A Review of Bayesian Neural Networks with an Application to Near Infrared Spectroscopy

Hans Henrik Thodberg
The Danish Meat Research Institute
Maglegaardsvej 2, DK-4000 Roskilde
e-mail thodberg@nn.dmri.dk

Revised version of Manuscript 1132E-part I, Feb 6, 1995

Abstract

MacKay's Bayesian framework for backpropagation is a practical and powerful means to improve the generalisation ability of neural networks. It is based on a Gaussian approximation to the posterior weight distribution. The framework is extended, reviewed and demonstrated in a pedagogical way. The notation is simplified using the ordinary weight decay parameter, and a detailed and explicit procedure for adjusting several weight decay parameters is given.

Bayesian backprop is applied in the prediction of fat content in minced meat from near infrared spectra. It outperforms "early stopping" as well as quadratic regression. The evidence of a committee of differently trained networks is computed, and the corresponding improved generalisation is verified. The error bars on the predictions of the fat content are computed. There are three contributors: The random noise, the uncertainty in the weights, and the deviation among the committee members. The Bayesian framework is compared to Moody's GPE. Finally, MacKay and Neal's Automatic Relevance Determination, in which the weight decay parameters depend on the input number, is applied to the data with improved results.

1 Introduction

1.1 Highlights

Bayesian neural networks or simply Bayesian backprop is the application of MacKay's Bayesian framework to feedforward networks implemented by the backpropagation algorithm. The main benefits are:

- Bayesian backprop adjusts the weight decay parameters automatically to their near-optimal values that gives the best generalisation. The adjustment is done during training, so the tedious and computer-intensive search for weight decay parameters is no longer needed.
- The Bayesian formalism estimates the *evidence* for each model. The evidence measures how probable the model is, given the data (assuming equal

prior probability). The evidence correlates well with the generalisation error and is used as a quality measure to select the best model. Networks converged to different local minima and networks with different numbers of nodes, layers and inputs can be compared and ranked according to the evidence. The evidence can also be used as a stop criterion for network growing or pruning. Pruning is treated in a separate paper [1].

- The instrument of the Bayesian analysis described in the above two items is the training data and the network itself. No separate validation set is used. Hence all available data can be used for training, which gives better models.

1.2 The Proper Treatment of Bayesian Backprop

Bayesian backprop was introduced by MacKay in 1991 [2, 3, 4, 5] as a radically different approach to the problem of overfitting and model comparison. The Bayesian framework for backprop originated in the field of Maximum Entropy [6] which develops better models for the analysis of empirical data from chemistry, geophysics, astronomy, physics, economics etc., and the approach is both theoretical and applied. However, the Maximum Entropy principle plays hardly any role in Bayesian backprop.

Bayesian backprop is based on the Bayesian school of statistics [7, 8]. This is distinct from mainstream "sampling theory" (or "frequentist") statistics, where the concept of probability must be attached to frequencies of samples drawn from a distribution. In contrast, Bayesians also use probabilities to describe *degrees of belief* in parameter values or models. With this interpretation Bayes' formula acquires a deeper meaning since it is now a way to compute beliefs.

The Bayesian analysis computes the evidence which expresses our posterior belief in each model. Empirically the evidence correlates well with the generalisation error of the model (there is a strong negative correlation), so the evidence can be used as a quality measure for models. At present, the relation between evidence and test error cannot be proved formally. However, that one must ultimately base inference on basic principles that cannot themselves be proved is widely accepted.

This paper is intended to bridge the gap between academic theory and its industrial application. The proper balance between the theoretical and applied aspects is emphasised. MacKay's formalism is recapitulated using the ordinary weight decay parameters. The mechanism of multiple weight decay parameters and the estimation of the Hessian are spelled out at the implementation level. The framework is demonstrated on real-life data. While MacKay studied learning problems with one or two input units, we consider problems with 8-18 inputs. This is a more appropriate domain of application for multilayer perceptrons.

1.3 Ockham's Razor

Bayesian model comparison addresses a question which has been discussed for centuries. William of Ockham (1285-1349) was already analysing and comparing theories of Nature and God. At the time, philosophers wanted to unify Aristotelian philosophy with Catholicism. William disliked overcomplicated thinking systems – why invoke God in an explanation of a physical phenomenon? He systematically applied a principle of economy in explanations, which was later given the name Ockham's Razor. Ockham never formulated the principle explicitly, so we take the liberty to state our own interpretation of Ockham's Razor:

> *If several theories account for a phenomenon we should prefer the simplest which describes the data sufficiently well.*

The principle states that a model has two virtues: simplicity and goodness of fit. But William did not specify the meaning of "sufficiently well" - the optimal balance between the two virtues was left to the intuition of the scientist. People are in fact rather good at assessing this balance in situations related to daily life. However, in other matters, like abstract scientific theories and neural network models, it is useful to formalise this balance.

Bayesian model comparison does just that. We express our belief in a model as its probability given the data, and use Bayes' formula:

$$P(\mathcal{H}\,|\,D) = \frac{P(D\,|\,\mathcal{H})P(\mathcal{H})}{P(D)} \tag{1}$$

We assume that the prior belief $P(\mathcal{H})$ is the same for all models, so we can compare models by comparing $P(D\,|\,\mathcal{H})$ which is called *the evidence for* $\mathcal{H}$, and acts as a quality measure in Bayesian model comparison. To compute $P(D\,|\,\mathcal{H})$ we integrate over the tunable parameters $\mathbf{w}$ of the model:

$$P(D\,|\,\mathcal{H}) = \int P(D\,|\,\mathbf{w},\mathcal{H})P(\mathbf{w}\,|\,\mathcal{H})d\mathbf{w}$$

The integrand is assumed to possess a pronounced maximum at $\mathbf{w}_{\mathrm{MP}}$ of width (or volume) $\Delta\mathbf{w}_{\mathrm{posterior}}$, and writing the prior probability for $\mathbf{w}$ as $P(\mathbf{w}\,|\,\mathcal{H}) = 1/\Delta\mathbf{w}_{\mathrm{prior}}$ we get:

$$\begin{aligned} P(D\,|\,\mathcal{H}) &= P(D\,|\,\mathbf{w}_{\mathrm{MP}},\mathcal{H})\frac{\Delta\mathbf{w}_{\mathrm{posterior}}}{\Delta\mathbf{w}_{\mathrm{prior}}} \\ \text{Evidence} &= \text{Likelihood} \times \text{OckhamFactor} \\ \text{ModelQuality} &= \text{DataFit} \times \text{Simplicity} \end{aligned} \tag{2}$$

We have split the evidence for the model into two factors:

- The best fit likelihood, i.e. the probability of the data given the model parameters tuned to that data. It measures how well the *tuned* model fits the data.

- The ratio of the available parameters volume after and before they are tuned to the data, i.e. the collapse of the available parameter space when the data is taken into account. This factor is small when the model has many parameters or when some parameters must be tuned very accurately to fit the data. This measures the simplicity of the model and is therefore called *the Ockham Factor* [1].

[1] The Ockham factor is not a penalty or complexity factor - it is a benefit or simplicity factor

Thus the Bayesian framework is essentially a mathematical formulation of Ockham's Razor.

1.4 The Four Levels of Bayesian Inference

MacKay organised the Bayesian inference for neural networks into the following four levels [2]:

Level 1 Make predictions including error bars for new input data.

Level 2 Estimate the weight parameters and their uncertainties.

Level 3 Estimate the weight decay parameters and their uncertainties

Level 4 Repeat level 2 and 3 with different initial conditions and different network architectures. Select the architecture and $\mathbf{w}$-minimum with highest evidence. Optionally select a committee to reflect the uncertainty on this level.

Level 1 – to make predictions – is the goal of the inference. The higher levels are designed to exploit the training data optimally in making predictions.

On level 2 the posterior weight distribution near one local minimum of the cost function is approximated by a Gaussian. The Gaussian approximation is the main weakness of the approach. MacKay found that the Gaussian approximation breaks down when the number of weights exceeds one third of the number of training cases [3]. This is confirmed in the present study.

Level 3 is particular to MacKay's approach and is discussed in detail below.

Level 4 deals with the model comparison discussed already in section 1.3. When we talk about a neural network model $\mathcal{H}$, we mean a specific architecture, the particular $\mathbf{w}$-minimum and the type of prior (for instance the number of weight decay parameters).

On level 3 the weight decay parameters (with uncertainties) are determined by maximising the evidence. An alternative approach was proposed by Buntine and Weigend [9], who analytically integrate over the weight decay parameters. The proper treatment of weight decay parameters (also called hyperparameters) has caused some controversy which is best summarised by quoting Neal (who was not involved in the controversy):

"MacKay's approach to handling hyperparameters is computationally equivalent to the "ML-II" method of prior selection (Section 3.5.4 in [8]). From a fully Bayesian viewpoint it is only an approximation to the true answer, which would be obtained by integrating over the hyperparameters as well as the parameters, but experience has shown that it is often a good approximation. Wolpert [10] criticizes the use of this procedure for neural networks on the grounds that by analytically integrating over the hyperparameters, in the manner of Buntine and Weigend, one can obtain the relative posterior probability densities for different values of the network parameters *exactly*, without the need for any approximation. This criticism is based on a failure to appreciate the nature of the task. The posterior probability densities for different parameter values are, in themselves, of no interest - all that matters is how well the predictive distribution is approximated. MacKay [11] shows that in approximating this predictive distribution, it is more important to integrate over the large number of parameters in the network than over the typically small number of hyperparameters.

This controversy has perhaps distracted from consideration of other problems with Gaussian approximation methods that are in my opinion more significant." (Page 23-24 in [12]).

Neal replaces the Gaussian approximation with a Monte Carlo method. In this paper, on the other hand, we pursue MacKay's method but in accordance with Neal, we keep an open eye on the Gaussian approximation, which is recognised as the main limitation of the approach.

Section 3.4 continues the discussion of the four levels of inference from the point of view of posterior uncertainties on each of these levels.

2 Elements of Bayesian Backprop

2.1 Probabilistic Interpretation of Neural Networks

The Likelihood

The training set D consists of N cases of the form $(\mathbf{x}, t)$ assumed to be independently drawn from a distribution $P(\mathbf{x}, t)$. We model t as a function of $\mathbf{x}$, $t = y(\mathbf{x}) + \nu$, assuming the noise ν to be Gaussian with a mean of zero.

A neural network $\mathcal{H}$ with weights $\mathbf{w}$ is used to implement $y(\mathbf{x})$ which is an estimate of $E(t\,|\,\mathbf{x})$, the *regression* of t on $\mathbf{x}$. The noise scale is a free parameter $\beta = 1/\sigma_\nu^2$. The probability of a single datum under this model is:

$$P(t\,|\,\mathbf{x}, \mathbf{w}, \beta, \mathcal{H}) = \sqrt{\frac{\beta}{2\pi}} \exp(-\tfrac{1}{2}\beta(y-t)^2)$$

The data error is defined as

$$E_D \equiv \tfrac{1}{2} \sum (y-t)^2 \tag{3}$$

where the sum extends over the N cases. The formalism also applies to models with several outputs with equal noise level. Then the sum in (3) extends over outputs as well as cases and N is the number of terms.

The probability of the data, traditionally called *the likelihood*, is

$$\begin{aligned} P(D\,|\,\mathbf{w}, \beta, \mathcal{H}) &= \frac{1}{Z_D(\beta)} \exp(-\beta E_D) \\ Z_D(\beta) &\equiv (\frac{2\pi}{\beta})^{N/2} \end{aligned}$$

Here $P(D \mid \ldots)$ denotes the probability of the t values. The $\mathbf{x}$ values are understood to be part of the condition, but are suppressed in the notation.

The Prior Weight Distribution

So far we have described the likelihood of ordinary statistics. We now introduce the genuinely Bayesian features, starting with the prior distribution of the model parameters. The model contains k adjustable parameters $\mathbf{w}$, called weights, which are in general split into G groups $\{\mathcal{G}_g, g = 1, \ldots, G\}$ with k_g weights in group g. A weight is assigned a prior Gaussian distribution:

$$P(w \,|\, \beta, \xi_g, \mathcal{H}) = \sqrt{\frac{\beta\xi_g}{2\pi}} \exp(-\tfrac{1}{2}\beta\xi_g w^2) \text{ for } w \in \mathcal{G}_g$$

where ξ_g will turn out to be *the weight decay parameter* for group g.

This prior is based on our experience that positive and negative weights are equally frequent and that the weights have a finite variance. The Gaussian prior is the natural description of such a compact distribution. It is a matter of experience that this prior is adequate as long as the regression is reasonably smooth.

We call β and the ξ_g's *the scale parameters* of the model. The scale parameters are free parameters determined by the data but on a higher level than the weights. Scale parameters are sometimes called hyperparameters. Defining what we metaphorically call the "weight error"

$$E_{Wg} \equiv \tfrac{1}{2} \sum_{w \in \mathcal{G}_g} w^2 \tag{4}$$

the prior of all the weights $\mathbf{w}$ is

$$\begin{aligned} P(\mathbf{w} \,|\, \beta, \Xi, \mathcal{H}) &= \frac{1}{Z_W(\beta, \Xi)} \exp(-\sum_{g=1}^{G} \beta\xi_g E_{Wg}) \qquad (5) \\ Z_W(\beta, \Xi) &\equiv \prod_{g=1}^{G} (\frac{2\pi}{\beta\xi_g})^{k_g/2} \end{aligned}$$

where $\Xi \equiv \{\xi_1, \ldots, \xi_G\}$.

Separate weight decay parameters for the layers of a network are needed for several reasons. The simplest reason is that the inputs and outputs have arbitrary scales. If all inputs are multiplied by 2, the weights of the first layer must be divided by 2 to keep the mapping unchanged and hence the weight decay parameter must be multiplied by 4 to maintain its strength. Usually we do our best to scale the data properly. We center the inputs and scale them to a standard deviation of $\frac{1}{4}$, to match the scale of the hidden unit outputs, which are in the interval $(-\frac{1}{2}, \frac{1}{2})$. Similarly, in regression (where we apply linear output units) the target t is centered and scaled to unit standard deviation to match a typical net input to a hidden unit. But even after these preparations, the optimal weight decay parameters typically differ from each other by factors up to 10.

The Posterior Weight Distribution

We want to adjust the weights to their most probable values given the data, some values of the scale parameters (to be determined later) and the model. We compute the posterior weight distribution using Bayes' rule (the *swapping* *rule*) in its general form $P(A\,|\,B,F) = P(B\,|\,A,F)P(A\,|\,F)/P(B\,|\,F)$:

$$\begin{aligned} P(\mathbf{w}\,|\,D,\beta,\Xi,\mathcal{H}) &= \frac{P(D\,|\,\mathbf{w},\beta,\Xi,\mathcal{H})P(\mathbf{w}\,|\,\beta,\Xi,\mathcal{H})}{P(D\,|\,\beta,\Xi,\mathcal{H})} \\ &= \frac{\exp(-\beta C)}{Z_C(\beta,\Xi,\mathcal{H})} \end{aligned} \tag{6}$$

with

$$C \equiv E_D + \sum_g \xi_g E_{Wg} \tag{7}$$

$$Z_C(\beta,\Xi,\mathcal{H}) \equiv \int \exp(-\beta C)d\mathbf{w} \tag{8}$$

C is recognised as the well-known cost function for backprop with weight decay. MacKay uses $\alpha_g = \beta\xi_g$ and β as independent scale parameters. We use (β,ξ_g) because ξ_g is the weight decay parameter of ordinary weight decay, which is familiar to many.

So *the most probable – or maximum posterior – weights according to the probabilistic interpretation are identical to the weights obtained by minimising the cost function C*. Assuming the maximum posterior value to be a good approximation to the average of the posterior distribution we have a nice Bayesian account for the success of weight decay. It has been used for linear models in statistics for many years under the name ridge regression [13].

2.2 Determination of Scale Parameters

The Posterior Probability of the Scale Parameters

Above we determined the most probable (or maximum posterior, MP) weights given some values of the scale parameters, which we haven't determined yet. We now determine the MP scale parameters given the model and the data. To this end we use the MP weights found earlier. In fact we set up a loop, where we alternately determine

- the MP weights given the MP scale parameters, the data and the model
- the MP scale parameters given the data and the model.

This converges unless the model is completely wrong (see section 6.2). Sections 1.4 and 3.4 discuss these two levels of inference in perspective.

To evaluate the posterior probability for a generic scale parameter s, we swap the scale parameter and the data:

$$P(\log s\,|\,D,\mathcal{H}) = \frac{P(D\,|\,s,\mathcal{H})P(\log s\,|\,\mathcal{H})}{P(D\,|\,\mathcal{H})}$$

We write $P(\log s\,|\,\ldots)$, to indicate a probability density for $\log s$, which is related to the density for s by $P(s\,|\,\ldots)ds = P(\log s\,|\,\ldots)d\log s = P(\log s\,|\,\ldots)\frac{1}{s}ds$. (When s appears in the condition it makes of course no difference whether we use s or $\log s$.) A scale parameter describes the ratio of two quantities in the model. A weight decay parameter is the ratio between two kinds of error, and β is the ratio between the noise level and the arbitrary unit of the target values t. It should make no difference whether we model the ratio or its inverse, i.e. the prior should have the same form. This implies that the prior and the posterior are naturally expressed in terms of the logarithm of the scale parameter. We assume that the prior $P(\log s \mid \mathcal{H})$ is uniformly distributed in an interval containing $\log s_{\mathrm{MP}}$. This implies that s_{MP} is characterised as the maximum of $P(D \mid s, \mathcal{H})$.

We call $P(D \mid \mathbf{w}, s, \mathcal{H})$ *the likelihood of the* $\mathbf{w}$*-parameters*, following standard terminology. We denote $P(D \mid s, \mathcal{H})$ *the evidence for* $(s, \mathcal{H})$ because it is used as a quality measure for the model $(s, \mathcal{H})$.

To compute the evidence we use *the rule of divide and conquer* $P(A) = \sum_i P(A \mid B_i) P(B_i)$ where $\{B_i\}$ is a division of the whole:

$$\begin{aligned} P(D \mid \beta, \Xi, H) &= \int P(D \mid \mathbf{w}, \beta, \Xi, \mathcal{H}) P(\mathbf{w} \mid \beta, \Xi, \mathcal{H}) d\mathbf{w} \\ &= \int \frac{\exp(-\beta E_D)}{Z_D(\beta)} \frac{\exp(-\sum_{g=1}^{G} \beta \xi_g E_{Wg})}{Z_W(\beta, \Xi)} d\mathbf{w} \\ &= \frac{Z_C(\beta, \Xi, \mathcal{H})}{Z_D(\beta) Z_W(\beta, \Xi)} \end{aligned} \tag{9}$$

This is the evidence for $(\beta, \Xi, \mathcal{H})$.

The Gaussian Approximation to the Evidence

To evaluate Z_C we approximate the cost function by the quadratic form $C(\mathbf{w}) = C(\mathbf{w}_{\mathrm{MP}}) + \frac{1}{2}(\mathbf{w} - \mathbf{w}_{\mathrm{MP}})^{\mathrm{T}}(\mathbf{B} + \sum_g \xi_g \mathbf{I}_g)(\mathbf{w} - \mathbf{w}_{\mathrm{MP}})$. $\mathbf{B}$ is the Hessian matrix of the data error, $\mathbf{B} \equiv \nabla\nabla E_D$, or

$$B_{ij} \equiv \frac{\partial^2 E_D(\mathbf{w})}{\partial w_i \partial w_j}$$

evaluated at $\mathbf{w}_{\mathrm{MP}}$, and $\mathbf{I}_g = \nabla\nabla E_{Wg}$ is a diagonal matrix which is zero except that $(\mathbf{I}_g)_{ii} = 1$ for $w_i \in \mathcal{G}_g$.

Then Z_C of equation (8) is a Gaussian integral[2]

$$Z_C(\beta, \Xi, \mathcal{H}) = \exp(-\beta C) \frac{(2\pi)^{k/2}}{\sqrt{\det \beta(\mathbf{B} + \sum_g \xi_g \mathbf{I}_g)}} \tag{10}$$

(where C is evaluated at $\mathbf{w}_{\mathrm{MP}}$). Now from (9) we get

$$\begin{aligned} \log P(D \mid \beta, \Xi, \mathcal{H}) = &- \beta(E_D + \sum_g \xi_g E_{Wg}) \\ &- \frac{N}{2} \log \frac{2\pi}{\beta} - \sum_g \frac{k_g}{2} \log \frac{1}{\xi_g} \\ &- \tfrac{1}{2} \mathrm{Tr} \log(\mathbf{B} + \sum_g \xi_g \mathbf{I}_g) \end{aligned} \tag{11}$$

This is the basis for the determination of the scale parameters and the evidence.

---

[2]In the following we use some algebra for symmetric, positive definite matrices: $\log\det \mathbf{B} = \mathrm{Tr} \log \mathbf{B}$ and $(d/ds)\mathrm{Tr} \log \mathbf{B} = \mathrm{Tr}\mathbf{B}^{-1}(d/ds)\mathbf{B}$ and $\int \exp(-\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{B}\mathbf{w}) d^k\mathbf{w} = (2\pi)^{k/2} \frac{1}{\sqrt{\det \mathbf{B}}}$.

Differentiation of the Evidence

The MP noise scale parameter is determined by differentiation of (11):

$$\frac{\partial}{\partial \beta} \log P(D \,|\, \beta, \Xi, \mathcal{H}) = -(E_D + \sum_g \xi_g E_W^g) + \frac{N}{2\beta} \tag{12}$$

Setting this to zero gives:

$$\beta_{\mathrm{MP}} = \frac{N}{2C} \tag{13}$$

We also want the MP values of the weight decay parameters, so we differentiate (11) with respect to ξ_g:

$$\frac{\partial}{\partial \xi_g} \log P(D \,|\, \beta, \Xi, \mathcal{H}) = \frac{k_g}{2\xi_g} - \beta E_{Wg} - \tfrac{1}{2} \mathrm{Tr} \frac{1}{\mathbf{B} + \sum_h \xi_h \mathbf{I}_h} \mathbf{I}_g \tag{14}$$

assuming $\partial \mathbf{B} / \partial \xi_g = 0$[3].

Diagonalising the Hessian

To evaluate (11) and (14) we diagonalise the total Hessian $\mathbf{B} + \sum_g \xi_g \mathbf{I}_g$. We want to do this via diagonalising the Hessian of the data error $\mathbf{B}$. This is possible if all weight decay parameters are equal. We therefore reparametrise the network by performing the following change of variables:

$$\begin{aligned} w &\rightarrow w/s_g \text{ for } w \in \mathcal{G}_g \\ \xi_g &\rightarrow \xi \end{aligned}$$

where the *weight scales* are defined as

$$s_g \equiv \sqrt{\xi / \xi_g} \tag{15}$$

and $\xi \equiv \xi_1$. Notice that the weight scales are introduced only for convenience; they are not new independent variables but are simple given in terms of the ξ_g's through (15). Quantities like $\xi_g E_{Wg}$, E_D, β and $\log P(D \,|\, \beta, \Xi, \mathcal{H})$ are invariant under this reparametrisation.

We diagonalise $\mathbf{B}$ of the reparametrised network. Let the eigenvalues be $\{\lambda_i, i = 1, \ldots, k\}$ with corresponding eigenvectors $\{\mathbf{s}_i\}$. We introduce the matrices $\mathbf{\Lambda} \equiv \mathrm{diag}(\lambda_i) + \xi \mathbf{I}$, and $\mathbf{S} \equiv (\mathbf{s}_1, \ldots, \mathbf{s}_k)$, with $\mathbf{S}^T \mathbf{S} = \mathbf{I}$. From $(\mathbf{B} + \xi \mathbf{I})\mathbf{S} = \mathbf{S}\mathbf{\Lambda}$ we get $\frac{1}{\mathbf{B}+\xi\mathbf{I}} = \mathbf{S}\mathbf{\Lambda}^{-1}\mathbf{S}^T$ so

$$\begin{aligned} \mathrm{Tr} \frac{1}{\mathbf{B} + \xi \mathbf{I}} \mathbf{I}_g &= \mathrm{Tr}(\frac{1}{\mathbf{\Lambda}} \mathbf{S}^T \mathbf{I}_g \mathbf{S}) \\ &= \sum_{i=1}^{k} \frac{1}{\lambda_i + \xi} (\mathbf{S}^T \mathbf{I}_g \mathbf{S})_{ii} \end{aligned}$$

[3] This approximation is not always good. When ξ is increased, $\mathbf{w}_{\mathrm{MP}}$ changes. The Hessian $\nabla\nabla E_D$ is evaluated at $\mathbf{w} = \mathbf{w}_{\mathrm{MP}}$ and depends on $\mathbf{w}$ except when E_D is a quadratic form as in a linear network. The approximation $\partial \mathbf{B} / \partial \xi_g = 0$ is at the heart of MacKay's formula (18) for ξ_{MP}. The approximation can be tested in the following way (in the case of a single weight group): Determine the evidence $P(D \,|\, \xi, \mathcal{H})$ for $\log \xi = \log \xi_{\mathrm{MP}}$ given by (18). Then evaluate the evidences at $\log \xi = \log \xi_{\mathrm{MP}} \pm 4\sigma_{\log \xi}$, remembering that changing ξ requires a new value of $\mathbf{w}_{\mathrm{MP}}$, i.e. the network must be retrained to the new minimum of C. Then fit the evidence to a quadratic function of $\log \xi$ and check whether the maximum occurs close to ξ_{MP}.

The quantity $\pi_g^i \equiv (\mathbf{S}^T \mathbf{I}_g \mathbf{S})_{ii} = \sum_j S_{ji} (I_g)_{jj} S_{ji} = \sum_{j \in \mathcal{G}_g} S_{ji}^2$ is the square of the projection of eigenvector $\mathbf{s}_i$ on the subspace of weight group g.

If the network has converged to a local minimum of C, the eigenvalues of BB should obey $\lambda_i + \xi \geq 0$. An eigenvalue $\lambda_i = -\xi$ corresponds to a zero eigenvalue of $\nabla\nabla C$, which occurs for instance when the cost in weight space is like a ball in a Mexican hat potential: it can move around in the valley without changing the cost. More generally, an eigenvalues λ_i smaller than 0 corresponds to a posterior weight uncertainty which exceeds the prior uncertainty. This is counter–intuitive, and negative eigenvalues are therefore neglected in the end.

Determination of ξ

Setting (14) to zero determines the MP values of ξ_g:

$$2\xi_g \beta E_{Wg} = k_g - \xi_g \mathrm{Tr} \frac{1}{\mathbf{B} + \sum_h \xi_h \mathbf{I}_h} \mathbf{I}_g$$

The first two terms in this equation are invariant under the reparametrisation, therefore the last term must be as well, so we can evaluate it in reparametrised form:

$$\begin{aligned} \xi_g \mathrm{Tr} \frac{1}{\mathbf{B} + \sum_h \xi_h \mathbf{I}_h} \mathbf{I}_g &= \xi \mathrm{Tr} \frac{1}{\mathbf{B} + \xi \mathbf{I}} \mathbf{I}_g \\ &= \sum_i \frac{\xi}{\lambda_i + \xi} \pi_g^i \end{aligned}$$

The MP value of ξ_g is denoted $\xi_{g\mathrm{MP}}$ and using $k_g = \sum_i \pi_g^i$ we get:

$$2\beta \xi_{g\mathrm{MP}} E_{Wg} = \gamma_g \tag{16}$$

where we have introduced the *number of well-determined parameters in weight group g*

$$\gamma_g \equiv \sum_i \frac{\lambda_i}{\lambda_i + \xi_{\mathrm{MP}}} \pi_g^i \tag{17}$$

Negative λ_i's are – as argued above – omitted from this sum. From (13) and (16) we get

$$\frac{\xi_{g\mathrm{MP}} E_{Wg}}{C} = \frac{\gamma_g}{N}, \quad g = 1, \ldots, G \tag{18}$$

This is a main result by MacKay. It states that *the proportion of the cost due to the weight error should equal the number of well-determined weights per training example.* Solving the set of G equations (18) with respect to the ξ_g's, we get the set of equations

$$\xi_{g\mathrm{MP}} = \frac{E_D}{E_{Wg}} \frac{\gamma_g}{N - \sum_h \gamma_h} \tag{19}$$

which are used to reestimate the weight decay parameters (simultaneously) during training.

This completes the determination of the scale parameters. During the derivation we have computed the evidence for $(\beta, \Xi, \mathcal{H})$ which serves as the basis for computing the evidence for $\mathcal{H}$ in section 2.5.

2.3 The Ockham Factors for the Scale Parameters

For a generic scale parameter s (for instance β or ξ_g), the logarithm of the evidence for $(s, \mathcal{H})$ is assumed to be a quadratic form in $\log s$, the natural scale variable:

$$\log P(D\,|\,s, \mathcal{H}) = \log P(D\,|\,s_{\mathrm{MP}}, \mathcal{H}) - \tfrac{1}{2}\frac{(\log s - \log s_{\mathrm{MP}})^2}{\sigma^2_{\log s}}$$

where

$$(\sigma_{\log s})^{-2} = -s^2 \frac{\partial^2}{\partial s^2} \log P(D\,|\,s, \mathcal{H}) \tag{20}$$

at $s = s_{\mathrm{MP}}$, so

$$\begin{aligned} P(D\,|\,\mathcal{H}) &= \int P(D\,|\,s, \mathcal{H}) P(\log s\,|\,\mathcal{H}) d \log s \\ &= P(D\,|\,s_{\mathrm{MP}}, \mathcal{H}) \int \exp(-\tfrac{1}{2}\frac{(\log s - \log s_{\mathrm{MP}})^2}{\sigma^2_{\log s}}) \frac{1}{\log \Omega} d \log s \end{aligned}$$

which gives

$$P(D\,|\,\mathcal{H}) = P(D\,|\,s_{\mathrm{MP}}, \mathcal{H}) \mathrm{Ock}(s) \tag{21}$$

$$\mathrm{Ock}(s) \equiv \frac{\sqrt{2\pi}\sigma_{\log s}}{\log \Omega} \tag{22}$$

We assumed that the prior of the scale is uniform on the logarithmic scale within an interval of length $\log \Omega$ containing the MP value. The parameter Ω is set to 10^3, which is a subjective estimate of the prior ignorance of a scale.

Ock(p) denotes the *Ockham Factor* for a parameter p. It accounts for the uncertainty of the parameter value after it has been tuned to the data, relative to the prior uncertainty:

$$\mathrm{Ock}(p) = \frac{\sigma_{\mathrm{posterior}}(p)}{\sigma_{\mathrm{prior}}(p)}$$

It costs an Ockham Factor to *marginalise* (i.e. "integrate away") a parameter from the evidence. A parameter which is determined well by the data gives a small Ockham Factor and hence a large penalty in the evidence. It may seem unfair to punish a model for the accuracy of its parameters when these are estimated from the data. But, in fact, it agrees with common sense. We don't like explanations which are accurately tuned to the phenomena in question. We regard them as improbable or contrived.

From (20), (12) and (14) we now find[4]

$$\sigma_{\log \beta} = \sqrt{\frac{2}{N}} \tag{23}$$

$$\sigma_{\log \xi_g} \approx \sqrt{2/\gamma_g} \tag{24}$$

[4] A more accurate determination of the posterior volume of the scale parameters is obtained by taking the correlations between the posterior uncertainties of the scale parameters into account. However, the accuracy of (23) and (24) is sufficient for computing the evidence given the inaccuracy of the Gaussian approximation.

2.4 Symmetries of the Network

When we evaluated Z_C in equation (10) we did not take into account that a minimum of C at $\mathbf{w}$ can be obtained with other $\mathbf{w}$'s related to $\mathbf{w}$ by two kinds of symmetries:

- By changing the sign of all incoming and outgoing weights of a hidden unit, we get an identical mapping (we use a hidden unit activation function which is symmetric about zero). With h hidden units this gives 2^h equivalent realisations of the solution.
- The hidden units can be permuted according to the symmetry group of the network topology. For a fully connected network with h hidden units in one hidden layer this gives $h!$ times as many realisations of the solution.

The Gaussian integral in Z_C measures the posterior $\mathbf{w}$ volume while the Gaussian integral Z_W measures the prior $\mathbf{w}$ volume. The ratio of these volumes is the Ockham Factor for $\mathbf{w}$. Since the prior volume Z_W includes all the symmetric solutions, the posterior volume must as well, so Z_C should be multiplied by the factor

$$2^h h! \tag{25}$$

for a fully connected network with one hidden layer.

The solution itself possesses a symmetry if two hidden units have identical weights to and from the unit, or in general whenever the posterior distribution for the weights for two units overlap to a significant degree. Then the factor (25) should be reduced accordingly. This complicates the computation of the evidence and may even deteriorate the generalisation ability. Pruning can be used to remedy this problem [1].

2.5 Collecting the terms of the Evidence

We are now ready to put everything together and compute the evidence of the model. The term $-\sum_g \frac{k_g}{2}\log\frac{1}{\xi_g} - \frac{1}{2}\mathrm{Tr}\log(\mathbf{B}+\sum_g \xi_g \mathbf{I}_g)$ in (11) is reparametrisation invariant since the rest of the equation is invariant. Therefore we can evaluate it in reparametrised form:

$$\begin{aligned}
-\sum_g \frac{k_g}{2}\log\frac{1}{\xi_g} - \tfrac{1}{2}\mathrm{Tr}\log(\mathbf{B}+\sum_g \xi_g \mathbf{I}_g) &= \frac{k}{2}\log\frac{1}{\xi} - \tfrac{1}{2}\mathrm{Tr}\log(\mathbf{B}+\xi\mathbf{I}) \\
&= \tfrac{1}{2}\sum_i \log\frac{\xi}{\lambda_i+\xi}
\end{aligned}$$

The evidence for $\mathcal{H}$, denoted $\mathrm{Ev}(\mathcal{H})$, is obtained from the evidence for $(\beta, \Xi, \mathcal{H})$ (11) by marginalisation of β and ξ_g. How to do this was derived in (21) and (22), and the posterior uncertainties of the scale parameters were computed in (23) and (24). Using (13) we obtain:

$$\begin{aligned}
\log \mathrm{Ev}(\mathcal{H}) \equiv \log P(D\,|\,\mathcal{H}) &= -\frac{N-\gamma}{2} - \frac{N}{2}\log\frac{4\pi C}{N} \\
&\quad + \log \mathrm{Ock}(\mathbf{w})
\end{aligned}$$

$$
\begin{aligned}
& + \log \mathrm{Ock}(\beta) + \sum_g \log \mathrm{Ock}(\xi_g) & (26) \\
\log \mathrm{Ock}(\mathbf{w}) \quad = \quad & \tfrac{1}{2} \sum_{i=1}^{k} \log \frac{\xi}{\xi + \lambda_i} - \frac{\gamma}{2} & \\
& + \log h! + h \log 2 & (27) \\
\mathrm{Ock}(\beta) \quad = \quad & \frac{\sqrt{4\pi/N}}{\log \Omega} & (28) \\
\mathrm{Ock}(\xi_g) \quad = \quad & \frac{\sqrt{4\pi/\gamma_g}}{\log \Omega} & (29) \\
\gamma \quad \equiv \quad & \sum_h \gamma_h & (30)
\end{aligned}
$$

As in the simple expression (2), the evidence consists of two parts, the best fit likelihood $-\frac{N-\gamma}{2} - \frac{N}{2} \log \frac{4\pi C}{N}$, and the Ockham Factors.

The Ockham (or simplicity) Factor for the weights $\mathrm{Ock}(\mathbf{w})$ is large when the eigenvalues λ_i are small, corresponding to poorly-determined weights. Eigenvalues close to or smaller than $-\xi$ are called *zero-modes*. They render the Ockham Factor for $\mathbf{w}$ divergent. Therefore, as in (17), the negative eigenvalues are neglected[5].

3 Further topics of Bayesian Backprop

3.1 Use of Validation

A critical issue in the application of the evidence for selection of models is its high correlation with the test error. It is recommended to check this relation in all applications. This is done by dividing the data analysis into two phases.

In the first phase the available data are split into a training and a test set. The Bayesian analysis is done on the training set and the evidences of the various models are computed and compared with the error on the test set. This can reveal inconsistencies in the models and help cure them. The goal of the first phase is to obtain a high correlation between the evidence and the test error.

In the second phase the data sets are merged to a single training set, and the final set of models is developed. Now the best model – or a committee of the best models – can be selected with confidence according to the evidence.

The first phase not only improves the model space, it is also a test of the basic assumption that high evidence implies good generalisation. Notice that

[5] A more detailed treatment of the zero-modes is obtained by introducing a variable cut-off Z: If an Ockham Factor $\sqrt{\frac{\xi}{\xi+\lambda_i}}$ exceeds Z it is truncated to Z. Neglect of the negative eigenvalues corresponds to $Z = 1$. The uncertainty of the evidence from the zero-modes can be estimated by evaluating the evidence $\mathrm{Ev}(Z)$ for different values of Z: Uncertainty $= |\frac{1}{2}(\log \mathrm{Ev}(2Z) - \log \mathrm{Ev}(Z/2))|$. A large uncertainty indicates that the Gaussian approximation is inaccurate. The zero-modes are partly due to numerical problems, e.g. computing the Hessian from finite differences of gradients. But the zero-modes are also partly real, indicating wide posterior weight distributions. We still want to cut these off at some point because they lead to overlapping posterior distributions for two units, which is in conflict with the factor (25).

cross validation is not used in the selection of the final model, but rather to validate the approach in general.

3.2 A Committee of Networks

For a given data set we usually train several networks with different numbers of hidden units and different initial weights. Several of these networks have evidence near or at the maximal value, but the networks differ in their predictions. This is a source of distress to some: How can neural networks be useful if they don't agree among themselves? However, the probabilistic (or Bayesian) interpretation naturally turns this ambiguity into an advantage. The different solutions are interpreted as components of the posterior distribution, and the correct Bayesian answer is obtained by averaging over the solutions weighted by their posterior probabilities, i.e. their evidences.

At this point it is important to take into account the uncertainty of the computed evidence as a measure of our posterior belief in the particular solution. The total uncertainty is modelled as a normally distributed noise on $\log \mathrm{Ev}$ with dispersion σ. The primary source of uncertainty is the Gaussian approximation.

A consequence of this uncertainty is that the $\log \mathrm{Ev}$ ranking must be considered a random permutation of the "true" ranking. Integration with Monte Carlo technique over these random shifts we find that the probabilistic weight of a solution $\mathcal{H}$ to a good approximation is given by

$$p(\mathcal{H}) \propto \exp(\log \mathrm{Ev}(\mathcal{H})/f) \qquad (31)$$

with

$$f^4 = 1 + (\sigma/2)^4 \qquad (32)$$

which for large σ reduces to $f = \sigma/2$.

As a practical approximation to the use of all the solutions with weights given by (31) we define a *committee* as the set of networks $\mathcal{H}$ with $\log \mathrm{Ev}(\mathcal{H}) > \max \log \mathrm{Ev}(\mathcal{H}) - f$, where $\max \log \mathrm{Ev}(\mathcal{H})$ is the maximal log evidence among the solutions.

This paper does not present a Bayesian approach to select σ or f. They can be determined by various methods outside the framework. A practical approach is to determine f in the first phase of the analysis described in section 3.1. There the plot of the test set error versus log evidence is available and f can be defined as the spread of evidence for networks with approximately the same generalisation error. This value of f is then used in the final phase, where all the data are used for training. Another approach is to use the best quarter of the solutions.[6].

The committee gives two advantages.

- The predictions of the committee, i.e. the average prediction of the committee gives a better generalisation than the average network in the committee [14].

[6] The uncertainty on the log evidence due to zero-modes mentioned in footnote 5 is another way to estimate σ.

- The degree of dissent within the committee contributes to the uncertainty of the predictions. By including this we get more reliable error bars on the predictions. This is treated in section 3.5 [9].

To shed light on the first item we compute the evidence $\mathrm{Ev}(\mathcal{C})$ of the committee in terms of the average log evidence of the members $\overline{\log \mathrm{Ev}(\mathcal{H})}$. Assume for simplicity that all networks in the committee $\mathcal{C}$ have the same architecture. We estimate the number $N_{\mathcal{C}}$ of truly different solutions in the committee. Of course we count symmetric realisations only once. The posterior volume, i.e. the Ockham Factor, for the weights is now $N_{\mathcal{C}}$ times larger. Furthermore, a committee with a suitably chosen f typically gives a better fit to the data, i.e. it has lower E_D, so let $\Delta E_D = \overline{E_D(\mathcal{H})} - E_D(\mathcal{C})$ denote the difference between the average E_D of the members and the E_D of the committee. Then the likelihood is $\exp(\beta \Delta E_D)$ times larger for the committee. So both the Ockham factor and the best fit likelihood are larger for the committee than for the average member, and the evidence of the committee is:

$$\log \mathrm{Ev}(\mathcal{C}) \approx \overline{\log \mathrm{Ev}(\mathcal{H})} + \log N_{\mathcal{C}} + \beta \Delta E_D \tag{33}$$

Since the evidence is correlated with the generalisation error, we expect the committee to generalise better than the average committee member. This is confirmed by the simulations in section 4.2.

The expression (33) is mainly intended as a formal statement of the situation where we have a greater belief in the committee than in a single model. The expression was not used specifically, e.g. to optimise f.

3.3 The Meaning of the Evidence

Empirically there is a high correlation between the evidence and the generalisation error. According to common sense, it is not surprising that the most probable model of the data is the best at making predictions in the same domain and indeed the correlation is usually higher at the upper end of the evidence scale. But we have not deduced this relation.

Section 3.1 recommended to alter the set of models to maximise the correlation between evidence and test error (especially at high evidence) in the first phase of the analysis, but there are several reasons for this correlation to be imperfect. They are presented here together with some means to examine or overcome them:

- If the test set is small, it gives an uncertain estimate of the true test error. This effect can be eliminated by using a test set which is much larger than the training set.

- Some models contain assumptions which are in conflict with the data, e.g. the weights or the residual errors are not Gaussian. The evidence of such models is underestimated. This situation calls for a modification of the models. It is a symptom that the best model may not have been found yet, due to wrong assumptions in the modelling.

  It is however acceptable to include some inconsistent models as long as the correct models are represented as well. In the end we select the models with highest evidence. The inconsistent models are shifted towards lower values of evidence and they are therefore unlikely to be chosen.

- Often the training set is limited so that the model parameters are poorly estimated. This is a general condition of inference from small samples. The evidence is particularly well suited to the problem of model selection under these circumstances, because it encompasses an exact, universal and consistent treatment of the overfitting problem.

- There are approximations in the evidence which may be more or less accurate, in particular approximating E_D as a quadratic form near $\mathbf{w}_{\rm MP}$ (the Gaussian approximation). In addition the MP solution may not be representative of the posterior distribution, i.e. the value of a predicted quantity at the mode (the maximum) is not always a good approximation to the integral over the posterior distribution. To overcome these problems Neal has developed a Monte Carlo technique [15].

- The evidence as a quality measure could reflect a mixture of virtues, of which the generalisation error is just one. Other virtues could be correct architecture, explanatory power or correct estimate of the error bars.

3.4 Uncertainties on the Four Levels of Inference

A key concept of Bayesian inference is to account for the posterior uncertainty and to integrate over it when making predictions. This is discussed in terms of the four levels of inference introduced in section 1.4:

Level 1 Make predictions including error bars for new input data.

Level 2 Estimate the weight parameters and their uncertainties.

Level 3 Estimate the weight decay parameters and their uncertainties

Level 4 Repeat level 2 and 3 with different initial conditions and different network architectures. Select a particular architecture and a particular $\mathbf{w}$-minimum. Optionally select a committee to reflect the uncertainty on this level.

Large uncertainties on level 2–4 are reflected by large Ockham (or simplicity) factors. Large uncertainty on level 1, i.e. large $1/\beta$, gives small likelihood. Since the evidence is the product of the Ockham factors and the likelihood, we draw the initially counter–intuitive conclusion that large uncertainties on level 2–4 gives *large* evidence while large uncertainty on level 1 gives *small* evidence.

On level 4, when we talk about a model $\mathcal{H}$ we mean a particular architecture (and prior) and the environment of a particular minimum $\mathbf{w}_{\rm MP}$. Uncertainty at this level corresponds to the presence of several minima and architectures with non-negligible evidence. The integration over this uncertainty corresponds to the use of a *committee* as described in section 3.2.

Level 3 deals with the inference of the the scale parameters β and ξ_g. It was found that the posterior distribution of the scale parameters has a narrow peak if there are many observations and parameters (equations (24) and (23)), i.e. the uncertainty at this level is relatively small.

Level 2 determines the weight parameters. The Bayesian weights are given by the posterior distribution $P(\mathbf{w}\,|\,D,\mathcal{H})$. This is a complicated entity and we seek a compact description in terms of the centre of gravity of the relevant part of it, and some uncertainty. MacKay has clarified [11] that the true maximum posterior value is a poor approximation to the centre of gravity. The desired description is instead obtained by introducing the extra level 3 of inference where the MP scales are determined, and then on level 2 defining $\mathbf{w}_{\mathrm{MP}}$ as the maximum of $P(\mathbf{w}\mid\beta_{\mathrm{MP}},\Xi_{\mathrm{MP}},D,\mathcal{H})$. The uncertainty is given by the Hessian evaluated at $\mathbf{w}_{\mathrm{MP}}$.

Level 1 deals with the predictions of the model on new data - the purpose of the modelling. For a given value of the weights $\mathbf{w}$ the network implements a mapping of input $\mathbf{x}$ onto a prediction $y(\mathbf{x};\mathbf{w})$. This mapping is an estimate of $E(t\,|\,\mathbf{x})$, the *regression* of t on $\mathbf{x}$. As described in section 2.1 our modelling assumes that the actual output t is obtained by adding to $y(\mathbf{x};\mathbf{w})$ an independent Gaussian noise term ν with variance $1/\beta_{\mathrm{MP}}=\sigma_\nu^2$, which describes the uncertainty due to the presumed noise process.

3.5 Bayesian Error Bars on Predictions

When making predictions we use not only the MP weight $\mathbf{w}_{\mathrm{MP}}$, but also take into account the posterior distribution of weights given in (6):

$$\begin{aligned} P_{\mathrm{posterior}}(\mathbf{w}) &\propto \exp(-\beta_{\mathrm{MP}}C) \\ &= \exp(-\tfrac{1}{2}(\mathbf{w}-\mathbf{w}_{\mathrm{MP}})^{\mathrm{T}}\beta_{\mathrm{MP}}(\mathbf{B}+\xi_{\mathrm{MP}}\mathbf{I})(\mathbf{w}-\mathbf{w}_{\mathrm{MP}})) \qquad (34) \end{aligned}$$

where $\mathbf{B}=\nabla\nabla E_D$ is the Hessian of the data error and *the weights are scaled* such that all weight decay parameters are equal to ξ_{MP}. Eqn.(34) is a k-dimensional normal distribution with mean $\mathbf{w}_{\mathrm{MP}}$ and covariance matrix $\mathbf{D}=(\beta_{\mathrm{MP}}(\mathbf{B}+\xi_{\mathrm{MP}}\mathbf{I}))^{-1}$. The mean over the posterior weight distribution of the network prediction is denoted $y(\mathbf{x})$ and the variance of the network prediction due to the weight uncertainty (WU) is denoted $\sigma_{\mathrm{WU}}(\mathbf{x})^2$. These are given by:

$$\begin{aligned} y(\mathbf{x}) &= \int y(\mathbf{x};\mathbf{w})P_{\mathrm{posterior}}(\mathbf{w})d\mathbf{w} \\ \sigma_{\mathrm{WU}}(\mathbf{x})^2 &= \int (y(\mathbf{x};\mathbf{w})-y(\mathbf{x}))^2 P_{\mathrm{posterior}}(\mathbf{w})d\mathbf{w} \end{aligned}$$

To compute these integrals we assume that the network predictions depend *linearly on* $\mathbf{w}$ within the bulk of the posterior $\mathbf{w}$ distribution:

$$y(\mathbf{x};\mathbf{w}) = y(\mathbf{x};\mathbf{w}_{\mathrm{MP}}) + \mathbf{g}(\mathbf{x})\cdot(\mathbf{w}-\mathbf{w}_{\mathrm{MP}})$$

where $\mathbf{g}(\mathbf{x})=\frac{\partial y(\mathbf{x};\mathbf{w}=\mathbf{w}_{\mathrm{MP}})}{\partial\mathbf{w}}$. Then the average prediction equals the prediction at $\mathbf{w}_{\mathrm{MP}}$: $y(\mathbf{x})=y(\mathbf{x};\mathbf{w}_{\mathrm{MP}})$.

The WU variance of the prediction is computed by introducing the variables $\mathbf{u}\equiv\mathbf{w}-\mathbf{w}_{\mathrm{MP}}$ and the normalisation constant $a=\sqrt{(2\pi)^k\det\mathbf{D}}$:

$$\sigma_{\mathrm{WU}}(\mathbf{x})^2 = \int (\mathbf{g}(\mathbf{x})\cdot\mathbf{u})^2\frac{1}{a}\exp(-\tfrac{1}{2}\mathbf{u}^{\mathrm{T}}\frac{1}{\mathbf{D}}\mathbf{u})d\mathbf{u}$$

$$
\begin{aligned}
&= \sum_{i=1}^{k}\sum_{j=1}^{k} g_i(\mathbf{x})g_j(\mathbf{x}) \int u_i u_j \frac{1}{a}\exp(-\tfrac{1}{2}\mathbf{u}^{\mathrm{T}}\frac{1}{\mathbf{D}}\mathbf{u})d\mathbf{u} \\
&= \sum_{i=1}^{k}\sum_{j=1}^{k} g_i(\mathbf{x})g_j(\mathbf{x})D_{ij} \\
&= \mathbf{g}(\mathbf{x})\cdot\mathbf{D}\mathbf{g}(\mathbf{x}) \qquad (35)
\end{aligned}
$$

The gradient $\mathbf{g}(\mathbf{x}) = \frac{\partial y(\mathbf{x};\mathbf{w}=\mathbf{w}_{\mathrm{MP}})}{\partial \mathbf{w}}$ is well-known as a building block of the gradient of the cost function computed with backpropagation. To compute $\sigma_{\mathrm{WU}}(\mathbf{x})^2$ we need $\mathbf{D}$, the inverse of $\mathbf{B}+\xi_{\mathrm{MP}}\mathbf{I}$. In order for this to be well-defined we set negative eigenvalues of $\mathbf{B}$ to zero corresponding to the zero-mode cut-off in section 2.5. The explicit formula is given in the recipe in section 4.1.

Thus the WU variance is obtained by propagating the uncertainty on level 2 to level 1. The uncertainty on level 3 is believed to have negligible influence on level 1. Finally let us propagate the uncertainty on level 4 to level 1. The committee prediction is the average of the member predictions. For a committee of networks of the same architecture the members are defined by their respective MP weights: $\mathbf{w}_{\mathrm{MP}}^j, j = 1, \ldots, N_{\mathcal{C}}$ so the average prediction is

$$
y(\mathbf{x}) = \frac{1}{N_{\mathcal{C}}}\sum_{j=1}^{N_{\mathcal{C}}} y(\mathbf{x}, \mathbf{w}_{\mathrm{MP}}^j) \qquad (36)
$$

The committee members disagree to a certain extent and this committee uncertainty (CU) gives the following contribution to the prediction variance:

$$
\sigma_{\mathrm{CU}}(\mathbf{x})^2 = \frac{1}{N_{\mathcal{C}}-1}\sum_{j=1}^{N_{\mathcal{C}}}(y(\mathbf{x}) - y(\mathbf{x}, \mathbf{w}_{\mathrm{MP}}^j))^2 \qquad (37)
$$

The total prediction variance is

$$
\sigma_{\mathrm{total}}(\mathbf{x})^2 = \sigma_\nu^2 + \sigma_{\mathrm{WU}}(\mathbf{x})^2 + \sigma_{\mathrm{CU}}(\mathbf{x})^2 \qquad (38)
$$

where σ_ν^2 and $\sigma_{\mathrm{WU}}(\mathbf{x})^2$ are understood to be averaged over the committee. The significance of these contributions is illustrated with the application (see figure 2).

An apparent ambiguity with the WU error bars is the zero-mode cut-off. This could underestimate the error bars. However, this potential loss is remedied by the CU error bars according to the following argument: A zero-mode indicates a poorly defined minimum. When several networks are trained it is likely that the committee members populate different parts of such a flat "basin of attraction". Limiting the extent through the zero-mode cut-off is therefore allowed *if* the CU contribution to the error bar is included. In fact some cut-off must be performed in order not to double-count the "flat basin" uncertainty.

4 Simulations

Here we give a recipe suitable for Bayesian backprop for networks requiring a minimum of programming. The recipe was used for all the simulations in this work, which required 3 days' computer time on a Sun SPARCstation 2 performing 1 Mflop per second.

4.1 Recipe for Bayesian Weight Decay

The simulations use the unscaled weights $\mathbf{w}$. There are k weights divided into G groups with k_g in each. Some quantities (the Hessian in step 8 and the gradient for the error bars in equation (41)) need to be computed in the scaled weights, but rather than introducing the scaled weights explicitly, it is convenient to introduce the G weight scales s_g, which are completely dependent on the G weight decay parameters ξ_g through $s_g = \sqrt{\xi/\xi_g}$.

1. Centre each of the inputs and scale them to standard deviation $\frac{1}{4}$. This is done to eliminate the potentially arbitrary units and offsets of the individual inputs. It is an "uninformed" normalisation.

2. Centre each of the outputs and scale them to unit standard deviation. Alternatively use the output unit activation function $y(h) = ah + b$ with a and b set to the empirical standard deviation and mean of the targets. Steps 1 and 2 adjust the problem to the natural dynamic range of network.

3. Define the *error scale* E equal to the expected standard training error $\sqrt{\frac{2E_D}{N}}$ based for instance on a previous simulation or a linear model.

4. Set the initial values of all the weight decay parameters $\xi_{\mathrm{MP}} = \xi_{g\mathrm{MP}} = E^2/4$. Initialise the weights to random values in the interval $(-\frac{1}{2}, \frac{1}{2})$. The small initial weights ensure the best convergence of the network. The initial weight decay parameters correspond to "mature" weights, $\sigma(w/s_g) = 2$ according to (18). This allows the weights to "grow up" in the next step.

5. Select a total number of training epochs T and train the network for $T/6$ epochs, i.e. perform a gradient descent towards a minimum of C. An epoch is one traversal of the training set; we typically use $T = 10,000$. The learning rate was set dynamically as in [16].

6. Initialise the number of well-determined parameters: $\gamma_g = \frac{9}{10}k_g$ for $g = 1, \ldots, G$.

7. Train the network for T/6 epochs. The weight decay parameters are re–estimated according to (19) after each epoch:

$$\xi_{g\mathrm{MP}} = \frac{E_D}{E_{W_g}} \frac{\gamma_g}{N - \sum_h \gamma_h}$$

8. This step computes the Hessian of E_D in the scaled weights by finite differences of gradients. Set the weight scales

$$s_g = \sqrt{\xi_{\mathrm{MP}}/\xi_{g\mathrm{MP}}}$$

   where $\xi_{\mathrm{MP}} \equiv \xi_{1\mathrm{MP}}$. Let $\mathbf{q}(\mathbf{w}) = \frac{\partial E_D(\mathbf{w})}{\partial \mathbf{w}}$ be the gradient of the data error at $\mathbf{w}$ (notice that $\mathbf{w}$ are the unscaled weights), and $\mathbf{e}_i$ a unit vector in the

i'th weight direction and let $s_{g(i)}$ be the weight scale of weight i, which is in group $g(i)$. The ij'th element of the Hessian is computed as:

$$B_{ij} = \frac{q_j(\mathbf{w} + \epsilon s_{g(i)}\mathbf{e}_i) - q_j(\mathbf{w} - \epsilon s_{g(i)}\mathbf{e}_i)}{2\epsilon} s_{g(j)} \tag{39}$$

It takes $2k$ epochs to compute the Hessian in this way[7]. If the step length ϵ is set too high we run into deviations from the quadratic behaviour of E_D, and if it is too small the numerical precision becomes critical, so the calculations must often be performed with double precision. The finite difference is symmetric around $\mathbf{w}$; an asymmetric difference, which is computationally twice as fast, is not sufficiently accurate.

The exact Hessian is symmetric, and the computed Hessian is symmetrised by replacing $\mathbf{B}$ by $\frac{1}{2}(\mathbf{B} + \mathbf{B}^T)$ (The asymmetry of the Hessian is an efficient way to check the precision of the computation). Finally $\mathbf{B}$ is diagonalised using the Jacobi method [18], which requires approximately $24k^3$ floating-point operations. This renders the recipe unpractical for networks considerably larger than 1000 connections[8].

9. Compute the number of well-determined parameters in group g:

$$\begin{aligned} \pi_g^i &= \sum_{j \in \mathcal{G}_g} S_{ji}^2 \\ \gamma_g &= \sum_i \frac{\lambda_i}{\lambda_i + \xi_{\mathrm{MP}}} \pi_g^i \end{aligned}$$

where S_{ji} is the j'th component of the eigenvector corresponding to λ_i. Negative eigenvalues are left out.

10. Compute the evidence of the model from (26-29):

$$\begin{aligned} \log \mathrm{Ev}(\mathcal{H}) = & -\frac{N}{2} - \frac{N}{2} \log \frac{4\pi C}{N} \\ & + \frac{1}{2} \sum_{i=1}^{k} \log \frac{\xi_{\mathrm{MP}}}{\xi_{\mathrm{MP}} + \lambda_i} \\ & + \log h! + h \log 2 \\ & + \log \frac{\sqrt{4\pi/N}}{\log \Omega} + \sum_g \log \frac{\sqrt{4\pi/\gamma_g}}{\log \Omega} \end{aligned} \tag{40}$$

where $\Omega = 10^3$ and h is the number of hidden units. There is a term $\log h! + h \log 2$ for each hidden layer. Negative eigenvalues are left out.

11. Steps 7 to 10 are repeated 5 times.

[7] The Hessian can be evaluated analytically using an extension to backprop developed by Bishop[17]. This involves some elaborate programming but then the Hessian is evaluated in just h epochs.

[8] This is according to the following argument: Assume that the number of cases is proportional to the number of weights, $N = 5k$; then the training time grows as k^2. With $k = 1000$ weights, 50% of the computer time is spent with the diagonalisation.

A *quick and dirty Bayesian weight decay procedure* [3] is obtained by skipping the Hessian and evidence calculations, i.e. step 8–10. Such a procedure is useful even for very large networks.

The noise parameter β_{MP} does not enter the training but can be computed by $\sigma_\nu^2 = 1/\beta_{\text{MP}} = 2C/N$.

The WU error bars on the predictions can be computed for an arbitrary $\mathbf{x}$. The gradient with respect to the unscaled weights $\mathbf{g}(\mathbf{x}) = \frac{\partial y(\mathbf{x};\mathbf{w}=\mathbf{w}_{\text{MP}})}{\partial \mathbf{w}}$ is computed with backpropagation. The gradient with respect to the *scaled* weights $g'_j(\mathbf{x}) = s_{g(j)} g_j(\mathbf{x})$ is projected onto the eigenvectors $\mathbf{s}_i$ to obtain the WU error bar according to (35):

$$\sigma_{\text{WU}}(\mathbf{x})^2 = \sigma_\nu^2 \sum_{i=1}^{k} \frac{(\mathbf{s}_i \cdot \mathbf{g}'(\mathbf{x}))^2}{\lambda_i + \xi_{\text{MP}}} \tag{41}$$

Negative λ_i's are set to zero.

The Hessian can be computed approximately by neglecting the nonlinearities of the sigmoid functions:

$$\mathbf{B} \approx \sum_{i=1}^{N} \mathbf{g}'(\mathbf{x}_i)\mathbf{g}'(\mathbf{x}_i)^{\text{T}} \tag{42}$$

where $\mathbf{x}_i$ is the input for training case i. This is sufficiently accurate for computing error bars and the number of well-determined parameters, but not for the evidence [3].

For a committee the CU error bars are computed as the variance of the committee predictions, see (37) and (38).

4.2 Application to Spectroscopic Data

We now apply Bayesian Backprop to a real-life application from the meat industry. The data were recorded by a Tecator near-infrared spectrometer (the Infratec Food and Feed Analyzer) which measured the spectrum of light transmitted through a sample of minced pork meat [19]. The spectrum consists af the absorbances at 100 wavelengths in the region 850-1050 nm. We want to calibrate the spectrometer to determine the fat content from the spectrum. The target values of the fat content are determined by wet-chemistry (the Soxhlet method) and ranges from 2 to 59%. The spectra are pre-processed using principal component analysis. The first 10 principal components are used as input to a neural network.

We give the error of a network as the *standard error*, i.e. the square root of the mean squared error. By SEP we denote the standard error of prediction.

In an earlier work [19] we have analysed these data with a linear model and with neural networks. The data were divided according to table 1. The data available for model estimation were split into a set C used for training and a "monitoring" set M. The error on M was monitored during training and the hitherto best network was saved. Several networks were trained and finally the network with the best error on M was selected. This training technique, denoted "early stopping", is rather successful for many applications, as it is an easy way to cope with the problem of overfitting [20]. The optimal architecture

Table 1: *The spectroscopic data sets and their use when training networks with early stopping*

Data set	Use	Number of cases
C	Training	129
M	Monitoring	43
T	Testing	43
E1	Extrapolation, Fat	8
E2	Extrapolation, Water	17

Table 2: *The Standard Error of Prediction on the data set T.*

Model	Model selection	SEP
Linear regression	F-test	2.78
Quadratic regression	F-test	0.80
Neural network	Early Stopping	0.65
Neural network (section 5)	GPE	0.60
Neural network	Evidence	0.55
Neural network committee	Evidence	0.52
ARD (section 7)	Evidence	0.36

had six hidden units and direct connections from input to output and the inputs were the first 10 principal components.

In this section we demonstrate that the Bayesian method outperforms early stopping and that a committee of networks improves the model even further. Table 2 summarises the results.

For the Bayesian analysis we combine the data sets C and M for the training. As in the earlier work we use networks with direct connections and we use 10 principal components based on a principal component analysis on the data set C. Three separate weight decay parameters are used: one for the weights and biases of the hidden layer, one for the connections from the hidden to the output layer, and one for the direct connections from the inputs to the output as well as the output bias. Like earlier, T is used only to compute SEP in table 2.

The relation between test error and log evidence is shown in figure 1[9]. We see that networks with 3 hidden units are most frequent among the networks with the highest evidence, so we select the 12 networks with 3 hidden units which have the highest evidence (larger than -270) for further investigation. The average SEP is 0.55, a 15% reduction relative to the method of early stopping (see table 2). There are several reasons for this improvement:

[9]One source of uncertainty in the evidence is the dependence on the cut-off Z mentioned in footnote 5. The increase of log evidence when Z is doubled is 1.5 for a network with 3 hidden units and 8 for a network with 6 hidden units. Thus the evidences of the larger networks are poorly defined. The networks with 4 hidden units have approximately $N/k = 3$, so we do expect the Gaussian approximation to break down for networks larger than that. Neal argues [12] that the larger networks may well be the more correct models, but we are reluctant to use them because working within the Gaussian aprroximation we cannot treat them reliably.

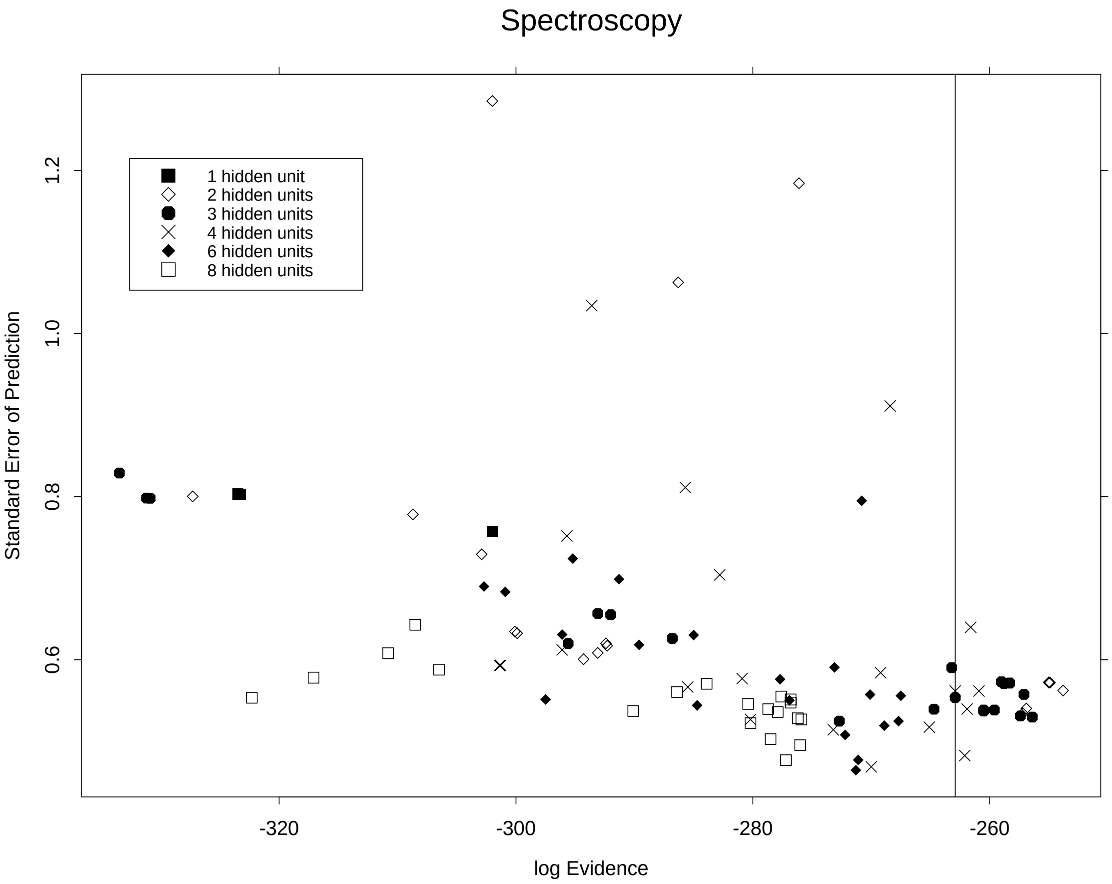

Figure 1: *The test error as a function of the log evidence for networks trained on the spectroscopic data. Networks with 2 and 3 hidden units attain the highest evidence. The 20 networks with highest evidence are delimited by the vertical line and have on average SEP=0.55.*

- The optimisation of three separate weight decay parameters gives a more careful regularisation of the complexity than early stopping.
- The use of C+M for training rather than C gives a larger training set.
- The evidence is a less noisy model selection criterion than the error on the monitoring set [2].
- The longer training times give a more extensive search for good networks than the shorter training times used with early stopping.

To improve the model further we form a committee of the 12 networks with 3 hidden units favoured by the evidence. This committee gives a further 6% reduction of the test error to SEP = 0.52. Notice that the size of the committee is determined in an ad hoc (non–Bayesian) manner.

Quadratic regression is a standard statistical method used for this kind of data. All products of pairs of the 10 principal components are included as 55 new input variables. The 65 variables are used as independent variables in

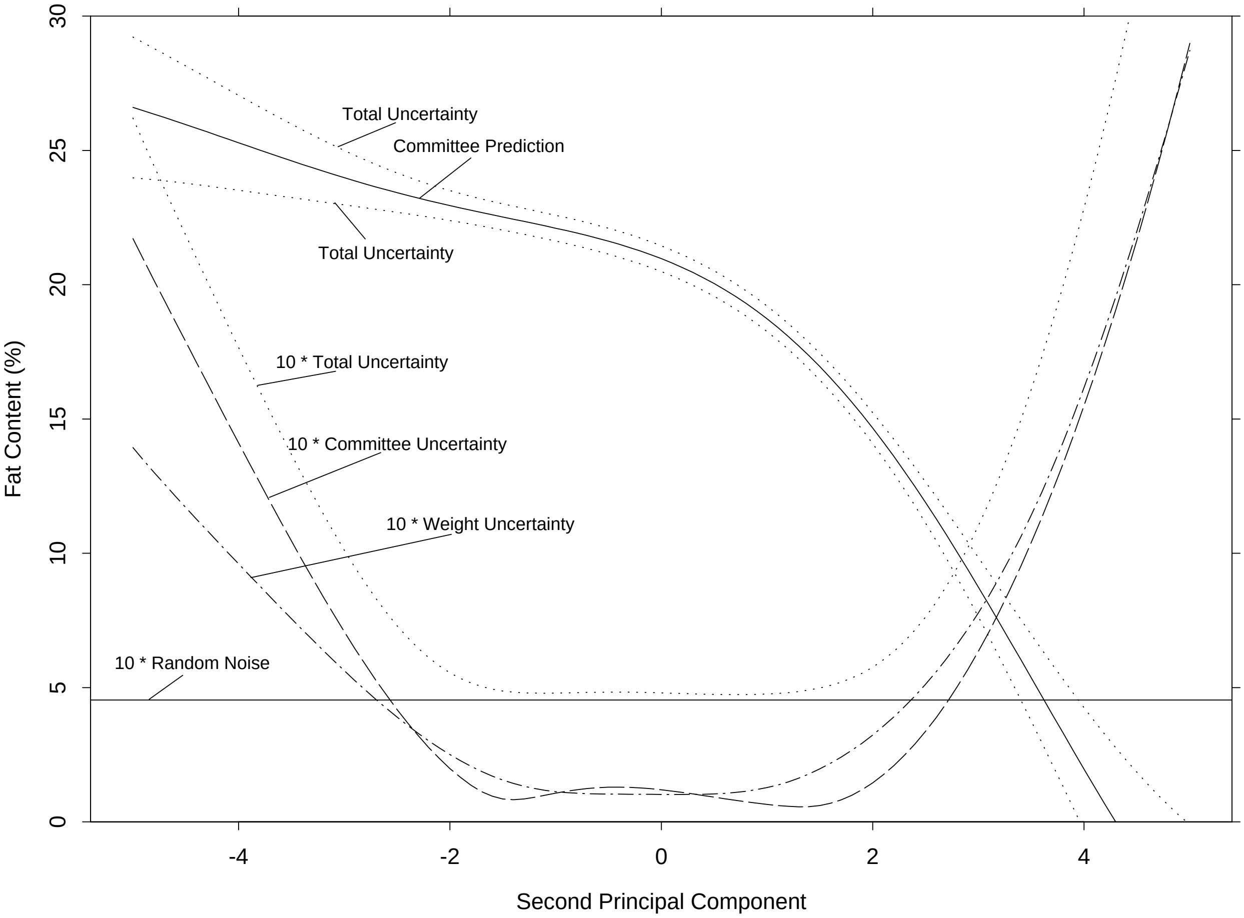

Figure 2: *Prediction of the fat content as a function of the second principal component p_2 of the NIR spectrum. 95% of the training data has $|p_2| < 2$. A committee of 12 networks with 3 hidden units was used. The total standard error bars are indicated by a "1 sigma" band with the dotted lines. The total standard errors and the standard errors of its contributions are shown separately, multiplied by a factor of 10.*

a stepwise regression (see e.g. [13]) using the data C+M. Model selection is performed with the F-test which tests the significance of adding or removing a variable in the model. This selects a model with 42 variables and SEP = 0.80. Hence quadratic regression is considerably less accurate than neural networks.

4.3 Error Bars for the Spectroscopic Data

The Bayesian error bars are illustrated for the spectroscopic data in figure 2. We study the model predictions along a line through input space defined by the second principal component axis, i.e. the second input is varied and all other inputs are set at zero. The principal component is scaled to unit variance so the training data has standard deviation 1 in this projection. The predictions within $(-1, +1)$ can be considered interpolations and as we go further away from 0 we are extrapolating to a larger and larger degree.

Figure 2 shows that the uncertainty increases dramatically when the model

Table 3: *The predicted and observed SEP (Standard Error of Prediction) on test samples grouped according to the predicted error bars (see figure 3). The observed SEPs are given with standard errors computed from χ^2-statistics.*

σ_{total}	Predicted SEP	Observed SEP
0.00–0.55	0.50	0.46±0.05
0.55–0.75	0.59	0.59±0.10
0.75–∞	1.01	1.80±0.42

is used outside the range of the training data. On the same figure are shown the total error amplified by a factor of 10 and its break-down into the three components, see eqn. (38)[10]. The random noise is constant, while the committee and weight uncertainties vary with input and are of the same order of magnitude. The committee uncertainty typically dominates for large extrapolations, but in general the contributions vary in a complicated manner. The general CU/WU proportion was found to be rather insensitive to the number of hidden units.

We now compare the computed error bars with the actual errors for test data. In addition to the test set T drawn from the same population as the training set C+M, there are two extrapolation data sets (see table 1). E1 contains samples with larger fat content than the training set. Similarly E2 consists of samples with larger water content. The fat predictions were computed with error bars for the 68 samples in T, E1 and E2 using the committee of networks with three hidden units. It was checked that the size of the residuals are independent of the predicted fat value on C, M and T [19]. The residuals are consistent with the error bars as shown in figure 3 and table 3.

5 Moody's GPE

Moody [21] proposed an estimator of the generalisation error for neural networks. Being a generalisation of Akaike's Final Prediction Error it is called the Generalised Prediction Error. GPE predicts the test error from the training error E_D and the "effective number of parameters" for which we substitute the number of well-determined parameters, $\gamma = \sum_g \gamma_g$:

$$\text{GPE} = \frac{2E_D}{N}\frac{N+\gamma}{N-\gamma} \tag{43}$$

GPE is compared to the actual test error in figure 4. A comparison with figure 1 shows that the evidence is slightly superior to the GPE when it comes to selecting the models with the best generalisation.

To summarise, here follows a comparison of GPE and the evidence:

Foundations. The GPE is a conventional statistical estimator of the generalisation error. Its properties are proved under specific assumptions. It is not clear whether these assumptions are fulfilled for neural networks.

[10]Note that the components are additive as variances but not as standard deviations

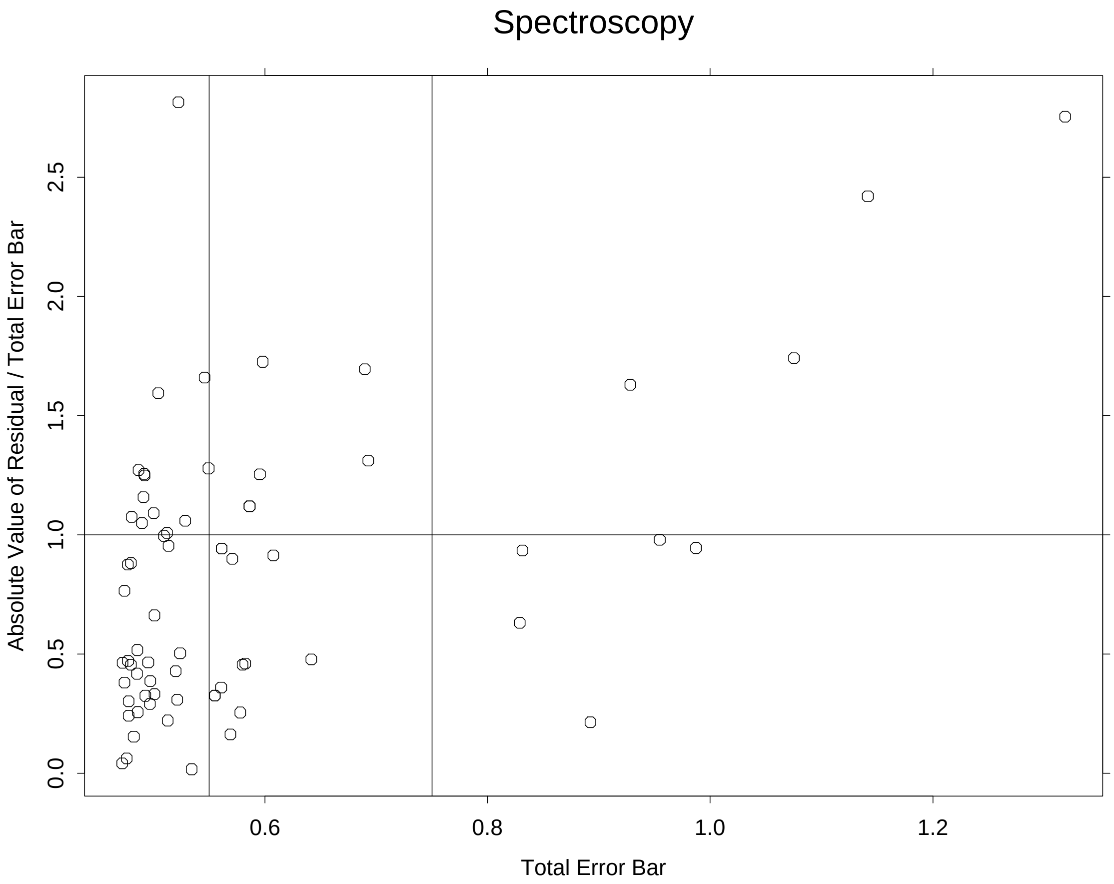

Figure 3: *The observed residuals of the predictions of fat on test samples divided by the Bayesian standard error bar, against the Bayesian standard error bars. The observations are centred on the horisontal line at 1, so the Bayesian error bars account well for the residuals. The test data are divided into three groups indicated by the vertical lines, and the observed SEP in each group is compared to the Bayesian estimate in table 3.*

The evidence has a different status. It is not proved to be an estimator but is a quality measure based on metaprinciples.

Performance. For the spectroscopic data GPE is a poorer selector of good networks than is the evidence. However, a comparison on other learning problems, for instance the robot arm problem in [3], shows that the GPE is as good as the evidence. So we limit our conclusion to saying that the GPE performs *no better* than the evidence in model comparison.

Properties. GPE has a simpler structure than the evidence. It does not involve the sometimes ill-defined determinant of the Hessian. The GPE scales differently from the evidence with γ and N (see [22])

Use in Regularisation. We used GPE for networks which were already regularised with the Bayesian method. It has not yet been demonstrated that

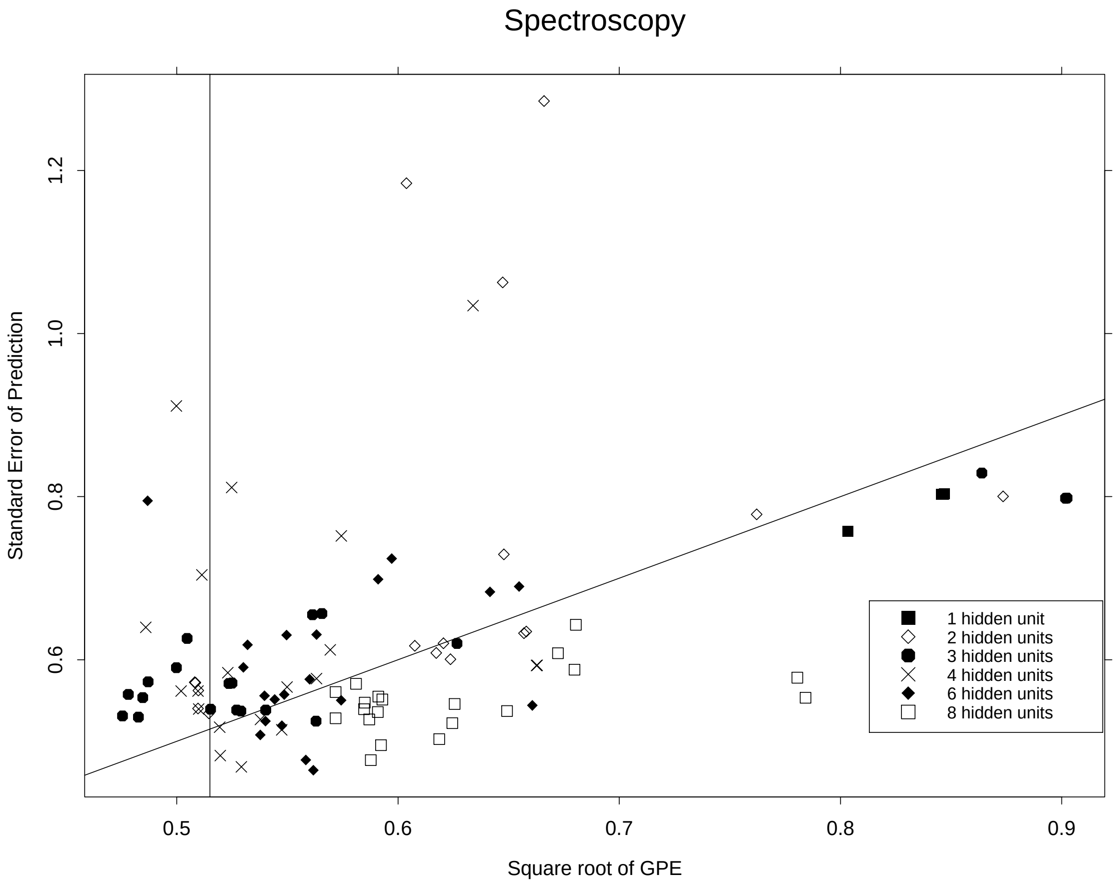

Figure 4: *The test error versus the square root of Moody's GPE (Generalised Prediction Error) for networks trained on the spectroscopic data. (The diagonal of the plot is drawn). The 20 networks with smallest GPE are delimited by the vertical line and have on average SEP=0.60. Compare with figure 1.*

GPE – like the Bayesian framework – can be used to set several weight decay parameters during training.

Generality. The Bayesian framework is more general than the GPE. It can be used to determine error bars on each prediction and it applies to classification problems and active learning [4, 5].

6 Practical Comments

This section discusses details pertaining to the practical application of Bayesian backprop.

6.1 Checking the Model

It is important to check the distribution of the weights and the residuals. The Bayesian framework presented here requires that these be Gaussian. If the weights of a group are not normally distributed, the group should be subdivided into groups with separate weight decay parameters. An excess of weights near zero is an indication that pruning may be a good idea, see [1]. Many eigenvalues near zero is another good reason for pruning the network.

If the residual errors are not Gaussian distributed it can be due to two causes:

- Some of the training examples are outliers, for instance measurement or typing errors. These should be removed from the training data (see [7] for a thorough Bayesian treatment).
- The noise level depends on the output or input. A possible solution is to transform the output variable or to introduce a noise level which depends on the output. A treatment of input dependent noise level is given in [23].

6.2 Testing for Non-linearities

Neural networks are well suited to test whether a data set defines a linear or a nonlinear regression. Linear models and neural networks are trained and the evidences are compared. The universal approximation capability of neural networks minimises the risk of overlooking a nonlinearity.

When neural networks are used to model a relation which is not significantly nonlinear they have been observed to degenerate in a sensible way: in a network with direct connections from input to output the connections of the first hidden layer are driven to zero by the Bayesian weight decay, i.e. the weight decay parameter diverges. In a network without direct input-output connections the weights of the first layer are driven to small finite values so that the sigmoid functions of the hidden layer are used in the linear region near zero.

7 Automatic Relevance Determination

7.1 Relevance of Inputs

The Tecator data pre-processed to the ten first principal components were used above to compare Bayesian and non-Bayesian methods. Alternative input–representations of the spectra where not considered and no knowledge of the input structure was used.

We expect that the relevance of the principal components decreases with increasing number of the component, and in this section we improve the model by incorporating this knowledge into the model prior. This is done using Automatic Relevance Determination (ARD) introduced by MacKay and Neal [24, 12]. ARD is simply a special case of the Bayesian framework, where each input corresponds to a weight group containing the weights from this input to the hidden layer. The model automatically adjusts the effect of an input through the weight decay parameter of the corresponding weight group. The philosophy of ARD is to avoid the sharp decision whether to include an input variable or not. This allows one to include extra variables which are more or less relevant to the problem. The advantages are that the relevance is determined automatically and that it allows for a situation where neither the inclusion nor the exclusion of the variable is optimal.

A large number of weight decay parameters gives raise to a more complex optimisation problem. As the training proceeds, the system follows a complicated path in a high–dimensional weight decay parameters space parallel with the path in weight space. This can lead to computational problems with large sensitivity to initial values as well as local minima. In addition, with few weights per weight group (e.g. 3), the weight decay parameters are more poorly determined and MacKay's method with the determination of MP values of the weight decay parameters may be a poor approximation.

7.2 Regularised ARD

To avoid the problems with the proliferation of weight decay parameters, we simplify ARD using our expectation of a decreasing relevance of the principal components and we force the decrease to be exponential, corresponding to the following relation between the weight decay parameters

$$\xi_i = \xi_0 \exp(si), \quad i = 1, \ldots, I \tag{44}$$

where i is the input number[11]. The parameters ξ_0 and s are determined from the data in the following way: We determine MP values of ξ_i's for weight groups for each input, but then we model $\log \xi_i$ as a linear function of the input number[12]:

$$\log \xi_i = \log \xi_0 + si$$

and we substitute the ξ_i's predicted by this equation.

The Ockham factors of ξ_0 and s are found in a heuristic way from the observation that we effectively determine two weight decay parameters for the weights in the first hidden layer. Let γ_{first} denote the effective number of parameters in the first layer. Each of the two weight decay parameters is considered to be determined by $\gamma_{\text{first}}/2$ parameters, so using (29) the Ockham factor is approximately $\text{Ock}(\xi_0, s) = (\sqrt{4\pi/(\gamma_{\text{first}}/2)}/\log\Omega)^2$.

To summarise, we use ARD, but the weight decay parameters are regularised to decrease exponentially with increasing input number.

7.3 Results

The performance of ARD is illustrated in figure 5 and table 2. The scheme is used with 8, 10, 12, 13, 14, 16 and 18 inputs and with 3 and 4 hidden units and without direct input–output connections. A committee of the 10 best networks with best evidence (out of 20 networks with 3 hidden units and 20 networks with 4 hidden units) is used for each number of input units[13].

With ARD, the largest evidence ($\log \text{Ev} = -218$) is obtained at 13 inputs, for which the SEP is 0.36. The s_{MP} is 0.33 on average, i.e. the weights from the

[11] The biases are associated with the weights from the first input in order to limit the number of weight groups

[12] A least square fit is used, corresponding to normally distributed uncertainties on $\log \xi_i$. This is a good approximation if each weight group contains at least 6 weights. Therefore, neighbouring weight groups are joined until the group size exceeds 5.

[13] The SEP for 10 inputs and no ARD is 0.04 better than obtained in section 4.2, due to different architecture and different committee selection.

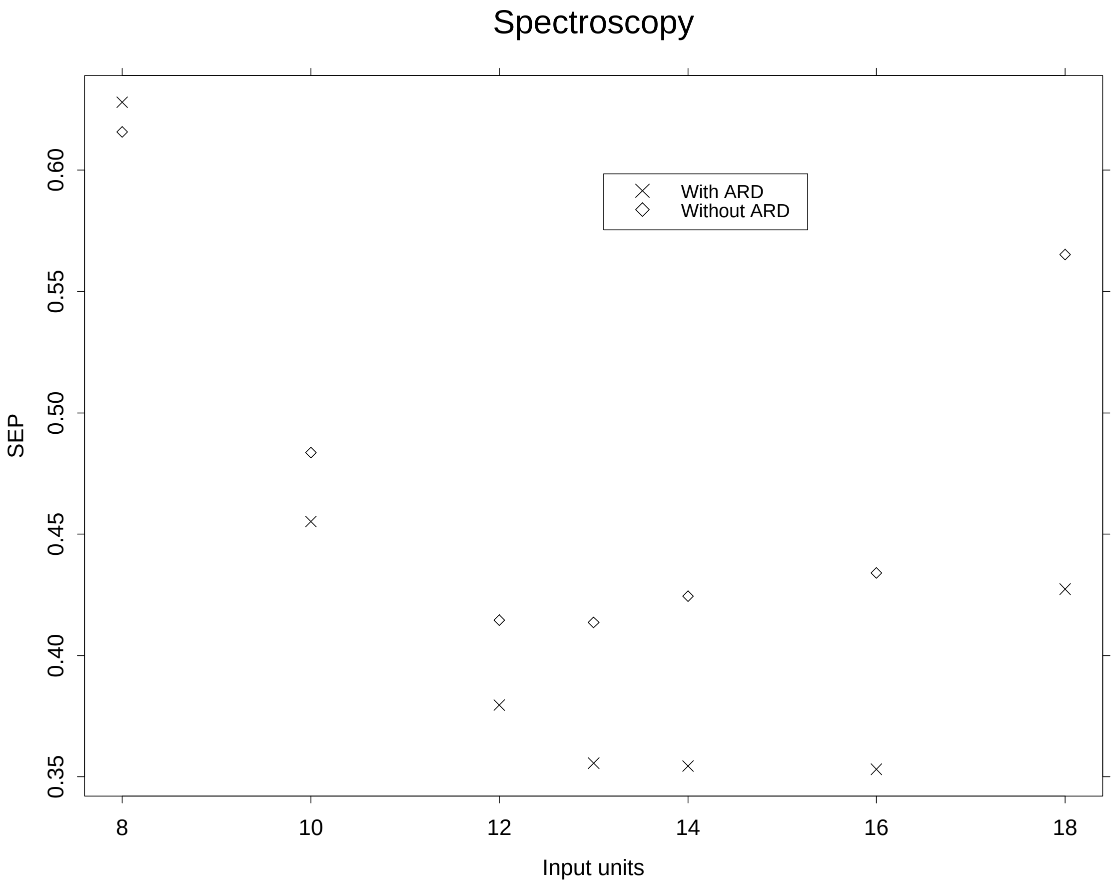

Figure 5: *The Standard error of prediction versus the number of principal components used as inputs to the network. With many inputs, the models with ARD are superior to the models with no ARD.*

13'th principal component are modelled as 7.2 times smaller than those from the first component. Without ARD, the maximum $\log \mathrm{Ev} = -236$ occurs at 12 hidden units corresponding to SEP=0.42. So ARD improves the model SEP by 0.06, a reduction of 14%.

When the number of inputs is increased above the optimal, both models deteriorate. Thus this version of ARD does not obviate the search for the optimal number of inputs. For 16 and 18 input units the committees are dominated by networks with 3 hidden units. A network with 16 inputs and 3 hidden units has approximately $N/k = 3$. We expect the Gaussian approximation to break down beyond 16 inputs, and indeed, the networks with 18 inputs perform poorly.

It is important to address the problem of how to incorporate our knowledge in the adaptive modelling. The solution here was to express it as an exponential prior on the weight decay parameters.

8 Conclusions

The Bayesian approach is often critizised for making assumptions which are not easily accepted. However, more conventional statistical methods are based on assumptions too – assumptions which are often not recognised. In general, inference must be based on assumptions which cannot be proved.

The main assumptions of the presented framework is that the evidence correlates well with the test error and that the Gaussian approximation to the posterior weight distribution is good.

It is proposed that the continued work with the Bayesian method should always consider the following two aspects in conjunction:

- The theoretical Bayesian framework must be presented in a convincing and consistent manner. The importance of a principled approach cannot be exaggerated.
- The correlation between the test error and the evidence must be studied for a wide range of applications. This is also a check on the validity of the Gaussian approximation.

If the applied aspect reveals inconsistencies, it can lead to a revision of the theoretical framework.

A detailed recipe for Bayesian backprop was given. With this it is easy to upgrade existing simulators. The Gaussian approximation was found to break down when the number of weights exceeds one third of the number of training cases.

With Bayesian weight decay, neural networks have matured. The unsatisfactory "early stopping" technique can be replaced by a principled and efficient treatment of the overfitting problem. For a wide range of applications, Bayesian backprop is likely to be the most efficient method of nonlinear regression.

ARD illustrates that the formalism is able to adapt to specific problems by incorporating knowledge into the prior

The spectroscopic data set has been made public and is available by ftp on the machine `lib.stat.cmu.edu` in the file `datasets/tecator`[14].

Acknowledgements

I am grateful to David MacKay for extensive discussions of many details of Bayesian Backprop. Also thanks to Radford Neal, Sara Solla and Carl Edward Rasmussen for valuable corrections to the manuscript and to Lars K.Hansen for ideas on generalisation.

References

[1] H.H.Thodberg, "A Bayesian Approach to Pruning of Neural Networks", submitted to *IEEE Trans. of Neural Networks* 1993.

[14] At the time of this writing, the file in `lib.stat.cmu.edu` is an old, corrupted version. Until the new version, which has the length 246934 bytes, is installed in `lib.stat.cmu.edu`, the data can be fetched from `nn.dmri.dk` in the file `pub/tecator`.

[2] D.J.C.MacKay, "Bayesian Interpolation" *Neural Comp.* **4** (1992) 415-447.

[3] D.J.C.MacKay, "A Practical Bayesian Framework for Backpropagation Networks" *Neural Comp.* **4** (1992) 448-472.

[4] D.J.C.MacKay, "Information-Based Objective Functions for Active Data Selection" *Neural Comp.* **4** (1992) 590-604.

[5] D.J.C.MacKay, "The Evidence Framework Applied to Classification Networks" *Neural Comp.* **4** (1992) 720-736.

[6] S.F.Gull "Bayesian inductive inference and maximum entropy", in *Maximum Entropy and Bayesian Methods in science and engineering, vol 1: Foundations*, eds. G.J.Ericson and C.R.Smith, Kluwer 1988.

[7] G.E.P.Box and G.C.Tiao, *Bayesian Inference in statistical analysis*, Addison-Wesley 1973.

[8] J.O.Berger, *Statistical Decision Theory and Bayesian Analysis*, Springer 1985.

[9] W.L.Buntine and A.S.Weigend, "Bayesian Back-Propagation", *Complex Systems* **5**, (1991) 603-643.

[10] D.H.Wolpert, "On the Use of Evidence Neural Networks", *Neural Information Processing Systems, Vol.5* ed. C.L.Giles, S.J.Hanson and J.D.Cowan (Morgan Kaufmann, San Mateo, 1993) 539-546.

[11] D.J.C.MacKay, "Hyperparameters: Optimise or integrate out?", *Maximum entropy and Bayesian Methods*, G.Heidbreder ed., (Kluwer, Dordrecht, 1993).

[12] R.M.Neal, *Bayesian Learning for Neural Networks*, PhD Thesis, Computer Science Dep., Univ. of Toronto, 1995

[13] N.R.Draper and H.Smith, *Applied Regression Analysis*, second ed., Wiley 1981.

[14] L.K.Hansen and P.Salamon, "Neural Network Ensembles", *IEEE Trans. PAMI* **12** (1990) 993-1001.

[15] R.M.Neal, "Bayesian Learning via Stochastic Dynamics", *Neural Information Processing Systems, Vol.5* ed. C.L.Giles, S.J.Hanson and J.D.Cowan (Morgan Kaufmann, San Mateo, 1993), and "Bayesian training of backpropagation networks by the hybrid Monte Carlo Method", CRG-TR-92-1, Dept.of Computer Science, Univ. of Toronto (1992).

[16] H.H.Thodberg, "Improving Generalization of Neural Networks through Pruning", *Int.J.Neural Systems* **1** (1991) 317-325.

[17] C.Bishop, "Exact Calculation of the Hessian Matrix for the Multilayer Perceptron" *Neural Comp.* **4** (1992) 494-501.

[18] W.H.Press et al., *Numerical Recipes in C*, Cambridge Univ. Press, Cambridge 1988.

[19] C.Borggaard and H.H.Thodberg, "Optimal Minimal Neural Interpretation of Spectra", *Analytic Chemistry* **64** (1992) 545-551.

[20] A.S.Weigend, B.A.Huberman and D.E.Rumelhart, "Predicting the future: a connectionist approach", *Int.J.Neural Systems* **1** (1990) 193-209.

[21] J.Moody, "The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems", *Neural Information Processing Systems, Vol.4* ed. J.E.Moody et al (Morgan Kaufmann, San Mateo, 1992).

[22] D.J.C.MacKay, "Bayesian Model Comparison and Backprop Nets", *Neural Information Processing Systems, Vol.4* ed. J.E.Moody et al (Morgan Kaufmann, San Mateo, 1992) 839-846.

[23] D.J.C.MacKay, *Bayesian Methods for Adaptive Models*, PhD thesis, California Inst.Tech. Pasadena 1991, chapter 6.

[24] D.J.C.MacKay, "Bayesian Non–linear modelling for the energy prediction competition", preprint, 1993.