Bayesian neural networks for classification: how useful is the evidence framework ?

William D. Penny and Stephen J. Roberts.
Department of Electrical and Electronic Engineering,
Imperial College,
London SW7 2BT, UK.

March 3, 1999

Abstract

This paper presents an empirical assessment of the Bayesian evidence framework for neural networks using four synthetic and four real-world classification problems. We focus on three issues; model selection, automatic relevance determination (ARD) and the use of committees. Model selection using the evidence criterion is only tenable if the number of training examples exceeds the number of network weights by a factor of five or ten. With this number of available examples, however, cross-validation is a viable alternative. The ARD feature selection scheme is only useful in networks with many hidden units and for data sets containing many irrelevant variables. ARD is also useful as a hard feature selection method. Results on applying the evidence framework to the real-world data sets showed that committees of Bayesian networks achieved classification accuracies similar to the best alternative methods. Importantly, this was achievable with a minimum of human intervention.

1 Introduction

The Bayesian evidence framework proposed in MacKay (1992a) provides a unified theoretical treatment of learning in neural networks. Its practical benefits include principled methods for determining optimal weight decay coefficients, methods for soft feature selection and methods for model selection. It also provides a framework for using committees of networks and for calculating error bars.

Despite these apparent benefits the evidence framework has been applied to only a handful of problems; assessing the fat content of carcasses (Thodberg, 1995), vowel recognition and classification of thyroid disease (Gutjahr, 1997), prediction of energy consumption (MacKay, 1995a), classification of number plates (Oldfield, 1995) and classification of EEG data (Sykacek, 1997).

This paper attempts to find out if there are any inherent problems with the evidence framework and, in the absence of these, to promote its wider application.

Throughout the paper we consider only the original Bayesian evidence framework described in MacKay (1992a). The more general Bayesian framework due to Neal (1996) is examined in a companion paper (Husmeier, 1998).

In evaluating the evidence framework we focus on three issues; model selection, soft feature selection and the use of committees for accurate classification.

Model selection is an important issue because if we use a network which is too simplistic it will not be able to adequately represent the input-output mapping underlying the data. On the other hand, if we use a network which is too complex it will extract features from the training set which are peculiar to that set and will not generalise well when faced with new data.

An obvious solution to the model selection problem is the one offered by cross-validation (see Ripley, 1996): networks with different structures, and therefore of different complexity, are tested on a validation set (data not used in training) and the networks which perform best are selected. This does mean, however, that not all of the data can be used for training: some of it must be put aside for validation. The evidence framework offers a different solution. Networks are selected which have the highest 'evidence'. This is a quantity which can be calculated from the training set only. No validation set is required. This is, potentially, a great advantage, especially if only small amounts of data are available.

The complexity of a model can also be constrained by regularization. This involves training a network on a cost function that includes, for example, a weight decay term. This scheme, however, requires the setting of a weight decay parameter which is often chosen, again, by cross-validation. A benefit of the evidence framework is that weight decay parameters can be set automatically. No validation set is required. In this paper we experiment with schemes where different groups of weights have different weight decay parameters. This type of regularization culminates in the Automatic Relevance Determination algorithm which performs soft feature selection.

In the Bayesian framework it is natural to consider not just a single neural network model but a whole ensemble of models. This leads to the use of committees of networks where the overall prediction on a new data point is the combined prediction of many networks. In this paper we use Bayesian committees on a number of real world classification problems to assess their accuracy in relation to other approaches.

In the next section we briefly explain what the evidence framework is and how it is applied to neural network classifiers. In section three we present results from the application of Bayesian networks to real and synthetic problems, focussing on the issues of model selection, soft feature selection and committees.

Throughout the paper we consider the neural network models to be Perceptrons or Multi-Layer Perceptrons (MLPs) as described in Bishop (1995, chapters 3 and 4), although the evidence framework is applicable to a broad range of classification and regression models (MacKay 1992a,b).

2 The Evidence Framework

This section presents a summary of the key ideas of the evidence framework in the context of training MLPs on classification problems. Derivations of formulae are omitted as they may be found in MacKay (1992b) and Thodberg (1995). Readers requiring a full tutorial on the evidence framework are referred to Bishop (1995, chapter 10).

2.1 Model selection

Conventional (maximum likelihood/minimum error) neural network learning produces a single weight vector, $\boldsymbol{w}$. But given that we have only a finite amount of data we cannot really be certain as to what that weight vector should be; adding or taking away even a single training pattern would result in a different learnt weight vector. Thus, there is some uncertainy in the value of the weight vector. This uncertainty is captured in Bayesian learning by a probability distribution over weight vectors that expresses our beliefs concerning how likely the different weight values are.

To start the Bayesian learning process we define a network structure, H, which in this paper specifies the number of hidden units in a MLP network. We also define a prior distribution for the weights in network H, $p(\boldsymbol{w})$, that expresses our initial beliefs about the weights before any data has arrived. When the data, D, is then observed the prior distribution is updated to a posterior distribution according to Bayes' theorem

$$p(\boldsymbol{w} \mid D, H) = \frac{p(D \mid \boldsymbol{w}, H)p(\boldsymbol{w} \mid H)}{p(D \mid H)} \tag{1}$$

This posterior distribution combines the likelihood function, $p(D \mid \boldsymbol{w}, H)$, which contains information about $\boldsymbol{w}$ from observation, and the prior, which contains information about $\boldsymbol{w}$ from background knowledge. The term in the denominator, $p(D \mid H)$, is known as the evidence for model H.

Given a set of candidate networks, H_i, which may have different numbers of hidden units, the posterior probability of each model can be expressed as

$$p(H_i \mid D) = \frac{p(D \mid H_i)p(H_i)}{p(D)} \tag{2}$$

If the models are considered equiprobable before we see any data then $p(H_i)$ is the same for all models. Since $p(D)$ does not depend on the model then the most probable model is the one with the highest evidence, $p(D \mid H)$. The evidence can therefore be used to select between different MLP networks.

For a network with weights $\boldsymbol{w}$ and K weight decay regularizers or 'hyperparameters', $\{\alpha_k\}$, which control the magnitude of weights in K different weight groups, the log of the evidence is (Thodberg, 1995)

$$\begin{aligned} LogEv &= -C(\boldsymbol{w}) + Log(Occ_w) + Log(Occ_\alpha) && (3)\\ C(\boldsymbol{w}) &= G(\boldsymbol{w}) + \sum_k \alpha_k E^k(\boldsymbol{w}) && (4)\\ Log(Occ_w) &= -\frac{1}{2}\ln \mid \boldsymbol{A} \mid + \ln N_h! + N_h \ln 2 + \sum_k \frac{W_k}{2} \ln \alpha_k && (5)\\ Log(Occ_\alpha) &= \sum_k \frac{1}{2} \ln\left(\frac{4\pi}{\gamma_k}\right) - K\ln(\ln \Omega) && (6) \end{aligned}$$

The first term in equation (3) is the log likelihood. This is equal to the negative of the regularised cost function $C(\boldsymbol{w})$ which is defined in equation 4 as the sum of $G(\boldsymbol{w})$, the usual cross-entropy term (Bishop 1995, p.237), and a weight decay term. The term $E^k(\boldsymbol{w})$ is the sum of squares of weights in the kth group.

The next two terms in equation (3) are 'Occam factors'. An Occam factor is the ratio of a posterior volume to a prior volume. Larger volume means greater uncertainty about the parameters. Large networks have a large prior volume and thus a small Occam factor. They therefore have lower evidence; the Occam factors act to penalize complex models. Networks with low posterior volume are also penalised. This may initially seem unreasonable but is justifiable on the grounds that networks with low posterior volume have had to be finely tuned to the data. This sort of brittleness is undesirable. The evidence is therefore seen to embody a trade-off between accuracy and complexity. For an expanded discussion of these issues see MacKay (1992a).

The first Occam factor, defined in equation 5, is the Occam factor for the weights. The first term in equation (5) is the negative log determinant of the Hessian matrix, $\boldsymbol{A}$. This term measures the posterior volume in weight space. It derives from the assumption that the posterior distribution, $p(\boldsymbol{w} \mid D, H)$, is Gaussian (a central tenet of the evidence framework); the inverse Hessian is therefore equivalent to the posterior covariance matrix of weight uncertainties. The next two terms in equation (5) arise from the redundancy of representation in a single hidden-layer MLP having N_h hidden units. By this we mean that the same function can be represented in a network by permuting the positions of hidden units (which can be done in $N_h!$ ways) and by reversing the sign of hidden unit outputs (which can be done in 2^{N_h} ways). These terms act as corrections to the posterior volume. The last term in equation (5) measures the (negative) prior volume in weight space. It is calculated by summing the inverse weight variance of a weight in the kth group, α_k (see next section), over the number of weights in that group, W_k, and over all groups.

The second Occam factor, defined in equation 6, is the Occam factor for the hyperparameters. The first term in equation (6) captures the posterior uncertainty in the hyperparameters. This is expressed in terms of a parameter γ_k, which, as discussed in the next section, is the number of 'well-determined' weights in group k. The second term in equation (6) captures the prior uncertainty in the hyperparameters. This is expressed in terms of a parameter Ω which captures our prior belief in the range of scales within which we believe each hyperparameter to lie. This is subjectively set to 10^3 (Thodberg, 1995), meaning that before any data is seen we believe we know the value of each α_k to within three orders of magnitude.

Once a network has been trained, equations (3) to (6) can be used to calculate the (log) evidence for that network. This is used for model selection.

2.2 Training and regularization

Network training proceeds in the usual manner using standard optimization algorithms (Bishop, chapter 7) with the novelty that training is periodically halted for the weight decay parameters to be updated. Specifically, a network with weights $\boldsymbol{w}$ and K weight decay regularizers, $\{\alpha_k\}$, is trained on the cost function in equation (4) (the regularization term in this cost function derives from the assumption that, in each weight group, each weight is drawn from a zero mean Gaussian distribution with variance $1/\alpha_k$).

The hyperparameters are initialised to small arbitrary values. This is important as the network must be allowed to find interesting structure in the data before any regularization takes place. The network is then trained to find the maximum posterior weight vector, $\boldsymbol{w}_{MP}$, by minimising the cost function. Training is stopped when the training error tolerance (fractional change in error between epochs) falls below some pre-specified value. The hyperparameters are then updated in a re-estimation step. This involves the calculation of the Hessian matrix (second derivative of the cost function)

$$\boldsymbol{A} = \nabla\nabla G + \sum_k \alpha_k \boldsymbol{I}_k \tag{7}$$

where $\boldsymbol{I}_k$ is a diagonal matrix having ones along the diagonal that pick off weights in the kth group. We also need to calculate the number of 'well-determined' weights in each weight group, γ_k. This is defined as the number of weights whose values are determined by the data rather than by the prior and is given by (MacKay, 1992a)

$$\gamma_k = W_k - \alpha_k \sum_j \left(\boldsymbol{I}_k \boldsymbol{A}^{-1} \boldsymbol{I}_k\right)_{jj} \tag{8}$$

The quantity γ_k is evaluated using the *old* value of α_k. The *new* value of α_k is then calculated according to the formula (MacKay, 1992a)

$$\alpha_k = \frac{\gamma_k}{2E^k(\boldsymbol{w})} \tag{9}$$

Once the regularizer has been re-estimated, the network is trained from where it left off until a specified lower training error tolerance is achieved. Re-estimation and further training continue according to some tolerance regime until a minimum tolerance is reached. The re-estimation scheme means that the network is automatically regularized and so will not overfit the data. Bayesian regularization eliminates the need for a separate validation set for choosing optimal weight decay values.

The regularization scheme can be used for single or multiple hyperparameters. A single hyperparameter corresponds to the standard global weight decay scheme. However, for networks where the number of hidden units is very different from the number of inputs, this prior is unsuitable. This is because the weights in the output layer have to be of a different magnitude than those in the hidden layer in order for the node activations to be in the same range. For these networks, a 'two-layer' prior is more suitable. This uses two independent hyperparameters: one for the hidden-layer weights, and one for the output-layer weights. A further group could also be used to regularize the bias weights. Some practitoners consider this desirable as unregularized parameters constitute values drawn from an improper prior. This may lead to problems when comparing models, as the evidence given these values is zero (Bishop, 1995). In this paper, however, we do not regularize the bias weights as they will be regularized indirectly; they will take on values in proportion to the magnitude of the other weights in their respective nodes in order that the (tanh) node outputs will on average be zero (if this didn't happen then the nodes would be either 'off' or 'on' all the time). In evaluating the evidence, the bias weights are ignored.

This scheme can be extended such that the group of weights leaving each input has its own hyperparameter. The resulting method, called Automatic Relevance Determination (ARD) by MacKay (1995a) and Neal (1996), performs soft feature selection; weights connected to irrelevant inputs are automatically set to small values. We investigate the ARD method in our experiments.

In this paper we restrict our experimentation to the use of (single or multiple) weight-decay regularizers which derive from the assumption of zero-mean Gaussian priors on the weights. It is worth noting, however, that other choices of prior are equally valid, if not more so. Williams (1995), for example, examines the use of a Laplace prior which results in automatic pruning of redundant weights. Gutjahr and Nautze (1997) consider groups of Gaussian priors with means which are inferred from the data. This can also implement a form of weight pruning.

2.3 Committees

In the Bayesian framework it is natural to consider not just a single neural network but a whole ensemble of networks. This leads to the use of committees of networks where the overall prediction on a new data point is the combined prediction of many networks. Models can be combined by weighting predictions according to the evidence of each network. In this way, the Bayesian framework provides a natural way of forming committees. However, if the Gaussian approximation is not valid (see next section) then the evidence will not be accurately determined. In practice therefore the evidence is used to select the best networks which are then used in an unweighted committee (Thodberg, 1995). The question then arises as to how many networks to use in the committee. This issue is addressed in our experiments.

2.4 Practical issues

The Gaussian approximation to the posterior distribution is central to the evidence framework. Walker (1969) has shown that in the limit of an infinite training set the posterior distribution does in fact become Gaussian. With a finite number of data points, however, the approximation breaks down. MacKay (1992a) notes that the approximation can be tested by looking at the correlation between evidence and test error. This will be investigated in our experiments.

A second issue is the choice of prior. Sometimes a single Gaussian prior will be adequate. In other cases a two-layer prior or an ARD prior may be more suitable. The correct choice of prior may be made by looking at the correlation between the evidence and classification error on a validation set. In practice, however, we would always use an ARD prior unless W_k, the number of weights per weight group, (an upper limit on γ_k) is less than three (MacKay, 1994).

When calculating the hyperparameters and when evaluating the evidence, the Hessian matrix $\boldsymbol{A}$ should first be reconstructed using the positive eigenvalues only. This is because, if the network is not exactly at a local minima of C then some eigenvalues of $\boldsymbol{A}$ may be negative. This implies that the posterior variance in some directions of weight space is negative. This is handled by ignoring these problematic directions and just considering the distribution in the non-negative directions. A second related problem is due to numerical round-off error in the evaluation of the determinant and therefore of the evidence. In networks with redundant weights it is possible that the Hessian is singular or nearly singular. In these cases some eigenvalues will be of the order of machine precision and the calculation of the determinant, which involves the product of eigenvalues, will be unreliable. For this reason, Thodberg (1995) considers reconstructing the Hessian using only those eigenvalues above a certain threshold. He then considers the dependence of the evidence on this threshold. In this paper, we simply reconstruct the Hessian using all positive eigenvalues.

A fourth issue is the choice of subjective prior when determining the Occam factor for the hyperparameters; why do we set Ω to 10^3 and not say 10^4. In practice, however, this term tends to be only a minor factor in the overall evidence calculation.

Estimation of hyperparameters and the evidence requires the evaluation and storage of the Hessian matrix. For problems of even moderate size, say 300 weights, this becomes problematic. Furthermore, to estimate the hyperparameters requires evaluation of the trace of the inverse Hessian (equation 9) and to evaluate the evidence requires the determinant of the Hessian (equation 5). For large matrices these computations become a bottleneck. This burden can be eased with the use of Lanczos methods which generate a low-dimensional tri-diagonal representation of a larger matrix (Oldfield, 1995 p. 57).

3 Results

All networks are trained with the conjugate gradient algorithm as described in Press et al. (1992). We use this training method because it is orders of magnitude faster than gradient descent and it does not require a learning rate parameter - the learning rate is determined automatically by the local curvature of the error surface. The only parameter it does require is the convergence criterion.

We define the error tolerance as the fractional change in error from one epoch to the next. If the error tolerance falls below some specified value the network is deemed to have converged. The regularizers are then re-estimated using equations (7-9). Because each new value of α_k gives rise to a new value of $\boldsymbol{A}$ (equation 7), equations 7 to 9 are applied a number of times to ensure convergence of $\boldsymbol{A}$ and α_k. In this paper this number is arbitrarily set to ten. Each network is then trained, with the hyperparameters now kept fixed, until a lower error tolerance value is reached.

The error tolerance is reduced according to the following regime: the initial error tolerance is 10^{-3} which is reduced by a factor of two after each re-estimation. Training and re-estimation is finally stopped at a tolerance of 10^{-6}.

We need to calculate the Hessian matrix in order to re-estimate the regularizers and evaluate the evidence. This is evaluated by an exact formula from Bishop (1995, p.154).

For each data set, we train an ensemble of single-hidden layer MLPs. For each classsifier structure 10 systems are trained in order to examine the effect of local minima on solutions and to construct committees. Networks with a single hidden unit, $N_h = 1$, are functionally equivalent to a single perceptron (logistic regression unit). In the experiments that follow we therefore use logistic regression in their place.

3.1 Description of data sets

We examine a total of eight data sets; four synthetic and four real. The synthetic data sets are 'XOR', 'Yin-Yang', 'Neal' and 'Ripley'. The real data sets are 'Diabetes', 'Tremor', 'Ionosphere' and 'Vowel'.

The XOR data set is a continuous version of the exclusive-or logic function. The Yin-Yang data set is a synthetic two-class problem generated by the authors. It is a continuous version of a data set investigated by Coetzee and Stonick (1997). The Neal and Ripley data sets are synthetic problems from Neal (1997) and Ripley (1994). The Neal data contains two useful inputs and two noise inputs.

The Diabetes data describes a population of women of Pima Indian heritage who were tested for diabetes. The data were collected by the US National Institute of Diabetes and Digestive and Kidney Diseases and are described more fully in Ripley (1996). The Tremor data set is a two-class medical classification problem. The data set was collected by Spyers-Ashby (1998) and consists of two input features derived from measurements of arm muscle tremor and a class label representing patient or non-patient. The Ionosphere data contains information collected by a radar system and previously analysed by Sigillito et al. (1989). It is a two-class classification problem; positive radar returns are those showing evidence of some type of structure in the ionosphere, negative returns are those that do not. The Vowel data set is a classification problem containing eleven classes each of which corresponds to an English vowel. The data was generated by fifteen different speakers and characterised with a linear predictive filter to form ten input features (Robinson, 1988).

Table 1 shows the dimensions of each problem and the number of examples available. The two-input data sets are shown in Figures 1, 2, 3, 4 and 5 along with decision boundaries from MLP networks trained on them.

3.2 Evidence as a model selection criterion

Figure 6 shows the relative importance of the different factors which contribute to the evidence for different networks trained on the XOR problem. As the number of hidden units increases the likelihood increases but the Occam factor for the weights decreases. This trend was the same for all the data sets, as expected. The Occam factor for the hyperparameters was negligible. This is also true of all the other data sets.

3.2.1 Effect of number of hidden units

Figure 7 shows plots of evidence versus number of hidden units for each of the data sets. For each plot there is a number of hidden units (or range of) for which the evidence is a maximum. We note, however, that the position of this maximum depends not just on the nature of the data set but also on the number of training examples available; with more data the maximum occurs at a larger number of hidden units. In the XOR data, for example, networks with two hidden units have the highest average evidence. This makes sense as two hidden units are sufficient to solve the XOR problem. If, however, the number of training examples is increased from 100 to 500 this maximum shifts to $N_h = 3$ and $N_h = 4, 5$ networks have higher evidence [1].

For the Yin-Yang data set, MLPs with 9-hidden units have the highest average evidence. This seems a reasonable choice given the geometry of the problem (see Figure 2). Out of all the classifiers trained on the Ripley data set, those with 3 and 4 hidden units had the highest evidence. This concurs with Ripley (1994) who found that MLPs with 3 hidden units had the lowest test error. For the Diabetes data, the logistic model (plotted at $N_h = 1$ in Figure 7e) has the highest evidence. This concurs with Ripley (1996) who found that MLPs were no better than logistic regression on this data. This is the because the data, like many real classification problems (Penny and Frost, 1996), is intrinsically linearly separable. The Tremor data, as can be seen from Figure 3, is however clearly nonlinear and networks with three hidden units have the highest average evidence.

All plots in Figure 7 were obtained from networks trained with a single regularizer, except for networks trained on the Ionosphere data. This is because, for all but the Ionosphere data, the number of network inputs was not significantly different from the number of hidden units. In these cases, a single regularizer is sufficient (see section 2.2). In the case of the Ionosphere data, however, there are many more inputs than hidden units and it was also believed that not all of the inputs were relevant. We also note that, for the other data sets, plots of evidence versus number of hidden units using networks trained with ARD priors were not significantly different from those shown in Figure 7.

3.2.2 Relation to test error

Figure 8 shows plots of test error (percentage of test data misclassified) versus evidence for each of the data sets. The expected trend that networks with higher evidence tend to have lower generalisation error is only generally observed for two of the data sets; Yin-Yang and Vowel (Figures 8b and 8h). It is no coincidence that these data sets also have the largest number of training examples. For some of the other data sets the correlation between evidence and test error, r, is good when the number of hidden units is low; XOR ($N_h = 2, r = -0.95$), Neal ($N_h = 3, r = -0.94; N_h = 4, r = -0.90; N_h = 5, r = -0.88$) for example. For the XOR and Tremor data sets there is a more fundamental reason why a strong correlation is not generally observable. For the XOR data, nearly all of the $N_h > 2$ networks have a very low test error. Thus, it is not possible to observe a graded decline of test error with increasing evidence. For the Tremor data, the situation is similar; logistic models generalise poorly but all the MLP models generalise well. Plots of the test log likelihood (instead of test error) versus evidence are somewhat smoother but a strong correlation is not observed. The same is true for the log likelihood of marginalised outputs, where each marginalised output is obtained by integrating over the posterior weight distribution (MacKay, 1992a).

3.2.3 Effect of training set size

We now turn to the important issue of training set size by looking at three of the synthetic data sets (Yin-Yang, Neal and Ripley) and measuring the correlation between test error and evidence as the size of the training set is increased. Figure 9 shows that the correlation

[1] For the XOR data there is another effect of increasing the number of training examples; the average evidence of 2-hidden unit networks decreases. This is because some of the 2-hidden unit MLPs reach a poor local minima solution. As the number of hidden units is increased, to say $N_h = 3$, the sub-optimal solutions are more easily avoided.

becomes significantly non-zero when R, the ratio of number of training examples to number of network weights, exceeds five or ten.

Our results so far indicate that the evidence is a good model selection criterion, provided sufficient examples are available. Although this is indeed the case, we note that the training error is also a good model selection criterion at similarly large values of R. This is demonstrated in Figure 10.

Thus, although correlations between evidence and generalization error are observable we need a disappointingly large number of training examples; with this number of examples cross-validation error, for example, is a realistic alternative model selection criterion.

3.3 Automatic Relevance Determination for feature selection

We now compare networks trained with and without ARD priors on a number of the data sets. Networks trained without an ARD prior use a two-layer Gaussian prior, that is, with independent hyperparameters for each layer (see section 2.2).

3.3.1 Effect of training set size

Firstly, we look at artificial data sets to assess the dependence of ARD on training set size. The Neal data set contains two useful inputs and two noise inputs. We also consider two extensions of this data set; 'Neal-5' and 'Neal-12'. The 'Neal-5' data set consists of the two useful inputs plus three extra variables, each one being equal to the original first input variable plus additive Gaussian noise of increasing variance. Thus, in this new data set, inputs x_1, x_3, x_4 and x_5 are of gradually decreasing relevance to the classification problem. Input x_2 is the same as in the original data set. The 'Neal-12' data set consists of the two original useful inputs plus ten extra noisy input variables.

We then trained a number of 4-hidden unit MLPs on each data set with varying amounts of training data. For each different training set size, ten networks were trained. Figure 11 shows plots of the inferred weight decay parameters versus R. The plots show that irrelevant inputs are assigned large weights decay coefficients, as expected, and that this is observable even for R as low as one. This was also observed on other data sets (Ripley, Yin-Yang) where we added spurious noisy input variables. In contrast, effective evidence-based model selection requires a much larger value of R. This is understandable because the calculation of each weight in the network is based on N_d/γ_{tot} 'effective' data points where N_d is the number of training data points and γ_{tot} is the total number of well-determined weights in the network ($\sum_k \gamma_k$). The calculation of each hyperparameter, however, is based on $W_k.N_d/\gamma_{tot}$ 'effective' data points where W_k is the number of weights in the kth group. The calculation of the hyperparameters is therefore statistically more reliable than the calculation of the evidence - because W_k times as many effective data points are used. It is also computationally more reliable. This is because the evaluation of the trace of a matrix, upon which the hyperparameters are based, is less prone to numerical round-off error than the evaluation of its determinant, upon which the calculation of the evidence is based. Bayesian regularization is therefore tenable with a small number of training examples whereas evidence-based model selection is not.

3.3.2 Relation to test error

For the Neal and Neal-5 data sets, however, the impact of the ARD scheme on the test error was negligible. There are two main reasons for this. Firstly, ARD is only effective in networks having many hidden units. This is because the evaluation of each ARD hyperparameter is based on $N_h.N_d/\gamma_{tot}$ effective data points (for an ARD weight group $W_k = N_h$).

Secondly, ARD can only show an improvement if non-ARD networks pick up spurious correlations in the data. This is more likely to happen if there are many irrelevant inputs. This last point is illustrated in Figure 12 which shows that for the Neal-12 data set, which has ten spurious inputs, there is a marginal but consistent reduction in test error over a range of R values [2]. This was'nt observed on the original Neal data set which has only two irrelevant inputs.

3.3.3 Hard feature selection

An alternative use of ARD is as a hard feature selection method where inspection of the hyperparameters leads to selection of a subset of variables which is used to train a new ARD network. In these smaller networks the calculation of evidence will be more accurate. This approach, which we call 'hard-ARD', was found to be useful on the Ionosphere data. Inputs 1, 5, 7, 8, 9, 16, 21, 23, 25, 27 and 31 were selected after a subjective cut-off value was chosen. The subjective choice of this cut-off value is, however, a major drawback of the hard-ARD method. Nevertheless, an evidence-ranked committee of ARD networks trained on this reduced data set achieved the lowest test error of all methods.

Table 2 compares classification results on four of the data sets with and without ARD. ARD is only seen to be beneficial for the Ionosphere data. For the other data sets the number of spurious inputs was not sufficient to upset non-ARD methods.

3.4 Committees of networks

The error made by a classifier may be split up into two components; a bias component and a variance component (Breiman, 1996). If the classifiers are of sufficient complexity, such as MLPs with large numbers of hidden units, then the bias component will be small. The variance component, however, will be large. But if the networks are used in committees the variance component can be reduced, thus reducing the overall prediction error.

3.4.1 Effect of committee size

In this section we consider the procedure of forming unweighted committees from the M networks with the highest evidence. We call this an 'evidence-ranked' committee of size M. Figure 13 shows the effect of M on test error.

Only for the Ionosphere data is there a sensitive dependence on M; a committee composed of ten networks has the mimumum test error. Larger committees, unusually, have a larger test error. The optimal ten network committee is composed mainly of two hidden unit networks (see Figure 8g). As the committee size is increased, three and four hidden unit networks are included which have a higher test error. These networks which have relatively few hidden unit and a large test error introduce an incorrect bias into the committee. This bias component results in an overall increase in prediction error. For the Vowel data the prediction error decreases significantly as the size of the committee is increased. The networks in this committee have more hidden units than for the Ionosphere problem and the reduction in variance is therefore the dominant effect of increasing committee size. For the other data sets there seems little benefit in using a committee.

3.4.2 Comparison with other methods

Table 3 summarizes the percentage classification error of committees on five of the data sets. For all data sets the results are quoted for networks using two-layer Gaussian priors, except

[2] The data sets for $R = 4, 5$ required more than the 400 data points avaiable in the original Neal data set. The extra data points were generated according to the method described in Neal (1997).

for the Ionosphere data where we quote the result from networks with soft-ARD priors. The first column (MaxEv) is the error of the single network with the highest evidence. The second column (MaxEvH) is the error from a committee formed from all h-hidden unit networks where h-hidden unit networks have the highest average evidence on that problem. The third and fourth columns show the median and minimum test errors over all values of M for the evidence-ranked committees. The last column indicates the classification error obtained with 'Other' classifiers. This covers a broad spectrum of methods. The result on the Tremor data set was obtained with a committee of Radial Basis Function classifiers (Roberts and Penny, 1997). The result on the Neal data set is from a Gaussian Process classifier (Neal, 1997). The result on the Ripley data set was obtained with a three-hidden unit MLP trained with a single weight decay regularizer (Ripley, 1994). The result on the Ionosphere data is with a nearest-neighbour classifier [3]. The result on the Vowel data set was obtained with an 88-hidden unit MLP trained with early stopping (Robinson, 1988).

We note that these 'Other' results were obtained after a good deal of experimentation by the respective authors, and generally speaking, the best results were reported. By focussing on our best results, the MinErr column in Table 3, we could conclude that committees of MLPs trained according to the evidence framework were the most accurate classifiers on every data set (except Neal's). This would, however, be misleading. The MedianErr and MaxEvH results are more representative and indicate that committees of networks trained according to the Bayesian evidence framework provide similar classification accuracies as the best alternative methods. Importantly, this is achievable with a minimum of human intervention.

4 Conclusions

The Bayesian evidence framework provides a unified theoretical treatment of learning in neural networks. To date, however, it has been applied to only a handful of problems. In order to find out whether this is due to inherent flaws in the paradigm or to lack of awareness of the methods we have provided an empirical assessment of the technique. We have focussed on three particular issues; model selection, feature selection using ARD and the use of committees of networks.

Experiments on synthetic data have shown that the correlation between training error and test error is as good as the correlation between the evidence and test error. We conclude that it is therefore unnecessary to calculate the evidence to decide between models - this can be done using the training error alone. Model selection using the evidence or training error is only tenable, however, if the number of training examples exceeds the number of network weights by a factor of five or ten. With this number of available examples cross-validation is a viable alternative.

The ARD feature selection scheme was able pick out relevant inputs and to correctly rank them in order of relevance on a number of synthetic data sets. ARD only had an impact on test error, however, in networks having many hidden units and when there were many irrelevant network inputs. ARD will only be beneficial for data sets and networks satisfying these constraints. ARD was also shown to be useful as a hard feature selection method. Overall, only one of our four real-world data sets benefitted from the use of ARD.

Results on applying the evidence framework to the real-world data sets showed that committees of Bayesian networks achieved similar classification rates to the best alternative methods. Although the classification results were no better, they were achieved with a min-

[3] In their original paper, Sigillito et al. (1989) quote a test error of 4 per cent. We did not include this in the table, however, as minimum test error was used as a stopping criterion in an 'early-stopping' training scheme. This therefore constitutes a biased result.

imum of intervention by the authors. Our overall conclusion is that the evidence framework is useful for precisely this reason; no parameters have to be hand-crafted to ensure that the classifiers work well. This is due to the automatic regularization scheme.

The overhead of Bayesian regularisation is the calculation, storage and inversion of Hessian matrices. In all of the networks we investigated this calculation was not found to be a bottleneck; the optimization algorithms took orders of magnitude more computer time.

Acknowledgements

William D. Penny is a research fellow funded by the UK Engineering and Physical Sciences Research Council (EPSRC) (GR/K79062), whose support we gratefully acknowledge. Our experiments used software based, in part, on the Netlab package written by Chris Bishop and Ian Nabney. Other code written by the authors is made available via http://www.ee.ic.ac.uk/research/neural/wpenny.html.

References

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

Breiman, L. (1996) Bias, Variance and Arcing classifiers. *Technical Report 460*, Statistics Department, University of California.

Coetzee, F.M. and Stonick, V.L. (1996) 488 solutions to the XOR problem. In *M.C. Mozer et. al. (Eds.) Neural Information Processing Systems - Natural and Synthetic.*

Husmeier, D. Penny, W.D. and Roberts, S.J. (1998). An Empirical Evaluation of Bayesian Sampling with Hybrid Monte Carlo for Training Neural Network Classifiers. Submitted to Neural Networks, May 1998.

MacKay, D.J.C. (1992a) The evidence framework applied to classification problems. *Neural Computation* **4**(5):720-736.

MacKay, D.J.C. (1992b) Bayesian Interpolation. *Neural Computation* **4**(3):415-447.

MacKay, D.J.C. (1994) Hyperparameters: optimise or integrate out ? In G. Heidbreder (Ed.), *Maximum Entropy and Bayesian Methods, Santa Barbara 1993*, Dordrecht: Kluwer.

MacKay, D.J.C. (1995a) Bayesian non-linear modelling for the 1993 energy prediction competition. In G. Heidbreder (Ed.), *Maximum Entropy and Bayesian Methods, Santa Barbara 1993*, Dordrecht: Kluwer.

Penny, W.D. and Frost, D. (1996) Neural networks in clinical medicine. *Medical Decision Making*, **16**(4):386-398.

Gutjahr, S. and Nautze, C. (1997) Extended Bayesian Learning. *Proceedings of ESANN 97, European Symposium on Artificial neural networks, Bruges*, pp: 321-326.

Neal, R. (1996) Bayesian learning for neural networks. *Springer-Verlag New York.*

Neal, R. (1997) Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification. Technical Report 9702, Department of Statistics, University of Toronto.

Oldfield, M.J. (1995) Advances in Probabilistic Modelling. *PhD Thesis, Trinity College, Cambridge.*

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

Ripley, B.D. (1994) Neural networks and related methods for classification. *Journal of the Royal Statistical Society B* **56**(3):409-456.

Ripley, B.D. (1996) Pattern recognition and neural networks.*Cambridge University Press.*

Roberts, S.J. and Penny, W. (1997) Novelty, confidence and errors in connectionst systems. *Technical report, Department of Electrical Engineering, Imperial College.* Available from http://www.ee.ic.ac.uk/research/neural/wpenny.html.

Robinson, A.J. and Fallside, F. (1988) A Dynamic Connectionist Model for Phoneme Recognition. *Proceedings of nEuro 1988, Paris.*

Sigillito, V. G., Wing, S. P., Hutton, L. V., & Baker, K. B. (1989). Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, **10**(3), 262-266.

Spyers-Ashby, J.M., Bain, P. and Roberts, S.J. (1998). A comparison of fast Fourier transform (FFT) and autoregressive (AR) spectral estimation techniques for the analysis of tremor data. *Journal of Neuroscience Methods*, Special issue on Waveform and Systems Analysis, Ed. Michael Gresty, **83**, 35-43.

Sykacek, P., Dorffner, G., Rappelsberger, P. and Zeitlhofer. J.(1997) Evaluating confidence measures in a neural network based sleep stager, *Technical Report* TR-97-21, OFAI, 1997.

Thodberg, H.H. (1995) A review of Bayesian neural networks with an application to near infrared spectroscopy. *IEEE Transactions on Neural Networks*, **7**, 56-72.

Walker, A.M. (1969) On the asymptotic behaviour of posterior distributions. *Journal of the Royal Statistical Society, B* **31**(1):80-88.

Williams, P.M. (1995) Bayesian regularization and pruning using a Laplace prior. *Neural Computation* **7**(1):117-143.

Table 1: *Description of data sets.*

Data Set	Origin	Inputs	Outputs	Training examples	Test examples
XOR	Synthetic	2	2	100	100
Yin-Yang	Synthetic	2	2	500	500
Neal	Synthetic	4	3	400	600
Ripley	Synthetic	2	2	250	1000
Diabetes	Real	7	2	200	332
Tremor	Real	2	2	179	178
Ionosphere	Real	34	2	200	150
Vowel	Real	10	11	528	462

Table 2: *Median percentage test error of evidence-ranked committees. In these tests the Tremor and Ripley data sets each contained two extra noisy input variables.*

Problem	No ARD	Soft ARD	Hard ARD
Neal	14.5	15.2	15.2
Ripley	10.2	10.2	9.3
Tremor	15.7	16.3	16.3
Ionosphere	7.0	7.3	4.0

Table 3: *Percentage test error for MaxEv (single network with the highest evidence), MaxEvH (a committee formed from all h-hidden unit networks where h-hidden unit nets have the highest average evidence for that problem), MedianErr (the committee with median test error), MinErr (the committee with minimum test error), Other method.*

Problem	MaxEv	MaxEvH	MedianErr	MinErr	Other
Neal	14.6	15.0	15.0	14.5	13.0
Ripley	9.3	9.4	9.3	8.9	9.4
Tremor	16.3	15.2	16.3	14.6	15.5
Ionosphere	7.3	3.3	7.3	7.3	7.3
Vowel	70.1	50.9	48.7	46.1	49.0

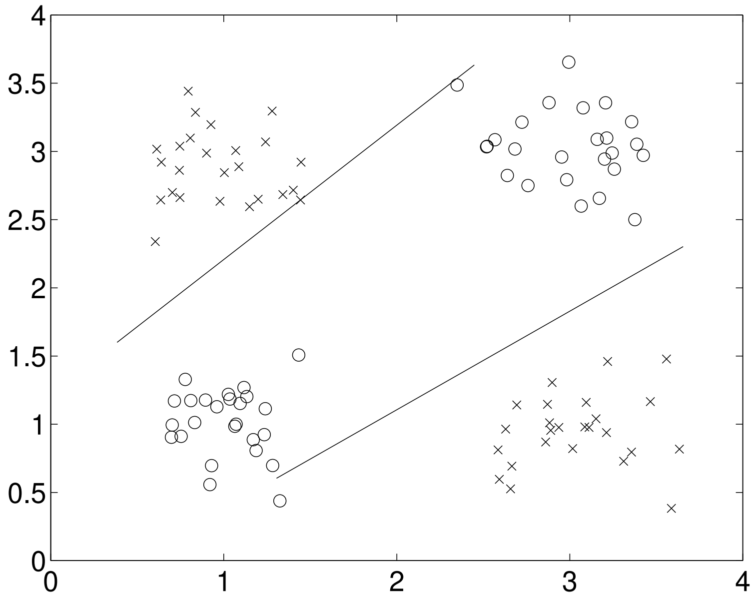


Figure 1: XOR data. The solid lines show the orientation of each hidden unit in a two-hidden-unit MLP.

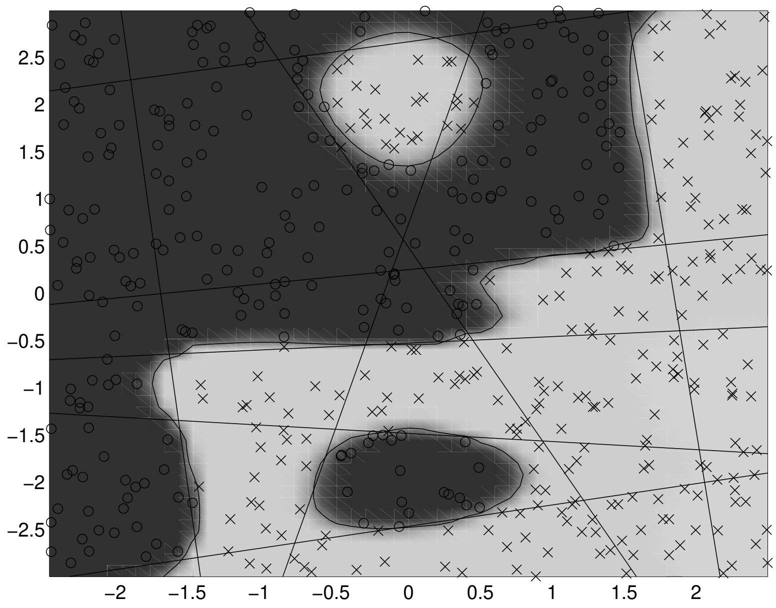


Figure 2: Yin-Yang data. The straight solid lines show the orientation of each hidden unit in a nine-hidden-unit MLP. The curvy solid lines show the overall decision boundary. The shade of gray codes the output of the network with dark gray being 0 and light gray being 1.

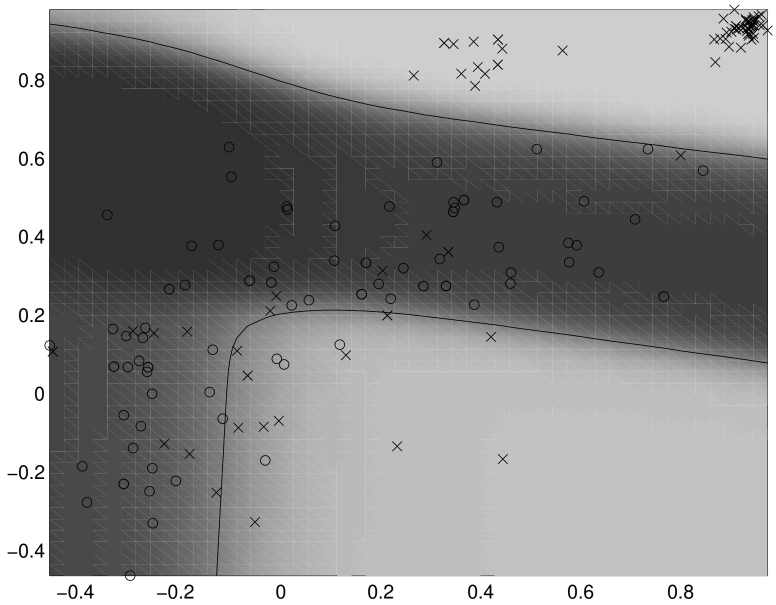


Figure 3: Tremor data. Crosses represent data points from patients, zeros indicate data points from normal subjects. The solid line shows the overall decision boundaries formed by a MLP with three hidden units. The shade of gray codes the output.

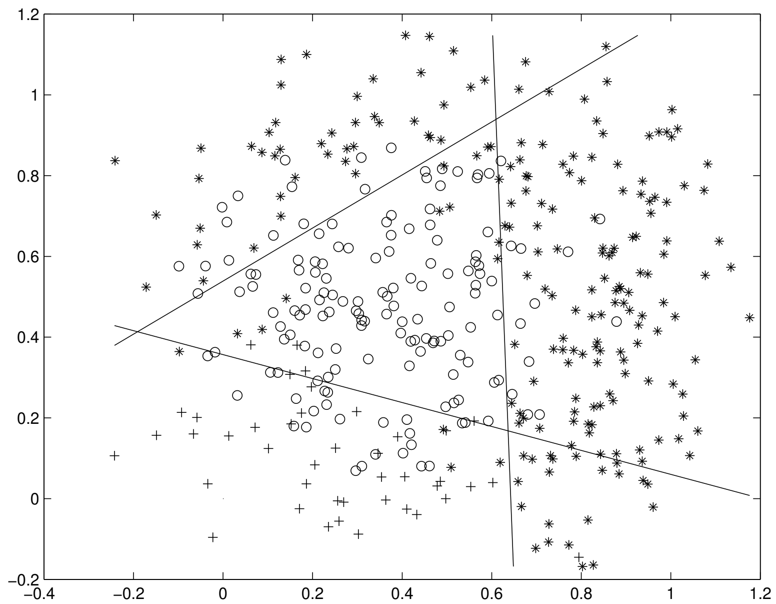


Figure 4: Neal's synthetic 3-class problem. The data are plotted against the two useful (non-noise) inputs. The solid lines show the orientation of each of the 3 hidden units in an MLP.

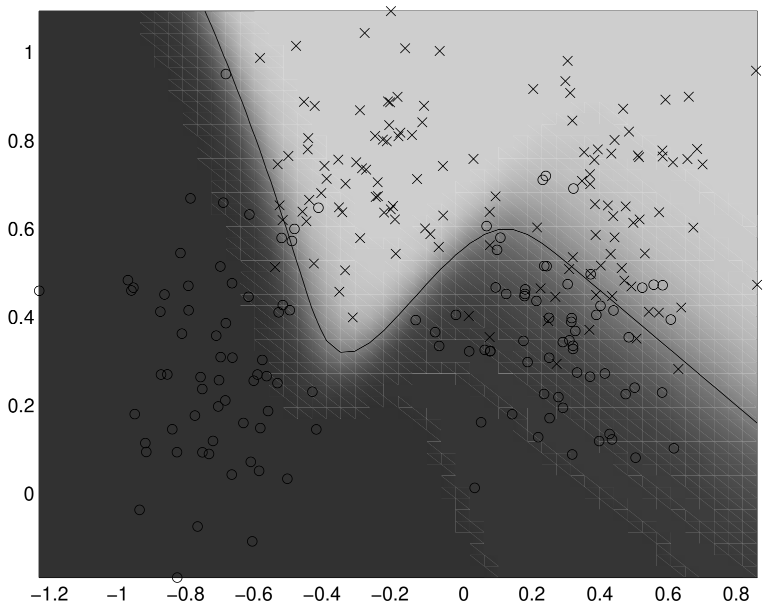


Figure 5: Ripley's synthetic data. The solid line shows the overall decision boundary formed by a MLP with three hidden units. The shade of gray codes the output.

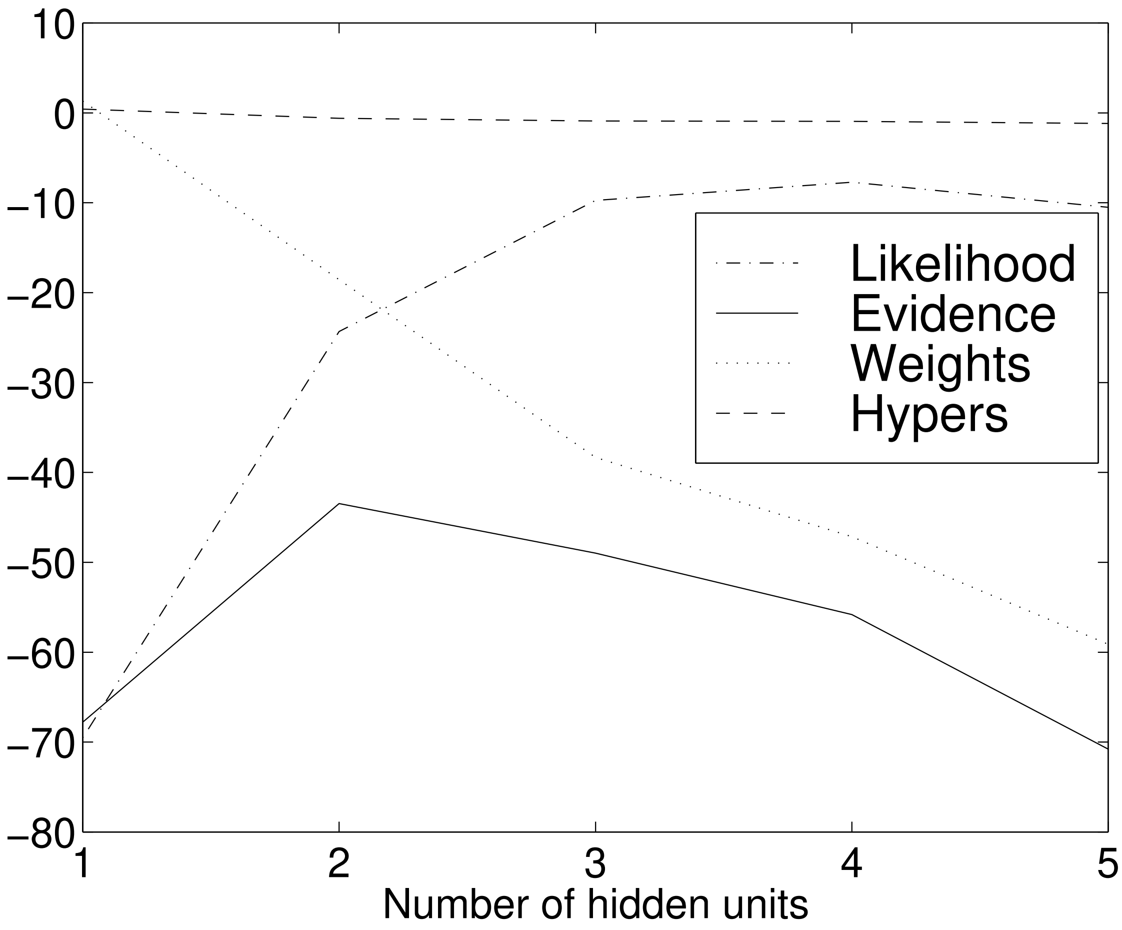


Figure 6: Terms in the evidence for the XOR problem versus number of hidden units; total evidence (solid line), log likelihood (dash-dotted line), log Occam factor for the weights (dotted line), log Occam factor for the hyperparameters (dashed line). Each point is averaged over 10 networks. Networks with 2 hidden units have the highest average evidence.

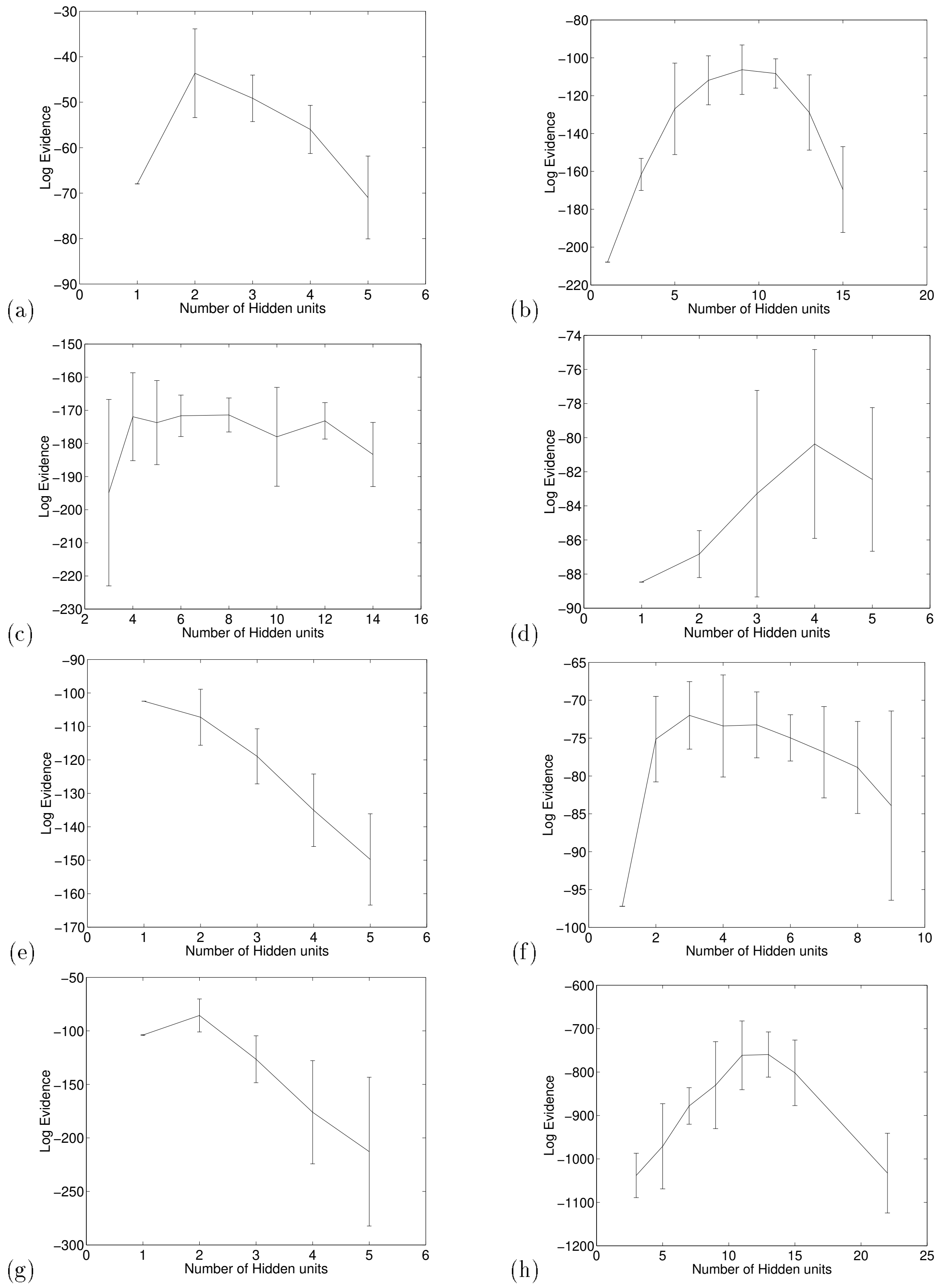


Figure 7: Evidence versus number of hidden units for (a) XOR, (b) Yin-Yang, (c) Neal, (d) Ripley, (e) Diabetes, (f) Tremor, (g) Ionosphere and (h) Vowel data sets. The solid line shows the evidence averaged over the ten networks in each committee. The error bars are at plus and minus one standard deviation.

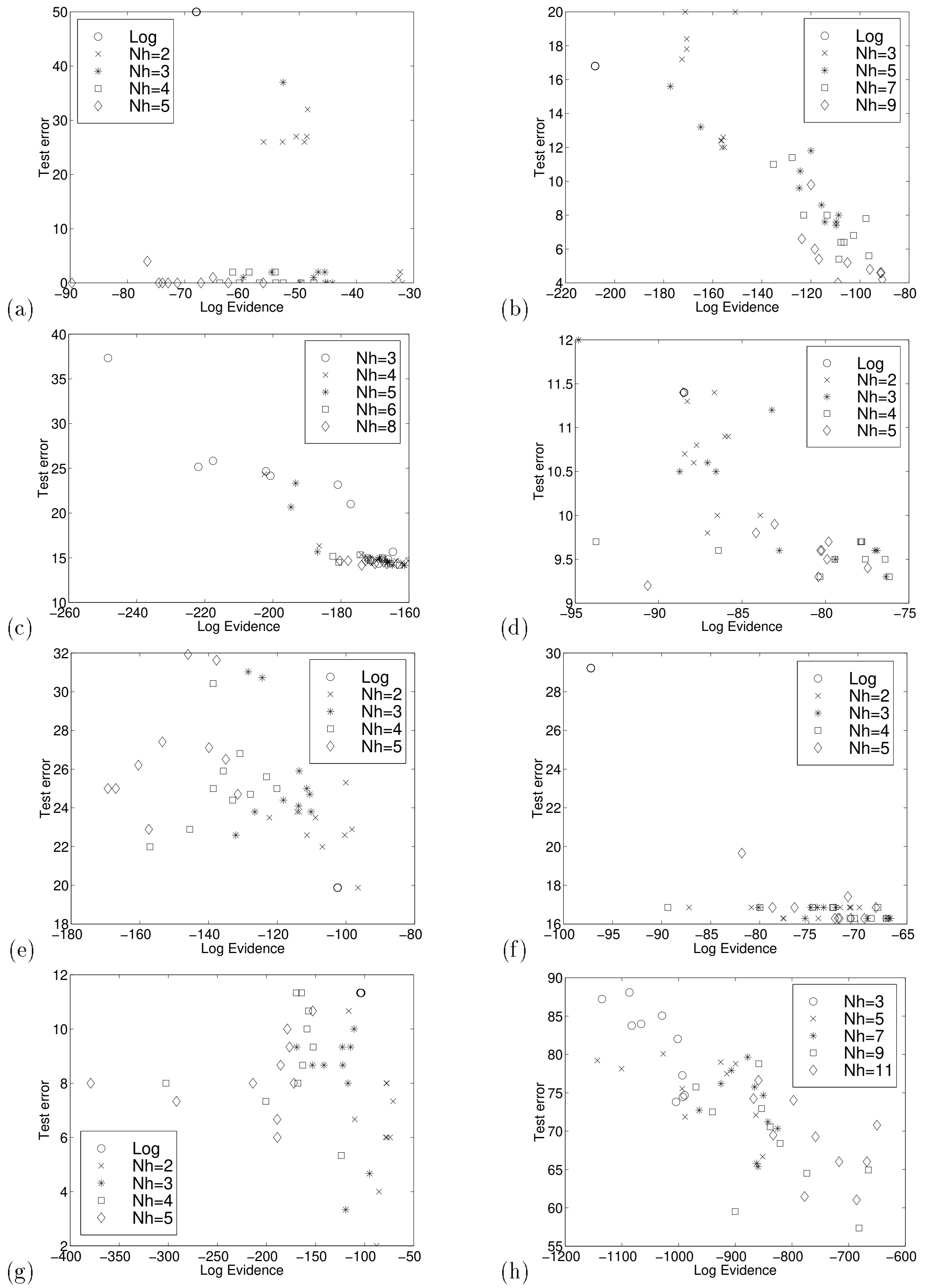


Figure 8: Test error rate versus evidence for (a) XOR, (b) Yin-Yang, (c) Neal, (d) Ripley, (e) Diabetes, (f) Tremor, (g) Ionosphere and (h) Vowel data sets. Different symbols represent networks with different numbers of hidden units, N_h. In each plot every symbol appears ten times, once for each of the ten networks trained, except for the Logistic classifiers (marked 'Log' in the legends) which, though also trained ten times, always converged to the same solution.

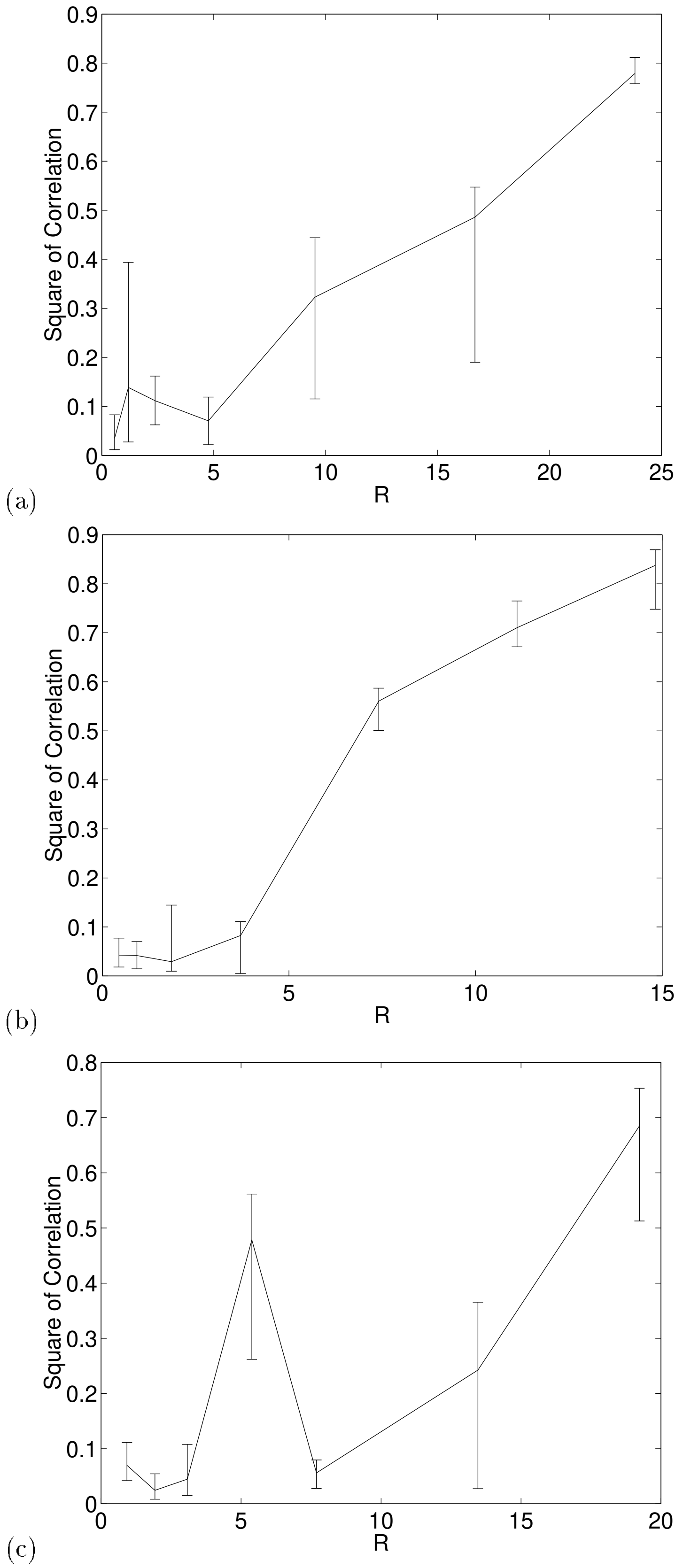
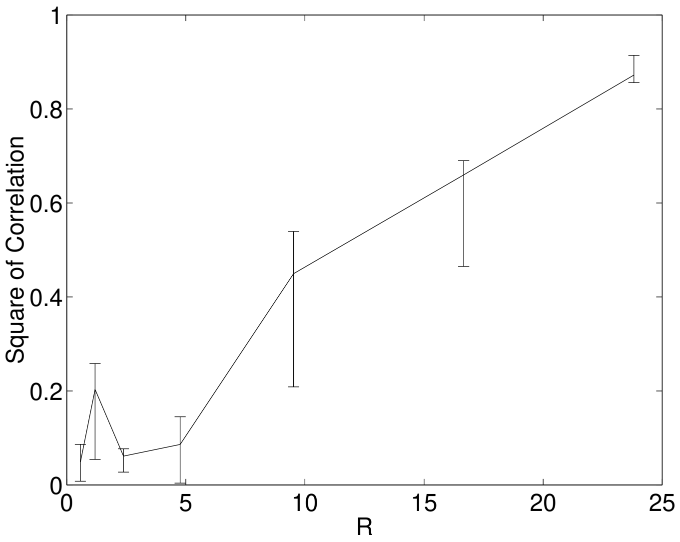
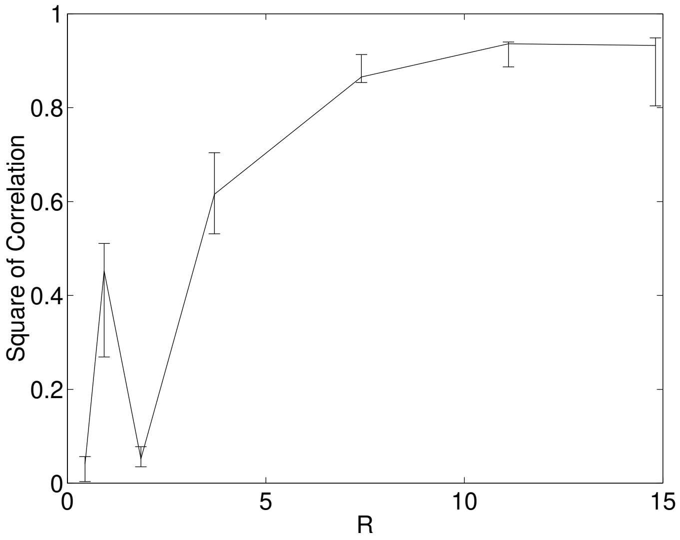


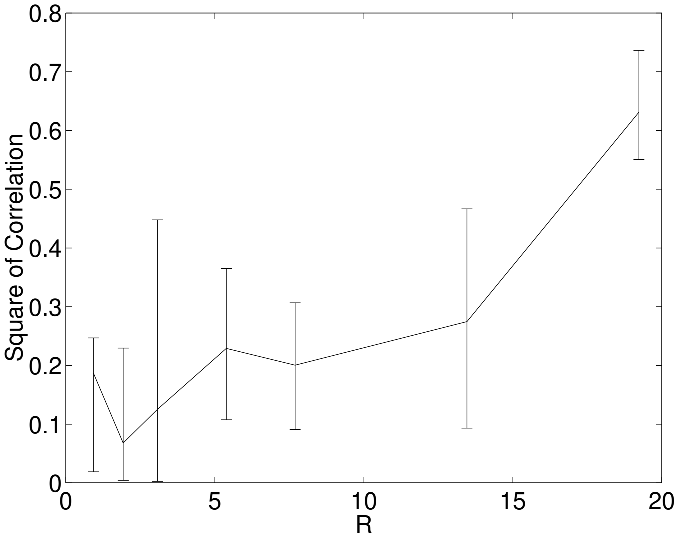
Figure 9: Square of correlation between *evidence* and test error versus R, the ratio of training examples to network weights for (a) Yin-Yang data learnt by a 5-hidden unit MLP, (b) Neal data learnt by a 4-hidden unit MLP and (c) Ripley data learnt by a 3-hidden unit MLP. At each value of R, the correlations were calculated from a committee composed of 10 networks. Ten such committees were trained giving rise to ten estimates of correlation. The solid lines show the median correlation and the error bars show the 30th and 70th percentiles.



(a)



(b)



(c)

Figure 10: Square of correlation between *training error* and test error versus R, the ratio of training examples to network weights for (a) Yin-Yang data learnt by a 5-hidden unit MLP, (b) Neal data learnt by a 4-hidden unit MLP and (c) Ripley data learnt by a 3-hidden unit MLP. At each value of R, the correlations were calculated from a committee composed of 10 networks. Ten such committees were trained giving rise to ten estimates of correlation. The solid lines show the median correlation and the error bars show the 30th and 70th percentiles.

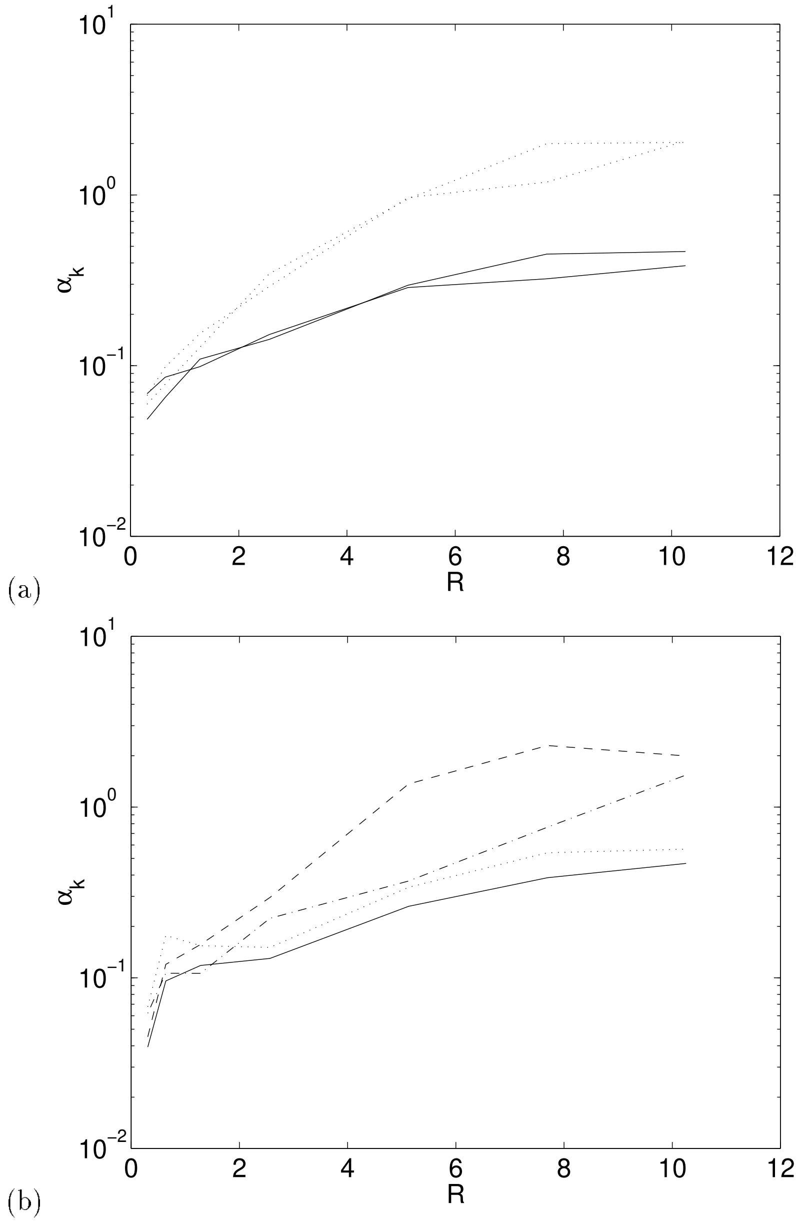


Figure 11: Effect of training set size on ARD. A plot of the inferred weight decay parameters, α_k, versus R, the ratio of training examples to network weights for (a) the original Neal data set containing two relevant inputs (solid lines) and two irrelevant inputs (dotted lines) and for (b) the Neal-5 data containing inputs $k = 1, 3, 4, 5$ of decreasing relevance indicated by solid, dotted, dash-dotted and dashed lines respectively. Bayesian regularization assigns larger weight decay parameters to less relevant inputs.

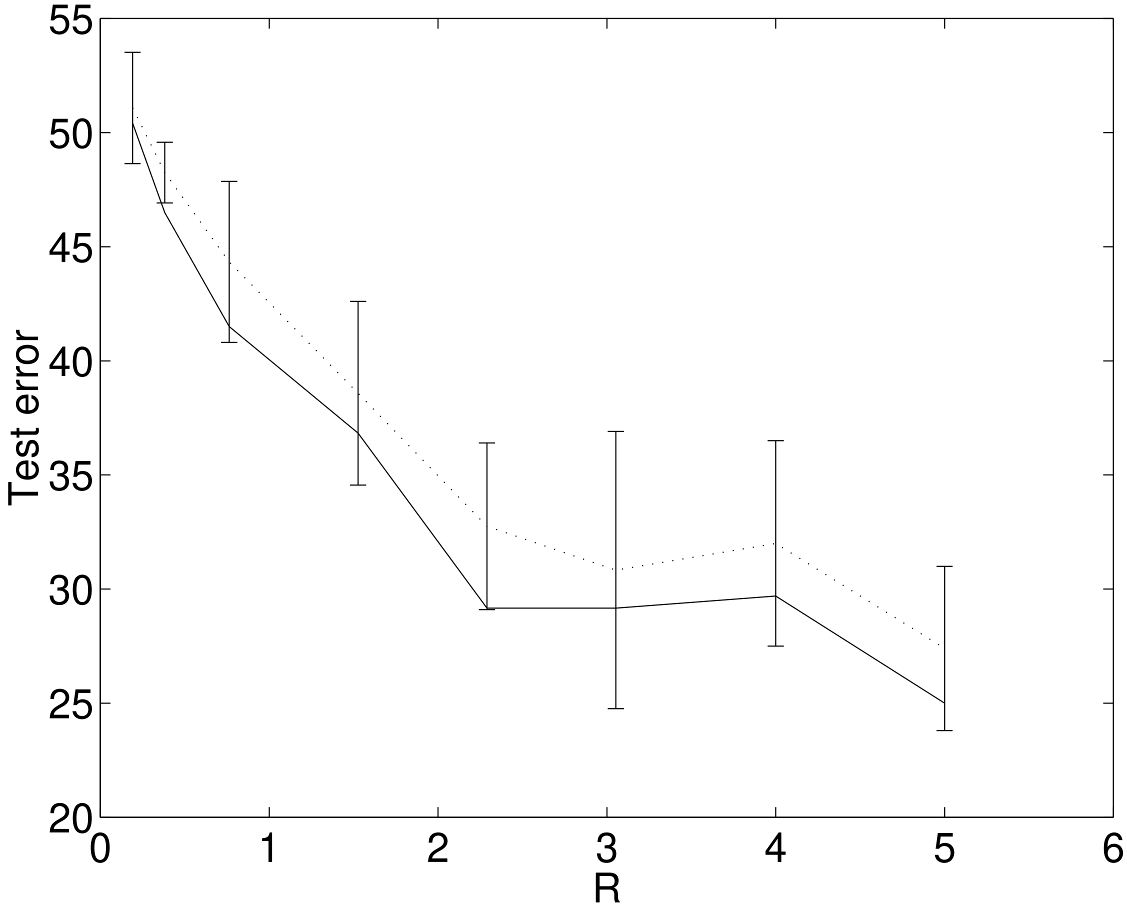


Figure 12: Test error versus R, the ratio of training examples to network weights on the 'Neal-12' data set which contains ten spurious inputs. The networks have eight hidden units. The solid lines are for nets with an ARD prior and the dotted lines for no ARD. The reduction in test error using ARD at each point is of the order of one standard deviation. However, as the ARD test errors are this much lower at *every* point this reduction is significant.

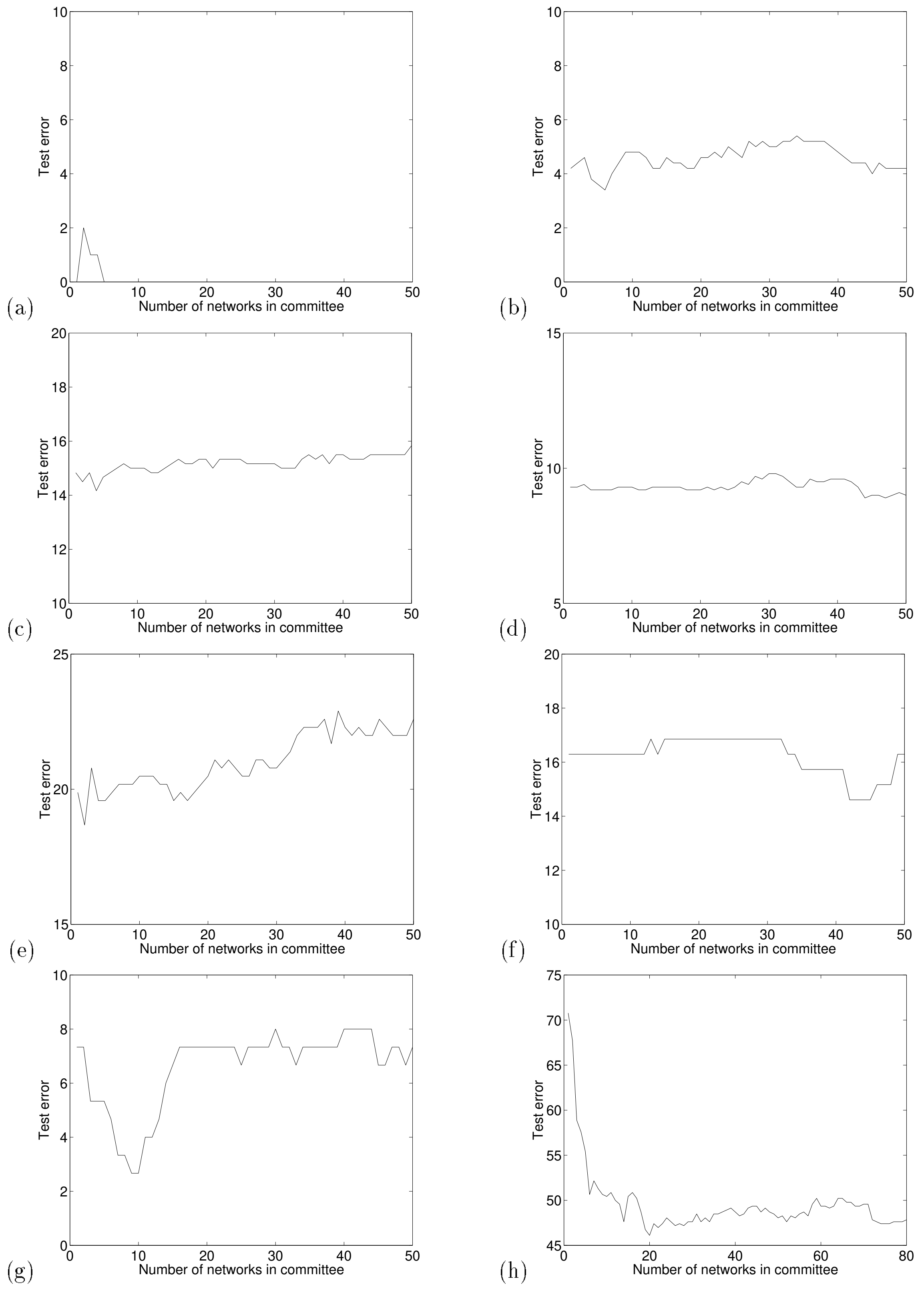


Figure 13: Committee test error versus number of networks in an evidence-ranked committee for (a) XOR, (b) Yin-Yang, (c) Neal, (d) Ripley, (e) Diabetes, (f) Tremor, (g) Ionosphere and (h) Vowel data sets.