Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks

David J C MacKay
Cavendish Laboratory,
Cambridge, CB3 0HE. United Kingdom.
mackay@mrao.cam.ac.uk

Abstract. Bayesian probability theory provides a unifying framework for data modelling. In this framework the overall aims are to find models that are well-matched to the data, and to use these models to make optimal predictions. Neural network learning is interpreted as an inference of the most probable parameters for the model, given the training data. The search in model space (i.e., the space of architectures, noise models, preprocessings, regularizers and weight decay constants) can then also be treated as an inference problem, in which we infer the relative probability of alternative models, given the data. This review describes practical techniques based on Gaussian approximations for implementation of these powerful methods for controlling, comparing and using adaptive networks.

1. Probability theory and Occam's razor

Bayesian probability theory provides a unifying framework for data modelling. A Bayesian data-modeller's aim is to develop probabilistic models that are well-matched to the data, and make optimal predictions using those models. The Bayesian framework has several advantages.

Probability theory forces us to make explicit all our modelling assumptions, whereupon our inferences are mechanistic. Once a model space has been defined, then, whatever question we wish to pose, the rules of probability theory give a unique answer which consistently takes into account all the given information. This is in contrast to orthodox (also known as 'frequentist' or 'sampling theoretical') statistics, in which one must invent 'estimators' of quantities of interest and then choose between those estimators using some criterion measuring their sampling properties; there is no clear principle for deciding which criterion to use to measure the performance of an estimator; nor, for most criteria, is there any systematic procedure for the construction of optimal estimators.

Bayesian inference satisfies the likelihood principle (Berger 1985): our inferences depend only on the probabilities assigned to the data that were received, not on properties of other data sets which might have occurred but did not.

Probabilistic modelling handles uncertainty in a natural manner. There is a unique prescription (marginalization) for incorporating uncertainty about parameters into our predictions of other variables.

Finally, Bayesian model comparison embodies **Occam's razor**, the principle that states a preference for simple models. This point will be expanded on in a moment.

The remainder of section 1 reviews Bayesian model comparison, with particular emphasis on the automatic complexity control that it provides. In section 2 the Bayesian interpretation of neural network learning is given. Section 3 then discusses the use of probability theory to control the complexity of a neural network, and section 4 discusses Bayesian comparison of neural network models.

In section 5 another important Bayesian theme is explored, namely the incorporation of parameter uncertainty (error bars) into neural network predictions.

Sections 6 and 7 review Bayesian formulae for 'pruning' (parameter deletion) and for automatic determination of the relevance of multiple inputs. Section 8 reviews the prior probability distribution over functions that is implicit in the traditional use of neural networks with weight decay regularization.

Most of the methods reviewed in this paper employ Gaussian approximations to the probability distribution of the network parameters. Section 9 offers computational short-cuts for reducing the expense of approximating Bayesian inference. Finally, section 10 compares the Bayesian framework with other theories and methods that have been applied to neural networks, and highlights differences with the conventional dogma of learning theory and statistics. This discussion is concluded by a sketch of some of the frontiers of current research in Bayesian methods and neural networks.

For background reading on Bayesian methods, the following references may be helpful. Bayesian methods are introduced and contrasted with orthodox statistics in (Jaynes 1983; Gull 1988; Loredo 1990). The Bayesian Occam's razor is demonstrated on model problems in (Gull 1988; MacKay 1992a). Useful textbooks are (Box and Tiao 1973; Berger 1985).

1.1. Motivations for Occam's razor

Occam's razor is the principle that states a preference for simple theories. If several explanations are compatible with a set of observations, Occam's razor advises us to buy the least complex explanation. This principle is often advocated for one of two reasons: the first is aesthetic ("A theory with mathematical beauty is more likely to be correct than an ugly one that fits some experimental data" (Paul Dirac)); the second reason is the supposed empirical success of Occam's razor. Here I will discuss a different justification for Occam's razor, namely:

Coherent inference embodies Occam's razor automatically and quantitatively.

1.2. Probability theory

The language of coherent inference is probability theory (Cox 1946). All coherent beliefs and predictions can be mapped onto probabilities. I will use the following notation for *conditional* probabilities: $P(A|B,\mathcal{H})$ is pronounced 'the probability of A, given B and $\mathcal{H}$'. The statements B and $\mathcal{H}$ list the conditional assumptions on which this measure of plausibility is based. For example, imagine that Joe has a test for a nasty disease; if A is the proposition "the test result is positive", and B is "Joe has the disease", then the quantity $P(A|B,\mathcal{H})$ is a number between 0 and 1 which expresses how likely we think the test would be to give the right answer, assuming that Joe did have the disease, and given the overall assumptions $\mathcal{H}$ about the reliability of the test.

There are two rules of probability. The *product rule* relates the *joint* probability of A *and* B, $P(A,B|\mathcal{H})$ to the conditional probability above:

$$P(A,B|\mathcal{H}) = P(A|B,\mathcal{H})P(B|\mathcal{H}). \tag{1}$$

Thus the probability that Joe has the disease (B) *and* the test is positive (A) is the product of the probability that Joe has the disease, and the probability that the test detects the disease, given that he has got it.

The *sum rule* relates the *marginal* probability distribution of A, $P(A|\mathcal{H})$, to the joint and conditional distributions:

$$P(A|\mathcal{H}) = \sum_B P(A, B|\mathcal{H}) = \sum_B P(A|B, \mathcal{H})P(B|\mathcal{H}).$$

Here we sum over (or 'marginalize over') the alternative values of B. For example, if Joe either has the disease (B) or does not ($\bar{B}$), then the sum rule states $P(A|\mathcal{H}) = P(A, B|\mathcal{H}) + P(A, \bar{B}|\mathcal{H})$, i.e., the probability of obtaining a positive result is the sum of the probabilities of the two alternative explanations: the probability that the result is positive and Joe has the disease, plus the probability that the result is positive and Joe does not in fact have the disease. If B is a continuous variable then the sum is replaced by an integral, and the probability $P(B|\mathcal{H})$ is a probability density.

Having specified the joint probability of all variables as in equation (1), we can then use the rules of probability to evaluate how our beliefs and predictions should change when we gain new information, i.e., as we change the conditioning statements to the right of the "|" symbol. For example, given that Joe's test result is positive, we might wish to know how plausible it is that Joe has the disease; this is measured by the probability $P(B|A, \mathcal{H})$, which can be obtained by Bayes' theorem,

$$P(B|A, \mathcal{H}) = \frac{P(A|B, \mathcal{H})P(B|\mathcal{H})}{P(A|\mathcal{H})}.$$

Here, our overall model of the situation, $\mathcal{H}$, is a conditioning statement on the right hand side of all the probabilities. In this sense, Bayesian inferences are 'subjective', in that it is not possible to reason about data without making assumptions. At the same time, Bayesian inferences are objective, in that anyone who shares the same assumptions $\mathcal{H}$ will draw identical inferences; there is only one answer to a well-posed problem. Bayesian methods thus force us to make all tacit assumptions explicit, and then provide rules for reasoning consistently given those assumptions. Note that the conditioning notation does not imply causation. $P(B|A)$ does not mean 'the probability that Joe's illness is caused by his positive test result'. Rather, it measures the plausibility of proposition B, assuming that the information in proposition A is true.

1.3. Model comparison and Occam's razor

We evaluate the plausibility of two alternative theories $\mathcal{H}_1$ and $\mathcal{H}_2$ in the light of data D as follows: using Bayes' theorem, we relate the plausibility of model $\mathcal{H}_1$ given the data, $P(\mathcal{H}_1|D)$, to the predictions made by the model about the data, $P(D|\mathcal{H}_1)$, and the prior plausibility of $\mathcal{H}_1$, $P(\mathcal{H}_1)$. This gives the following probability ratio between theory $\mathcal{H}_1$ and theory $\mathcal{H}_2$:

$$\frac{P(\mathcal{H}_1|D)}{P(\mathcal{H}_2|D)} = \frac{P(\mathcal{H}_1)}{P(\mathcal{H}_2)} \frac{P(D|\mathcal{H}_1)}{P(D|\mathcal{H}_2)}. \tag{2}$$

The first ratio ($P(\mathcal{H}_1)/P(\mathcal{H}_2)$) on the right hand side measures how much our initial beliefs favoured $\mathcal{H}_1$ over $\mathcal{H}_2$. The second ratio expresses how well the observed data were predicted by $\mathcal{H}_1$, compared to $\mathcal{H}_2$.

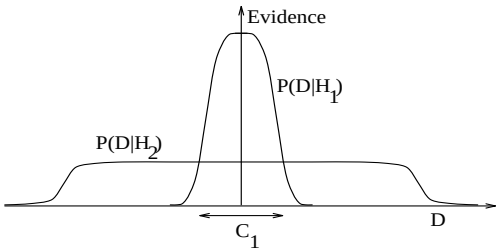


Figure 1. Why Bayesian inference embodies Occam's razor. This figure gives the basic intuition for why complex models can turn out to be less probable. The horizontal axis represents the space of possible data sets D. Bayes' theorem rewards models in proportion to how much they *predicted* the data that occurred. These predictions are quantified by a normalized probability distribution on D. In this paper, this probability of the data given model $\mathcal{H}_i$, $P(D|\mathcal{H}_i)$, is called the evidence for $\mathcal{H}_i$.

A simple model $\mathcal{H}_1$ makes only a limited range of predictions, shown by $P(D|\mathcal{H}_1)$; a more powerful model $\mathcal{H}_2$, that has, for example, more free parameters than $\mathcal{H}_1$, is able to predict a greater variety of data sets. This means, however, that $\mathcal{H}_2$ does not predict the data sets in region C_1 as strongly as $\mathcal{H}_1$. Suppose that equal prior probabilities have been assigned to the two models. Then, if the data set falls in region C_1, the *less powerful* model $\mathcal{H}_1$ will be the *more probable* model.

How does this relate to Occam's razor, when $\mathcal{H}_1$ is a simpler model than $\mathcal{H}_2$? The first ratio $(P(\mathcal{H}_1)/P(\mathcal{H}_2))$ gives us the opportunity, if we wish, to insert a prior bias in favour of $\mathcal{H}_1$ on aesthetic grounds, or on the basis of experience. This would correspond to the aesthetic and empirical motivations for Occam's razor mentioned earlier. But such a prior bias is not necessary: the second ratio, the data-dependent factor, embodies Occam's razor *automatically*. Simple models tend to make precise predictions. Complex models, by their nature, are capable of making a greater variety of predictions (figure 1). So if $\mathcal{H}_2$ is a more complex model, it must spread its predictive probability $P(D|\mathcal{H}_2)$ more thinly over the data space than $\mathcal{H}_1$. Thus, in the case where the data are compatible with both theories, the simpler $\mathcal{H}_1$ will turn out more probable than $\mathcal{H}_2$, without our having to express any subjective dislike for complex models. Our subjective prior just needs to assign equal prior probabilities to the possibilities of simplicity and complexity. Probability theory then allows the observed data to express their opinion.

Let us turn to a simple example. Here is a sequence of numbers:

$$-1,\ 3,\ 7,\ 11$$

The task is to predict what the next two numbers are likely to be, and infer what the underlying process probably was, that gave rise to this sequence. A popular answer to this question is the prediction "15, 19", with the explanation "add 4 to the previous number".

What about the alternative answer "$-19.9, 1043.8$" with the underlying rule being: "get the next number from the previous number, x, by evaluating $-x^3/11 + 9/11x^2 + 23/11$"? I assume that this prediction seems rather less plausible. But the second rule fits the data (-1, 3, 7, 11) just as well as the rule "add 4". So why should we find it less plausible? Let us give labels to the two general theories:

$\mathcal{H}_a-$ the sequence is an **arithmetic** progression, 'add n', where n is an integer.

$\mathcal{H}_c$– the sequence is generated by a **cubic** function of the form $x \to cx^3 + dx^2 + e$, where c, d and e are fractions.

One reason for finding the second explanation, $\mathcal{H}_c$, less plausible, might be that arithmetic progressions are more frequently encountered than cubic functions. This would put a bias in the prior probability ratio $P(\mathcal{H}_a)/P(\mathcal{H}_c)$ in equation (2). But let us give the two theories equal prior probabilities, and concentrate on what the data have to say. How well did each theory predict the data?

To obtain $P(D|\mathcal{H}_a)$ we must specify the probability distribution that each model assigns to its parameters. First, $\mathcal{H}_a$ depends on the added integer n, and the first number in the sequence. Let us say that these numbers could each have been anywhere between -50 and 50. Then since only the pair of values $\{n=4, \text{first number}=-1\}$ give rise to the observed data D = (-1, 3, 7, 11), the probability of the data, given $\mathcal{H}_a$, is:

$$P(D|\mathcal{H}_a) = \frac{1}{101}\frac{1}{101} = 0.00010$$

To evaluate $P(D|\mathcal{H}_c)$, we must similarly say what values the fractions c, d and e might take on. [I choose to represent these numbers as fractions rather than real numbers because if we used real numbers, the model would assign, relative to $\mathcal{H}_a$, an infinitessimal probability to D. Real parameters are the norm however, and are assumed in the rest of this paper.] A reasonable prior might state that for each fraction the numerator could be any number between -50 and 50, and the denominator is any number between 1 and 50. As for the initial value in the sequence, let us leave its probability distribution the same as in $\mathcal{H}_a$. There are four ways of expressing the fraction $c = -1/11 = -2/22 = -3/33 = -4/44$ under this prior, and similarly there are four and two possible solutions for d and e, respectively. So the probability of the observed data, given $\mathcal{H}_c$, is found to be:

$$\begin{aligned} P(D|\mathcal{H}_c) &= \left(\frac{1}{101}\right)\left(\frac{4}{101}\frac{1}{50}\right)\left(\frac{4}{101}\frac{1}{50}\right)\left(\frac{2}{101}\frac{1}{50}\right) \\ &= 0.0000000000025 = 2.5 \times 10^{-12} \end{aligned}$$

Thus comparing $P(D|\mathcal{H}_c)$ with $P(D|\mathcal{H}_a) = 0.00010$, even if our prior probabilities for $\mathcal{H}_a$ and $\mathcal{H}_c$ are equal, the odds, $P(D|\mathcal{H}_a) : P(D|\mathcal{H}_c)$, in favour of $\mathcal{H}_a$ over $\mathcal{H}_c$, given the sequence D = (-1, 3, 7, 11), are about forty million to one.

This answer depends on several subjective assumptions; in particular, the probability assigned to the free parameters n, c, d, e of the theories. Bayesians make no apologies for this: there is no such thing as inference or prediction without assumptions. However, the quantitative details of the prior probabilities have no effect on the qualitative Occam's razor effect; the complex theory $\mathcal{H}_c$ always suffers an 'Occam factor' because it has more parameters, and so can predict a greater variety of data sets (figure 1). This was only a small example, and there were only four data points; as we move to larger and more sophisticated problems the magnitude of the Occam factors typically increases, and the degree to which our inferences are influenced by the quantitative details of our subjective assumptions becomes smaller.

1.4. Bayesian methods and data analysis

Let us now relate the discussion above to real problems in data analysis.

There are countless problems in science, statistics and technology which require that, given a limited data set, preferences be assigned to alternative models of differing complexities. For example, two alternative hypotheses accounting for planetary motion are Mr. Inquisition's geocentric model based on 'epicycles', and Mr. Copernicus's simpler model of the solar system. The epicyclic model fits data on planetary motion at least as well as the Copernican model, but does so using more parameters. Coincidentally for Mr. Inquisition, two of the extra epicyclic parameters for every planet are found to be identical to the period and radius of the sun's 'cycle around the earth'. Intuitively we find Mr. Copernicus's theory more probable.

1.5. The mechanism of the Bayesian razor: the evidence and the Occam factor

Two levels of inference can often be distinguished in the process of data modelling. At the first level of inference, we assume that a particular model is true, and we fit that model to the data, i.e., we infer what values its free parameters should plausibly take, given the data. The results of this inference are often summarized by the most probable parameter values, and error bars on those parameters. This analysis is repeated for each model. The second level of inference is the task of model comparison. Here we wish to compare the models in the light of the data, and assign some sort of preference or ranking to the alternatives.†

Bayesian methods are able consistently and quantitatively to solve both the inference tasks. There is a popular myth that states that Bayesian methods only differ from orthodox statistical methods by the inclusion of subjective priors which are difficult to assign, and usually don't make much difference to the conclusions. It is true that, at the first level of inference, a Bayesian's results will often differ little from the outcome of an orthodox attack. What is not widely appreciated is how a Bayesian performs the second level of inference; this section will therefore focus on Bayesian model comparison.

Model comparison is a difficult task because it is not possible simply to choose the model that fits the data best: more complex models can always fit the data better, so the maximum likelihood model choice would lead us inevitably to implausible, over-parameterized models which generalize poorly. Occam's razor is needed.

Let us write down Bayes' theorem for the two levels of inference described above, so as to see explicitly how Bayesian model comparison works. Each model $\mathcal{H}_i$ is assumed to have a vector of parameters $\mathbf{w}$. A model is defined by a collection of probability distributions: a 'prior' distribution $P(\mathbf{w}|\mathcal{H}_i)$ which states what values the model's parameters might be expected to take; and a set of conditional distributions, one for each value of $\mathbf{w}$, defining the predictions $P(D|\mathbf{w}, \mathcal{H}_i)$ that the model makes about the data D.

(i) **Model fitting.** At the first level of inference, we assume that one model, the ith, say, is true, and we infer what the model's parameters $\mathbf{w}$ might be, given the

† Note that both levels of *inference* are distinct from *decision theory*. The goal of inference is, given a defined hypothesis space and a particular data set, to assign probabilities to hypotheses. Decision theory typically chooses between alternative *actions* on the basis of these probabilities so as to minimize the expectation of a 'loss function'. This paper concerns inference alone and no loss functions are involved. This paper will often discuss model comparison, which should not be construed as implying model *choice*. Ideal Bayesian predictions do not involve choice between models; rather, predictions are made by summing over all the alternative models, weighted by their probabilities (section 5).

data D. Using Bayes' theorem, the **posterior probability** of the parameters $\mathbf{w}$ is:

$$P(\mathbf{w}|D, \mathcal{H}_i) = \frac{P(D|\mathbf{w}, \mathcal{H}_i)P(\mathbf{w}|\mathcal{H}_i)}{P(D|\mathcal{H}_i)}, \tag{3}$$

that is,

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}.$$

The normalizing constant $P(D|\mathcal{H}_i)$ is commonly ignored since it is irrelevant to the first level of inference, i.e., the inference of $\mathbf{w}$; but it becomes important in the second level of inference, and we name it the **evidence** for $\mathcal{H}_i$. It is common practice to use gradient-based methods to find the maximum of the posterior, which defines the most probable value for the parameters, $\mathbf{w}_{\text{MP}}$; it is then usual to summarize the posterior distribution by the value of $\mathbf{w}_{\text{MP}}$, and error bars or confidence intervals on these best fit parameters. Error bars can be obtained from the curvature of the posterior; evaluating the Hessian at $\mathbf{w}_{\text{MP}}$, $\mathbf{A} = -\nabla\nabla \log P(\mathbf{w}|D, \mathcal{H}_i)|_{\mathbf{w}_{\text{MP}}}$, and Taylor-expanding the log posterior probability with $\Delta\mathbf{w} = \mathbf{w} - \mathbf{w}_{\text{MP}}$:

$$P(\mathbf{w}|D, \mathcal{H}_i) \simeq P(\mathbf{w}_{\text{MP}}|D, \mathcal{H}_i) \exp\left(-\tfrac{1}{2}\Delta\mathbf{w}^{\text{T}}\mathbf{A}\Delta\mathbf{w}\right), \tag{4}$$

we see that the posterior can be locally approximated as a Gaussian with covariance matrix (equivalent to error bars) $\mathbf{A}^{-1}$. [Whether this approximation is good or not will depend on the problem we are solving. Indeed, the maximum and mean of the posterior distribution have no fundamental status in Bayesian inference—they both change under non-linear reparameterizations. Maximization of a posterior probability is only useful if an approximation like equation (4) gives a good summary of the distribution.]

(ii) **Model comparison.** At the second level of inference, we wish to infer which model is most plausible given the data. The posterior probability of each model is:

$$P(\mathcal{H}_i|D) \propto P(D|\mathcal{H}_i)P(\mathcal{H}_i). \tag{5}$$

Notice that the data-dependent term $P(D|\mathcal{H}_i)$ is the evidence for $\mathcal{H}_i$, which appeared as the normalizing constant in (3). The second term, $P(\mathcal{H}_i)$, is the subjective prior over our hypothesis space which expresses how plausible we thought the alternative models were before the data arrived. Assuming that we choose to assign equal priors $P(\mathcal{H}_i)$ to the alternative models, *models $\mathcal{H}_i$ are ranked by evaluating the evidence.* The normalizing constant $P(D) = \sum_i P(D|\mathcal{H}_i)P(\mathcal{H}_i)$ has been omitted from equation (5) because in the data modelling process we may develop new models after the data have arrived, when an inadequacy of the first models is detected, for example. Inference is open ended: we continually seek more probable models to account for the data we gather.

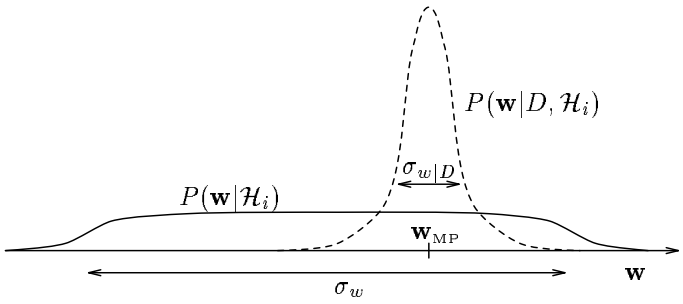


Figure 2. The Occam factor. This figure shows the quantities that determine the Occam factor for a hypothesis $\mathcal{H}_i$ having a single parameter $\mathbf{w}$. The prior distribution (solid line) for the parameter has width σ_w. The posterior distribution (dashed line) has a single peak at $\mathbf{w}_{\rm MP}$ with characteristic width $\sigma_{w|D}$. The Occam factor is $\sigma_{w|D} P(\mathbf{w}_{\rm MP}|\mathcal{H}_i) = (\sigma_{w|D}/\sigma_w)$.

To reiterate the key concept: to rank alternative models $\mathcal{H}_i$, a Bayesian evaluates the evidence $P(D|\mathcal{H}_i)$. This concept is very general: the evidence can be evaluated for parametric and 'non-parametric' models alike; whatever our data modelling task, a regression problem, a classification problem, or a density estimation problem, the evidence is a transportable quantity for comparing alternative models. In all these cases the evidence naturally embodies Occam's razor.

1.6. Evaluating the evidence

Let us now study the evidence more closely to gain insight into how the Bayesian Occam's razor works. The evidence is the normalizing constant for equation (3):

$$P(D\,|\mathcal{H}_i) = \int P(D|\mathbf{w},\mathcal{H}_i) P(\mathbf{w}|\mathcal{H}_i)\,\mathrm{d}\mathbf{w}. \tag{6}$$

For many problems, including interpolation, it is common for the posterior $P(\mathbf{w}|D,\mathcal{H}_i) \propto P(D|\mathbf{w},\mathcal{H}_i)P(\mathbf{w}|\mathcal{H}_i)$ to have a strong peak at the most probable parameters $\mathbf{w}_{\rm MP}$ (figure 2). Then, taking for simplicity the one-dimensional case, the evidence can be approximated by the height of the peak of the integrand $P(D|\mathbf{w},\mathcal{H}_i)P(\mathbf{w}|\mathcal{H}_i)$ times its width, $\sigma_{w|D}$:

$$\begin{aligned} P(D\,|\mathcal{H}_i) \;&\simeq\; \underbrace{P(D\,|\mathbf{w}_{\rm MP},\mathcal{H}_i)}\times\underbrace{P(\mathbf{w}_{\rm MP}|\mathcal{H}_i)\,\sigma_{w|D}}\,. \\ \text{Evidence} \;&\simeq\; \text{Best fit likelihood} \times \text{Occam factor} \end{aligned} \tag{7}$$

Thus the evidence is found by taking the best fit likelihood that the model can achieve and multiplying it by an 'Occam factor' (Gull 1988), which is a term with magnitude less than one that penalizes $\mathcal{H}_i$ for having the parameter $\mathbf{w}$.

1.7. Interpretation of the Occam factor

The quantity $\sigma_{w|D}$ is the posterior uncertainty in $\mathbf{w}$. Suppose for simplicity that the prior $P(\mathbf{w}|\mathcal{H}_i)$ is uniform on some large interval σ_w, representing the range of values

of $\mathbf{w}$ that were possible *a priori*, according to $\mathcal{H}_i$ (figure 2). Then $P(\mathbf{w}_{\rm MP}|\mathcal{H}_i) = 1/\sigma_w$, and

$$\text{Occam factor} = \frac{\sigma_{w|D}}{\sigma_w},$$

i.e., *the Occam factor is equal to the ratio of the posterior accessible volume of $\mathcal{H}_i$'s parameter space to the prior accessible volume,* or the factor by which $\mathcal{H}_i$'s hypothesis space collapses when the data arrive (Gull 1988; Jeffreys 1939). The model $\mathcal{H}_i$ can be viewed as consisting of a certain number of exclusive submodels, of which only one survives when the data arrive. The Occam factor is the inverse of that number. The logarithm of the Occam factor is a measure of the amount of information we gain about the model's parameters when the data arrive.

A complex model having many parameters, each of which is free to vary over a large range σ_w, will typically be penalized by a larger Occam factor than a simpler model. The Occam factor also penalizes models which have to be finely tuned to fit the data, favouring models for which the required precision of the parameters $\sigma_{w|D}$ is coarse. The magnitude of the Occam factor is thus a measure of complexity of the model but, unlike the V-C dimension (Abu-Mostafa 1990), it relates to the complexity of the predictions that the model makes in data space. This depends not only on the number of parameters in the model, but also on the prior probability that the model assigns to them. Which model achieves the greatest evidence is determined by a trade-off between minimizing this natural complexity measure and minimizing the data misfit. In further contrast to alternative measures of model complexity, the Occam factor for a model is straightforward to evaluate: it simply depends on the error bars on the parameters, which we already evaluated when fitting the model to the data.

Figure 3 displays an entire hypothesis space so as to illustrate the various probabilities in the analysis. There are three models, $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$, which have equal prior probabilities. Each model has one parameter $\mathbf{w}$ (each shown on a horizontal axis), but assigns a different prior range σ_w to that parameter. $\mathcal{H}_3$ is the most 'flexible' or 'complex' model, assigning the broadest prior range. A one-dimensional data space is shown by the vertical axis. Each model assigns a joint probability distribution $P(D, \mathbf{w}|\mathcal{H}_i)$ to the data and the parameters, illustrated by a cloud of dots. These dots represent random samples from the full probability distribution. The total number of dots in each of the three model subspaces is the same, because we assigned equal prior probabilities to the models.

When a particular data set D is received (horizontal line), we infer the posterior distribution of $\mathbf{w}$ for a model ($\mathcal{H}_3$, say) by reading out the density along that horizontal line, and normalizing. The posterior probability $P(\mathbf{w}|D, \mathcal{H}_3)$ is shown by the dotted curve at the bottom. Also shown is the prior distribution $P(\mathbf{w}|\mathcal{H}_3)$ (c.f. figure 2). [In the case of model $\mathcal{H}_1$ which is very poorly matched to the data, the shape of the posterior distribution will depend on the details of the tails of the prior $P(\mathbf{w}|\mathcal{H}_1)$ and the likelihood $P(D|\mathbf{w}, \mathcal{H}_1)$; the curve shown is for the case where the prior falls off more strongly.]

We obtain figure 1 by marginalizing the joint distributions $P(D, \mathbf{w}|\mathcal{H}_i)$ onto the D axis at the left hand side. For the data set D shown by the dotted horizontal line, the evidence $P(D|\mathcal{H}_3)$ for the more flexible model $\mathcal{H}_3$ has a smaller value than the evidence for H_2. This is because $\mathcal{H}_3$ placed less predictive probability (fewer dots) on that line. In terms of the distributions over $\mathbf{w}$, model $\mathcal{H}_3$ has smaller evidence because the Occam factor $\sigma_{w|D}/\sigma_w$ is smaller for $\mathcal{H}_3$ than for $\mathcal{H}_2$. The simplest model $\mathcal{H}_1$ has the smallest evidence of all, because the best fit that it can achieve to the data D is

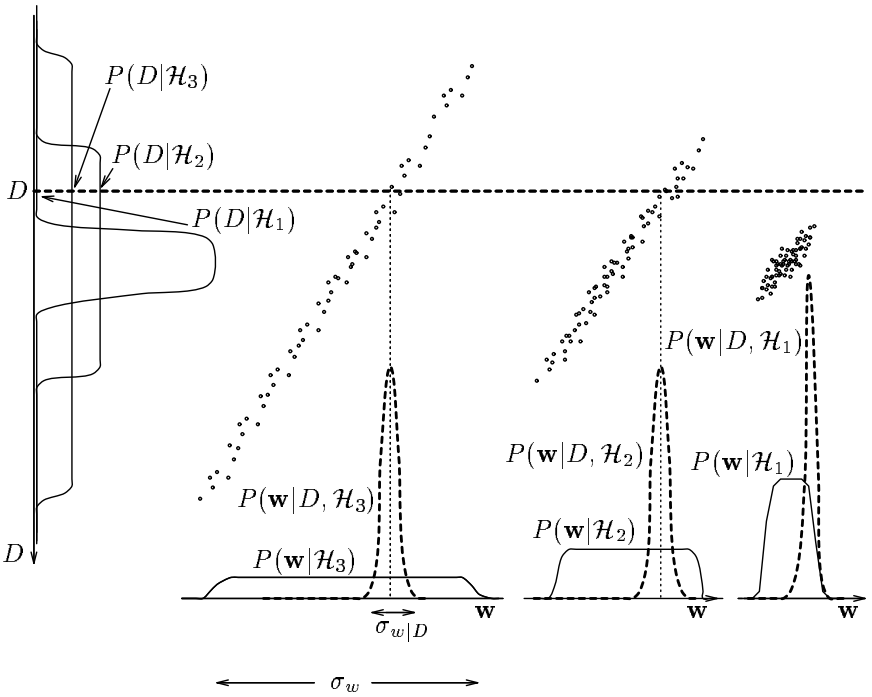


Figure 3. A hypothesis space consisting of three exclusive models, each having one parameter $\mathbf{w}$, and a one-dimensional data set D. The 'data set' is a single measured value which differs from the parameter $\mathbf{w}$ by a small amount of additive noise. Typical samples from the joint distribution $P(D, w, \mathcal{H})$ are shown by dots. (NB, these are not data points.) The observed 'data set' is a single particular value for D shown by the dashed horizontal line. The dashed curves below show the posterior probability of $\mathbf{w}$ for each model given this data set (c.f. figure 1). The evidence for the different models is obtained by marginalizing onto the D axis at the left hand side (c.f. figure 2).

very poor. Given this data set, the most probable model is $\mathcal{H}_2$.

1.8. Occam factor for several parameters

If the posterior is well approximated by a Gaussian, then the Occam factor is obtained from the determinant of the corresponding covariance matrix (c.f. equation (7)):

$$\begin{aligned} P(D\,|\mathcal{H}_i) \;&\simeq\; \underbrace{P(D\,|\mathbf{w}_{\rm MP}, H_i)}_{\text{Best fit likelihood}} \times \underbrace{P(\mathbf{w}_{\rm MP}|\mathcal{H}_i)\det{}^{-\frac{1}{2}}(\mathbf{A}/2\pi)}_{\text{Occam factor}}, \\ \text{Evidence} \;&\simeq\; \text{Best fit likelihood} \times \qquad \text{Occam factor} \end{aligned} \tag{8}$$

where $\mathbf{A} = -\nabla\nabla \log P(\mathbf{w}|D, \mathcal{H}_i)$, the Hessian which we evaluated when we calculated the error bars on $\mathbf{w}_{\rm MP}$ (equation 4). As the number of data collected, N, increases, this Gaussian approximation is expected to become increasingly accurate.

In summary, Bayesian model selection is a simple extension of maximum likelihood model selection: *the evidence is obtained by multiplying the best fit likelihood by the Occam factor.*

To evaluate the Occam factor we need only the Hessian $\mathbf{A}$, if the Gaussian approximation is good. Thus the Bayesian method of model comparison by evaluating the evidence is no more demanding computationally than the task of finding for each model the best fit parameters and their error bars.

1.9. Bayesian methods meet neural networks

The two ideas of neural network modelling and Bayesian statistics might seem uneasy bed-fellows. Neural networks are non-linear parallel computational devices inspired by the structure of the brain. 'Backpropagation networks' are able to learn, by example, to solve prediction and classification problems. Such a neural network is typically viewed as a black box which finds by hook or by crook an incomprehensible solution to a poorly understood problem. In contrast, Bayesian methods are characterized by an insistence on coherent inference based on clearly defined axioms; in Bayesian circles, an 'ad hockery' is a capital offense. Thus Bayesian statistics and neural networks might seem to occupy opposite extremes of the data modelling spectrum.

However, there is a common theme uniting the two. Both fields aim to create models which are well-matched to the data. Neural networks can be viewed as more flexible versions of traditional regression techniques. Because they are more flexible (non-linear) they are able to model regularities in the data that linear models cannot capture. The problem with neural networks is that an over-flexible network can be duped by stray correlations within the data into 'discovering' non-existent structure. This is where Bayesian methods play a complementary role. Using Bayesian probability theory one can automatically infer how flexible a model is warranted by the data; the Bayesian Occam's razor automatically suppresses the tendency to discover spurious structure in data.

Occam's razor is needed in neural networks for the reason illustrated in figure 4*a-d*. Consider a control parameter which influences the complexity of a model (for example a regularization constant). As the control parameter is varied to increase the complexity of the model (descending from figure 4*a-c* and going from left to right across figure 4*d*), the best fit to the **training** data that the model can achieve becomes increasingly good. However, the empirical performance of the model, the **test error**, has a minimum as a function of the control parameters. *An over-complex model overfits the data and generalizes poorly.* This problem may also complicate the choice of the number of hidden units in a multilayer perceptron, the radius of the basis functions in a radial basis function network, and the choice of the input variables themselves in any regression problem. Finding values for model control parameters that are appropriate for the data is therefore an important and non-trivial problem.

A central message of this paper is illustrated in figure 4*e*. If we give a probabilistic interpretation to the model, then we can evaluate the evidence for the control parameters. We find the Bayesian Occam's razor at work. Over-complex models are less probable, because they predict the data less strongly. Thus the evidence $P(\text{Data}|\text{Control Parameters})$ can be used as an objective function for optimization of model control parameters.

Bayesian optimization of model control parameters has four important advantages. (1) No 'test set' or 'validation set' is involved, so all available training data can be devoted to both model fitting and model comparison. (2) Regularization constants can be optimized on-line, i.e., simultaneously with the optimization of ordinary model parameters. (3) The Bayesian objective function is not noisy, in contrast to a cross-

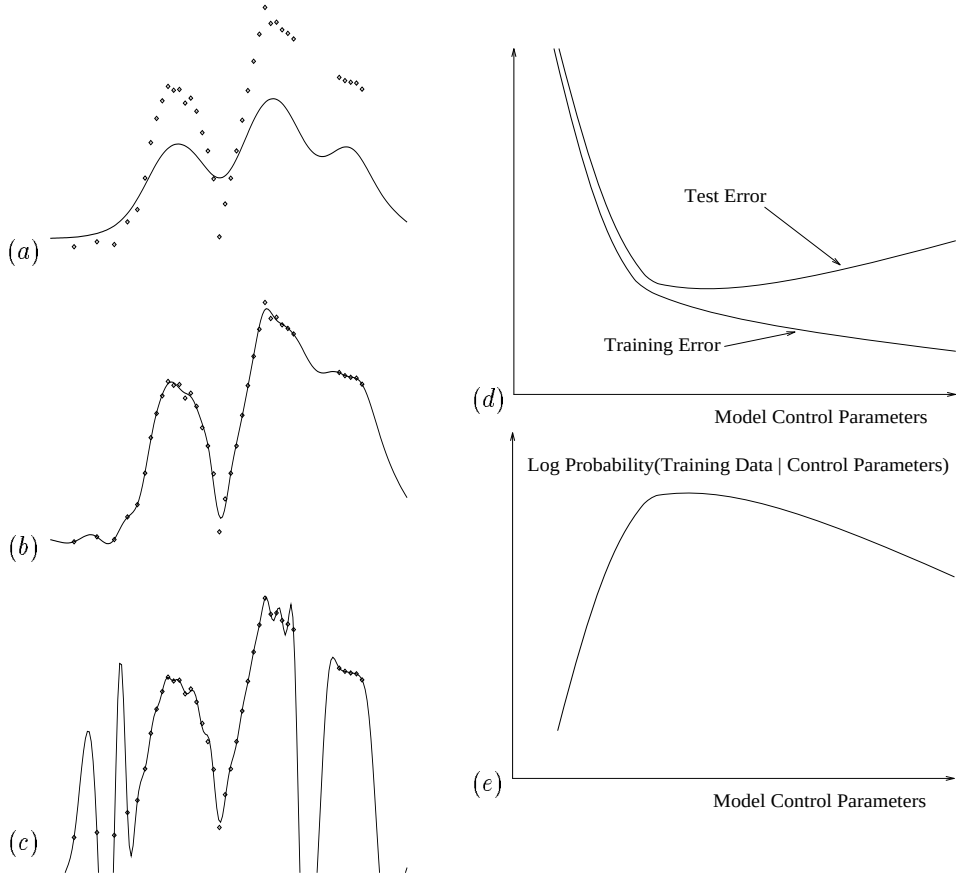


Figure 4. Optimization of model complexity. Figures $a - c$ show a radial basis function model interpolating a simple data set with one input variable and one output variable. As the regularization constant is varied to increase the complexity of the model (from a to c), the interpolant is able to fit the training data increasingly well, but beyond a certain point the generalization ability (test error) of the model deteriorates. Probability theory allows us to optimize the control parameters without needing a test set.

validation measure. (4) The gradient of the evidence with respect to the control parameters can be evaluated, making it possible to simultaneously optimize a large number of control parameters.

Other benefits of the Bayesian framework for neural networks will also be reviewed here. But my aim is not to advocate the sole use of Bayesian methods. For practical purposes it can be beneficial to look at a modelling problem from other perspectives too. Bayesian methods are typically quite sensitive to erroneous modelling assumptions, so it can sometimes turn out that Bayesian methods give different model choices from empirical performance measures such as cross-validation. Such differences may then give useful insights into poor implicit assumptions, and guide the modeller to the creation of new and superior models.

2. Neural networks as probabilistic models

A supervised neural network is a non-linear parameterized mapping from an input $\mathbf{x}$ to an output $\mathbf{y} = \mathbf{y}(\mathbf{x}; \mathbf{w}, \mathcal{A})$. The output is a continuous function of the parameters $\mathbf{w}$ (discontinuous functions may be defined but do not lend themselves to practical gradient-based optimization). The architecture of the net, i.e., the functional form of the mapping, is denoted by $\mathcal{A}$. Such networks can be 'trained' to perform regression and classification tasks.

2.1. Regression networks

In the case of a regression problem, the mapping for a network with one hidden layer may have the form:

$$\begin{array}{ll} \text{Hidden layer:} & a_j^{(1)} = \sum_l w_{jl}^{(1)} x_l + \theta_j^{(1)}; \quad h_j = f^{(1)}(a_j^{(1)}) \\ \text{Output layer:} & a_i^{(2)} = \sum_j w_{ij}^{(2)} h_j + \theta_i^{(2)}; \quad y_i = f^{(2)}(a_i^{(2)}) \end{array} \tag{9}$$

where, for example, $f^{(1)}(a) = \tanh(a)$, and $f^{(2)}(a) = a$. The 'weights' w and 'biases' θ together make up the parameter vector $\mathbf{w}$. The non-linear 'sigmoid' function $f^{(1)}$ at the hidden layer gives the neural network greater computational flexibility than a standard linear regression model.

This network is trained using a data set $D = \{\mathbf{x}^{(m)}, \mathbf{t}^{(m)}\}$ by iteratively adjusting $\mathbf{w}$ so as to minimize an objective function, e.g., the sum squared error,

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_m \sum_i \left(t_i^{(m)} - y_i(\mathbf{x}^{(m)}; \mathbf{w}) \right)^2 . \tag{10}$$

This minimization is based on repeated evaluation of the gradient of E_D using 'backpropagation' (the chain rule) (Rumelhart *et al.* 1986). Often, regularization (also known as 'weight decay') is included, modifying the objective function to:

$$M(\mathbf{w}) = \beta E_D + \alpha E_W , \tag{11}$$

where, for example, $E_W = \frac{1}{2} \sum_i w_i^2$. This additional term favours small values of $\mathbf{w}$ and decreases the tendency of a model to 'overfit' noise in the training data.

2.2. Neural network learning as inference

The neural network learning process above can be given the following probabilistic interpretation. The error function is interpreted as minus the log likelihood for a noise model:

$$P(D|\mathbf{w}, \beta, \mathcal{H}) = \frac{1}{Z_D(\beta)} \exp(-\beta E_D). \tag{12}$$

Thus, the use of the sum-squared error E_D (10) corresponds to an assumption of Gaussian noise on the target variables, and the parameter β defines a noise level $\sigma_\nu^2 = 1/\beta$.

Similarly the regularizer is interpreted in terms of a log prior probability distribution over the parameters:

$$P(\mathbf{w}|\alpha, \mathcal{H}) = \frac{1}{Z_W(\alpha)} \exp(-\alpha E_W). \tag{13}$$

If E_W is quadratic as defined above, then the corresponding prior distribution is a Gaussian with variance $\sigma_w^2 = 1/\alpha$. The probabilistic model $\mathcal{H}$ specifies the functional form $\mathcal{A}$ of the network (equation 9), the likelihood (12), and the prior (13).

The objective function $M(\mathbf{w})$ then corresponds to the *inference* of the parameters $\mathbf{w}$, given the data:

$$P(\mathbf{w}|D, \alpha, \beta, \mathcal{H}) = \frac{P(D|\mathbf{w}, \beta, \mathcal{H})P(\mathbf{w}|\alpha, \mathcal{H})}{P(D|\alpha, \beta, \mathcal{H})} \tag{14}$$

$$= \frac{1}{Z_M} \exp(-M(\mathbf{w})). \tag{15}$$

The $\mathbf{w}$ found by (locally) minimizing $M(\mathbf{w})$ is thus interpreted as the (locally) most probable parameter vector, $\mathbf{w}_{\rm MP}$.

Why is it natural to interpret the error functions as *log* probabilities? Error functions are usually additive. For example, E_D is a *sum* of squared errors. Probabilities, on the other hand, are multiplicative: for independent events A and B, the joint probability is $P(A, B) = P(A)P(B)$. The logarithmic mapping maintains this correspondence.

The interpretation of $M(\mathbf{w})$ as a log probability adds little new at this stage. But new tools will emerge when we proceed to other inferences. First, though, let us establish the probabilistic interpretation of classification networks, to which the same tools apply.

2.3. Binary classification networks

If the targets t in a data set are binary classification labels (0,1), it is natural to use a neural network whose output $y(\mathbf{x}; \mathbf{w}, \mathcal{A})$ is bounded between 0 and 1, and is interpreted as a probability $P(t=1|\mathbf{x}, \mathbf{w}, \mathcal{A})$. For example, a network with one hidden layer could be described by equation (9), with $f^{(2)}(a) = 1/(1 + \mathrm{e}^{-a})$. The error function βE_D is replaced by the log likelihood:

$$G(\mathbf{w}) = \sum_m t^{(m)} \log y(\mathbf{x}^{(m)}; \mathbf{w}) + (1 - t^{(m)}) \log(1 - y(\mathbf{x}^{(m)}; \mathbf{w})).$$

The total objective function is then $M = -G + \alpha E_W$. Note that this includes no parameter β.

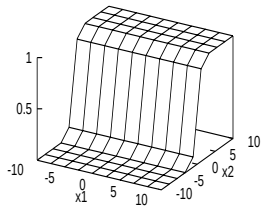
2.4. Multi-class classification networks

For a classification problem with $I > 2$ classes, we can represent the targets by a vector, $\mathbf{t}$, in which a single element t_i is set to 1, indicating the correct class, and all other elements are set to 0. In this case it is appropriate to use a 'softmax' network (Bridle 1989) having coupled outputs which sum to one and are interpreted as class probabilities $y_i = P(t_i=1|\mathbf{x}, \mathbf{w}, \mathcal{A})$. The last part of equation (9) is replaced by:

$$y_i = \exp(a_i) \Bigg/ \sum_{i'=1}^{I} \exp(a_{i'}) \ . \tag{16}$$

The log likelihood in this case is

$$G(\mathbf{w}) = \sum_m \sum_i t_i \log y_i(\mathbf{x}^{(m)}; \mathbf{w}).$$



$$\mathbf{w} = (0, 2)$$

Figure 5. Output of simple neural network as a function of its input.

As in the case of the regression network, the minimization of the objective function $M(\mathbf{w}) = -G + \alpha E_W$ corresponds to an inference of the form (15).

2.5. The probabilistic interpretation of neural network learning: a simple example

Assume we are studying a binary classification problem using a minimalist neural network whose output is the following function of $\mathbf{x}$:

$$P(t=1|\mathbf{x}, \mathbf{w}, \mathcal{H}) \equiv y(\mathbf{x}; \mathbf{w}, \mathcal{H}) = \frac{1}{1 + \mathrm{e}^{-\mathbf{w}\cdot\mathbf{x}}} \tag{17}$$

This form of function is known to statisticians as a linear logistic, and in neural networks as a single neuron. We view the output of the neuron as defining (when its parameters $\mathbf{w}$ are specified) the probability that an input $\mathbf{x}$ belongs to class $t = 1$, rather than the alternative $t = 0$.

Figure 5 shows the output of the neuron as a function of the input vector, for $\mathbf{w} = (0, 2)$. Each possible value of $\mathbf{w}$ defines a different sub-hypothesis about the probability of class 1 relative to class 0 as a function of $\mathbf{x}$. These different sub-hypotheses are depicted in the two-dimensional **weight space**, that is, the parameter space of the network, in figure 6. Each *point* in $\mathbf{w}$ space corresponds to a *function* of t and $\mathbf{x}$. For a selection of different values of the parameter vector $\mathbf{w}$, the inset figures show the function of $\mathbf{x}$ performed by the network (as in figure 5). Notice that the gradient of the sigmoid function increases as the magnitude of $\mathbf{w}$ increases.

Now imagine that we receive some data as shown in the left column of figure 7. Each data point consists of a two dimensional input vector $\mathbf{x}$ and a t value indicated by $\times$ ($t = 1$) or $\square$ ($t = 0$).

In the traditional view of learning, a single parameter vector $\mathbf{w}$ evolves under the learning rule from an initial starting point $\mathbf{w}^0$ to a final optimum $\mathbf{w}^*$, in such a way as to minimize an objective function measuring data misfit plus a regularizer such as $\alpha \sum_i w_i^2/2$.

The objective function that measures how well parameters $\mathbf{w}$ predict the observed data is the probability assigned to the observed values $D = \{t\}$ by the model with parameters set to $\mathbf{w}$. This likelihood function is shown as a function of $\mathbf{w}$ in the second column. It is a product of functions of the form (17), $P(D|\mathbf{w}, \{\mathbf{x}\}, \mathcal{H}) = \prod_n P(t^{(n)}|\mathbf{w}, \mathbf{x}^{(n)}, \mathcal{H})$; these functions are now viewed as functions of $\mathbf{w}$, with $\{\mathbf{x}, t\}$ fixed.

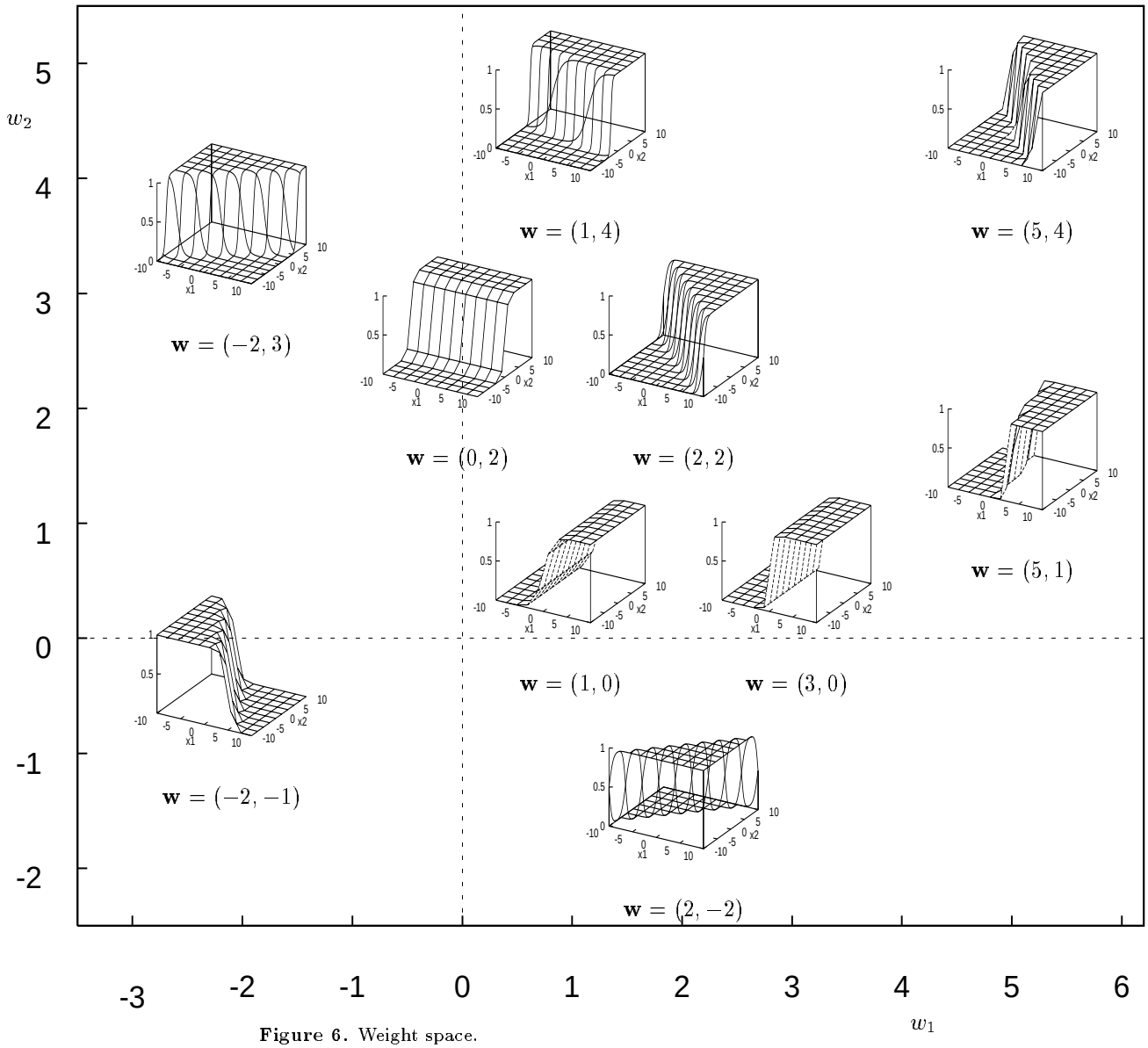


Figure 6. Weight space.

In the traditional view, the product of learning is a point in $\mathbf{w}$-space, the 'estimator' $\mathbf{w}^*$, which minimizes the objective function. In contrast, in the Bayesian view, the product of learning is an *ensemble* of plausible parameter values (bottom right of figure 7). We do not choose one particular sub-hypothesis $\mathbf{w}$; rather we evaluate their posterior probabilities, which by Bayes' theorem are:

$$P(\mathbf{w}|D,\{\mathbf{x}\},\mathcal{H}) = \frac{P(D\,|\,\mathbf{w},\{\mathbf{x}\},\mathcal{H})P(\mathbf{w}|\mathcal{H})}{P(D\,|\{\mathbf{x}\},\mathcal{H})}.$$

The posterior distribution is thus obtained by multiplying the likelihood by a prior distribution over $\mathbf{w}$ space (shown as a broad Gaussian at upper right of figure 7). The

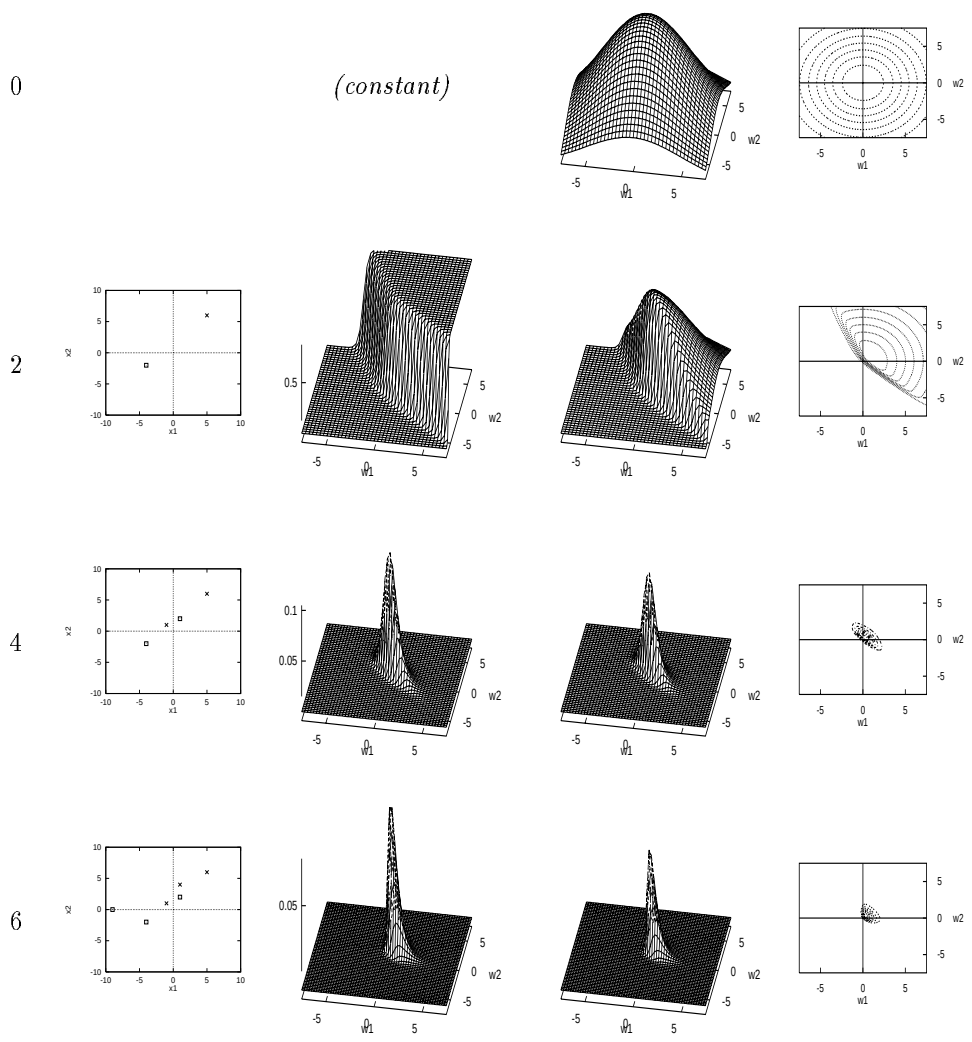


Figure 7. Evolution of the probability distribution over parameters as data arrives.

posterior ensemble (within a multiplicative constant) is shown in the third column of figure 7, and as a contour plot in the fourth column. As the amount of data increases (from top to bottom), the posterior ensemble becomes increasingly concentrated around the most probable value $\mathbf{w}^*$. This illustrates the Bayesian interpretation and generalization of traditional neural network learning.

2.6. Implementation

Let us now study the variety of useful results that can be built on the Bayesian interpretation of neural net learning. The results will refer to regression models; the corresponding results for classification models are obtained by replacing βE_D by $-G$, and $Z_D(\beta)$ by 1.

Bayesian inference for data modelling problems may be implemented by analytical methods, by Monte Carlo sampling, or by deterministic methods employing Gaussian approximations. For neural networks there are few analytic methods. Sophisticated Monte Carlo methods which make use of gradient information have been applied to model problems by Neal (1993). The methods reviewed here are based on Gaussian approximations to the posterior distribution.

3. Setting regularization constants α and β

We now return to the general case specified in equations (11–15). The control parameters α and β determine the complexity of the model. The term model here refers to a set of three assumptions: the network architecture; the form of the prior on the parameters; and the form of the noise model. Different values for the hyperparameters α and β define different sub-models. To infer α and β given the data, we simply apply the rules of probability theory:

$$P(\alpha, \beta|D, \mathcal{H}) = \frac{P(D|\alpha, \beta, \mathcal{H})P(\alpha, \beta|\mathcal{H})}{P(D|\mathcal{H})}. \tag{18}$$

The data-dependent factor $P(D|\alpha, \beta, \mathcal{H})$ is the normalizing constant from our previous inference (14); we call this factor the 'evidence' for α and β.

Assuming we have only weak prior knowledge about the noise level and the smoothness of the interpolant, the evidence framework optimizes the constants α and β by finding the maximum of the evidence for α and β. If we can approximate the posterior probability distribution in equation (15) by a single Gaussian,

$$P(\mathbf{w}|D, \alpha, \beta, \mathcal{H}) \simeq \frac{1}{Z'_M} \exp\left(-M(\mathbf{w}_{\rm MP}) - \frac{1}{2}(\mathbf{w} - \mathbf{w}_{\rm MP})^{\rm T}\mathbf{A}(\mathbf{w} - \mathbf{w}_{\rm MP})\right), \tag{19}$$

where $\mathbf{A} = -\nabla\nabla \log P(\mathbf{w}|D, \alpha, \beta, \mathcal{H})|_{\mathbf{w}_{\rm MP}}$, then the evidence for α and β can be written down:

$$\log P(D|\alpha, \beta, \mathcal{H}) = \log \frac{Z'_M}{Z_W(\alpha) Z_D(\beta)} \tag{20}$$

$$= -M(\mathbf{w}_{\rm MP}) - \frac{1}{2}\log\det\left(\frac{\mathbf{A}}{2\pi}\right) - \log Z_W(\alpha) - \log Z_D(\beta). \tag{21}$$

The terms $-\frac{1}{2}\log\det(\mathbf{A}/2\pi) - \log Z_W(\alpha)$ constitute the log of a volume factor which penalizes small values of α. The maximum of the evidence has some elegant properties which allow it to be located efficiently by on-line re-estimation techniques. Technically, there may be multiple evidence maxima, but this is not common when the model space is well matched to the data. As shown in (Gull 1989; MacKay 1992a), the maximum evidence $\alpha = \alpha_{\rm MP}$ satisfies the following implicit equation:

$$1/\alpha_{\rm MP} = \sum_i {w_i^{\rm MP}}^2/\gamma \tag{22}$$

where $\mathbf{w}^{\rm MP}$ is the parameter vector which minimizes the objective function $M = \beta E_D + \alpha E_W$, and γ is the 'number of well-determined parameters', given by

$$\gamma = k - \alpha \mathrm{Trace}(\mathbf{A}^{-1}). \tag{23}$$

Here k is the total number of parameters, and the matrix $\mathbf{A}^{-1}$ is the variance-covariance matrix that defines error bars on the parameters $\mathbf{w}$. Thus $\gamma \to k$ when the parameters are all well-determined in relation to their prior range, which is defined by α. The quantity γ always lies between 0 and k. Recalling that α corresponds to the variance $\sigma_w^2 = 1/\alpha$ of the assumed distribution for $\{w_i\}$, equation (22) specifies an intuitive condition for matching the prior to the data: the variance is estimated by $\sigma_w^2 = \langle w^2 \rangle$, where the average is over the γ effective well determined parameters, the other $k - \gamma$ effective parameters having been set to zero by the prior.

Similarly, in a regression problem with a Gaussian noise model, the maximum evidence value of β satisfies:

$$1/\beta_{\rm MP} = 2E_D/(N - \gamma). \tag{24}$$

Since $2E_D$ is the sum of squared residuals, this expression can be recognized as a variance estimator with the number of degrees of freedom set to γ.

Equations (22) and (24) can be used as re-estimation formulae for α and β. The computational overhead for these Bayesian calculations is not severe: it is only necessary to evaluate properties of the error bar matrix, $\mathbf{A}^{-1}$. One might argue that this matrix should be evaluated anyway, in order to compute the uncertainty in a model's predictions. This matrix may be evaluated explicitly (MacKay 1992b; Thodberg 1993; Bishop 1992; Hassibi and Stork 1993), which does not take significant time when the number of parameters is small (a few hundred). For large problems these calculations can be performed more efficiently using algorithms which evaluate products $\mathbf{A}\mathbf{v}$ without explicitly evaluating $\mathbf{A}$ (Skilling 1993; Pearlmutter 1994).

Thodberg (1993) combines equations (22) and (24) into a single re-estimation formula for the ratio α/β. This ratio is all that matters if only the best-fit parameters are of interest. An advantage of keeping α and β distinct, however, is that knowledge from other sources (bounds on the value of the noise level, for example) can be explicitly incorporated. Also, if we move to noise models more sophisticated than a Gaussian, a separation of these two hyperparameters is essential. Finally, if we wish to construct error bars, or generate a sample from the posterior parameter distribution for use in a Monte Carlo estimation procedure, the separate values of α and β become relevant.

3.1. Relationship to ideal hierarchical Bayesian modelling

Bayesian probability theory has been used above to *optimize* the hyperparameters α and β. This procedure of setting the hyperparameters of a model using the data is known in some circles as 'generalized maximum likelihood' or 'empirical Bayes'. Ideally we would *integrate over* these 'nuisance parameters' to obtain the posterior distribution over the parameters $P(\mathbf{w}|D, \mathcal{H})$ and the predictive distribution $P(\mathbf{t}^{(N+1)}|D, \mathcal{H})$; however, the optimization of the hyperparameters can be viewed as an accurate approximation to this ideal procedure. If a hyperparameter is well determined by the data, integrating over it is very much like estimating the hyperparameter from the data and then using that estimate in our equations (Bretthorst 1988; Gull 1988;

MacKay 1995b). The intuition is that if, in the predictive distribution

$$P(\mathbf{t}^{(N+1)}|D,\mathcal{H}) = \int \mathrm{d}\alpha\; P(\mathbf{t}^{(N+1)}|D,\alpha,\mathcal{H})P(\alpha|D,\mathcal{H}),$$

the probability $P(\alpha|D,\mathcal{H})$ is sharply peaked at $\alpha = \alpha_{\mathrm{MP}}$ with width $\sigma_{\log\alpha|D}$, and if the distribution $P(\mathbf{t}^{(N+1)}|D,\alpha,\mathcal{H})$ varies slowly with $\log\alpha$ on a scale of $\sigma_{\log\alpha|D}$, then $P(\alpha|D,\mathcal{H})$ is effectively a delta function, so that:

$$P(\mathbf{t}^{(N+1)}|D,\mathcal{H}) \simeq P(\mathbf{t}^{(N+1)}|D,\alpha_{\mathrm{MP}},\mathcal{H}).$$

Now the error bars on $\log\alpha$ and $\log\beta$, found by differentiating $\log P(D|\alpha,\beta,\mathcal{H})$ twice, are (MacKay 1992a):

$$\sigma^2_{\log\alpha|D} \simeq 2/\gamma\,; \quad \sigma^2_{\log\beta|D} \simeq 2/(N-\gamma). \tag{25}$$

Thus the error introduced by optimizing α and β is expected to be small for $\gamma \gg 1$ and $N-\gamma \gg 1$. How large γ needs to be depends on the problem, but for many neural network problems a value of γ even as small as 3 may suffice, since the predictions of an optimized network are often insensitive to an e-fold change in α.

It is often possible to integrate over α and β early in the calculation, obtaining a true prior and a true likelihood (Bretthorst 1988; Gull 1988). Some authors have recommended this procedure (Buntine and Weigend 1991; Wolpert 1993), but it is counterproductive as far as practical manipulation is concerned (Gull 1988; MacKay 1995b): the resulting true posterior is a skew-peaked distribution, and apart from Monte Carlo methods there are currently no computational techniques which can cope directly with such distributions.

Later a correction term will be given which approximates the integration over α and β when predictions are made, i.e., as a final step in the calculations.

3.2. Multiple regularization constants

For simplicity, it has so far been assumed that there is only a single class of weights, which are modelled as coming from a single Gaussian prior with $\sigma_w^2 = 1/\alpha$. However, in dimensional terms, weights usually fall into three or more distinct classes; for the network of equation (9), for example, the three dimensional classes are $\{w^{(1)}\}$, $\{\theta^{(1)}\}$ and $\{w^{(2)}, \theta^{(2)}\}$. For consistency, weights from different classes should not be modelled as coming from a single prior. Assuming a Gaussian prior for each class, we can define $E_{W(c)} = \sum_{i\in c} w_i^2/2$, and assign the prior:

$$P(\{w_i\}|\alpha_c,\mathcal{H}) = \frac{1}{\prod Z_{W(c)}} \exp\left(-\sum_c \alpha_c E_{W(c)}\right), \tag{26}$$

where each class now has its own scale parameter or 'weight decay rate' α_c. It is often found that network performance is enhanced by this division of weights into different classes. The automatic relevance determination model (section 7) uses this prior.

The evidence framework optimizes the decay constants by finding their most probable value, i.e., the maximum over $\{\alpha_c\}$ of $P(D|\{\alpha_c\},\mathcal{H})$. And, as before, the maximum evidence $\{\alpha_c\}$ satisfy the implicit equations:

$$1/\alpha_c^{\mathrm{MP}} = \sum_{i\in c} {w_i^{\mathrm{MP}}}^2/\gamma_c, \tag{27}$$

where $\mathbf{w}^{\rm MP}$ is the parameter vector which minimizes the objective function $M = \beta E_D + \sum_c \alpha_c E_{W(c)}$, and γ_c is the number of well-determined parameters in class c, $\gamma_c = k_c - \alpha_c {\rm Trace}_c(\mathbf{A}^{-1})$; k_c is the number of parameters in class c, and the trace is over those parameters only.

For simplicity, the following discussion will assume once more that there is only a single parameter α.

4. Model comparison

The evidence framework divides our inferences into distinct 'levels of inference', of which we have now completed the first two.

Level 1: Infer the parameters $\mathbf{w}$ for given values of α, β:

$$P(\mathbf{w}|D, \alpha, \beta, \mathcal{H}) = \frac{P(D|\mathbf{w}, \alpha, \beta, \mathcal{H}) P(\mathbf{w}|\alpha, \beta, \mathcal{H})}{P(D|\alpha, \beta, \mathcal{H})}. \tag{28}$$

Level 2: Infer α, β:

$$P(\alpha, \beta|D, \mathcal{H}) = \frac{P(D|\alpha, \beta, \mathcal{H}) P(\alpha, \beta|\mathcal{H})}{P(D|\mathcal{H})}. \tag{29}$$

Level 3: Compare models:

$$P(\mathcal{H}|D) \propto P(D|\mathcal{H}) P(\mathcal{H}).$$

There is a pattern in these three applications of Bayes' theorem: at each of the higher levels 2 and 3, the data-dependent factor [e.g., at level 2, $P(D|\alpha, \beta, \mathcal{H})$] is precisely the normalizing constant (the 'evidence') from the preceding level of inference. This pattern of inference continues when we compare different models $\mathcal{H}$, which might use different architectures, preprocessings, regularizers or noise models. Alternative models are ranked by evaluating $P(D|\mathcal{H})$, the normalizing constant of inference (29).

In the preceding section we reached level 2 by using a Gaussian approximation to $P(\mathbf{w}|D, \alpha, \beta, \mathcal{H})$. We now evaluate the evidence for $\mathcal{H}$. Using a separable Gaussian approximation for $P(\log \alpha, \log \beta|D, \mathcal{H})$, we obtain the estimate

$$P(D|\mathcal{H}) \simeq P(D|\alpha_{\rm MP}, \beta_{\rm MP}, \mathcal{H}) P(\log \alpha_{\rm MP}, \log \beta_{\rm MP}|\mathcal{H}) \, (2\pi)^{\frac{1}{2}} \sigma_{\log \alpha|D} \, (2\pi)^{\frac{1}{2}} \sigma_{\log \beta|D}, \tag{30}$$

where $P(D|\alpha_{\rm MP}, \beta_{\rm MP}, \mathcal{H})$ is obtained from equation (21), and the error bars on $\log \alpha$ and $\log \beta$ are as given in equation (25). This Gaussian approximation over α and β holds good for $\gamma \gg 1$ and $N - \gamma \gg 1$ (MacKay 1995b).

4.1. Multi-modal distributions

The preceding exposition falls into difficulty if the posterior distribution $P(\mathbf{w}|D, \alpha, \beta, \mathcal{H})$ is significantly multi-modal; this is usually the case for multi-layer neural networks. However we can persist with the use of Gaussian approximations if we introduce two modifications.

First, we recognize that a typical optimum $\mathbf{w}_{\rm MP}$ will be related to a number of equivalent optima by symmetry operations, such as interchange of hidden units and inversion of signs of weights. When evaluating the evidence using a local Gaussian approximation, a symmetry factor should be included in equation (20) to take into account these equivalent islands of probability mass. In the case of a net with one hidden layer of H units (equation 9), the appropriate permutation factor is $H!\,2^H$, for general $\mathbf{w}_{\rm MP}$: a factor of $H!$ for permutations of the H hidden units, and a factor of 2 for the reversal of sign of all weights into and out of each hidden unit.

Second, there are multiple optima which are not related to each other by model symmetries. We modify the above framework by changing our goals; specifically, we view each of the local probability peaks as a distinct model. Instead of inferring the posterior over α, β for the entire model $\mathcal{H}$, we allow each local peak of the posterior to choose its own optimal value for these parameters. Similarly, instead of evaluating the evidence for the entire model $\mathcal{H}$, we aim to calculate the posterior probability mass in each local optimum. This procedure is natural if in the final implementation of the model the parameter vector will be set either to a particular value, or a small set of values, with error bars. In this case the probability of an entire model is not as important as the probability of the local solutions we find. The same method of chopping up a complex model space is used in the unsupervised classification system, AutoClass (Hanson *et al.* 1991).

Henceforth, the term 'model' will refer to a pair $\{\mathcal{H}, S_{\mathbf{w}^*}\}$, where $\mathcal{H}$ denotes the model specification and $S_{\mathbf{w}^*}$ specifies a solution neighbourhood around an optimum $\mathbf{w}^*$. Adopting this shift in objective, the Gaussian integrals above can be used without alteration to set α and β and to compare alternative solutions, assuming that the posterior probability consists of well separated islands in parameter space that are roughly Gaussian.

For general α and β the Gaussian approximation over $\mathbf{w}$ will not be accurate; however we only need it to be accurate for the small range of α and β close to their most probable value. For sufficiently large amounts of data compared to the number of parameters, this approximation is expected to hold. Practical experience indicates that this is a useful approximation for many real problems.

5. Error bars and predictions

Having progressed up the three levels of modelling, the next inference task is to make predictions with our adapted model. It is common practice simply to use the most probable values of $\mathcal{H}, \mathbf{w}$, etc., when making predictions, but this is not optimal, as we shall see. Bayesian prediction of a new datum $\mathbf{t}^{(N+1)}$ involves *marginalizing* over our uncertainty at all levels:

$$P(\mathbf{t}^{(N+1)}|D) = \sum_{\mathcal{H}} \int \mathrm{d}\alpha\, \mathrm{d}\beta \int \mathrm{d}^k\mathbf{w}\; P(\mathbf{t}^{(N+1)}|\mathbf{w}, \alpha, \beta, \mathcal{H}) P(\mathbf{w}, \alpha, \beta, \mathcal{H}|D).$$

The evaluation of the distribution $P(\mathbf{t}^{(N+1)}|\mathbf{w}, \alpha, \beta, \mathcal{H})$ is generally straightforward, requiring a single forward pass through the network. Typically, marginalization over $\mathbf{w}$ and $\mathcal{H}$ affects the predictive distribution significantly, but integration over α and β has a lesser effect.

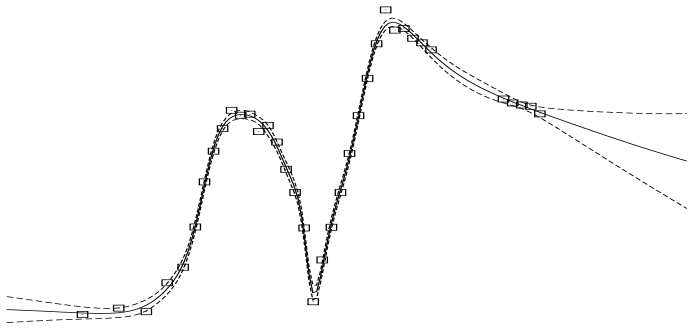


Figure 8. Error bars on the predictions of a trained regression network. The solid line gives the predictions of the best fit parameters of a multilayer perceptron trained on the data points shown. The error bars (dotted lines) are those produced by the uncertainty of the parameters $\mathbf{w}$. Notice that the error bars become larger where the data are sparse.

5.1. Implementation

Marginalization can sometimes be done analytically. When this is not possible, Monte Carlo methods (Neal 1993) may be used. The average of a function of an uncertain parameter $\mathbf{q}$, $t(\mathbf{q})$, under $P(\mathbf{q})$, can be estimated with tolerable error by obtaining a small number of samples from $P(\mathbf{q})$ and evaluating the mean of the observed values of $t(\mathbf{q})$. The variance of this estimator is independent of the dimensionality of $\mathbf{q}$, and scales inversely with the sample size. A cheap and cheerful way of obtaining such samples from $P(\mathbf{w}|D, \alpha, \beta, \mathcal{H})$ is described later in section (9). Here, methods based on Gaussian approximations are described.

5.2. Error bars in regression

Integrating first over $\mathbf{w}$ for fixed α and β, the predictive distribution is:

$$P(\mathbf{t}^{(N+1)}|D, \alpha, \beta, \mathcal{H}) = \int \mathrm{d}^k\mathbf{w}\, P(\mathbf{t}^{(N+1)}|\mathbf{w}, \beta, \mathcal{H}) P(\mathbf{w}|D, \alpha, \beta, \mathcal{H}). \qquad (31)$$

Let us first take the case of prediction of a single target t. If a Gaussian approximation is made for the posterior $P(\mathbf{w}|D, \alpha, \beta, \mathcal{H})$; if the noise model is Gaussian; and if a local linearization of the output is made as a function of the parameters,

$$y(\mathbf{x}^{N+1}, \mathbf{w}) \simeq y(\mathbf{x}^{N+1}; \mathbf{w}_{\mathrm{MP}}) + \mathbf{g} \cdot (\mathbf{w} - \mathbf{w}_{\mathrm{MP}}),$$

where $\mathbf{g}$ is the 'sensitivity' of the output to the parameters,

$$\mathbf{g} = \left.\frac{\partial y}{\partial \mathbf{w}}\right|_{\mathbf{x}^{N+1}, \mathbf{w}_{\mathrm{MP}}};$$

then the predictive distribution (31) is a straightforward Gaussian integral. This distribution has mean $y(\mathbf{x}^{N+1}, \mathbf{w}_{\mathrm{MP}})$, and variance $\sigma^2_{t|\alpha,\beta} = \mathbf{g}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{g} + \sigma^2_\nu$, where $\mathbf{A} = -\nabla\nabla \log P(\mathbf{w}|D, \alpha, \beta, \mathcal{H})$. Figure 8 illustrates the error bars corresponding to the interesting term in this expression, $\mathbf{g}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{g}$, for the predictions of a trained multilayer perceptron with ten hidden units.

5.2.1. Joint error bars on multiple predictions. Often one wishes simultaneously to predict several targets $\{t^{(u)}\}_{u=1}^{U}$, when a single network has multiple outputs, or when we have a sequence of different inputs and wish to predict all the corresponding targets, or both. Let the sensitivities of these outputs to the parameters be $\mathbf{g}_{(u)} \equiv \frac{\partial y^{(u)}}{\partial \mathbf{w}}$. Then the covariance matrix of the values $\{y^{(u)}\}$ is

$$\mathbf{Y} = \mathbf{G}^{\mathrm{T}} \mathbf{A}^{-1} \mathbf{G}, \tag{32}$$

where the matrix $\mathbf{G} = \left[\mathbf{g}_{(1)} \mathbf{g}_{(2)} \ldots \mathbf{g}_{(U)}\right]$. This matrix expresses, within the Gaussian approximation, the expected variances and covariances of the predicted variables. These correlated error ellipsoids have been demonstrated for a two output regression network in MacKay (1992b).

5.2.2. Additional uncertainty produced by uncertainty of hyperparameters. Integration over the regularization constants α and β contributes an additional variance in only one direction; to leading order in γ^{-1}, $P(\mathbf{t}^{(N+1)}|D, \mathcal{H})$ is normal, with variance (MacKay 1995b):

$$\sigma_t^2 = \mathbf{g}^{\mathrm{T}} \left(\mathbf{A}^{-1} + (\sigma^2_{\log \alpha | D} + \sigma^2_{\log \beta | D}) \mathbf{w}'_{\mathrm{MP}} \mathbf{w}'^{\mathrm{T}}_{\mathrm{MP}} \right) \mathbf{g} + \sigma_\nu^2,$$

where $\mathbf{w}'_{\mathrm{MP}} \equiv \partial \mathbf{w}_{\mathrm{MP}|\alpha} / \partial(\log \alpha) = \alpha \mathbf{A}^{-1} \mathbf{w}_{\mathrm{MP}}$, and $\sigma^2_{\log \alpha | D} = \frac{2}{\gamma}$ and $\sigma^2_{\log \beta | D} = \frac{2}{N-\gamma}$.

5.3. Integrating over models: committees

If we have multiple regression models $\mathcal{H}$, then our predictive distribution is obtained by summing together the predictive distribution of each model, weighted by its posterior probability. If a single prediction is required and the loss function is quadratic, the optimal prediction is a weighted mean of the models' predictions $y(\mathbf{x}^{(N+1)}; \mathbf{w}_{\mathrm{MP}}, \mathcal{H})$. The weighting coefficients are the posterior probabilities, which are obtained from the evidences $P(D|\mathcal{H})$. If we cannot evaluate these accurately then alternative pragmatic prescriptions for the weighting coefficients exist (Thodberg 1993; Breiman 1992; MacKay 1994).

5.4. Error bars in classification

In the case of linearized regression discussed above, the mean of the predictive distribution (31) was identical to the prediction made by the mean, $\mathbf{w}_{\mathrm{MP}}$. This is not the case in classification problems. The best fit parameters give over-confident predictions. A non-Bayesian approach to this problem is to down-weight all predictions by an empirically determined factor (Copas 1983). But a Bayesian viewpoint helps us to understand the cause of the problem, and provides a straightforward solution that is demonstrably superior to this ad hoc procedure.

This issue is illustrated for a simple two-class problem in figure 9. Figure 9*a* shows a binary data set, which, in figure 9*b* is modelled with a linear logistic function as defined in equation (17), except including a bias θ so that the decision boundary is not obliged to pass through the origin. The best fit parameter values give predictions shown by three contours. Are these reasonable predictions? Consider new data arriving at points A and B. The best-fit model assigns both of these examples probability 0.9 of being in class 1. But intuitively we might be inclined to assign a less confident probability (closer to 0.5) at point B, since it is further from the training data.

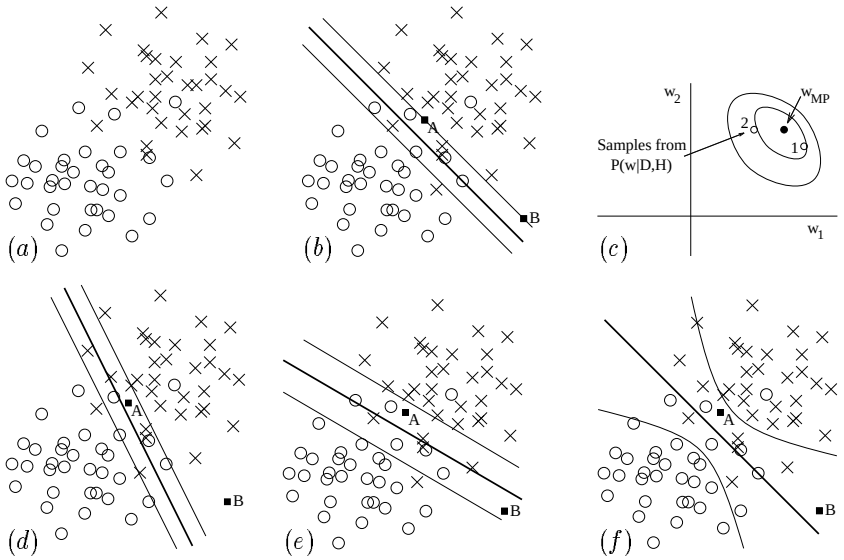


Figure 9. Integrating over error bars in a classifier. *a)* A binary data set. The two classes are denoted by $\times$=1, $\circ$=0. *b)* The data are modelled with a linear logistic function. Here the *best-fit* model is shown by its 0.1, 0.5 and 0.9 predictive contours. The best fit model assigns probability 0.9 of being in class 1 to both inputs A and B. *c)* The posterior probability distribution of the model parameters, $P(\mathbf{w}|D,\mathcal{H})$ (schematic; the third parameter, the bias, is not shown). The parameters are not perfectly determined by the data. Two typical samples from the posterior are indicated by the points labeled 1 and 2. The following two panels show the corresponding classification contours. *d)* Sample 1. *e)* Sample 2. Notice how the point B is classified differently by these different plausible classifiers, whereas the classification of A is relatively stable. *f)* We obtain the Bayesian predictions by integrating over the posterior distribution of $\mathbf{w}$. The width of the decision boundary increases as we move away from the data (point B). See text for further discussion.

Precisely this result is obtained by marginalizing over the parameters, whose posterior probability distribution is depicted in figure 9*c*. Random samples from the posterior define different classification surfaces, as illustrated by two samples in figures 9*d*, *e*. The point B is classified differently by these different plausible classifiers, whereas the classification of A is relatively stable. We obtain the Bayesian predictions (figure 9*f*) by averaging together the predictions of the plausible classifiers. The resulting 0.5 contour remains similar to that for the best-fit parameters. However, the width of the decision boundary region increases as we move away from the data, in full accordance with intuition.

The Bayesian approach is superior because the best-fit model's predictions are *selectively* down-weighted, to a different degree for each test case. The consequence is that a Bayesian classifier is better able to identify the points where the classification is uncertain. This pleasing behaviour results simply from a mechanical application of the rules of probability.

For a binary classifier, a numerical approximation to the integral over a Gaussian posterior distribution is given in (MacKay 1992c). An equivalent approximation for a multi-class classifier is yet to be implemented.

This marginalization can also be done by Monte Carlo methods. A disadvantage of a straightforward Monte Carlo approach would be that it is a poor way of estimating the probability of an improbable event, i.e., a $P(t|D,\mathcal{H})$ that is very close to zero, if the improbable event is most likely to occur in conjunction with improbable parameter values. In such cases one might instead temporarily add the event in question to the data set, and evaluate the evidence $P(D,t^{(N+1)}|\mathcal{H})$. The desired probability is obtained by comparing this with either the previous evidence $P(D|\mathcal{H})$, or the evidence assuming the complementary virtual data set $P(D,\overline{t^{(N+1)}}|\mathcal{H})$.

6. Pruning

The evidence can serve as a guide for *pruning*, i.e. changing the model by setting selected parameters to zero. Thodberg (1993) has done this in the straightforward way: each parameter in the network is tentatively pruned, then the new model is optimized and the evidence is evaluated to decide whether to accept the pruning.

Here an alternative procedure using the Gaussian approximation is described. Whether pruning is in fact a good idea is questioned later in sections 7 and 10.4.

Suppose that the model is locally linear, and α and β are well-determined, so that the model's parameters have prior and posterior distributions which are exactly Gaussian (for brevity α and β are omitted from the conditioning propositions in this section):

$$P(\mathbf{w}|\mathcal{H}) = \frac{1}{Z_W}\exp\left(-\alpha\tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{I}\mathbf{w}\right),$$
$$P(\mathbf{w}|D,\mathcal{H}) = \frac{1}{Z_M}\exp\left(-M_{\mathrm{MP}}-\tfrac{1}{2}\Delta\mathbf{w}^{\mathrm{T}}\mathbf{A}\Delta\mathbf{w}\right),$$

where $\Delta\mathbf{w} = \mathbf{w}-\mathbf{w}_{\mathrm{MP}}$. The log evidence for $\mathcal{H}$ is

$$\log P(D|\mathcal{H}) = -M_{\mathrm{MP}} - \frac{1}{2}\log\det\mathbf{A} + \frac{1}{2}\log\det\alpha\mathbf{I} + \text{const.}$$

We are interested in evaluating the difference in evidence between this model $\mathcal{H}$ and an alternative model $\mathcal{H}_{\bar{s}}$, where the subscript $\bar{s}$ denotes the setting to zero of parameter s. The remaining parameters of $\mathcal{H}_{\bar{s}}$ still have a Gaussian distribution, but are confined to the constraint surface $\mathbf{w}\cdot\mathbf{e}_s = 0$, where $\mathbf{e}_s$ is the unit vector in the direction of the deleted parameter.

We can evaluate the difference in evidence between $\mathcal{H}$ and $\mathcal{H}_{\bar{s}}$ by (a) finding the location of the new optimum $\mathbf{w}_{\bar{s}}^{\mathrm{MP}}$ and evaluating the change in M_{MP}, $\Delta M_{\mathrm{MP}} = -\frac{1}{2}\Delta\mathbf{w}^{\mathrm{T}}\mathbf{A}\Delta\mathbf{w}$ there; and (b) evaluating the change in log determinant of the distribution.

The first task is accomplished by introducing a Lagrange multiplier. We find:

$$\mathbf{w}_{\bar{s}}^{\mathrm{MP}} = \mathbf{w}_{\mathrm{MP}} - \frac{w_s}{\sigma_s^2}\mathbf{A}^{-1}\mathbf{e}_s; \quad \Delta M_{\mathrm{MP}} = \frac{w_s^2}{2\sigma_s^2},$$

where the marginal error bars on parameter w_s are $\sigma_s^2 = \mathbf{e}_s^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{e}_s = \mathbf{A}^{-1}_{ss}$. The quantity ΔM_{MP} is the saliency term which has been advocated as a guide for 'optimal brain damage' (LeCun *et al.* 1990; Hassibi and Stork 1993). The change in evidence, however, involves a second 'Occam factor' term which is simple to calculate. The change in evidence when a single parameter s is deleted is:

$$\log P(D|\mathcal{H}) - \log P(D|\mathcal{H}_{\bar{s}}) = \frac{w_s^2}{2\sigma_s^2} + \log\frac{\sigma_s}{\sigma_w},$$

where σ_w^2 is the prior variance for the parameter w_s. This objective function can be used to select which parameter to delete. It also tells us to stop pruning (or, to be precise, that pruning is yielding a less probable model) once it is positive, for all parameters, s.

An equivalent expression can be worked out for the case of simultaneous pruning of multiple parameters. Consider pruning of k_s parameters. We obtain the joint $(k_s \times k_s)$ covariance matrix for the pruned parameters, Σ_s, by reading out the appropriate sub-matrix of $\mathbf{A}^{-1}$. Then the evidence difference is

$$\log P(D|\mathcal{H}) - \log P(D|\mathcal{H}_{\bar{s}}) = \frac{1}{2}\mathbf{w}_s^{\mathrm{T}}\Sigma_s^{-1}\mathbf{w}_s + \log \frac{\det^{1/2}\Sigma_s}{\prod_1^{k_s}\sigma_w}.$$

Thus the Bayesian formulae incorporate small additional volume terms not included in the 'brain surgery' literature. In my opinion the pruning technique is now superseded by the use of more sophisticated regularizers, as discussed in section 7.

7. Automatic relevance determination

The automatic relevance determination (ARD) model (D J C MacKay and R M Neal, unpublished) can be implemented with the methods described in the previous sections.

Suppose that in a regression problem there are many input variables, of which some are irrelevant to the prediction of the output variable. Because a finite data set will show random correlations between the irrelevant inputs and the output, any conventional neural network (even with weight decay) will fail to set the coefficients for these junk inputs to zero. Thus the irrelevant variables will hurt the model's performance, particularly when the variables are many and the data are few. Selecting the best input variables is a non-trivial problem.

In the ARD model we define a prior over the regression parameters that embodies the concept of uncertain relevance, so that the model is effectively able to infer which variables are relevant and then switch the others off. This is achieved in a simple and 'soft' way by introducing multiple weight decay constants, one 'α' associated with each input. The decay rates for junk inputs will automatically be inferred to be large, preventing those inputs from causing significant overfitting.

The ARD model uses the prior of equation (26). For a network having one hidden layer, the weight classes are: one class for each input, consisting of the weights from that input to the hidden layer; one class for the biases to the hidden units; and one class for each output, consisting of its bias and all the weights from the hidden layer. Control of the ARD model can be implemented using equation (27).

Automatic relevance determination has been found to be a useful alternative to the technique of pruning (section 6), which also embodies the concept of relevance, but in a discrete manner. Possible advantages of ARD include:

(i) Pruning using Bayesian model comparison requires the evaluation of determinants or inverses of large Hessian matrices, which may be ill-conditioned. ARD, on the other hand, can be implemented using evaluations of the trace of the Hessian alone, which is more robust. In a stochastic dynamics implementation of ARD, in fact, no matrix computations are needed at all: the conditional distribution of α_c given $\mathbf{w}$ is a Gamma distribution, so a Gibbs sampling procedure can be used, alternately sampling $\{\alpha_c\}$ given $\mathbf{w}$ and $\mathbf{w}$ given $\{\alpha_c\}$.

(ii) Compared with a non-Bayesian cross-validation method, ARD simultaneous infers the utility of large numbers of possible input variables. With only a single cross-validation measure, one might have to explicitly prune one variable at a time in order to estimate which variables are useful. In contrast, ARD returns two *vectors* measuring the relevance of all input variables x_i: the regularization constants α_i and the 'well-determinednesses' γ_i, and it suppresses the irrelevant inputs without further intervention.

(iii) ARD allows large numbers of input variables of unknown relevance to be left in the model without harm.

Practical problems found in implementing the ARD model using Gaussian approximations are:

(i) If irrelevant variables are not explicitly pruned from a large model, then computation times remain wastefully large.

(ii) The presence of large numbers of irrelevant variables in a model hampers the calculation of the evidence for different models. Numerical problems arise with calculation of determinants of Hessians. This does not interfere with the Bayesian optimization of regularization constants, but it prevents the use of Bayesian model comparison methods.

(iii) Although the ARD model is intended to embody a soft version of pruning, the approximations of the evidence framework can lead to singularities with an α_c going to ∞, if the signal to noise ratio is low; this causes inputs to be irreversibly shut off.

In spite of these reservations, I am confident that the right direction for adaptive modelling methods lies in the replacement of discrete model choices by continuous control parameters.

A common concern is whether the extra hyperparameters $\{\alpha_c\}$ might cause overfitting. There is no cause for worry. There are two reasons. First, if we can evaluate the evidence, then we can evaluate objectively whether the new model is more probable, given the data. The extra parameters are penalized by Occam factors so, eventually, if we massively increased the number of hyperparameters, an evidence maximum would be reached. In fact the Occam factors for regularization constants are very weak; the error bars on $\log \alpha_c$ scale only as $1/\sqrt{\gamma_c}$. This fact relates to the second reason why the extra hyperparameters $\{\alpha_c\}$ are incapable of causing overfitting of the data. Only the parameters $\mathbf{w}$ can overfit noise, and the worst overfitting occurs when the regularization constants α_c are all switched to zero. Thus the extra hyperparameters have no effect on the worst-case capacity of the model. Their effect is a positive one, namely a damping out of unneeded degrees of freedom in the model. There is a weak probabilistic penalty for the extra parameters, simply because they increase the variety of simple data sets that the model is capable of predicting. A model with only one hyperparameter α is capable of realising only one 'flavour of simplicity', namely 'all parameters w_i are small', and one flavour of complexity, 'most parameters w_i are big'. A model having, say, three hyperparameters $\{\alpha_c\}$, can predict a total of $2^3 = 8$ flavours of simplicity and complexity including the two above as special cases.

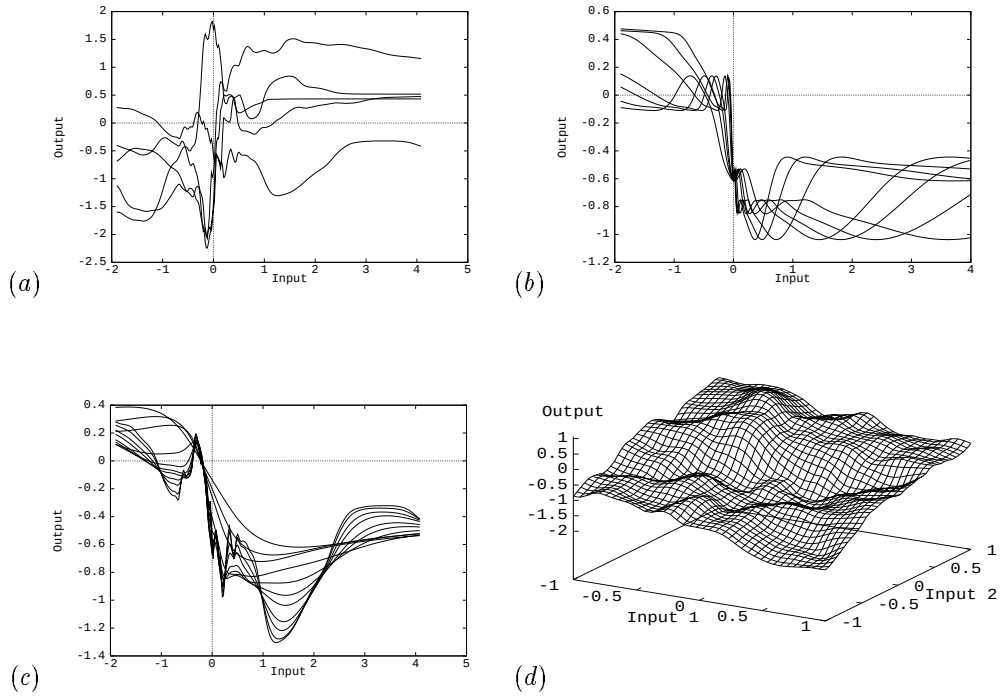


Figure 10. Samples from the prior of (a–c) a one input network; and (d) a two input network.

a) **Varying number of hidden units:** For each curve a different number of hidden units, H, is used: 100, 200, 400, 800 and 1600. The regularization constants for the input weights and hidden unit biases are fixed at $\sigma^w_{\rm in} = 40$ and $\sigma^w_{\rm bias} = 8$. The output weights have $\sigma^w_{\rm out} = 1/\sqrt{H}$ to keep the dynamic range of the function constant.

b) **Varying $\sigma^w_{\rm in}$:** Varying $\sigma^w_{\rm in}$ alone changes both the characteristic scale length of the oscillations and the overall width of the region in input space in which the action occurs. $\{H, \sigma^w_{\rm bias}, \sigma^w_{\rm out}\} = \{400, 2.0, 0.05\}$. $\sigma^w_{\rm in} = 40$, 30, 20, 10, 8, 6, 4. The smaller the value of $\sigma^w_{\rm in}$, the less steep the function.

c) **Varying $\sigma^w_{\rm bias}$:** $H = 400$, $\sigma^w_{\rm out} = 0.05$. The same seed was used for all these samples, so that all the weights are simply scaled by the regularization constants as the 'movie' progresses. The ratio $\sigma^w_{\rm in}/\sigma^w_{\rm bias} = 5.0$ in all cases, so as to keep the action in the horizontal range ± 5.0. The constant $\sigma^w_{\rm bias}$ took the values: 8, 6, 4, 3, 2, 1.6, 1.2, 0.8, 0.4, 0.3, 0.2. This constant determines the total number of fluctuations in the function. The constant $\sigma^w_{\rm in}$ determines the input scale on which the fluctuations occur.

d) Two input network with $\{H, \sigma^w_{\rm in}, \sigma^w_{\rm bias}, \sigma^w_{\rm out}\} = \{400, 8.0, 8.0, 0.05\}$.

8. Implicit priors

It is interesting to examine what sort of functions are generated when networks' parameters are set by sampling from the prior distributions of equations (13),(26). The study of these prior distributions provides guidelines for the expected scaling behaviour of regularization constants with the number of hidden units, H. It also identifies which control parameters are responsible for controlling the 'complexity' of

the function. For regression nets with one hidden layer of tanh units and Gaussian priors on the three dimensional classes of weights, $\{w^{(1)}\}$, $\{\theta^{(1)}\}$ and $\{w^{(2)}, \theta^{(2)}\}$, we find the following interesting result (Neal 1994).

In the limit as $H \to \infty$, the complexity of the functions generated by the prior is independent of the number of hidden units. The prior on the input to hidden weights determines the spatial scale (over the inputs) of variations in the function. The prior on the biases of the hidden units determines the characteristic number of fluctuations in the function. The prior on the output weights determines the vertical scale of the output.

Figures $10a-c$ illustrate samples from priors for a one input, one output network with a large number of hidden units. Figure $10a$ illustrates that as the number of hidden units H is increased, while keeping $\{\sigma^w_{\rm in}, \sigma^w_{\rm bias}, (\sigma^w_{\rm out}\sqrt{H})\}$ fixed, the properties of a random sample from the prior remain stable. (The output weights must get smaller in accordance with $\sigma^w_{\rm out} \propto 1/\sqrt{H}$, in order to keep constant the vertical range of the function, which is a sum of H independent random variables with finite variance.) In fact, in this limit, the prior over functions tends to a non-trivial Gaussian process(Neal 1994). Figure $10b$ illustrates the effect of varying $\sigma^w_{\rm in}$ alone. Finally, figure $10c$ illustrates the effect of varying both $\sigma^w_{\rm in}$ and $\sigma^w_{\rm bias}$ in such a way as to keep constant the range of the 'action' over the input variable $\sim \sigma^w_{\rm bias}/\sigma^w_{\rm in}$. The parameter $\sigma^w_{\rm bias}$ determines the total number of fluctuations in the function.

Progressing to multiple inputs, we obtain figure $10d$ by setting the weights into a 2:400:1 net to random values and plotting the output of the net. The picture shows that a typical sample is a homogeneous 'random-looking' function even though the hidden units' activities are based on linear functions of the inputs.

The prior distribution over functions is symmetrical about zero, both in the input space and the output space. It is therefore wise, if this Bayesian model is used, to preprocess the inputs and targets so that zero is at the expected centre of the action.

9. Cheap and cheerful implementations

The following methods can be used to solve the tasks of automatic optimization of $\{\alpha_c\}$ and β (section 3) and calculation of error bars on parameters and predictions (section 5) without calculation of Hessians, or sophisticated Monte Carlo methods. These methods depend on the same Gaussian assumptions as does the rest of this paper; further approximations are also made.

9.1. Cheap approximations for optimization of α and β

On neglecting the distinction between well determined and poorly determined parameters, we obtain the following update rules for α and β (cf. equations (27) and (24)):

$$\begin{aligned} \alpha_c &:= k_c/2E^c_W \\ \beta &:= N/2E_D . \end{aligned}$$

This easy-to-program procedure is expected to give singular behaviour when there are a large number of poorly determined parameters.

9.2. Cheap generation of predictive distributions

A simple way of obtaining random samples from the posterior probability distribution of the parameters has been used in Bayesian image reconstruction (Skilling *et al.* 1991). This approximate procedure is accurate when the noise is Gaussian, and when the model can be treated as locally linear.

(i) Start with a converged network, with parameters $\mathbf{w}^*$, trained on the true data set $D^* = \{\mathbf{x}^{(m)}, \mathbf{t}^{(m)}\}$. Estimate the Gaussian noise level from the residuals using, for example, $\sigma_\nu^2 = \sum(t - y(\mathbf{w}^*))^2/(N-k)$; alternatively estimate σ_ν^2 from a test set.

(ii) Now define a new data set D_1 by adding artificial Gaussian noise of magnitude σ_ν to the outputs in the true data set D^*. Thus $D_1 = \{\mathbf{x}^{(m)}, \mathbf{t}_1^{(m)}\}$, where $\mathbf{t}_1^{(m)} = \mathbf{t}^{(m)} + \nu$, with $\nu \sim \text{Normal}(0, \sigma_\nu^2)$. No noise is added to the inputs.

(iii) Starting from $\mathbf{w}^*$, train a new network on D_1. Call the converged weight vector $\mathbf{w}_1$. Because the data set will be changed little by the added noise, $\mathbf{w}_1$ will be close to $\mathbf{w}^*$, and this optimization should not take long.

(iv) Repeat steps 2 and 3 twelve times, generating a new data set D_j from the original data set D^* each time to obtain a new $\mathbf{w}_j$. Save the list of vectors $\mathbf{w}_j$.

(v) Separately, use each of $\mathbf{w}_1$, $\mathbf{w}_2$, $\ldots \mathbf{w}_{12}$ to make predictions. For example, in the case of time series continuation, use each $\mathbf{w}_j$ by itself to generate an entire continuation.

These predictions can be viewed as samples from the model's predictive distribution. They might be summarized by measuring their mean and variance.

In order to get a true sample from the posterior, we should also perturb the prior. For each weight, the mean to which each weight decays, ordinarily zero, should be randomized at step 2 by sampling from a Gaussian of variance $\sigma_w^2 = 1/\alpha$.

The above method should be particularly useful for obtaining error bars when neural nets are used to forecast a time-series by bootstrapping the network with its own predictions. A full Bayesian treatment of time-series modelling with neural nets has not yet been made.

10. Discussion

10.1. Applications

The methods of sections 2 to 7 have been successfully applied to several practical problems.

Thodberg (1993) has applied these methods to an industrial problem, the prediction of pork fat content from spectroscopic data. The evidence framework yields better performance than standard techniques involving cross-validation. This improvement is attributed to the fact that the Bayesian framework needs no validation set: all the available data can be used for parameter fitting, for optimization of model complexity, and for model comparison.

The automatic relevance determination model (section 7) has also been used to win a recent prediction competition, involving modelling of the energy consumption of

a building (MacKay 1994). Here the success is attributed to the fact that the evidence framework can be used to simultaneously optimize multiple regularization constants $\{\alpha_c\}$ on-line. Over twenty regularization constants were involved in these networks.

10.2. Modelling insights

An advantage of the Bayesian framework for data modelling is that it forces one to make explicit the assumptions made in constructing the model. When a poor modelling assumption is identified, the probabilistic viewpoint makes it easy to design coherent modifications to the model.

For example, in (MacKay 1992b), the standard weight decay model with only one regularization constant α was applied to a two layer regression network. The evidence for different solutions was found to be poorly correlated with their empirical performance. This failure forced home the insight that, for dimensional consistency, at least three different α's are required: one for the input to hidden weights, one for the biases of the hidden units, and one for all the connections to the outputs. The model with multiple regularizers produced solutions with slightly improved empirical performance; and more importantly, the evidence for these solutions was much better correlated with their generalization error. The automatic relevance determination model (section 7) is a straightforward extension to a larger number of hyperparameters.

Here are some examples of other model modifications that are easily motivated from the probabilistic viewpoint. The use of a sum-squared error corresponds to the assumption that the residuals are Gaussian and uncorrelated among the different target variables. In time-series modelling, this may well be a poor model for residuals, which might show local trends. A better model would, for example, assume Gaussian correlations between residuals, such that the data error βE_D is replaced by:

$$\sum_m \left(\beta_0 (t_m - y)^2 + \beta_1 (t_m - y)(t_{m+1} - y) + \beta_2 (t_m - y)(t_{m+2} - y) + \ldots\right).$$

This would modify the 'backprop' algorithm, so that the propagated error signal at each frame would be a weighted combination of the residuals at neighbouring frames. The network would then experience less of an urge to fit local trends in the data. And, when predictions are made, the model of correlations among residuals would be able to capture the current trend and modify the net's predictions accordingly. The evidence would be used to optimize the correlation model's parameters $\beta_0, \beta_1, \beta_2$, etc.

The Gaussian noise model might also be modified to include the possibility of outliers, using a Bayesian robust noise model (Box and Tiao 1973). Probability theory allows us to infer from the data how heavy the tails of the noise model ought to be.

Another assumption is that the output noise level is the same for all input vectors. As discussed in (MacKay 1991:Chapter 6), one could construct a parameterized model of the noise level $\beta(\mathbf{x})$ which could be learned by evidence maximization.

A final example of a probabilistic motivation for a model modification lies in image analysis. If we use a neural net for character recognition, say, then we might expect a well-trained net to have input weights that are spatially correlated. It is desirable to incorporate this prior expectation into the model adaptation process, because such priors on parameters damp out unnecessary degrees of freedom, reduce overfitting, and give rise to more probable models. One way of creating such a correlation is to pre-blur the data through a linear filter before feeding it into the network, and use the traditional uncorrelated prior on the parameters. This is equivalent to keeping the

original inputs and having a correlated prior on the parameters, where the correlations are defined implicitly by the properties of the pre-blur. This procedure has been used fruitfully in character recognition work (Guyon *et al.* 1992).

10.3. Relationship to theories of generalization

Bayesian model comparison assesses within a well defined hypothesis space *how probable* a set of alternative models are. However, we often want to know how well each model is expected to *generalize*. Empirically, the correlation between the evidence and generalization error is often good (MacKay 1992b; Thodberg 1993). But a theoretical connection between the two is not yet established. Here, a brief discussion is given of similarities and differences between the evidence and quantities arising in work on prediction of generalization error.

10.3.1. Relation to 'G.P.E.' Moody's (1992) 'Generalized Prediction Error' is a generalization of Akaike's 'Final Prediction Error' to non-linear regularized models. These are both estimators of generalization error which can be derived without making assumptions about the distribution of residuals between the data and the true interpolant, and without assuming that the true interpolant belongs to some class. Both are derived under the assumption that the observed distribution over the inputs in the training set gives a good approximation to the distribution of future inputs.

The difference between F.P.E. and G.P.E. is that the total number of parameters k in F.P.E. is replaced by an effective number of parameters, which is in fact identical to the quantity γ arising in the Bayesian analysis (23). If E_D is half the sum squared error, then the predicted error per data point is:

$$\text{G.P.E.} = \left(E_D + \sigma_\nu^2 \gamma\right)/N. \tag{33}$$

The added term $\sigma_\nu^2\gamma$ has an intuitive interpretation in terms of overfitting. For every parameter that is well-determined by the data, we unavoidably overfit one 'direction' of noise. This has two effects: it makes E_D smaller than it 'ought to be', by $\sigma_\nu^2/2$, on average; and it means that our predictions vary from the ideal predictions (those that we would make if we had infinite data) so that our prediction error on the same N input points would on average be worse by $\sigma_\nu^2/2$. The sum of these two terms, multiplied by the effective number of well-determined parameters γ, gives the correction term.

Like the log evidence, the G.P.E. has the form of the data error plus a term that penalizes complexity. However, although the same quantity γ arises in the Bayesian analysis, the Bayesian Occam factor does *not* have the same scaling behaviour as the G.P.E. term (see below). And, empirically, the G.P.E. is not always a good predictor of generalization. One reason is that, in the derivation of G.P.E., it is effectively assumed that test samples will be drawn only at the $\mathbf{x}$ locations at which we have already received data. The consequences of this false assumption are most serious for over-parameterized and over-flexible models. An additional distinction between the G.P.E. and the evidence framework is that the G.P.E. is defined for regression problems only; the evidence can be evaluated for regression, classification and density models.

10.3.2. Relation to the effective V-C dimension. Recent work on 'structural risk minimization' (Guyon *et al.* 1992) uses empirically tuned expressions of the form:

$$E_{\rm gen} \simeq E_D/N + c_1 \frac{\log(N/\gamma) + c_2}{N/\gamma} \tag{34}$$

where γ is the 'effective V-C dimension' of the model, and is identical to the quantity in (23). The constants c_1 and c_2 are determined by experiment. The structural risk theory is currently intended to be applied only to nested families of classification models (hence the absence of β: E_D is dimensionless, like G) with monotonic effective V-C dimension, whereas the evidence can be evaluated for any models. Interestingly, the scaling behaviour of this expression (34) is identical to the scaling behaviour of the log evidence (21), subject to two assumptions. First, that the value of the regularization constant satisfies (22). Second, that the significant eigenvalues ($\lambda_a > \alpha$) scale as $\lambda_a \sim N\alpha/\gamma$. (This scaling holds for various simple interpolation models.) Then it can be shown that the scaling of the log evidence is:

$$-\log P(D|\alpha,\beta,\mathcal{H}) \sim \beta E_D^{\rm MP} + \frac{1}{2}\left(\gamma \log(N/\gamma) + \gamma\right). \tag{35}$$

[Readers familiar with Minimum Description Length (MDL) will recognize the dominant $\frac{\gamma}{2}\log N$ term; MDL and Bayesian inference are equivalent, as discussed later.] Thus the scaling behaviour of the log evidence is identical to the structural risk minimization expression (34), provided that $c_1 = \frac{1}{2}$ and $c_2 = 1$. Isabelle Guyon (personal communication) has confirmed that the empirically determined values for c_1 and c_2 are indeed close to these Bayesian values. It will be interesting to try to understand and develop this relationship.

10.4. Contrasts with conventional dogma in learning theory and statistics

10.4.1. 'Representation theorems'. It is popular to assert the utility of a particular model by demonstrating that the model has arbitrary representational power. For example, 'three layer neural networks are good interpolation tools because they can implement any smooth function given enough hidden units'.

A Bayesian data modeller takes a different attitude (as, to be fair, do other learning theory researchers). The objective of data modelling is to find a model that is well-matched to the data. A model that could match an arbitrary function is too flexible and will generalize poorly; and in Bayesian terms such a model is improbable compared to simpler models which fit the data nearly as well. Probability theory in fact favours models which are as *inflexible* as possible: just flexible enough to capture the real structure in the data, but no more.

Those who appreciate that models with universal representational power are not necessarily a good thing are often led astray in the other direction by a second popular myth, the supposed need to limit the complexity of a model when there is little data.

10.4.2. 'The complexity of the model should be matched to the amount of data'. A central tenet of the Vapnik-Chervonenkis theory is that, when there is little data, it is good to use a model with few parameters, even if that model is known to be incapable of representing the true function. An attempt is made to match the 'capacity' of the model to the number of data points. This is sometimes used as the motivation for 'pruning' a neural network (Hassibi and Stork 1993).

A Bayesian needs never do this; the choice of which models to consider is a matter of prior belief, and should not depend on the amount of data collected. In a domain such as interpolation, a typical prior belief is that the real underlying function, although smooth, would require a very large number of parameters to describe it exactly. There will never be enough data to determine all the parameters of the true model. But all the same, we should use the model we truly believe in. There are two possible outcomes. It may be that the prior knowledge of smoothness, etc., included in the true model, may constrain the ill-determined parameters sufficiently that the predictions are useful. On the other hand, it may turn out that the predictions are ill-determined, so that the true model's predictions have huge error bars. In this case, it would surely be most unwise to use a simpler model: its predictions would be better determined, but they would be *incorrect*!

The practice of deliberately using a simple model when data are scarce is probably one of the two main causes of dangerously overconfident predictions in orthodox statistics. The second cause is the practice of 'accepting' a null hypothesis (and using it alone for prediction) when a test of that hypothesis gives a result that is 'not significant'.

It is now common practice for Bayesians to fit models that have more parameters than the number of data points (MacKay 1992a; Neal 1994; Weir 1991). The recommended philosophy (Box and Tiao 1973) is to aim to incorporate every imaginable possibility into the model space: for example, if it is conceivable that a very simple model might be able to explain the data, one should include simple models in the model space; if the noise might have a long-tailed distribution, one should include a hyperparameter which controls the heaviness of the tails of the distribution; if an input variable might be irrelevant to a regression, include it in the regression anyway, with a sophisticated regularizer embodying the concept of uncertain relevance. The inclusion of remote possibilities in the model space is 'safe', because our inferences will home in on the sub-models which are best matched to the data. The inclusion in our initial model space of bizarre models which are subsequently ruled out by the data is not expected significantly to influence predictive performance.

10.5. Minimum description length (MDL)

A complementary view of Bayesian model comparison is obtained by replacing probabilities of events by the lengths in bits of messages which communicate the event without loss to a receiver. Message lengths $L(\mathbf{x})$ correspond to a probabilistic model over events $\mathbf{x}$ via the relations:

$$P(\mathbf{x}) = 2^{-L(\mathbf{x})}, \quad L(\mathbf{x}) = -\log_2 P(\mathbf{x}). \tag{36}$$

Non-integer coding lengths can be handled by the arithmetic coding procedure (Witten *et al.* 1987).

The MDL principle (Wallace and Boulton 1968) states that one should prefer models which can communicate the data in the smallest number of bits. Consider a message which states which model, $\mathcal{H}$, is to be used, and then communicates the data D within that model, to some pre-arranged precision δD. This produces a message of length $L(D, \mathcal{H}) = L(\mathcal{H}) + L(D|\mathcal{H})$. The lengths $L(\mathcal{H})$ for different $\mathcal{H}$ define an implicit prior $P(\mathcal{H})$ over the alternative models. Similarly $L(D|\mathcal{H})$ corresponds to a density $P(D|\mathcal{H})$. Thus, a procedure for assigning message lengths can be mapped

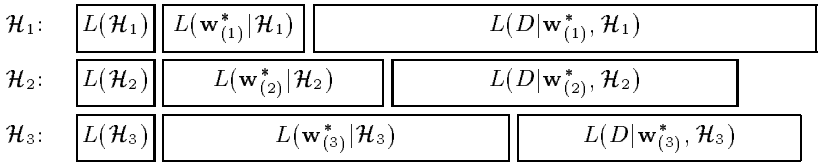


Figure 11. A popular view of model comparison by minimum description length. Each model $\mathcal{H}_i$ communicates the data D by sending the identity of the model, sending the best fit parameters of the model $\mathbf{w}^*$, then sending the data relative to those parameters. As we proceed to more complex models the length of the parameter message increases. On the other hand, the length of the data message decreases, because a complex model is able to fit the data better, making the residuals smaller. In this example the intermediate model $\mathcal{H}_2$ achieves the optimum trade-off between these two trends.

onto posterior probabilities:

$$\begin{aligned} L(D, \mathcal{H}) &= -\log P(\mathcal{H}) - \log(P(D|\mathcal{H})\delta D) \\ &= -\log P(\mathcal{H}|D) + \text{const.} \end{aligned}$$

In principle, then, MDL can always be interpreted as Bayesian model comparison and *vice versa*. However, this simple discussion has not addressed how one would actually evaluate the key data-dependent term $L(D|\mathcal{H})$, which corresponds to the evidence for $\mathcal{H}$. Often, this message is imagined as being subdivided into a parameter block and a data block (figure 11). Models with a small number of parameters have only a short parameter block but do not fit the data well, and so the data message (a list of large residuals) is long. As the number of parameters increases, the parameter block lengthens, and the data message becomes shorter. There is an optimum model complexity ($\mathcal{H}_2$ in the figure) for which the sum is minimized.

This picture glosses over some subtle issues. We have not specified the precision to which the parameters $\mathbf{w}$ should be sent. This precision has an important effect (unlike the precision δD to which real-valued data D are sent, which, assuming δD is small relative to the noise level, just introduces an additive constant). As we decrease the precision to which $\mathbf{w}$ is sent, the parameter message shortens, but the data message typically lengthens because the truncated parameters do not match the data so well. There is a non-trivial optimal precision. In simple Gaussian cases it is possible to solve for this optimal precision (Wallace and Freeman 1987), and it is closely related to the posterior error bars on the parameters, $\mathbf{A}^{-1}$, where $\mathbf{A} = -\nabla\nabla \log P(\mathbf{w}|D, \mathcal{H})$. It turns out that the optimal parameter message length is virtually identical to the log of the 'Occam factor' in equation (8). (The random element involved in parameter truncation means that the encoding is slightly sub-optimal.)

With care, therefore, one can replicate Bayesian results in MDL terms. Although some of the earliest work on complex model comparison involved the MDL framework (Patrick and Wallace 1982), MDL has no apparent advantages over the direct probabilistic approach.

MDL does have its uses as a pedagogical tool. The description length concept is useful for motivating prior probability distributions. Also, different ways of breaking down the task of communicating data using a model can give helpful insights into the modelling process, as will now be illustrated.

10.5.1. On-line learning and cross-validation. The log evidence can be decomposed as a sum of 'on-line' predictive performances:

$$\log P(D|\mathcal{H}) = \log P(\mathbf{t}^{(1)}|\mathcal{H}) + \log P(\mathbf{t}^{(2)}|\mathbf{t}^{(1)}, \mathcal{H}) + \log P(\mathbf{t}^{(3)}|\mathbf{t}^{(1)}, \mathbf{t}^{(2)}, \mathcal{H}) \cdots + \log P(\mathbf{t}^{(N)}|\mathbf{t}^{(1)} \ldots \mathbf{t}^{(N-1)}, \mathcal{H}).$$

This decomposition can be used to explain the difference between the evidence and 'leave one out cross-validation' as measures of predictive ability. Cross-validation examines the average value of just the last term, $\log P(\mathbf{t}^{(N)}|t^{(1)} \ldots \mathbf{t}^{(N-1)}, \mathcal{H})$, under random re-orderings of the data. The evidence, on the other hand, sums up how well the model predicted all the data, starting from scratch.

10.5.2. The 'bits back' encoding method. Another MDL thought experiment (Hinton and van Camp 1993) involves incorporating random bits into our message. The data are communicated using a parameter block and a data block. The parameter vector sent is a random sample from the posterior distribution $P(\mathbf{w}|D, \mathcal{H}) = P(D|\mathbf{w}, \mathcal{H})P(\mathbf{w}|\mathcal{H})/P(D|\mathcal{H})$. This sample $\mathbf{w}$ is sent to an arbitrary small granularity $\delta\mathbf{w}$ using a message length $L(\mathbf{w}|\mathcal{H}) = -\log[P(\mathbf{w}|\mathcal{H})\delta\mathbf{w}]$. The data are encoded relative to $\mathbf{w}$ with a message of length $L(D|\mathbf{w}, \mathcal{H}) = -\log[P(D|\mathbf{w}, \mathcal{H})\delta D]$. Once the data message has been received, the random bits used to generate the sample $\mathbf{w}$ from the posterior can be deduced by the receiver. The number of bits so recovered is $-\log[P(\mathbf{w}|D, \mathcal{H})\delta\mathbf{w}]$. These recovered bits need not count towards the message length, since we might use some other optimally encoded message as a random bit string, thereby communicating that message at the same time. The net description cost is therefore:

$$\begin{aligned} L(\mathbf{w}|\mathcal{H}) + L(D|\mathbf{w}, \mathcal{H}) - \text{'Bits back'} &= -\log \frac{P(\mathbf{w}|\mathcal{H})P(D|\mathbf{w}, \mathcal{H})\delta D}{P(\mathbf{w}|D, \mathcal{H})} \\ &= -\log P(D|\mathcal{H}) - \log \delta D. \end{aligned}$$

Thus this thought experiment has yielded the optimal description length.

10.6. Ensemble Learning

The posterior distribution $P(\mathbf{w}|D, \mathcal{H})$ may be a very complicated density. The methods described in this paper have assumed that, in local regions that contain significant probability mass, the posterior can be well approximated by a Gaussian found by making a quadratic expansion of $\log P(\mathbf{w}|D, \mathcal{H})$ around a local maximum. (For brevity I here omit the parameters α and β.)

An interesting idea that has been implemented by Hinton and van Camp (1993) is to try to improve the quality of this type of approximation by optimizing the entire posterior approximation. The idea is that a Gaussian fitted somewhere other than the mode of $P(\mathbf{w}|D, \mathcal{H})$ might in some sense be a better approximation to the posterior. Consider a parameterized approximation $Q(\mathbf{w}; \theta)$ to the true posterior distribution $P(\mathbf{w}|D, \mathcal{H})$. For example, the parameters θ for a Gaussian approximation would be its mean and covariance matrix. We can measure the quality of fit of Q to P by the 'variational free energy':

$$F(\theta) = -\int \mathrm{d}\mathbf{w}\, Q(\mathbf{w}; \theta) \log \frac{P(\mathbf{w}|D, \mathcal{H})}{Q(\mathbf{w}; \theta)}.$$

F has a lower bound of zero which can only be realised if there are parameters θ such that Q matches P exactly. This measure can be motivated by generalizing the MDL 'bits back' thought experiment (section 10.5.2), with the random sample $\mathbf{w}$ drawn from Q instead of from P (Hinton and van Camp 1993).

Now the task is to minimize $F(\theta)$. This minimization, called 'ensemble learning', is in general a challenging task. However, Hinton and van Camp have shown that exact derivatives of F with respect to θ can be obtained for a neural net with one non-linear hidden layer and a linear output, if the Gaussian approximation $Q(\mathbf{w};\theta)$ is restricted to have no correlation among the weights.

The weakness of ensemble learning by free energy minimization is that if the approximating distribution $Q(\mathbf{w};\theta)$ has only a simple form then the free energy objective function favours distributions which are extremely conservative, placing no probability mass in regions where $P(\mathbf{w})$ is small. For example, if a strongly correlated Gaussian P is modelled by a separable Gaussian Q, then the free energy solution sets the curvature of $\log Q$ to be the same as the diagonal elements of the curvature of $\log P$. This gives an approximating distribution which covers far too small a region of $\mathbf{w}$ space, so that the outcome of ensemble learning would be essentially identical to the outcome of traditional optimization of a point estimate.

A possible extension of Hinton and van Camp's idea to more complex models Q is to include in θ an adaptive linear preprocessing of the inputs. Denote the coefficients of this linear mapping from inputs to 'sub-inputs' by $\mathbf{U}$, and the parameters from the sub-inputs to the hidden units by $\mathbf{V}$; the effective input weights are given by the product $\mathbf{VU}$. A separable Gaussian prior can now be applied to the parameters $\mathbf{V}$, so that Hinton's exact derivatives can still be evaluated. Inclusion of the additional parameters $\mathbf{U}$ in θ defines a richer family of probability distributions $Q(\mathbf{w};\theta)$ over the effective parameters $\mathbf{w}$. It will be interesting to see if these distributions are powerful enough to yield Gaussian approximations superior to those produced by the evidence framework.

10.7. Future directions

The challenges that face the Bayesian approach to data modelling are the invention of good model spaces, and the creation of numerical techniques for inference in those spaces. These are both non-trivial tasks requiring skill and ingenuity.

The scaling up of Gaussian approximation methods to larger neural network problems will be helped by the use of implicit second order methods (Skilling 1993; Pearlmutter 1994); these are algorithms that make use of properties of the Hessian matrix $\mathbf{A}$, such as the value of $\mathbf{Av}$ for arbitrary vector $\mathbf{v}$, without explicitly evaluating $\mathbf{A}$.

Multilayer perceptrons are well established as probabilistic models for *regression* and *classification*, both of which are *conditional* modelling tasks: the *input* variables are assumed given, and we *condition* on their values when modelling the distribution over the *output* variables; no model of the density over input variables is constructed. In density modelling (or generative modelling), on the other hand, a density over *all* the observable quantities is constructed. Multi-layer perceptrons have not conventionally been used to create density models (though belief networks (Spiegelhalter and Lauritzen 1990) and other neural networks such as the Boltzmann machine (Hinton and Sejnowski 1986) do define density models). Several researchers are presently working on extending multilayer perceptrons to turn them into density

models (Hinton and Zemel 1994; MacKay 1995a).

Acknowledgments

I thank my colleagues at Caltech, the University of Toronto, and the University of Cambridge for invaluable discussions.

I am grateful to Radford Neal and Timothy Jervis for helpful comments on the manuscript.

References

Abu-Mostafa Y S 1990 The Vapnik–Chervonenkis dimension: information versus complexity in learning *Neural Computation* **1** (3) 312–317

Berger J 1985 *Statistical Decision theory and Bayesian Analysis* Springer

Bishop C M 1992 Exact calculation of the Hessian matrix for the multilayer perceptron *Neural Computation* **4** (4) 494–501

Box G E P and Tiao G C 1973 *Bayesian inference in statistical analysis* Addison–Wesley

Breiman L 1992 Stacked regressions Technical Report 367, Dept. of Stat., Univ. of Cal. Berkeley

Bretthorst G 1988 *Bayesian spectrum analysis and parameter estimation* Springer

Bridle J S 1989 Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition In *Neuro-computing: algorithms, architectures and applications*, ed F Fougelman-Soulie and J Hérault. Springer–Verlag

Buntine W and Weigend A 1991 Bayesian back–propagation *Complex Systems* **5** 603–643

Copas J B 1983 Regression, prediction and shrinkage (with discussion) *J. R. Statist.Soc B* **45** (3) 311–354

Cox R 1946 Probability, frequency, and reasonable expectation *Am. J. Physics* **14** 1–13

Gull S F 1988 Bayesian inductive inference and maximum entropy In *Maximum Entropy and Bayesian Methods in Science and Engineering, vol. 1: Foundations*, ed G Erickson and C Smith, pp 53–74, Dordrecht. Kluwer

—— 1989 Developments in maximum entropy data analysis In *Maximum Entropy and Bayesian Methods, Cambridge 1988*, ed J Skilling, pp 53–71, Dordrecht. Kluwer

Guyon I, Vapnik V N, Boser B E, Bottou L Y and Solla S A 1992 Structural risk minimization for character recognition In *Advances in Neural Information Processing Systems 4*, ed J E Moody, S J Hanson and R P Lippmann, pp 471–479, San Mateo, California. Morgan Kaufmann

Hanson R, Stutz J and Cheeseman P 1991 Bayesian classification with correlation and inheritance In *Proceedings of the 12th International Joint Conference on Artificial Intelligence, Sydney, Australia*

Hassibi B and Stork D G 1993 Second order derivatives for network pruning: Optimal brain surgeon In *Advances in Neural Information Processing Systems 5*, ed C L Giles, S J Hanson and J D Cowan, pp 164–171, San Mateo, California. Morgan Kaufmann

Hinton G E and Sejnowski T J 1986 Learning and relearning in Boltzmann machines In *Parallel Distributed Processing*, ed D. E Rumelhart and J E McClelland, pp 282–317. Cambridge Mass.: MIT Press

—— and van Camp D, 1993 Keeping neural networks simple by minimizing the description length of the weights To appear in: *Proceedings of COLT-93*

—— and Zemel R S 1994 Autoencoders, minimum description length and Helmholtz free energy In *Advances in Neural Information Processing Systems 6*, ed J D Cowan, G Tesauro and J Alspector, San Mateo, California. Morgan Kaufmann

Jaynes E T 1983 Bayesian intervals versus confidence intervals In *E.T. Jaynes. Papers on Probability, Statistics and Statistical Physics*, ed R D Rosencrantz, p 151. Kluwer

Jeffreys H 1939 *Theory of Probability* Oxford Univ. Press

LeCun Y, Denker J and Solla S A 1990 Optimal brain damage In *Advances in Neural Information Processing Systems 2*, ed D Touretzky, pp 598–605. Morgan Kaufmann

Loredo T J 1990 From Laplace to supernova SN 1987A: Bayesian inference in astrophysics In *Maximum Entropy and Bayesian Methods, Dartmouth, U.S.A., 1989*, ed P Fougere, pp 81–142. Kluwer

MacKay D J C, 1991 *Bayesian Methods for Adaptive Models* PhD thesis, California Institute of Technology

—— 1992a Bayesian interpolation *Neural Computation* **4** (3) 415–447

—— 1992b A practical Bayesian framework for backpropagation networks *Neural Computation* **4** (3) 448–472

—— 1992c The evidence framework applied to classification networks *Neural Computation* **4** (5) 698–714

—— 1994 Bayesian non-linear modelling for the prediction competition In *ASHRAE Transactions, V.100, Pt.2*, Atlanta Georgia. ASHRAE

—— 1995a Bayesian neural networks and density networks *Nuclear Instruments and Methods in Physics Research, Section A*

—— 1995b Hyperparameters: Optimize, or integrate out? In *Maximum Entropy and Bayesian Methods, Santa Barbara 1993*, ed G Heidbreder, Dordrecht. Kluwer

Moody J E 1992 The *effective* number of parameters: An analysis of generalization and regularization in nonlinear learning systems In *Advances in Neural Information Processing Systems 4*, ed J E Moody, S J Hanson and R P Lippmann, pp 847–854, San Mateo, California. Morgan Kaufmann

Neal R M 1993 Bayesian learning via stochastic dynamics In *Advances in Neural Information Processing Systems 5*, ed C L Giles, S J Hanson and J D Cowan, pp 475–482, San Mateo, California. Morgan Kaufmann

—— 1994 Priors for infinite networks Technical Report in preparation, Univ. of Toronto

Patrick J D and Wallace C S 1982 Stone circle geometries: an information theory approach In *Archaeoastronomy in the Old World*, ed D. C Heggie, pp 231–264. Cambridge Univ. Press

Pearlmutter B A 1994 Fast exact multiplication by the Hessian *Neural Computation* **6** (1) 147–160

Rumelhart D E, Hinton G E and Williams R J 1986 Learning representations by back–propagating errors *Nature* **323** 533–536

Skilling J 1993 Bayesian numerical analysis In *Physics and Probability*, ed W. T Grandy, Jr. and P Milonni, Cambridge. C.U.P.

——, Robinson D R T and Gull S F 1991 Probabilistic displays In *Maximum Entropy and Bayesian Methods, Laramie, 1990*, ed W T Grandy and L Schick, pp 365–368, Dordrecht. Kluwer

Spiegelhalter D J and Lauritzen S L 1990 Sequential updating of conditional probabilities on directed graphical structures *Networks* **20** 579–605

Thodberg H H 1993 Ace of Bayes: application of neural networks with pruning Technical Report 1132 E, Danish meat research institute

Wallace C and Boulton D 1968 An information measure for classification *Comput. J.* **11** (2) 185–194

Wallace C S and Freeman P R 1987 Estimation and inference by compact coding *J. R. Statist. Soc. B* **49** (3) 240–265

Weir N 1991 Applications of maximum entropy techniques to HST data In *Proceedings of the ESO/ST-ECF Data Analysis Workshop, April 1991*

Witten I H, Neal R M and Cleary J G 1987 Arithmetic coding for data compression *Communications of the ACM* **30** (6) 520–540

Wolpert D H 1993 On the use of evidence in neural networks In *Advances in Neural Information Processing Systems 5*, ed C L Giles, S J Hanson and J D Cowan, pp 539–546, San Mateo, California. Morgan Kaufmann