the case where $\mathbf{A}$ has the spectral representation $\mathbf{A} = \mathbf{U\Lambda U^{-1}}$. We bring together their results as follows, using $\odot$ to denote the Hadamard (entry-by-entry) product of similar matrices.

Theorem. *If* $\mathbf{A} = \mathbf{U\Lambda U}^{-1}$ *is the spectral representation of a square matrix* $\mathbf{A}$*, and* $\mathbf{V}$ *is an arbitrary matrix of the same size as* $\mathbf{A}$*, then*

$$\mathbf{D}_{\mathbf{V}}(f(\mathbf{A})) = \mathbf{U}\Big(\overline{\mathbf{V}} \odot \Delta_f(\mathbf{\Lambda})\Big)\,\mathbf{U}^{-1}$$

where $\overline{\mathbf{V}} = \mathbf{U}^{-1}\mathbf{V}\mathbf{U}$ *and* $\Delta_f(\boldsymbol{\Lambda})$ *is the symmetric matrix with* i, j*th entry*

$$\begin{array}{ll} \dfrac{f(\lambda_i) - f(\lambda_j)}{\lambda_i - \lambda_j} & \textit{if } \lambda_i \neq \lambda_j \\ f'(\lambda_i) & \textit{if } \lambda_i = \lambda_j. \end{array}$$

Proof. If $f(\mathbf{A}) = \sum_{n=0}^{\infty} c_n \mathbf{A}^n$ then

$$\mathbf{D}_{\mathbf{V}}(f(\mathbf{A})) = \sum_{n=0}^{\infty} c_n \mathbf{D}_{\mathbf{V}}(\mathbf{A}^n) \tag{34}$$

by linearity of (33). To compute directional derivatives of integer powers $\mathbf{A}^n$ we obtain

$$\mathbf{D}_{\mathbf{V}}(\mathbf{A}^n) = \sum_{r=1}^{n} \mathbf{A}^{n-r}\mathbf{V}\mathbf{A}^{r-1}$$

by considering the coefficient of h in the expansion of $(\mathbf{A} + h\mathbf{V})^n$ in (33). If now $\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{-1}$ so that $\mathbf{A}^s = \mathbf{U}\boldsymbol{\Lambda}^s\mathbf{U}^{-1}$ then

$$\begin{aligned} \mathbf{D}_{\mathbf{V}}(\mathbf{A}^n) &= \sum_{r=1}^{n} \Big(\mathbf{U}\boldsymbol{\Lambda}^{n-r}\mathbf{U}^{-1}\Big)\mathbf{V}\Big(\mathbf{U}\boldsymbol{\Lambda}^{r-1}\mathbf{U}^{-1}\Big) \\ &= \mathbf{U}\left(\sum_{r=1}^{n} \boldsymbol{\Lambda}^{n-r}\,\overline{\mathbf{V}}\,\boldsymbol{\Lambda}^{r-1}\right)\mathbf{U}^{-1} \qquad \text{(where } \overline{\mathbf{V}} = \mathbf{U}^{-1}\mathbf{V}\mathbf{U}\text{)} \\ &= \mathbf{U}\left(\sum_{r=1}^{n} \overline{\mathbf{V}} \odot \boldsymbol{\Psi}(r, n)\right)\mathbf{U}^{-1} \end{aligned}$$

where $\boldsymbol{\Psi}(r, n)$ is the matrix with i, jth element $\lambda_i^{n-r}\lambda_j^{r-1}$. Hence

$$\mathbf{D}_{\mathbf{V}}(\mathbf{A}^n) = \mathbf{U}\Big(\overline{\mathbf{V}} \odot \sum_{r=1}^{n} \boldsymbol{\Psi}(r, n)\Big)\mathbf{U}^{-1} = \mathbf{U}\Big(\overline{\mathbf{V}} \odot \boldsymbol{\Phi}(n)\Big)\mathbf{U}^{-1}$$

where $\boldsymbol{\Phi}(n)$ is the matrix with i, jth element $\phi_{ij}(n) = \sum_{r=1}^{n} \lambda_i^{n-r}\lambda_j^{r-1}$ so, by summation,

$$\phi_{ij}(n) = \left\{ \begin{array}{ll} \dfrac{\lambda_i^n - \lambda_j^n}{\lambda_i - \lambda_j} & \text{if } \lambda_i \neq \lambda_j \\ n\lambda_i^{n-1} & \text{if } \lambda_i = \lambda_j. \end{array} \right.$$

The theorem follows by substituting the expressions for $\mathbf{D}_{\mathbf{V}}(\mathbf{A}^n)$ into (34). ▮

To obtain the effect on each element of $f(\mathbf{A})$ of perturbing a single entry a_{ij} of $\mathbf{A}$, we have to consider the derivative in the elementary direction $\mathbf{E}_{ij}$, where $\mathbf{E}_{ij}$ is the matrix with 1 in the i, jth position and 0 elsewhere. If $\mathbf{A}$ is constrained to be symmetric, then the partial derivatives of interest are given by the matrix

$$\frac{\partial f(\mathbf{A})}{\partial a_{ij}} = \mathbf{D}_{\widehat{\mathbf{E}}_{ij}}(f(\mathbf{A}))$$

where $\widehat{\mathbf{E}}_{ij}$ is the symmetric elementary direction with 1 in the i, jth and j, ith positions and 0 elsewhere (hence a single 1 on the diagonal if $i = j$). The expressions in (7) for the partial derivatives of the log likelihood function (3) can now be obtained by straightforward manipulation.

B Weight prior

This appendix offers a justification for the use of the weight priors (14) (compare Buntine & Weigend, 1991; Williams, 1995).

B.1 Laplacian prior

Suppose that individual network weights are distributed with a Laplace or two-sided exponential density $p(w|\lambda) = (\lambda/2) \exp\{-\lambda\, |w|\}$ where λ^{-1} is a positive scale parameter equal to the expected absolute value of w. Suppose there are W components of the weight vector $\mathbf{w}$. Assuming independence, the prior density for the full weight vector $\mathbf{w}$ is then

$$p(\mathbf{w}|\lambda) = \left(\frac{\lambda}{2}\right)^W \exp\left\{-\lambda\, \|\mathbf{w}\|_1\right\} \tag{35}$$

where the unknown scale parameter λ can be eliminated using

$$p(\mathbf{w}) = \int_0^\infty p(\mathbf{w}|\,\lambda)\, p(\lambda)\, d\lambda \tag{36}$$

if we assume a suitable prior $p(\lambda)$. A natural choice is the conjugate prior (Berger, 1985; Bernardo & Smith, 1994) which, for the Laplace likelihood, is the gamma distribution

$$p(\lambda) = \frac{\beta^\alpha}{\Gamma(\alpha)}\, \lambda^{\alpha-1} \exp\{-\beta\lambda\} \tag{37}$$

for $\alpha, \beta > 0$. Substituting (35) and (37) into (36) we obtain another gamma integral, hence

$$p(\mathbf{w}) = K\left(\|\mathbf{w}\|_1 + \beta\right)^{-(W+\alpha)} \tag{38}$$

where

$$K \;=\; \frac{\beta^\alpha}{2^W}\, \frac{\Gamma(W+\alpha)}{\Gamma(\alpha)}.$$

As α and β approach zero, (37) approaches the improper $1/\lambda$ ignorance prior for λ. Correspondingly, in the limit $\alpha, \beta \to 0$, we have from (38)

$$p(\mathbf{w}) \;\propto\; \left(\|\mathbf{w}\|_1\right)^{-W}$$

which is (14) with $p = 1$ and $\gamma = W$.

B.2 Gaussian prior

Now suppose that individual network weights are distributed with independent zero-mean normal densities with common variance. The prior density for the full weight vector $\mathbf{w}$ is then

$$p(\mathbf{w}|\lambda) = \left(\frac{\lambda}{2\pi}\right)^{W/2} \exp\left\{-\frac{\lambda}{2}\,\|\mathbf{w}\|_2^2\right\} \tag{39}$$

where λ^{-1} is the unknown common variance. The conjugate prior is again the gamma distribution (37), so that after substitution into (36) and integration, we have

$$p(\mathbf{w}) = K\left(\|\mathbf{w}\|_2^2 + 2\beta\right)^{-(W/2+\alpha)} \tag{40}$$

where

$$K = \frac{(2\beta)^\alpha}{\pi^{W/2}}\,\frac{\Gamma(W/2+\alpha)}{\Gamma(\alpha)}.$$

In the limit $\alpha, \beta \to 0$, we have

$$p(\mathbf{w}) \propto \left(\|\mathbf{w}\|_2\right)^{-W}$$

which is (14) with $p = 2$ and $\gamma = W$.

Multiple classes. Note that the more general prior (15) can be derived similarly, in both cases, if we suppose that there may be different unknown characteristic scales $\lambda_1, \ldots, \lambda_C$ for different groups of weights $\mathcal{W}_1, \ldots, \mathcal{W}_C$.

References

Baldi, P., and Chauvin, Y. 1996. Hybrid modeling, HMM/NN architectures, and protein applications. *Neural Computation*, *8*, 1541–1565.

Berger, J. O. 1985. *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag.

Bernardo, J. M., and Smith, A. F. M. 1994. *Bayesian Theory.* John Wiley & Sons.

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

Bishop, C. M., and Legleye, C. 1995. Estimating conditional probability densities for periodic variables. In Tesauro, G., Touretzky, D., and Leen, T., eds., *Advances in Neural Information Processing Systems 7*, pp. 641–648. The MIT Press.

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

Chiu, T. Y. M., Leonard, T., and Tsui, K.-W. 1996. The matrix-logarithmic covariance model. *Journal of the American Statistical Association*, *91*(433), 198–210.

Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition* (second edition). Academic Press.

Ghahramani, Z., and Jordan, M. I. 1994. Supervised learning from incomplete data via an EM approach. In Cowan, J. D., Tesauro, G., and Alspector, J., eds., *Advances in Neural Information Processing Systems 6*, pp. 120–127. Morgan Kaufmann.

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

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

Kendall, M. G. 1980. *Multivariate Analysis* (second edition). Charles Griffin & Co. Ltd.

Leonard, T., and Hsu, J. S. J. 1992. Bayesian inference for a covariance matrix. *Annals of Statistics*, *20*(4), 1669–1696.

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

Najfeld, I., and Havel, T. F. 1995. Derivatives of the matrix exponential and their computation. *Advances in Applied Mathematics*, *16*, 321–375.

Nix, D. A., and Weigend, A. S. 1995. Learning local error bars for nonlinear regression. In Tesauro, G., Touretzky, D. S., and Leen, T. K., eds., *Advances in Neural Information Processing Systems 7*, pp. 489–496. The MIT Press.

Pinheiro, J. C., and Bates, D. M. 1996. Unconstrained parametrizations for variance-covariance matrices. *Statistics and Computing*, *6*, 289–296.

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

Ripley, B. D. 1996. *Pattern Recognition and Neural Networks.* Cambridge University Press.

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

Williams, P. M. 1996. Using neural networks to model conditional multivariate densities. *Neural Computation*, *8*, 843–854.

Williams, P. M. 1998. Modelling seasonality and trends in daily rainfall data. In *Advances in Neural Information Processing Systems 10.* The MIT Press.