A Variational Bayesian Committee of Neural Networks

Neil D. Lawrence [1]

Computer Laboratory
New Museums Site, Pembroke Street
Cambridge, CB2 3QG, U.K.
neil.lawrence@cl.cam.ac.uk

Mehdi Azzouzi

Neural Computing Research Group
Aston University, Aston Triangle, Birmingham, B4 7ET, U.K.
azzouzim@aston.ac.uk

Abstract

Exact inference in Bayesian neural networks is non analytic to compute, approximate methods such as the evidence procedure, Monte-Carlo sampling and variational inference have been proposed. In this paper we present a general overview of the Bayesian approach, with a particular emphasis on the variational procedure. We then present a new approximating distribution based on *mixtures* of Gaussian distributions and show how it may be implemented. We present results on a simple toy problem and on two real world data sets.

1 Introduction

Learning from data consists of developing a model, $\mathcal{M}$, with a particular parameterisation $\boldsymbol{\theta}$, which best explains a set of N data points D. In the case of a regression problem, this data consists of a set of inputs $\{\mathbf{x}_1 \dots \mathbf{x}_N\} \in \mathbb{R}^I$ and a set of targets $\{\mathbf{t}_1 \dots \mathbf{t}_N\} \in \mathbb{R}^P$ which are assumed to have come from some generating function $y(\mathbf{x})$. Learning in such a model consists of determining the parameters of the model through optimisation of an information theoretic criteria such as the likelihood. This single estimate is then used to

[1] Corresponding author.

Preprint submitted to Neural Networks *17 September 1999*

make predictions. Unfortunately, when the number of data points, N, is 'small' the parameters are poorly estimated by this approach. A complex model will typically better fit the training data without being able to give good predictions on unseen data, i. e. it will exhibit poor generalisation capabilities. This phenomenon is known as *over-fitting* and we must look to techniques such as regularisation in order to solve this problem. Such techniques often use an extra partition of the data known as a validation set in order to evaluate the generalisation capabilities of the model during the training phase.

In this work, we follow an alternative approach known as Bayesian inference. In the Bayesian approach we consider the parameters vector $\boldsymbol{\theta}$ of the model to be random variables. We seek to determine the form of the conditional distribution of $\boldsymbol{\theta}$ given the training data, D, and the structure of our model, $\mathcal{M}$. We must therefore define a *prior* distribution $p(\boldsymbol{\theta})$, and the required conditional or *posterior* distribution, $p(\boldsymbol{\theta}|D, \mathcal{M})$, may then be determined via Bayes' theorem:

$$p(\boldsymbol{\theta}|D, \mathcal{M}) = \frac{p(D|\boldsymbol{\theta}, \mathcal{M})p(\boldsymbol{\theta})}{p(D|\mathcal{M})}, \tag{1}$$

where $p(D|\boldsymbol{\theta}, \mathcal{M})$ is the likelihood of the data given the parameter vector as defined by our model. Predictions for unseen data can then be made by considering expectations under the posterior posterior distribution.

Regularisation can be given a natural interpretation within the Bayesian framework and regularisation coefficients may be determined without the need to resort to a validation set.

Unfortunately in many cases the required integrals are non-analytic, and we therefore must look to approximations to make progress. The multi-layer perceptron provides one example of a model for which exact Bayesian solutions are not possible. In response approximate methods such as the Laplace approximation (MacKay, 1992; Gull, 1988), and Monte-Carlo sampling (Neal, 1996) have been successfully applied for both regression and classification problems.

A further approach known as *ensemble learning* was introduced in the context of the multi-layer perceptron by Hinton and van Camp (1993). It involves the selection of a class of distributions which may represent good approximations to the posterior, yet are also simple enough to allow necessary integrals to be performed. The best approximation within that class is then selected through minimisation of the Kullback-Leibler (KL) divergence between the true posterior and the approximating distribution. Hinton and van Camp used a Gaussian distribution with a diagonal covariance matrix for the approximation, Barber and Bishop (1998) extended their approach and demonstrated how a Gaussian with a full covariance matrix could be utilised. In this work, we review the ensemble learning approach and demonstrate its implementation with an approximation based on *mixtures* of Gaussian distributions.

First we briefly review the different approaches to Bayesian learning in neural networks (section 2), and discuss in more depth ensemble learning from a variational perspective (section 3). We derive in general the rigorous lower bound on the marginal likelihood $p(D|\mathcal{M})$ and then demonstrate the implementation of this bound with the mixture of Gaussians as an approximating distribution. Finally in section 4 we apply the approach to both synthetic and real-world data and compare the performance of our approximation with other well established methods.

2 Bayesian learning of neural networks

We define our model to be a two-layer feed-forward neural network with I input nodes, H hidden nodes and a single output node. The network function may be written as:

$$f(\mathbf{x}, \mathbf{w}) = \sum_{h=1}^{H} v_h g(\mathbf{u}_h^{\mathrm{T}} \mathbf{x}), \tag{2}$$

where $\mathbf{w} = \{\mathbf{u}_1 \ldots \mathbf{u}_H, \mathbf{v}\}$ is a vector representing the parameters or 'weights' of the network. Where the input to hidden weights are represented by H vectors $\mathbf{u}_h$, each vector being the weights that 'fan-in' to hidden unit h. $\mathbf{v}$ is the vector of the hidden to output weights, consisting of H elements v_h. We account for hidden layer 'biases' by considering additional input and hidden nodes whose values are taken to be one at all times. The activation function g is often taken to be a hyperbolic tangent, for reasons of tractability though we follow Barber and Bishop (1998) in our use of the cumulative Gaussian distribution function:

$$g(x) = \sqrt{\frac{2}{\pi}} \int_0^x \exp(-t^2)\ dt \tag{3}$$

which has a similar form to the hyperbolic tangent.

Given our dataset $D = \{\mathbf{x}_n, t_n\}_{n=1}^{N}$ our task is to interpolate through the data, we model the data as being derived from a underlying true function $y(\mathbf{x})$ with Gaussian noise added. This leads us to consider a likelihood function of the form

$$p(D|\mathbf{w}, \beta) = \left(\frac{\beta}{2\pi}\right)^{N/2} \exp(-\beta E_D(D, \mathbf{w})), \tag{4}$$

where β is a hyper-parameter governing the inverse noise variance and $E_D(D, \mathbf{w})$

represents the error on the data set:

$$E_D(D,\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(t_n - f(\mathbf{x}_n,\mathbf{w}))^2. \tag{5}$$

We define a prior distribution $p(\mathbf{w}|\alpha)$ which is taken to be a zero mean Gaussian distribution:

$$p(\mathbf{w}|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{K/2}\exp(-\alpha E_w(\mathbf{w})), \tag{6}$$

where α is a hyper parameter governing the prior, K is the number of weights in the model and $E_w(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$. This choice of prior corresponds to a belief that positive and negative weights are equally probable and that small weights are more probable than larger ones, leading to smoother interpolations. More complex priors are possible in which the weights grouped according to their connectivity. Hyper-parameters are then associated with each separate grouping. This allows the implementation of techniques such as MacKay and Neal's automatic relevance determination (see Neal (1996)) which allows us to identify irrelevant inputs to the network and remove them. For simplicity, we constrain our investigations to priors including only a single hyper-parameter α, although extending the derivations for automatic relevance determination is straightforward.

The second level of inference will be concerned the hyper-parameters α and β. A full Bayesian treatment at this level makes use of hyper-priors. As they both represent inverse variances of Gaussian distributions a natural choice for both hyper-priors is the Gamma distribution:

$$p(\tau|a,b) = \frac{b^a}{\Gamma(a)}\tau^{a-1}\exp\left(-b\tau\right) \tag{7}$$

where a and b are constants, and $\Gamma(\dot{})$ is the Gamma function. The ratio a/b is the expectation of τ under this distribution and the value of a/b^2 is the variance of τ under the distribution.

The first level of inference deals with the posterior distribution

$$\begin{aligned} p(\mathbf{w}|D,\alpha,\beta) &= \frac{p(D|\mathbf{w},\beta)p(\mathbf{w}|\alpha)}{p(D|\alpha,\beta)} && (8)\\ &= \frac{1}{Z(\alpha,\beta)}\exp(-\alpha E_w(\mathbf{w}) - \beta E_D(D,\mathbf{w})) && (9)\end{aligned}$$

where we have dropped the conditioning on the model, $\mathcal{M}$ to avoid cluttering the notation and $Z(\alpha,\beta) = \int\exp(-\alpha E_w(\mathbf{w}) - \beta E_D(D,\mathbf{w}))d\mathbf{w}$ is a normalisation constant.

The second level of inference concerns the *hyper-posterior* distribution

$$p(\alpha, \beta|D) = \frac{p(D|\alpha, \beta)p(\alpha)p(\beta)}{p(D)}. \tag{10}$$

The term $p(D|\alpha, \beta)$ is often called the *evidence* for α and β. This is not to be confused with the model evidence which refers to $p(D|\mathcal{M})$. We avoid this confusion by referring to $p(D|\alpha, \beta)$ as the *hyper-likelihood* and the model evidence as the *model likelihood*. This term is the denominator of eqn. 8 and can be evaluated by integration over $\mathbf{w}$:

$$p(D|\alpha, \beta) = \int p(D|\mathbf{w}, \beta)p(\mathbf{w}|\alpha)\ d\mathbf{w}. \tag{11}$$

Finally network predictions on unseen data are obtained by looking at the expected value of the output under the posterior distribution

$$\langle f(\mathbf{x}, \mathbf{w})\rangle_p(\mathbf{w}|D) = \int f(\mathbf{x}, \mathbf{w})p(\mathbf{w}|D)\ d\mathbf{w} \tag{12}$$

for which we require the posterior distribution of the weights given the data

$$p(\mathbf{w}|D) = \int p(\mathbf{w}|D, \alpha, \beta)p(\alpha, \beta|D)\ d\alpha\ d\beta. \tag{13}$$

The variance of predications, $\langle f(\mathbf{x}, \mathbf{w})^2\rangle - \langle f(\mathbf{x}, \mathbf{w})\rangle^2$ may also be obtained and used to place error-bars on the predictions.

The Bayesian framework provides a principled way in which to handle generalisation and confidence intervals on predictions. Unfortunately exact Bayesian learning requires the computation of expectations under th posterior distribution $p(\mathbf{w}|D)$ which involves a high dimensional non-linear integral.

2.1 Markov chain Monte-Carlo sampling

Sampling approximations consist of approximating an integral $I = \int f(\mathbf{w})p(\mathbf{w}|D)d\mathbf{w}$ with a finite sum $I \approx \frac{1}{S}\sum_{s=1}^{S} f(\mathbf{w}_n)$ where $\{\mathbf{w}_s\}_{s=1}^{S}$ are samples from the posterior distribution $p(\mathbf{w}|D)$. The approximation becomes exact in the limit $S \to \infty$. The major difficulty relies on finding a representative set of samples. Markov chain Monte-Carlo algorithm creates a Markov chain which converges to the desired distribution $p(\mathbf{w}|D)$.

Neal (1996) applied hybrid Monte-Carlo sampling techniques to obtain samples from the posterior distribution for Bayesian neural networks. Although these samples are generally more representative than functional approximation of the posterior distribution, issues remain such as convergence of the sampler, and the lack of an approximation to the model likelihood.

Practical details of the implementation of this technique for regression and classification can be found respectively in Rasmussen (1996) and Husmeier *et al.* (1998).

2.2 The Laplace approximation

The evidence procedure (MacKay, 1992) involves making the Laplace approximation to the posterior and makes use of type II maximum likelihood (Gull, 1988) for determining the hyper-parameters.

The Laplace approximation involves finding a single local optimal point $\mathbf{w}^*$ in weight space and constructing a full covariance Gaussian approximation to the posterior distribution around this point.

$$p(\mathbf{w}|D,\alpha,\beta) \approx \frac{|\mathbf{H}|^{\frac{1}{2}}}{(2\pi)^{\frac{W}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{w}-\mathbf{w}^*)^{\mathrm{T}}\mathbf{H}(\mathbf{w}-\mathbf{w}^*)\right\}, \tag{14}$$

where the matrix $\mathbf{H}$ is the Hessian about point $\mathbf{w}^*$,

$$\mathbf{H} = \nabla\nabla(\beta E_D(D,\mathbf{w}) + \alpha E_w(\mathbf{w})). \tag{15}$$

The treatment of the hyper-parameters involves also an approximation of the posterior probability distribution $p(\alpha,\beta|D)$. It is assumed that the distribution is sharply peaked around the most probable values α^* and β^*. It is then possible to get re-estimation formulae for the hyper-parameters as a function of the eigenvalues of the Hessian matrix $\mathbf{H}$. A review of the approach can be found in MacKay (1995).

Thus, the evidence procedure is a two-step procedure. First, the maximum a posteriori value for the weights is obtained by maximising the penalised likelihood. This local maximum is then used for estimating the hyper-likelihood (eqn. 11). Re-estimation equations for α and β are obtained by maximising the hyper-likelihood. The whole process is repeated until convergence. Practical details of the implementation and some results for a regression problem can be found in Thodberg (1996).

While the Laplace approximation provides a quick and easy method of approximating the posterior distribution, the approximation is only locally valid, it is concentrated at a mode of the posterior. Unfortunately this mode may not be representative of the mass of the distribution, even if the posterior is unimodal. In the context of neural networks, the posterior distributions are known to be highly multi-modal. We therefore look for an alternative functional approximation which is more responsive to the true mass of the distribution.

3 Ensemble Learning

Ensemble learning was first applied in the context of neural networks by Hinton and van Camp (1993), where in was introduced through the implementation of the minimum description length principle. The approximation to the posterior in this approach is more responsive to the mass of the distribution than that of the evidence procedure and has the additional advantage of providing a rigorous lower bound on the model likelihood.

3.1 Variational techniques

Ensemble learning can be viewed as a special case of the general framework of variational techniques for inference in probabilistic models (Jordan *et al.*, 1998). Consider the model likelihood for a Bayesian neural network

$$p(D|\mathcal{M}) = \int p(D, \mathbf{w}, \alpha, \beta|\mathcal{M}) d\mathbf{w}\ d\alpha\ d\beta \tag{16}$$

$$= \int p(D|\mathbf{w}, \beta) p(\mathbf{w}|\alpha) p(\alpha) p(\beta)\ d\mathbf{w}\ d\alpha\ d\beta. \tag{17}$$

This likelihood may be bounded from below through the introduction of a variational distribution $q(\mathbf{w}, \alpha, \beta)$,

$$\log p(D) \geq \int q(\mathbf{w}, \alpha, \beta) \log \frac{p(D, \mathbf{w}, \alpha, \beta)}{q(\mathbf{w}, \alpha, \beta)}\ d\mathbf{w}\ d\alpha\ d\beta. \tag{18}$$

The difference between this bound and the true likelihood can easily be shown to be the Kullback-Leibler (KL) divergence between the approximating distribution $q(\mathbf{w}, \alpha, \beta)$ and the true posterior $p(\mathbf{w}, \alpha, \beta|D)$

$$\mathrm{KL}(q||p) = \int q(\mathbf{w}, \alpha, \beta) \log \frac{q(\mathbf{w}, \alpha, \beta)}{p(\mathbf{w}, \alpha, \beta|D)} d\mathbf{w}\ d\alpha\ d\beta. \tag{19}$$

The KL-divergence is always positive unless $q(\mathbf{w}, \alpha, \beta)$ is identical to $p(\mathbf{w}, \alpha, \beta|D)$ when it is zero. The bound 18 is maximised by the minimum KL-divergence between the distributions. However as it stands the bound requires further treatment. We follow MacKay (1995) and assume

$$q_{\mathbf{w}}(\mathbf{w}, \alpha, \beta) = q_{\mathbf{w}}(\mathbf{w}) q_{\alpha}(\alpha) q_{\beta}(\beta), \tag{20}$$

which corresponds to a factorised approximating distribution. We may now rewrite the bound

$$\log p(D) \geq \int q_{\mathbf{w}}(\mathbf{w}) q_{\alpha}(\alpha) q_{\beta}(\beta) \log p(D, \mathbf{w}, \alpha, \beta)\ d\mathbf{w}\ d\alpha\ d\beta$$
$$+\mathcal{H}(q_{\mathbf{w}}(\mathbf{w})) + \mathcal{H}(q_{\beta}(\beta)) + \mathcal{H}(q_{\alpha}(\alpha)), \tag{21}$$

where $\mathcal{H}(p(x))$ denotes the entropy of the distribution $p(x)$.

We now turn to the choice of the approximating distribution $q_{\mathbf{w}}(\mathbf{w})$, the other distributions will be dealt with in section 3.3.4.

3.2 Single Gaussian variational distribution

Hinton and van Camp (1993) selected a diagonal covariance Gaussian for the approximating distribution $q_{\mathbf{w}}$. The advantage of this choice is the simplicity of its implementation. In particular, if the activation function g of the network is chosen to be the cumulative Gaussian all but one of the required integrals are analytically tractable. In section B we show how the remaining integral may be approximated efficiently. However, such a simple representation can not account for correlations between network weights present in the posterior.

Barber and Bishop (1998) used a full covariance Gaussian in their implementation. This approach requires the use of a three dimensional look up table, or a one dimensional numerical integration. For such an approximation, some integrations are not tractable and the use of lookup tables or numerical integrals is required. The efficient implementation of this approach remains an unresolved issue.

3.3 Mixture of diagonal covariance Gaussian

In order to effectively model a multi-modal posterior distribution, we propose to implement a mixture distribution:

$$q_{\mathbf{w}}(\mathbf{w}) = \sum_{m=1}^{M} Q_{\mathbf{w}}(m) q_{\mathbf{w}}(\mathbf{w}|m), \tag{22}$$

where each component $q_{\mathbf{w}}(\mathbf{w}|m) = \mathcal{N}(\bar{\mathbf{w}}_m, \boldsymbol{\Sigma}_{q_m})$ is a diagonal covariance Gaussian. This representation is able to capture correlations between the weights and would also be able to represent skewness or kurtosis in the posterior. In practice however, strong correlations between weights may require many components for a good approximation to the posterior distribution and it may not be practical to optimise such large numbers of components. In particular, we might expect to under-perform an approach based on full covariance Gaussians when these correlations are present in the posterior distribution. It is possible to extend our approach for mixtures of full covariance Gaussians, however we consider that the resulting algorithm would be too cumbersome for practical application.

3.3.1 Lower bound on the entropy

The use of the mixture distribution does not complicate the calculation of the first term in bound 21 beyond what has been tackled in previous works. However, the entropy of the mixture distribution presents other problems, namely we are required to evaluate the expectation of the logarithm of a sum. We make progress by considering the mutual information, $I(m;\mathbf{w})$, between the component label m of the mixture distribution and an observed weight vector. The entropy may be rewritten as:

$$\begin{aligned}\mathcal{H}(q_{\mathbf{w}}(\mathbf{w})) &= -\sum_{m=1}^{M} Q_{\mathbf{w}}(m)\int q_{\mathbf{w}}(\mathbf{w}|m)\log\left[\frac{q_{\mathbf{w}}(\mathbf{w})}{q_{\mathbf{w}}(\mathbf{w}|m)}q_{\mathbf{w}}(\mathbf{w}|m)\right]d\mathbf{w}\\ &= \sum_{m=1}^{M} Q_{\mathbf{w}}(m)\mathcal{H}(q_{\mathbf{w}}(\mathbf{w}|m)) + I(m;\mathbf{w}). \qquad (23)\end{aligned}$$

The first term in eqn. 23 is a sum of entropies and is analytically tractable for Gaussian distributions [2] . The second term, the mutual information, takes the form

$$I(m;\mathbf{w}) = \sum_{m=1}^{M} Q_{\mathbf{w}}(m)\int q_{\mathbf{w}}(\mathbf{w}|m)\log\frac{q_{\mathbf{w}}(\mathbf{w}|m)}{q_{\mathbf{w}}(\mathbf{w})}\,d\mathbf{w}. \qquad (24)$$

The mutual information is always positive and therefore its inclusion over a naive committee [3] of networks improves the bound. We are still left with a logarithm of a sum in this second term and to make progress, we look to a further lower bound, first introduced in the context of discrete systems by Jaakkola and Jordan (1998) and applied to probabilistic graphical models by Bishop *et al.* (1998) and Lawrence *et al.* (1998) (see section B for the derivation of this bound).

[2] the entropy of each component is simply $\mathcal{H}(q_{\mathbf{w}}(\mathbf{w}|m)) = \frac{1}{2}\log|\mathbf{\Sigma}_{q_m}| + \frac{K}{2}(1+\log 2\pi)$

[3] We consider a naive committee of networks to be on which only considers a weighted sum of the model-likelihood of each committee member to form a overall model-likelihood.

$$
\begin{aligned}
I(m;\mathbf{w}) \geq & \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) \log r(\mathbf{w}|m)\ d\mathbf{w} \\
& - \sum_{m,m'}^{M} Q_{\mathbf{w}}(m)\lambda_{m'} \int r(\mathbf{w}|m') q_{\mathbf{w}}(\mathbf{w}|m)\ d\mathbf{w} \\
& - \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \log Q_{\mathbf{w}}(m) + \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \log \lambda_m + 1, \qquad (25)
\end{aligned}
$$

where we have introduced the variational distributions $r(\mathbf{w}|m)$ and the variational parameters λ_m for tightening the bound. The selection of the smoothing distributions $r(\mathbf{w}|m)$ is important. Indeed, if we were to take $r(\mathbf{w}|m) \propto q_{\mathbf{w}}(\mathbf{w}|m)/q_{\mathbf{w}}(\mathbf{w})$ and maximise over the variational parameters λ_m, we would recover the mutual information exactly. Such a choice however would lead to no reduction in complexity of the calculations, and we opt instead to use a Gaussian form, $r(\mathbf{w}|m) \propto \mathcal{N}(\bar{\mathbf{r}}_{r_m}, \mathbf{\Sigma}_{r_m})$ leading to the following bound

$$
\begin{aligned}
I(m;\mathbf{w}) \geq & -\frac{1}{2} \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \left\{ \mathrm{tr}(\mathbf{\Sigma}_{q_m} \mathbf{\Sigma}_{r_m}^{-1}) + (\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_{r_m})^{\mathrm{T}} \mathbf{\Sigma}_{r_m}^{-1} (\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_{r_m}) \right\} \\
& - \sum_{m,n}^{M} \frac{Q_{\mathbf{w}}(m)\lambda_{m'}}{|\mathbf{I} + \mathbf{\Sigma}_{q_m} \mathbf{\Sigma}_{r_{m'}}^{-1}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} (\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_{m'})(\mathbf{\Sigma}_{q_m} + \mathbf{\Sigma}_{r_{m'}})^{-1} (\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_{m'}) \right\} \\
& - \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \log Q_{\mathbf{w}}(m) + \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \log \lambda_m + 1 \qquad (26) \\
\equiv & \mathcal{I}(m;\mathbf{w}). \qquad (27)
\end{aligned}
$$

The smoothing distributions $r(\mathbf{w}|m')$ control the tightness bound. Optimisation with respect to the parameters of these distributions increases the bound. In our experiments, we used diagonal covariance matrices for r although the extension for full covariance is straightforward.

Note that the maximum possible value for the mutual information is $\log M$ (see section B). This upper bound corresponds to the case where there is no overlap between components. Therefore we may only improve our bound by a maximum of $\log 2$ by moving from one component to two.

3.3.2 Lower bound on the log-likelihood

We now have a rigorous lower bound on the model log-likelihood:

$$
\begin{aligned}
\log p(D) \geq & \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \langle \log p(D|\mathbf{w},\beta) \rangle_{q_m,S} + \sum_{m}^{M} Q_{\mathbf{w}}(m) \langle \log p(\mathbf{w}|\alpha) \rangle_{q_m,q_\alpha} \\
& + \langle \log p(\alpha) \rangle_{q_\alpha} + \langle \log p(\beta) \rangle_{q_\beta} \\
& + \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \mathcal{H}(q_{\mathbf{w}}(\mathbf{w}|m)) + \mathcal{I}(m;\mathbf{w}) + \mathcal{H}(q_\alpha(\alpha)) + \mathcal{H}(q_\beta(\beta)) \quad (28) \\
\equiv & \mathcal{L}(\mathbf{w},\alpha,\beta). \quad (29)
\end{aligned}
$$

For diagonal covariance Gaussian distributions and Gamma hyper-priors, all these expectations but one are analytically tractable. The intractable expectation concerns the square of the output function $\langle f(\mathbf{x},\mathbf{w})^2 \rangle_{q_{\mathbf{w}}}$ for which an approximation is required. For clarity, we relegate the mathematical derivations of these integrals to the appendix.

3.3.3 Optimising the approximation to the posterior

We now write the lower bound on the marginal log-likelihood using $p(\mathbf{w},\alpha,\beta|D) \propto p(D|\mathbf{w},\beta)p(\mathbf{w}|\alpha)p(\alpha)p(\beta)$,

$$
\begin{aligned}
\mathcal{L}(\mathbf{w},\alpha,\beta) = & - \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) q_\beta(\beta) \left\{ \frac{-\beta}{2} E_D(D,\mathbf{w}) + \frac{N}{2} \log \beta + \log p(\beta) \right\} dw \ d\beta \\
& - \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) q_\alpha(\alpha) \left\{ \frac{-\alpha}{2} E_w(\mathbf{w}) + \frac{K}{2} \log \alpha + \log p(\alpha) \right\} d\mathbf{w} \ d\alpha \\
& - \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \mathcal{H}(q_{\mathbf{w}}(\mathbf{w}|m)) - I(m;\mathbf{w}) - \mathcal{H}(q_\alpha(\alpha)) - \mathcal{H}(q_\beta(\beta)) + \text{const} \quad (30)
\end{aligned}
$$

where K is the total number of weight parameters, in the model. Consider first the dependence of $\mathcal{L}(\mathbf{w},\alpha,\beta)$ on $\mathbf{w}$

$$
\begin{aligned}
\mathcal{L}(\mathbf{w}) = & - \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) \left\{ -\bar{\beta} E_D(D,\mathbf{w}) - \frac{\bar{\alpha}}{2} \mathbf{w}^{\mathrm{T}} \mathbf{w} \right\} d\mathbf{w} \\
& - \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \mathcal{H}(q_{\mathbf{w}}(\mathbf{w}|m)) - I(m;\mathbf{w}) + \text{const}, \quad (31)
\end{aligned}
$$

where we have introduce the expectations $\bar{\alpha} = \int \alpha q_\alpha(\alpha) d\alpha$ and $\bar{\beta} = \int \beta q_\beta(\beta) d\beta$. Evaluating the derivatives of the above expression with respect to the parameters of the variational distributions can be undertaken analytically (section C). Together with the model likelihood bound, $\mathcal{L}(\mathbf{w})$, these derivatives are used in a non-linear optimisation algorithm.

3.3.4 Optimising the approximations to the hyper-posterior

So far, the distributions q_α and q_β have not been specified. Yet we need at minimum to estimate the moments $\bar{\alpha}$ and $\bar{\beta}$ in order to determine $q_{\mathbf{w}}(\mathbf{w})$. A free form optimisation over all possible forms of q_α leads us to the following result

$$q_\alpha = \frac{\exp\langle \ln p(D|\mathbf{w},\beta)p(\mathbf{w}|\alpha)p(\alpha)p(\beta)\rangle_{q_{\mathbf{w}}q_\beta}}{\langle \exp\langle \ln p(D|\mathbf{w},\beta)p(\mathbf{w}|\alpha)p(\alpha)p(\beta)\rangle_{q_{\mathbf{w}}q_\beta}\rangle_{q_\alpha}} \tag{32}$$

Substituting the appropriate form of $q_{\mathbf{w}}$ we find

$$q_\alpha = \frac{\bar{b}_\alpha^{\bar{a}_\alpha}}{\Gamma(\bar{a}_\alpha)}\tau^{\bar{a}_\alpha - 1}\exp\left(-\bar{b}_\alpha \alpha\right) \tag{33}$$

where

$$\bar{a}_\alpha = \frac{K}{2} + a_\alpha, \tag{34}$$

$$\bar{b}_\alpha = \frac{1}{2}\sum_m \left\{\mathrm{tr}(\mathbf{\Sigma}_{q_m}) + \bar{\mathbf{w}}_m^{\mathrm{T}}\bar{\mathbf{w}}_{q_m}\right\} + b_\alpha \tag{35}$$

A similar treatment for β reveals

$$q_\beta = \frac{\bar{b}_\beta^{\bar{a}_\beta}}{\Gamma(\bar{a}_\beta)}\tau^{\bar{a}_\beta - 1}\exp\left(-\bar{b}_\beta \beta\right) \tag{36}$$

where

$$\bar{a}_\beta = \frac{N}{2} + a_\beta, \tag{37}$$

$$\bar{b}_\beta = \sum_m \langle E_D(D,\mathbf{w})\rangle_{q_{\mathbf{w}}(\mathbf{w}|m)} + b_\beta. \tag{38}$$

Note that the last term of eqn. 38 has already been computed for the optimisation of $q_{\mathbf{w}}$.

The ensemble learning treatment of Bayesian neural networks can be undertaken as a two stage optimisation. First, we minimise the KL divergence with respect to the approximating distribution $q_{\mathbf{w}}(\mathbf{w})$, secondly we re-estimate the hyper-posteriors. The whole process is re-iterated until convergence of the model likelihood bound.

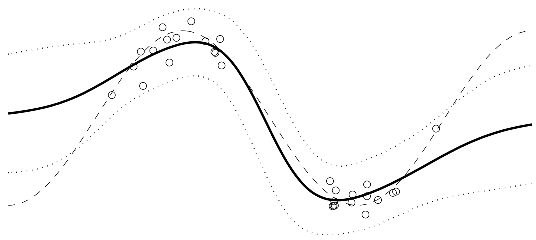

Fig. 1. Plot of the sine wave from which data was sampled (dashed line), the sample used for training (circles), the expected output of the network (solid line) and error bars representing two standard deviations (dotted line).

4 Experimental results

In this section, we present our approach on both synthetic and real-world data. We also compare the performance of the approximation with hybrid Monte-Carlo sampling and the evidence procedure. All the models are implemented with a single hyper-parameter α governing the weights of the network. The components of the mixture distribution for the variational approach are taken to be diagonal covariance Gaussians as are the variational smoothing distributions.

We chose not to optimise the mixture coefficients, $Q_{\mathbf{w}}(m)$, which are taken to be fixed during learning and equal to $\frac{1}{M}$. This is in the hope that the network will be forced to seek multi-modal approximations to the posterior, and will be prevented from simply fine tuning the shape of a unimodal approximation.

For all the Bayesian techniques, we chose broad hyper-priors on α and β by setting $a_\alpha = 3 \times 10^{-4}$, $b_\alpha = 1 \times 10^{-4}$ and $a_\beta = 0.05$, $b_\beta = 1 \times 10^{-4}$.

4.1 Synthetic data

As a simple illustration of our approach, we first present the algorithm applied to a toy problem. Thirty data points were sampled from the function $0.4 \sin(2\pi x)$ and noise of standard deviation 0.05 was added. A mixture of five diagonal covariance Gaussian containing three hidden nodes was trained on the data. Figure 1 shows the results obtained and error bars at two standard deviations.

4.2 Sunspot data

Our first real world data set involves the annual average[4] of sunspots from 1700 to 1920. The time series shows a strong seasonality with a time varying amplitude and has served as a benchmark in the statistical literature (Tong, 1995). The average time between maxima is 11 years and to date no principled theory exits for describing this behaviour, although it is known that sunspots are related to other activities of the sun.

We propose to model this time series within a Bayesian framework as the number of training points is relatively small. We compare results from three different Bayesian approaches and that of a standard neural networks trained by 'early stopping'. The goal of the simulations is to compare Bayesian techniques on a time series dataset. To facilitate comparison, the number of hidden nodes is taken to be fixed at eight and the input window is chosen arbitrarily to be 12, i. e. we modelled $x_n = f(x_{n-1}, \ldots x_{n-12})$. The yearly sunspot data from 1700 through 1920 (221 data points) was used for training and the data from 1921 to 1979 (59 data points) for evaluating the generalisation performance. These choices correspond to those of Weigend *et al.* (1990) who studied the performance of early stopping techniques on this time series. The dataset was normalised to have a zero mean and unit variance.

We first trained a standard neural network with the early stopping technique. The validation set contained 22 points chosen randomly from the training set. The training phase was stopped when the error on the validation reached a minimum. Ten different neural networks were trained and the best network is presented below.

For the hybrid Monte-Carlo sampling, 300 samples[5] of the posterior distribution were taken. Predictions were made by averaging the network samples from the posterior distribution, ignoring the first 100 samples.

For the evidence procedure, we trained ten models with different random initialisations. We then selected the best network according to the hyper-likelihood of the training data. Also, in order to have a fair comparison with the ensemble learning approach, we created a committee of three evidence networks. We selected the three networks with the best hyper-likelihood which were then used in an unweighted committee. The models were trained by 500 conjugate gradient iterations in weight space, followed by an update of the hyper-parameters. This cycle was completed five times.

[4] The data are daily, monthly and annually reported by the Royal Observatory of Belgium and can be found at `http://www.oma.be/KSB-ORB/SIDC/sidc_txt.html`.
[5] Runs with many more samples (up to 10,000) were also undertaken, but lead to no significant improvement in performance

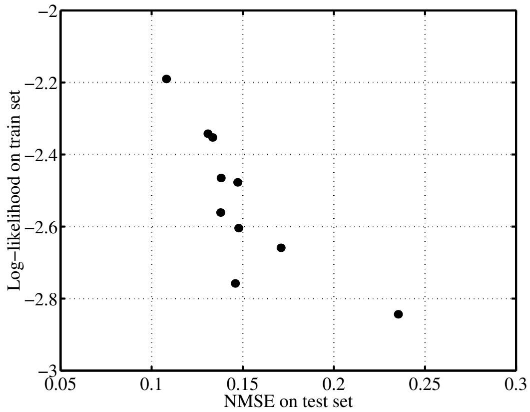

Fig. 2. Test error versus lower bound on the log-likelihood on the training set for the sunspot time series.

For the ensemble learning we took a similar approach. We trained both a one component model and a three component mixture model with ten different initialisations. For the mixture of three components. Each component was first independently initialised by performing 100 iterations of a quasi-newton optimiser, the components were then trained together as a mixture. As for the evidence procedure the parameters of the Gaussians where optimised over five hundred iterations followed by the single step optimisation of the hyper-posteriors. These two steps where again completed five times. We selected the best network from the initialisations according to the lower bound on the training set likelihood obtained for each network.

Figure 2 shows the lower bound on the training set log-likelihood versus the normalised mean square error on the test set for ten different models. Note the strong correlation between the bound on the model likelihood and the generalisation capabilities of the neural networks. Table 1 reports the NMSE obtained by each technique on the test set. The first result was reported by Weigend *et al.* (1990).

While the data set is too small to draw absolute conclusions, the results seem to indicate the utility of the Bayesian approach in general and show the competitiveness of the methods based on variational techniques.

committee with the one standard deviation error bars.

4.3 Tecator data

We also assessed the performance of our approach on a regression problem, the Tecator dataset (Thodberg, 1996). The data are recorded on a Tecator Infratec Food and Feed Analyser working in the wavelength range 850 - 1050 nm by the Near Infrared Transmission (NIT) principle. Each sample contains

Method	Test error
Early stopping	0.22
Evidence	0.18
Evidence committee	0.14
Variational (diagonal)	0.14
Variational committee	0.11
Hybrid Monte-Carlo	0.09

Table 1
Normalised mean squared error on the test set for the five techniques. For the evidence and ensemble learning techniques, we selected the best network according to the likelihood on the training set. The evidence committee contains three networks and the variational committee is composed of three components.

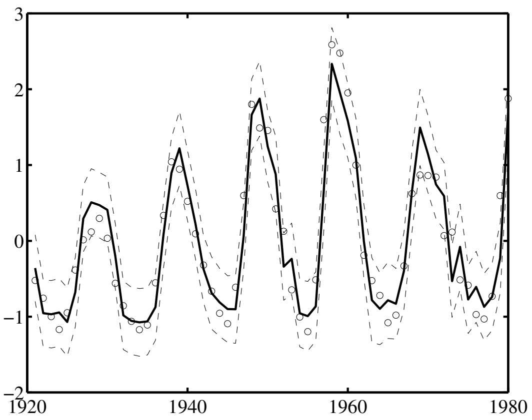

Fig. 3. One-step ahead prediction (solid line) with the one standard error bars (dashed line) obtained with an ensemble learning committed of three component. Circles represent the targets.

finely chopped pure meat with different moisture, fat and protein contents. The task is to predict the fat content of a meat sample on the basis of its near infrared absorbance spectrum. For each meat sample, the data consists of a 100 channel spectrum. the spectra are pre-processed using principal component analysis and the first ten principal components are used as inputs to a neural network. Thodberg (1996) demonstrated the benefits of the evidence approach compared to the early stopping technique.

The training dataset contains 172 samples and the test set is composed of 43 samples [6] . In his paper, Thodberg (1996) reported the square root of the mean square error and in order to have comparable results we used the same error measure. Note however that in this previous work, the neural networks had a direct connection for the inputs to the output. Our models do not contain

[6] The dataset is available from `http://temper.stat.cmu.edu/datasets/tecator`

Method	Test error
Linear regression	2.78
Quadratic regression	0.80
Early stopping	0.65
Evidence	0.60
Hybrid Monte-Carlo	0.59
Evidence committee (3)	0.57
Evidence committee (10)	0.57
Variational (diagonal)	0.56
Variational committee	0.54

Table 2
Square root of the mean squared error on the test set for the five techniques. For the evidence and ensemble learning techniques, again, we selected the best network according to the likelihood on the training set. The two evidence committees contains ten and three networks as labelled and the variational committee was composed of three components.

such connections. For the number of hidden units, we followed also (Thodberg, 1996) and chose to train three hidden units neural networks. The ensemble mixture is once again composed of three components. For the simulations, we used the same settings for the hyper-parameters as previously and the same approach for model selection.

Table tab:tecator reports the test error obtained by each technique. The first two results were reported by Thodberg (1996) Once again, the Bayesian approaches outperform ‘early stopping’ technique. The variational committee once again performs well.

5 Discussion

5.1 Conclusions

We have reviewed the Bayesian framework for regression neural networks with a particular focus on variational approaches. We introduced a new variational distribution based on mixtures of diagonal covariance Gaussians. This approximation enables the modelling of non-Gaussian posterior distributions. The results on real world data set were promising.

This framework is also applicable to classification networks, although this requires the use of further variational bounds. Mixtures of full covariance Gaussians are also possible within our framework, and would provide ground for further study.

Acknowledgements

We are indebted to Radford Neal for his implementation of hybrid Monte Carlo sampling (Neal, 1999) and Ian Nabney for the NETLAB package (Nabney, 1998). We thank Christopher Bishop for helpful discussions.

References

Barber, D. and C. M. Bishop (1998). Ensemble learning in Bayesian neural networks. In C. M. Bishop (Ed.), *Neural Networks and Machine Learning*, Volume 168 of *Series F: Computer and Systems Sciences*, pp. 215–237. Cambridge, U.K., NATO Advanced Study Institute: Springer-Verlag.

Bishop, C. M., N. D. Lawrence, T. S. Jaakkola, and M. I. Jordan (1998). Approximating posterior distributions in belief networks using mixtures. In M. I. Jordan, M. J. Kearns, and S. A. Solla (Eds.), *Advances in Neural Information Processing Systems*, Volume 10, pp. 416–422. Cambridge, MA: MIT Press.

Gull, S. F. (1988). Bayesian inductive inference and maximum entropy. In G. J. Erickson and C. R. Smith (Eds.), *Maximum-Entropy and Bayesian Methods in Science and Engineering*, Volume 1: Foundations, pp. 53–74. Dordrecht, The Netherlands: Kluwer.

Hinton, G. E. and D. van Camp (1993). Keeping neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Anuual Conference on Computational Learning Theory*, pp. 5–13.

Husmeier, D., W. D. Penny, and S. J. Roberts (1998). Empirical evaluation of Bayesian sampling for neural classifiers. In L. Niklason, M. Boden, and T. Ziemke (Eds.), *Proceedings of the 8th International Conference on Artificial Neural Networks*, Perspectives in Neural Computing, pp. 323–328. Springer Verlag.

Jaakkola, T. S. and M. I. Jordan (1998). Approximating posteriors via mixture models. See Jordan (1998), pp. 163–174.

Jordan, M. I. (Ed.) (1998). *Learning in Graphical Models*, Volume 89 of

Series D: Behavioural and Social Sciences, Erice, Italy. NATO Advanced Study Institute: Kluwer.

Jordan, M. I., Z. Gharamani, T. S. Jaakkola, and L. K. Saul (1998). An introduction to variational methods for graphical models. See Jordan (1998), pp. 105–162.

Lawrence, N. D., C. M. Bishop, and M. I. Jordan (1998). Mixture representations for inference and learning in Boltzmann machines. In G. F. Cooper and S. Moral (Eds.), *Uncertainty in Artificial Intelligence*, Volume 14, pp. 320–327. San Fransisco, CA: Morgan Kauffman.

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

MacKay, D. J. C. (1995). Developments in probabilistic modelling with neural networks—ensemble learning. In *Neural Networks: Artificial Intelligence and Industrial Applications. Proceedings of the 3rd Annual Symposium on Neural Networks, Nijmegen, Netherlands, 14-15 September 1995*, pp. 191–198. Berlin: Springer.

MacKay, D. J. C. (1995). Probable networks and plausible predictions – a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems* **6** (3), 469–505.

Nabney, I. T. (1998). NETLAB neural network software. available from `http://www.ncrg.aston.ac.uk/netlab/`.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer. Lecture Notes in Statistics 118.

Neal, R. M. (1999). Software for flexible Bayesian modeling and Markov chain sampling. available from `http://www.cs.utoronto.ca/~radford/fbm.software.html`.

Rasmussen, C. E. (1996). A practical Monte Carlo implementation of Bayesian learning. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems*, Volume 8, pp. 598–604. Cambridge, MA: MIT Press.

Thodberg, H. H. (1996, January). A review of Bayesian neural networks with an application to near infrared spectroscopy. *IEEE Transactions on Neural Networks* **7** (1), 56–72.

Tong, H. (1995). *Non-linear Time Series: a Dynamical System Approach*, Volume 6 of *Oxford Statistical Science Series*. Oxford: Clarendon Press.

Weigend, A. S., B. A. Huberman, and D. E. Rumelhart (1990). Predicting sunspots and exchange rates with connectionist networks. In S. Eubank and M. Casdagli (Eds.), *Proceedings of the 1990 NATO Workshop on Nonlinear Modeling and Foracsting, Santa Fe, New Mexico*. Addislon-Wesley.

A Lower bound on the entropy of a mixture distribution

The entropy of a mixture distribution $q_{\mathbf{w}}(\mathbf{w}) = \sum_m Q(m) q_{\mathbf{w}}(\mathbf{w}|m)$ is defined as:

$$\mathcal{H}(q_{\mathbf{w}}(\mathbf{w})) = -\sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) \log\left\{\sum_{m=1}^{M} Q_{\mathbf{w}}(m) q_{\mathbf{w}}(\mathbf{w}|m)\right\} d\mathbf{w}. \quad \text{(A.1)}$$

By introducing the mutual information between the component label m and the variable $\mathbf{w}$ in the distribution, the entropy can be rewritten:

$$\begin{aligned}
\mathcal{H}(q_{\mathbf{w}}(\mathbf{w})) &= -\sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) \log\left\{q_{\mathbf{w}}(\mathbf{w}|m) \frac{q_{\mathbf{w}}(\mathbf{w})}{q_{\mathbf{w}}(\mathbf{w}|m)}\right\} d\mathbf{w} \\
&= -\sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) \log q_{\mathbf{w}}(\mathbf{w}|m) d\mathbf{w} \\
&\quad + \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) \log \frac{q_{\mathbf{w}}(\mathbf{w}|m)}{q_{\mathbf{w}}(\mathbf{w})} d\mathbf{w} \\
&= \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \mathcal{H}(q_{\mathbf{w}}(\mathbf{w}|m)) + I(m;\mathbf{w}). \qquad \text{(A.2)}
\end{aligned}$$

We are still left with an average of a logarithm of a sum. To make progress, we look to a lower bound on the mutual information which has been first introduced by Jaakkola and Jordan (1998) for discrete systems:

$$\begin{aligned}
I(m;\mathbf{w}) &= \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) \log \frac{q_{\mathbf{w}}(\mathbf{w}|m)}{q_{\mathbf{w}}(\mathbf{w})} d\mathbf{w} \\
&= -\sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) \log\left\{\frac{Q_{\mathbf{w}}(m) r(\mathbf{w}|m)}{Q_{\mathbf{w}}(m) r(\mathbf{w}|m)} \frac{q_{\mathbf{w}}(\mathbf{w})}{q_{\mathbf{w}}(\mathbf{w}|m)}\right\} d\mathbf{w} \\
&= \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) \log r(\mathbf{w}|m) d\mathbf{w} - \sum_{m} Q_{\mathbf{w}}(m) \log Q_{\mathbf{w}}(m) \\
&\quad + \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) \left\{-\log \frac{r(\mathbf{w}|m)}{Q_{\mathbf{w}}(m)} \frac{q_{\mathbf{w}}(\mathbf{w})}{q_{\mathbf{w}}(\mathbf{w}|m)}\right\} d\mathbf{w}, \quad \text{(A.3)}
\end{aligned}$$

where we have introduced a smoothing distribution $r(\mathbf{w}|m)$. In order to avoid averaging the logarithm of the sum $q_{\mathbf{w}}(\mathbf{w})$, we now make use of the following convexity inequality:

$$-\log(x) \geq -\lambda x + \log(\lambda) + 1, \qquad \text{(A.4)}$$

and introduce a new variational parameter λ for each smoothing distribution r and obtain the bound

$$\begin{aligned}\mathcal{I}(m;\mathbf{w}) \geq & \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) \log r(\mathbf{w}|m) d\mathbf{w} \\ & - \sum_{m,m'}^{M} Q_{\mathbf{w}}(m) \lambda_{m'} \int q_{\mathbf{w}}(\mathbf{w}|m) r(\mathbf{w}|m') d\mathbf{w} \\ & - \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \log Q_{\mathbf{w}}(m) + \sum_{m=1}^{M} Q_{\mathbf{w}}(m) \log \lambda_m + 1.\end{aligned}$$

For the case of mixtures of Gaussian distributions $q_{\mathbf{w}}(\mathbf{w}|m) = \mathcal{N}(\bar{\mathbf{w}}_m, \mathbf{\Sigma}_{q_m})$ and we choose $r(\mathbf{w}|m') \propto \mathcal{N}(\bar{\mathbf{r}}_{m'}, \mathbf{\Sigma}_{r_{m'}})$ and the second term is then the integral across two Gaussians which and is itself Gaussian in form, for convenience we define

$$\Delta(m) = (\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_m)^{\mathrm{T}} \mathbf{\Sigma}_{r_m}^{-1} (\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_m), \tag{A.5}$$

$$\Delta(m, m') = (\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_{m'})^{\mathrm{T}} (\mathbf{\Sigma}_{q_m} + \mathbf{\Sigma}_{r_{m'}})^{-1} (\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_{m'}), \tag{A.6}$$

$$\mathbf{A}_{m,m'} = \mathbf{I} + \mathbf{\Sigma}_{q_m} \mathbf{\Sigma}_{r_{m'}}^{-1}, \tag{A.7}$$

giving

$$\langle r(\mathbf{w}|m') \rangle_{q_{\mathbf{w}}(\mathbf{w}|m)} = |\mathbf{A}_{m,m'}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\Delta(m, m')\right).$$

Also required is the expectation of the log of the smoothing distribution, again this may be computed analytically

$$\langle \log r(\mathbf{w}|m') \rangle_{q_{\mathbf{w}}(\mathbf{w}|m)} = -\frac{1}{2}\left\{ \mathrm{tr}(\mathbf{\Sigma}_{q_m} \mathbf{\Sigma}_{r_m}^{-1}) + \Delta(m) \right\}. \tag{A.8}$$

This last result additionally allows us to write down the entropy of each component of the mixture:

$$\mathcal{H}(q_{\mathbf{w}}(\mathbf{w}|m)) = \frac{1}{2} \log |\mathbf{\Sigma}_{q_m}| + \frac{K}{2}(1 + \log 2\pi). \tag{A.9}$$

This leads us to our lower bound on the entropy of a mixture of Gaussians:

$$\begin{aligned}
\mathcal{H}(q_{\mathbf{w}}(\mathbf{w})) \geq & \frac{1}{2}\sum_{m=1}^{M} Q_{\mathbf{w}}(m)\log|\mathbf{\Sigma}_{q_m}| + \frac{K}{2}(1+\log 2\pi) \\
& -\frac{1}{2}\sum_{m=1}^{M} Q_{\mathbf{w}}(m)\left\{\mathrm{tr}(\mathbf{\Sigma}_{q_m}\mathbf{\Sigma}_{r_m}^{-1}) + \Delta(m)\right\} \\
& -\sum_{m,n}^{M} Q_{\mathbf{w}}(m)\lambda_{m'}|\mathbf{A}_{m,m'}|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\Delta(m,m')\right\} \\
& -\sum_{m=1}^{M} Q_{\mathbf{w}}(m)\log Q_{\mathbf{w}}(m) + \sum_{m=1}^{M} Q_{\mathbf{w}}(m)\log\lambda_m + 1 \qquad \text{(A.10)} \\
\equiv & \hat{\mathcal{H}}(q_{\mathbf{w}}(\mathbf{w})). \qquad \text{(A.11)}
\end{aligned}$$

Additionally the mutual information may be upper bounded, consider

$$\begin{aligned}
I(m;\mathbf{w}) = & \sum_{m=1}^{M} Q_{\mathbf{w}}(m)\mathcal{H}(q_{\mathbf{w}}(\mathbf{w}|m)) \\
& -\sum_{m=1}^{M} Q_{\mathbf{w}}(m)\left\langle\log\left(Q_{\mathbf{w}}(m)q_{\mathbf{w}}(\mathbf{w}|m) + \sum_{m'\neq m}^{M} Q_{\mathbf{w}}(m')q_{\mathbf{w}}(\mathbf{w}|m')\right)\right\rangle_{q_{\mathbf{w}}(\mathbf{w}|m)} \text{(A.12)}
\end{aligned}$$

If we neglect the terms $\sum_{m'\neq m}^{M} Q_{\mathbf{w}}(m')q_{\mathbf{w}}(\mathbf{w}|m')$ in the logarithm [7]. We may write down an upper bound on the mutual information.

$$\begin{aligned}
I(m;\mathbf{w}) \leq & \sum_{m=1}^{M} Q_{\mathbf{w}}(m)\mathcal{H}(q_{\mathbf{w}}(\mathbf{w}|m)) \\
& -\sum_{m=1}^{M} Q_{\mathbf{w}}(m)\left(\log Q_{\mathbf{w}}(m) + \mathcal{H}(q_{\mathbf{w}}(\mathbf{w}|m))\right) \qquad \text{(A.13)} \\
\leq & -\sum_{m}^{M} Q_{\mathbf{w}}(m)\log Q_{\mathbf{w}}(m), \qquad \text{(A.14)}
\end{aligned}$$

The entropy of a discrete random variable with M states is upper bounded by $\log M$, therefore the logarithm of the number of components is an upper bound on the mutual information.

B Lower bound on the log-likelihood

Recall our bound 29,

[7] This corresponds to an assumption that there is no overlap between the separate components.

$$\log p(D) \geq \sum_m Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) q_\beta(\beta) \left\{ -\beta E_D(D, \mathbf{w}) + \frac{N}{2} \log \beta \right\} d\mathbf{w}\ d\beta$$
$$+ \sum_m Q_{\mathbf{w}}(m) \int q_{\mathbf{w}}(\mathbf{w}|m) q_\alpha(\alpha) \left\{ -\alpha E_w(\mathbf{w}) + \frac{K}{2} \log \alpha \right\} d\mathbf{w}\ d\alpha$$
$$+\mathcal{H}(q_{\mathbf{w}}(\mathbf{w})) + \mathcal{H}(q_\alpha(\alpha)) + \mathcal{H}(q_\beta(\beta)). \tag{B.1}$$

Using the following results

$$\int \Gamma(x|a, b) \log x \ dx = \psi(a) - \ln b, \tag{B.2}$$
$$-\int \Gamma(x|a, b) \log \Gamma(x|a, b) \ dx = -(a-1)\psi(a) - \ln b + a + \ln \Gamma(a), \tag{B.3}$$

where the function ψ is defined as

$$\psi(a) = \frac{d \ln \Gamma(a)}{da}. \tag{B.4}$$

we may obtain the entropy of the gamma distributions and the required expectations of the hyper-parameters. The expectation of the term derived from the prior is fount as

$$\int \mathbf{w}^{\mathrm{T}} \mathbf{w} q_{\mathbf{w}}(\mathbf{w}|m) d\mathbf{w} = \mathrm{tr}(\Sigma_{q_m}) + \frac{1}{2} \bar{\mathbf{w}}_m^{\mathrm{T}} \bar{\mathbf{w}}_m. \tag{B.5}$$

A more complex task is the computation of expectations of the network output and its square which are required in the terms coming from the likelihood, dropping the component label m we have

$$\int q_{\mathbf{w}}(\mathbf{w}) E_D(D, \mathbf{w}) d\mathbf{w} = \frac{1}{2} \sum_{n=1}^{N} \left\langle f(\mathbf{x}, \mathbf{w})^2 \right\rangle_q - \sum_{n=1}^{N} t_n \left\langle f(\mathbf{x}, \mathbf{w}) \right\rangle_q + \frac{1}{2} \sum_{n=1}^{N} t_n^2, \tag{B.6}$$

with $f(\mathbf{x}, \mathbf{w}) = \sum_h v_h \mathrm{erf}(\mathbf{u}_h^{\mathrm{T}} \mathbf{x})$. We initially follow Barber and Bishop (1998) in our treatment of the expectations,

$$q(\mathbf{u}, \mathbf{v}) = \frac{1}{(2\pi)^{\frac{K}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (\mathbf{u} - \bar{\mathbf{u}})^{\mathrm{T}} \mathbf{\Sigma_{uu}}^{-1} (\mathbf{u} - \bar{\mathbf{u}}) - \frac{1}{2} (\mathbf{v} - \bar{\mathbf{v}})^{\mathrm{T}} \mathbf{\Sigma_{vv}}^{-1} (\mathbf{v} - \bar{\mathbf{v}}) \right\}, \tag{B.7}$$

where we have used $\mathbf{u}$ to represent a vector formed from the vectors $\{\mathbf{u}_1 \ldots \mathbf{u}_H\}$ and we have introduced $\bar{\mathbf{u}}$ and $\bar{\mathbf{v}}$ which correspond to $\bar{\mathbf{w}}$ in the same way as $\mathbf{u}$ and $\mathbf{v}$ correspond to $\mathbf{w}$. The matrices $\mathbf{\Sigma_{vv}}$ and $\mathbf{\Sigma_{uu}}$ represent the associated covariance matrices, both of which are taken to be diagonal in this implementation [8] For convenience, we use the notation $\mathbf{u}^{\mathrm{T}} \mathbf{x}_h$ instead of $\mathbf{u}_h^{\mathrm{T}} \mathbf{x}$, where $\mathbf{x}_h$

[8] The calculations can be performed analytically for more complex forms of covariance matrix, the most advanced being one which constrains $\mathbf{\Sigma_{uu}}$ to correlations

is a vector of the same dimension as the concatenated vector $\mathbf{u}$ with zero components everywhere except for those that correspond to hidden unit h. Those elements contain $\mathbf{x}$. The expectations can now be rewritten as:

$$\langle f(\mathbf{x},\mathbf{w})\rangle_q = \sum_{h=1}^{H} \left\langle v_h \mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_h)\right\rangle_{q(\mathbf{u},\mathbf{v})}, \tag{B.8}$$

$$\left\langle f(\mathbf{x},\mathbf{w})^2\right\rangle_q = \sum_{i,j}^{H} \left\langle v_i v_j \mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_i)\mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_j)\right\rangle_{q(\mathbf{u},\mathbf{v})}. \tag{B.9}$$

As the components of $\mathbf{v}$ do not enter the non-linear function, we can immediately integrate over $\mathbf{v}$,

$$\langle f(\mathbf{x},\mathbf{w})\rangle_q = \sum_{h} \left\langle \bar{\mathbf{w}}_{\mathbf{v}_h} \mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_h)\right\rangle_{q(\mathbf{u})} \tag{B.10}$$

$$\left\langle f(\mathbf{x},\mathbf{w})^2\right\rangle_q = \sum_{i,j} \left\langle (\bar{\mathbf{w}}_{\mathbf{v}_i}\bar{\mathbf{w}}_{\mathbf{v}_j} + \mathbf{\Sigma}_{\mathbf{v}_{i,j}}) \mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_i)\mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_j)\right\rangle_{q(\mathbf{u})}, \tag{B.11}$$

which leaves us with expectations of the erf functions under Gaussian distributions. These may be reduced to one dimensional integrals by firstly transforming the Gaussian to have zero mean and an isotropic covariance,

$$\left\langle \mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_h)\right\rangle_{q(\mathbf{u})} = \left\langle \mathrm{erf}\left(\bar{\mathbf{u}}^{\mathrm{T}}\mathbf{x}_h + \mathbf{x}_h^{\mathrm{T}}\mathbf{\Sigma}_u^{\frac{1}{2}}\mathbf{s}\right)\right\rangle_{\mathcal{N}(0,\mathbf{I})} \tag{B.12}$$

$$\left\langle \mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_i)\mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_j)\right\rangle_{q(\mathbf{u})} = \left\langle \mathrm{erf}\left(\mathbf{x}_i^{\mathrm{T}}\bar{\mathbf{u}} + \mathbf{x}_i^{\mathrm{T}}\mathbf{\Sigma}_u^{\frac{1}{2}}\mathbf{s}\right)\mathrm{erf}\left(\mathbf{x}_j^{\mathrm{T}}\bar{\mathbf{u}} + \mathbf{x}_j^{\mathrm{T}}\mathbf{\Sigma}_u^{\frac{1}{2}}\mathbf{s}\right)\right\rangle_{\mathcal{N}(0,\mathbf{I})} \tag{B.13}$$

where $\mathbf{s} = \mathbf{\Sigma}_{\mathbf{uu}}^{\frac{1}{2}}(\mathbf{u}-\bar{\mathbf{u}})$.We now rotate the co-ordinate system. For the $\langle \mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_h)\rangle_{q(\mathbf{u})}$ we decompose $\mathbf{s}$ into $\mathbf{s} = s_{\parallel}\mathbf{d} + \mathbf{s}_{\perp}$ where $\mathbf{d}$ is a unit vector parallel to $\mathbf{\Sigma}_{\mathbf{uu}}\mathbf{x}_h$ and $\mathbf{s}_{\perp}$ is orthogonal to $\mathbf{d}$. This leads to a one-dimensional integral over $s_{\parallel}$

$$\left\langle \mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_h)\right\rangle_{q(\mathbf{u})} = \left\langle \mathrm{erf}(\sqrt{\mathbf{x}_h^{\mathrm{T}}\mathbf{\Sigma}_{\mathbf{uu}}\mathbf{x}_h}\, s_{\parallel} + \bar{\mathbf{u}}^{\mathrm{T}}\mathbf{x}_h)\right\rangle_{\mathcal{N}(0,1)}, \tag{B.14}$$

and through application of the identity:

$$\frac{1}{\sqrt{2\pi}}\int \mathrm{erf}(ax+b)\exp(-\frac{1}{2}x^2)dx = \mathrm{erf}\left(\frac{b}{\sqrt{1+a^2}}\right), \tag{B.15}$$

---

between weights which 'fan-in' to each hidden node, but allows all covariances in the hidden to output layer and additionally correlations between weights in the two different layers. To compute all elements of $\mathbf{\Sigma}_{\mathbf{uu}}$ though, one needs to resort to the numerical techniques discussed by Barber and Bishop.

we get the following result:

$$\langle f(\mathbf{x}, \mathbf{w})\rangle_q = \sum_{h=1}^{H} \bar{\mathbf{w}}_{\mathbf{v}_h} \operatorname{erf}\left(\frac{\bar{\mathbf{u}}^{\mathrm{T}}\mathbf{x}_h}{\sqrt{1+\mathbf{x}_h^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{uu}}\mathbf{x}_h}}\right). \tag{B.16}$$

For the expectation of the square, the approach is the same. We rotate the coordinate system so that $\mathbf{s} = s_1\mathbf{a} + s_2\mathbf{b} + \mathbf{s}_\perp$, where $\mathbf{a}$ and $\mathbf{b}$ are orthogonal. The arguments of the erf functions are independent of the component of $\mathbf{s}_\perp$. Barber and Bishop then treat the remaining integral by numerical means. However, due to the constraints we imposed on $\boldsymbol{\Sigma}$ we observe that $\boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{x}_i$ and $\boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{x}_j$ are also orthogonal. This allows us not to take into account the correlations amongst the weights and to perform the integrations separately over each vector $\mathbf{u}_i$:

$$\begin{aligned}\left\langle f(\mathbf{x}, \mathbf{w})^2\right\rangle_q =& \sum_{i\neq j}^{H}\left\{\bar{\mathbf{w}}_{\mathbf{v}_i}\bar{\mathbf{w}}_{\mathbf{v}_j} \operatorname{erf}\left(\frac{\bar{\mathbf{u}}^{\mathrm{T}}\mathbf{x}_i}{\sqrt{1+\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{uu}}\mathbf{x}_i}}\right) \operatorname{erf}\left(\frac{\bar{\mathbf{u}}^{\mathrm{T}}\mathbf{x}_j}{\sqrt{1+\mathbf{x}_j^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{uu}}\mathbf{x}_j}}\right)\right\} \\ &+ \sum_{h=1}^{H}\left(\bar{\mathbf{w}}_{\mathbf{v}_h}^2 + \boldsymbol{\Sigma}_{\mathbf{v}_h}\right)\left\langle \operatorname{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_h)^2\right\rangle_{\mathcal{N}(0,1)}\end{aligned} \tag{B.17}$$

The remaining integral $\left\langle \operatorname{erf}(\mathbf{u}_h^{\mathrm{T}}\mathbf{x})^2\right\rangle_{\mathcal{N}(0,1)}$ is still unfortunately analytically intractable. One can approximate this expectation by using lookup tables or numerical integration techniques, however a further solution, and the one preferred in our case is to use a parametric approximation to a neural network for approximating this function. We choose to $\operatorname{erf}(x)^2$ of the form $g(x) = 1 - \exp(-\frac{c}{2}x^2)$ where $c \approx 1.24$. This facilitates differentiation of the bound.

$$\left\langle \operatorname{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_h)^2\right\rangle_{q(\mathbf{u})} \approx 1 - \frac{1}{\sqrt{1+c(\mathbf{x}_h^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{u}}\mathbf{x}_h)}}\exp\left\{\frac{-c(\bar{\mathbf{u}}^{\mathrm{T}}\mathbf{x}_h)^2}{2(1+c(\mathbf{x}_h^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{u}}\mathbf{x}_h))}\right\} \tag{B.18}$$

We have now all the terms needed to compute the lower bound on the marginal log-likelihood.

C Derivatives

We now give some of the required derivatives for the optimisation of our lower bound . We first present the derivatives of the lower bound A.11, for the general case of full covariance matrices

$$
\begin{aligned}
\frac{\partial \hat{\mathcal{H}}(q_{\mathbf{w}})}{\partial \bar{\mathbf{w}}_m} &= -Q_{\mathbf{w}}(m)\boldsymbol{\Sigma}_{r_m}^{-1}(\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_m) \\
&\quad + Q_{\mathbf{w}}(m)\sum_{m'} \lambda_{m'} |\mathbf{A}_{m,m'}|^{-\frac{1}{2}} e^{-\frac{1}{2}\Delta(m,m')}(\boldsymbol{\Sigma}_{q_m} + \boldsymbol{\Sigma}_{r_{m'}})^{-1}(\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_{m'}) \qquad \text{(C.1)} \\
\frac{\partial \hat{\mathcal{H}}(q_{\mathbf{w}})}{\partial \bar{\mathbf{r}}_{m'}} &= Q_{\mathbf{w}}(m)\boldsymbol{\Sigma}_{r_{m'}}^{-1}(\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_m) \\
&\quad - \lambda_{m'} \sum_{m} Q_{\mathbf{w}}(m) |\mathbf{A}_{m,m'}|^{-\frac{1}{2}} e^{-\frac{1}{2}\Delta(m,m')}(\boldsymbol{\Sigma}_{q_m} + \boldsymbol{\Sigma}_{r_{m'}})^{-1}(\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_{m'}) \qquad \text{(C.2)}
\end{aligned}
$$

Using some general matrix derivative rules, we get also the following equations:

$$
\begin{aligned}
\frac{\partial \hat{\mathcal{H}}(q_{\mathbf{w}})}{\partial \boldsymbol{\Sigma}_{q_m}} &= \frac{1}{2} Q_{\mathbf{w}}(m)\Bigg\{\boldsymbol{\Sigma}_{q_m}^{-1} - \boldsymbol{\Sigma}_{r_m}^{-1} + \sum_{m'} \lambda_{m'} |\mathbf{A}_{m,m'}|^{-\frac{1}{2}} e^{-\frac{1}{2}\Delta(m,m')}\Big(\mathbf{A}_{m,m'}^{-1}\boldsymbol{\Sigma}_{r_{m'}}^{-1} \\
&\quad - (\boldsymbol{\Sigma}_{q_m} + \boldsymbol{\Sigma}_{r_{m'}})^{-1}(\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_{m'})^{\mathrm{T}}(\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_{m'})(\boldsymbol{\Sigma}_{q_m} + \boldsymbol{\Sigma}_{r_{m'}})^{-1}\Big)\Bigg\} \qquad \text{(C.3)} \\
\frac{\partial \hat{\mathcal{H}}(q_{\mathbf{w}})}{\partial \boldsymbol{\Sigma}_{r_{m'}}} &= \frac{1}{2}\alpha_{m'}\Big\{\boldsymbol{\Sigma}_{r_{m'}}^{-1}\boldsymbol{\Sigma}_{q_{m'}}\boldsymbol{\Sigma}_{r_{m'}}^{-1} + \boldsymbol{\Sigma}_{r_{m'}}^{-1}(\bar{\mathbf{w}}_{m'} - \bar{\mathbf{r}}_{m'})^{\mathrm{T}}(\bar{\mathbf{w}}_{m'} - \bar{\mathbf{r}}_{m'})\boldsymbol{\Sigma}_{r_{m'}}^{-1}\Big\} \\
&\quad - \frac{1}{2}\lambda_{m'}\Bigg\{\sum_{m} Q_{\mathbf{w}}(m) |\mathbf{A}_{m,m'}|^{-\frac{1}{2}} e^{-\frac{1}{2}\Delta(m,m')}\Big(\mathbf{A}_{m,m'}^{-1}\boldsymbol{\Sigma}_{r_{m'}}^{-1}\boldsymbol{\Sigma}_{q_m}\boldsymbol{\Sigma}_{r_{m'}}^{-1} \\
&\quad + (\boldsymbol{\Sigma}_{q_m} + \boldsymbol{\Sigma}_{r_{m'}})^{-1}(\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_{m'})^{\mathrm{T}}(\bar{\mathbf{w}}_m - \bar{\mathbf{r}}_{m'})(\boldsymbol{\Sigma}_{q_m} + \boldsymbol{\Sigma}_{r_{m'}})^{-1}\Big)\Bigg\} \qquad \text{(C.4)}
\end{aligned}
$$

Secondly we consider the derivatives of the output function expectations for the case of diagonal covariance Gaussian. The derivatives of $\langle \mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_h)\rangle_{q(\mathbf{u})}$ are similar and are omitted.

$$
\begin{aligned}
\frac{\partial \langle \mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_h)\rangle_{q(\mathbf{u})}}{\partial \bar{\mathbf{w}}_{\mathbf{u}_h}} &= \sqrt{\frac{2}{\pi}} \exp\left\{\frac{-(\bar{\mathbf{u}}^{\mathrm{T}}\mathbf{x}_h)^2}{2(1 + \mathbf{x}_h^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{uu}}\mathbf{x}_h)}\right\} \cdot \frac{x_h}{(1 + \mathbf{x}_h\boldsymbol{\Sigma}_{\mathbf{uu}}\mathbf{x}_h)^{\frac{1}{2}}} \\
\frac{\partial \langle \mathrm{erf}(\mathbf{u}^{\mathrm{T}}\mathbf{x}_h)\rangle_{q(\mathbf{u})}}{\partial \boldsymbol{\Sigma}_{\mathbf{u}_h}} &= \sqrt{\frac{2}{\pi}} \exp\left\{\frac{-(\bar{\mathbf{u}}^{\mathrm{T}}\mathbf{x}_h)^2}{2(1 + \mathbf{x}_h^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{uu}}\mathbf{x}_h)}\right] \cdot \frac{(\bar{\mathbf{u}}^{\mathrm{T}}\mathbf{x})(-x_i^2\boldsymbol{\Sigma}_{\mathbf{u}_h})}{(1 + \mathbf{x}_h^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{uu}}\mathbf{x}_h)^{\frac{3}{2}}} \qquad \text{(C.5)}
\end{aligned}
$$

We have also omitted the equations relative to the derivatives with respect to the mixture coefficients $Q_{\mathbf{w}}(m)$ and $\lambda_{m'}$. These derivatives are straightforward to obtain, and may be used to form fixed point update equations for these parameters.

With the lower bound on the model likelihood, these derivatives are used in a non-linear optimisation algorithm to find the parameters $\bar{\mathbf{w}}_m$, $\boldsymbol{\Sigma}_{q_m}$, $Q_{\mathbf{w}}(m)$, $\bar{\mathbf{r}}_{m'}$, $\boldsymbol{\Sigma}_{r_{m'}}$ and $\lambda_{m'}$ that represent the best fit for the mixture of Gaussians .