Matrix Logarithm Parametrizations for Regularized Neural Network Regression Models

Peter M Williams

School of Cognitive and Computing Sciences
Unive sity of Sussex
Falme , B ighton BN1 9QH, UK
email: pete w@cogs.susx.ac.uk

Novembe 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 with $|\rho| < 1$ then

$$\log \mathbf{\Sigma} \;=\; \frac{1}{2}\left[\begin{array}{cc} \log(1-\rho^2) & \log\dfrac{1+\rho}{1-\rho} \\ \log\dfrac{1+\rho}{1-\rho} & \log(1-\rho^2) \end{array}\right].$$

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} $(i \leq j)$ 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

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

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

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

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

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

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

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 TJi(()Tj/R212i1xd(w)999.340.0801039840Td(16-4.32984.119920-341nT.s)Tj/R22(

isthereE

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

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

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

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

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

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

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

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

$$\begin{array}{ll} \dfrac{f(\lambda_i) - f(\lambda_j)}{\lambda_i - \lambda_j} & \textit{if } \lambda_i \neq \lambda_j \\[2ex] f'(\lambda_i) & \textit{if } \lambda_i = \lambda_j. \end{array}$$

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 $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$ so that $\mathbf{A}^s = \mathbf{U}\mathbf{\Lambda}^s\mathbf{U}^{-1}$ then

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

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

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

where $\mathbf{\Phi}(n)$ 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,

$$\phi_{ij}(n) \;=\; \left\{ \begin{array}{l} \dfrac{\lambda_i^n - \lambda_j^n}{\lambda_i - \lambda_j} \quad \text{if } \lambda_i \neq \lambda_j\,{}_{i}) \; \varrho \qquad n \\ \\ \end{array} \right.$$

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 $\gamma = W$.

Multi le 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

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.