ESL-PA-94/07-02 (1)

OR-94-17-1

BAYESIAN NONLINEAR MODELING FOR THE PREDICTION COMPETITION

David J.C. MacKay, Ph.D.

ABSTRACT

The 1993 energy prediction competition involved the prediction of a series of building energy loads from a series of environmental input variables. Nonlinear regression using neural networks is a popular technique for such modeling tasks. Since it is not obvious how large the input time-window should be or what preprocessing of inputs is best, this can be viewed as a regression problem in which there are many possible input variables, some of which may actually be 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 regularization or "weight decay") will not set the coefficients for these junk inputs to zero. Thus, the irrelevant variables will hurt the model's performance.

The automatic relevance determination (ARD) model puts a prior probability distribution over the regression parameters that embodies the concept of relevance. This is done in a simple and "soft" way by introducing multiple regularization constants—one associated with each input. Using Bayesian methods, the regularization constants for junk inputs are automatically inferred to be large, preventing those inputs from causing significant overfitting.

An entry using the ARD model won the competition by a significant margin.

OVERVIEW OF BAYESIAN MODELING METHODS

A practical Bayesian framework for adaptive data modeling is described in MacKay (1992). In this framework, the overall aim is to develop probabilistic models that are well matched to the data and make optimal predictions with those models. Neural network learning, for example, is interpreted as an inference of the most probable parameters for a model, given the training data. The search in model space (i.e., the space of architectures, noise models, preprocessings, regularizers, and regularization constants) can then also be treated as an inference problem, where the relative probability of alternative models can be inferred, given the data. Bayesian model comparison naturally embodies *Occam's razor*, the principle that states a preference for simple models.

Bayesian optimization of model control parameters has four important advantages: (1) no validation set is needed, so all the 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, as is a cross-validation measure; and (4) because the gradient of the evidence with respect to the control parameters can be evaluated, it is possible to optimize a large number of control parameters simultaneously.

Bayesian inference for neural nets can be implemented numerically by a deterministic method involving Gaussian approximations, the "evidence" framework (MacKay 1992), or Monte Carlo methods (Neal 1993). The former framework is used here.

Neural Networks for Regression

A supervised neural network is a nonlinear parameterized mapping from an input $\mathbf{x}$ to an output $\mathbf{y} = \mathbf{y}(\mathbf{x}; \mathbf{w})$. Here, the parameters of the net are denoted by $\mathbf{w}$. Such networks can be "trained" to perform regression, binary classification, or multi-class classification tasks.

In the case of a regression problem, the mapping for a "two-layer network" may have the form:

$$h_j = f^{(1)}\left(\sum_k w_{jk}^{(1)} x_k + \theta_j^{(1)}\right),$$
$$y_i = f^{(2)}\left(\sum_j w_{ij}^{(2)} h_j + \theta_i^{(2)}\right) \qquad (1)$$

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 nonlinearity of $f^{(1)}$ at the "hidden layer" gives the neural network greater computational flexibility than a standard linear regression. Such a network is trained to fit a data set $D = \{\mathbf{x}^{(m)}, \mathbf{t}^{(m)}\}$ by minimizing an error function, e.g.,

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

This function is minimized using an optimization method that makes use of the gradient of E_D, which can be

David J.C. MacKay is the Royal Society Smithson Research Fellow at the Cavendish Laboratory, Cambridge, UK.

evaluated using "back propagation" (the chain rule) (Rumelhart et al. 1986). Often, regularization (weight decay) is included, modifying the objective function to

$$M(\mathbf{w}) = \beta E_D + \alpha E_W \tag{3}$$

where $E_W = 1/2 \sum_i w_i^2$. The additional term decreases the tendency of a model to "overfit" the details of the training data.

Neural Network Learning as Inference

This neural network learning process can be given the following probabilistic interpretation. The error function is interpreted as the log likelihood for a noise model, and the regularizer is interpreted as a prior probability distribution over the parameters:

$$P(D|\mathbf{w},\beta,\mathcal{H}) = \frac{1}{Z_D(\beta)}\exp(-\beta E_D); \qquad P(\mathbf{w}|\alpha,\mathcal{H}) = \frac{1}{Z_W(\alpha)}\exp(-\alpha E_W). \tag{4}$$

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

$$P(\mathbf{w}|D,\alpha,\beta,\mathcal{H}) = \frac{P(D|\mathbf{w},\beta,\mathcal{H})P(\mathbf{w}|\alpha,\mathcal{H})}{P(D|\alpha,\beta,\mathcal{H})} = \frac{1}{Z_M}\exp(-M(\mathbf{w})). \tag{5}$$

This interpretation adds little at this stage. But new ideas emerge when we proceed to higher levels of inference.

Setting Regularization Constants α and β

The control parameters α and β determine the flexibility of the model. Bayesian probability theory can tell us how to set these parameters. All one needs to do is write down the inference one wishes to make, namely, the probability of α and β given the data, and then use Bayes' theorem:

$$P(\alpha,\beta|D,\mathcal{H}) = \frac{P(D|\alpha,\beta,\mathcal{H})P(\alpha,\beta|\mathcal{H})}{P(D|\mathcal{H})}. \tag{6}$$

The data-dependent term, $P(D|\alpha,\beta,\mathcal{H})$, is the normalizing constant from the previous inference (Equation 5); this is called the "evidence" for α and β. This pattern of inference continues if one wishes to compare model $\mathcal{H}$ with other models, using different architectures, regularizers, or noise models. Alternative models are ranked by evaluating $P(D|\mathcal{H})$, the normalizing constant of the inference in Equation 6.

Assuming there is only weak prior knowledge about the noise level and the smoothness of the interpolant, the evidence framework optimizes constants α and β by finding the maximum of the evidence. If the posterior probability distribution can be approximated by a Gaussian,

$$P(\mathbf{w}|D,\alpha,\beta,\mathcal{H}) = \frac{1}{Z_M} \exp\left(-M(\mathbf{w}_{MP}) + \frac{1}{2}(\mathbf{w}-\mathbf{w}_{MP})^T \mathbf{A}(\mathbf{w}-\mathbf{w}_{MP})\right), \tag{7}$$

then the maximum of the evidence has elegant properties that allow it to be located on-line. Here the method for the case of a single regularization constant α is summarized. As shown in MacKay (1992), the maximum evidence α satisfies the following self-consistent equation:

$$1/\alpha = \sum_i w_i^{MP^2}/\gamma \tag{8}$$

where $\mathbf{w}^{MP}$ is the parameter vector that minimizes the objective function $M = \beta E_D + \alpha E_W$ and γ is the number of well-determined parameters, given by $\gamma = k - \alpha\mathrm{Trace}(\mathbf{A}^{-1})$, where k is the total number of parameters and $\mathbf{A} = -\nabla\nabla\log P(\mathbf{w}|D,\mathcal{H})$. The matrix $\mathbf{A}^{-1}$ measures the size of the error bars on the parameters w. Thus $\gamma \to k$ when the parameters are all well determined; otherwise, $0 < \gamma < k$. Noting that $1/\alpha$ correponds to the variance σ_w^2 of the assumed distribution for $\{w_i\}$, Equation 8 specifies an intuitive condition for matching the prior to the data, $\sigma_w^2 = \langle w^2 \rangle$, where the average is over the γ effective parameters (the other $k - \gamma$ effective parameters having been set to zero by the prior).

Equation 8 can be used as a re-estimation formula for α. The computational overhead for these Bayesian calculations is not severe: one only needs to evaluate properties of the error bar matrix, $\mathbf{A}^{-1}$. The author has evaluated this matrix explicitly; this does not take much time if the number of parameters is small (a few hundred). For large problems, these calculations can be performed more efficiently (Skilling 1993).

Automatic Relevance Determination

The automatic relevance determination (ARD) model (MacKay and Neal 1994) is a Bayesian model that can be implemented with the methods described in MacKay (1992).

Consider a regression problem in which there are many input variables, some of which are actually 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 regularization) will not set the coefficients for these junk

inputs to zero. Thus, the irrelevant variables will hurt the model's performance, particularly when there are many variables and few data.

What is needed is a model whose prior over the regression parameters embodies the concept of relevance so that the model is effectively able to infer which variables are relevant and switch the others off. A simple and "soft" way of doing this is to introduce multiple regularization constants—one α associated with each input—controlling the weights from that input to the hidden units. Two additional regularization constants are used to control the biases of the hidden units and the weights going to the outputs. Thus, in the ARD model, the parameters are divided into classes c, with independent scales α_c. Assuming a Gaussian prior for each class, $E_{W(c)} = \sum_{i \in c} w_i^2/2$ can be defined so the prior is

$$P(\{w_i\} | \{\alpha_c\}, \mathcal{H}_{ARD}) = \frac{1}{\prod Z_{W(c)}} \exp\left(-\sum_c \alpha_c E_{W(c)}\right) . \quad (9)$$

The evidence framework can be used to optimize all the regularization constants simultaneously by finding their most probable value, i.e., the maximum over $\{\alpha_c\}$ of the evidence, $P(D|\{\alpha_c\}, \mathcal{H}_{ARD})$. The regularization constants for junk inputs are inferred to be large, thus preventing those inputs from causing significant overfitting.

In general, caution should be exercised when simultaneously maximizing the evidence over a large number of hyperparameters; probability maximization in many dimensions can give results that are unrepresentative of the whole probability distribution. In this application, the relevances of the input variables are expected to be approximately independent, so the joint maximum over $\{\alpha_c\}$ is expected to be representative.

PREDICTION COMPETITION: PART A

The American Society of Heating, Refrigerating and Air-Conditioning Engineers organized a prediction competition that was active from December 1992 to April 1993. Both parts of the competition involved creating an empirical model (as distinct from a physical model) based on training data and making predictions for a test set. Part A involved three target variables, and the test set came from a different time period from the training set so that extrapolation was involved. Part B had one target variable and was an interpolation problem.

The Task

The training set consisted of hourly measurements from September 1, 1989, to December 31, 1989, of four input variables (temperature, humidity, solar flux, and wind) and three target variables (electricity, cooling water, and heating water)—2,926 data points for each target. The test set consisted of the input variables for the next 54 days—1,282 data points. The organizers requested predictions for the test set; no error bars on these predictions were requested. The performance measures for predictions were the coefficient of variation (CV, a sum-squared error measure normalized by the data mean) and the mean bias error (MBE, the average residual normalized by the data mean). The three target variables are displayed in their entirety, along with the final predictions and residuals of the author's models, in Figures 1 through 3.

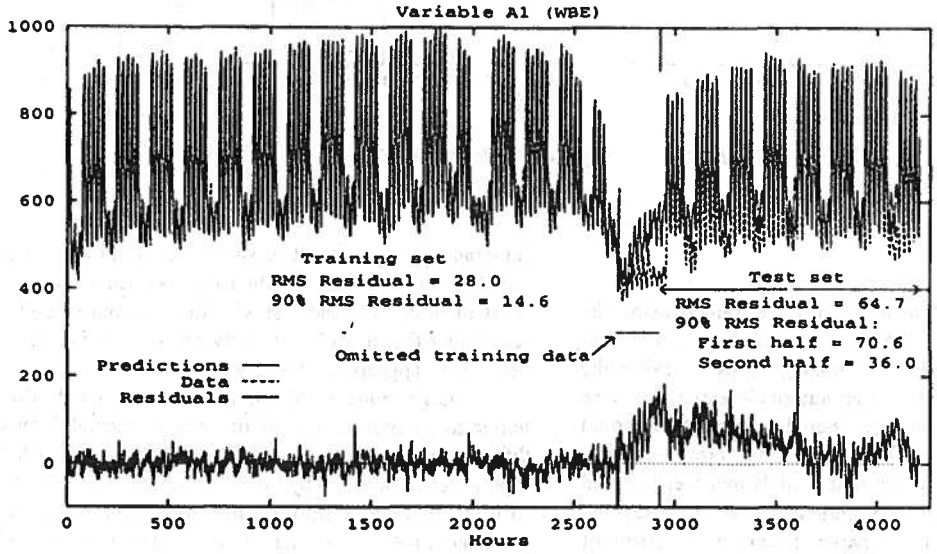

Figure 1 Target A1—electricity.

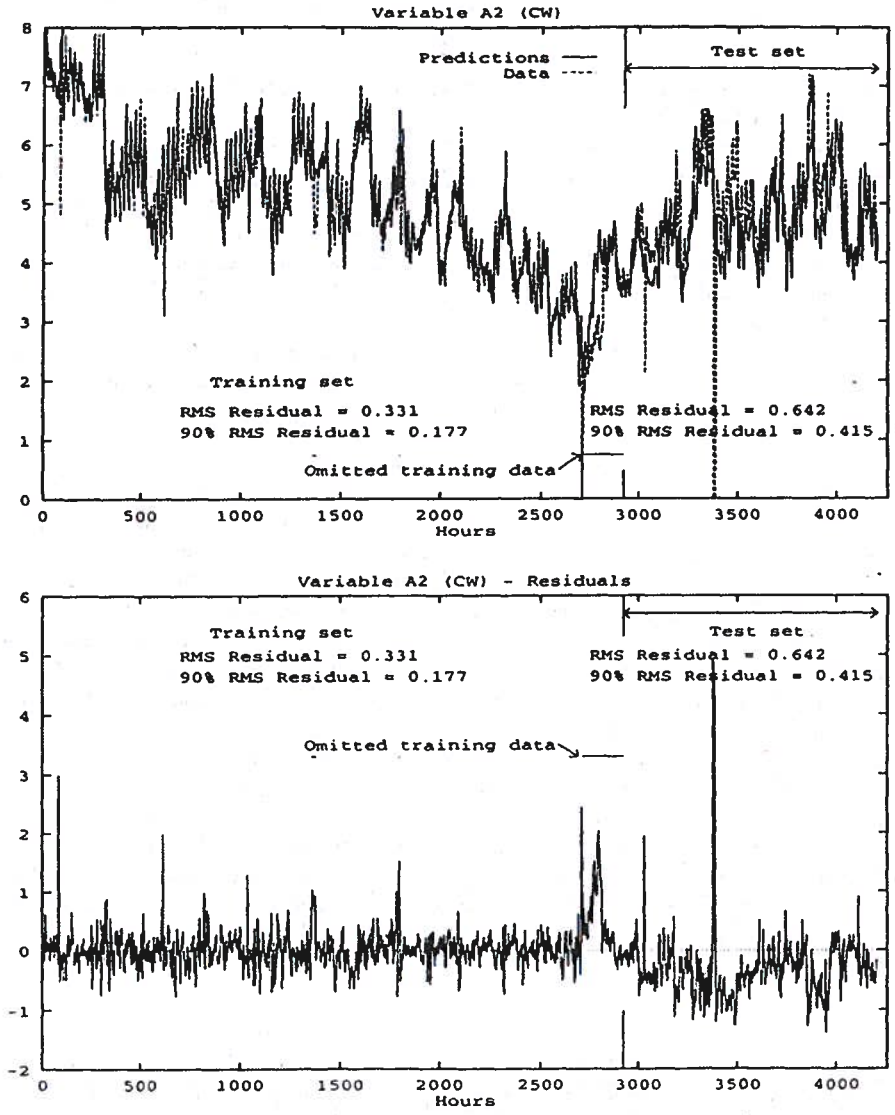

Figure 2 Target A2—cooling water.

Method

A large number of neural nets were trained using the ARD model for each of the prediction problems. The data seemed to include some substantial glitches. Because the author had not yet developed an automatic Bayesian noise model that anticipates outliers (though this certainly could be done [Box and Tiao 1973]), those data points that gave large residuals relative to the first models that were trained were omitted by hand. These omitted periods are indicated on some of the graphs in this paper. Twenty-five percent of the data was selected at random as training data, with the remainder being left out to speed the optimizations and to use as a validation set. All the networks had a single hidden layer of tanh units and a single linear output (Figure 4). It was found that models with between four and eight hidden units were appropriate for these problems.

A large number of inputs were included: different temporal preprocessings of the environmental inputs and different representations of time and holidays. All these inputs were controlled by the ARD model. The ARD proved to be a moderately useful guide for decisions concerning preprocessing of the data, in particular, how much time history to include. Moving averages of the environmental

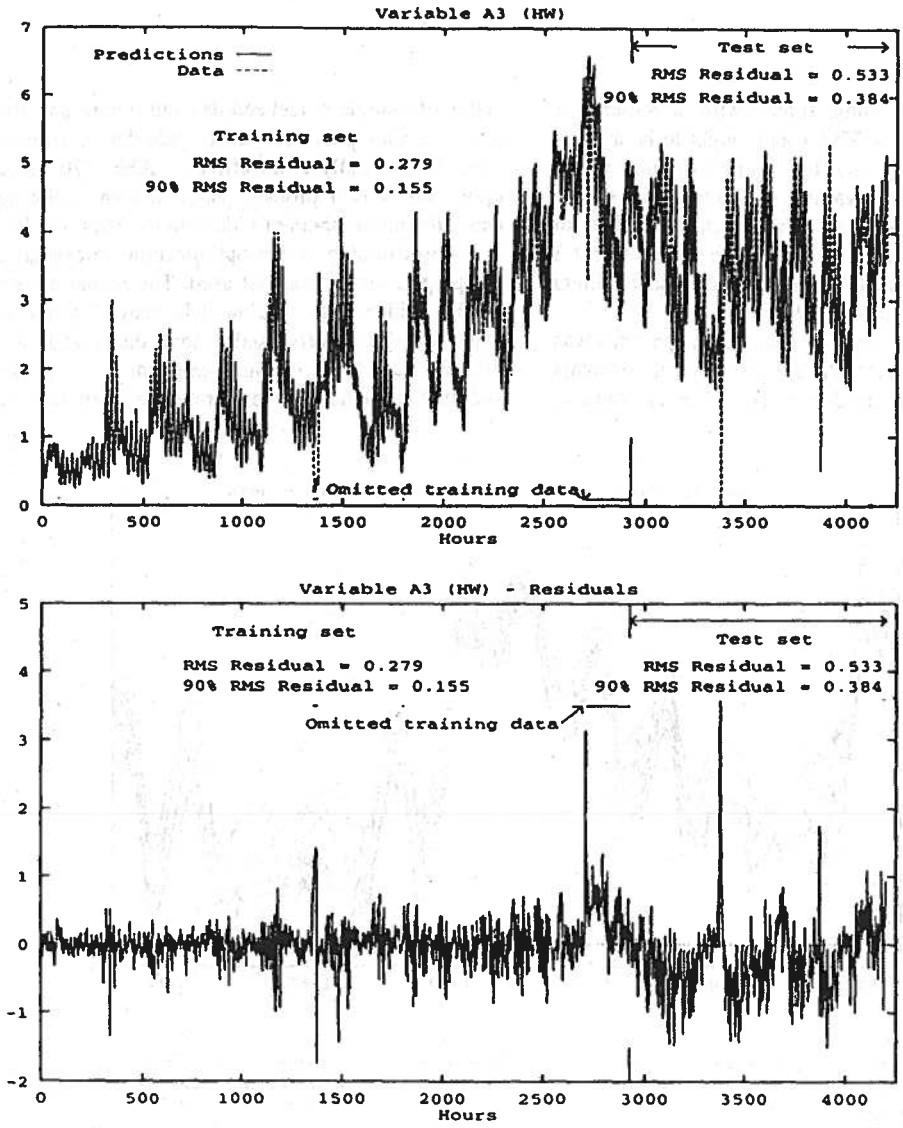

Figure 3 *Target A3—heating water.*

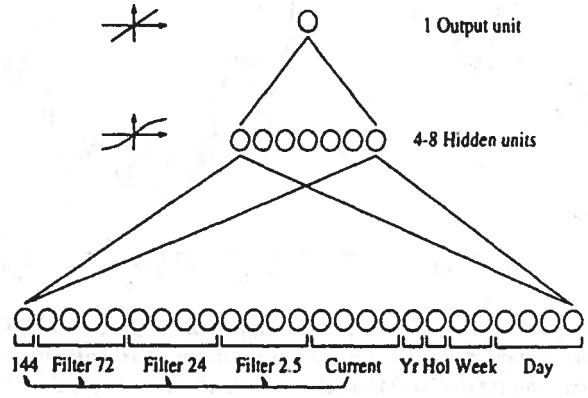

Figure 4 *A typical network used for problem A.*

variables were created using filters with a variety of exponential time constants. This was thought to be a more appropriate representation than time delays because filters suppress noise in the input variables, allowing one to use fewer filtered inputs with long time constants; and, with exponentially filtered inputs, it is easy to create (what I believe to be) a natural model, giving equal status to filters having time scales of 1, 2, 4, 8, 16, etc.

The on-line optimization of regularization constants was very successful. For problem A, 28 such control constants were simultaneously optimized in every model. The optimization of a single model and its control constants took about one day on a workstation, using code that could probably be made substantially more efficient. About 20 models were optimized for each problem using different initial conditions and different numbers of hidden units. Most models did not show overtraining as the optimization proceeded, so early stopping generally was not used. The numerical evaluation of the evidence for the models proved problematic, so validation errors were used to rank the models for prediction. For each task, a committee of models was assembled, and their predictions were averaged together (see Figure 5);

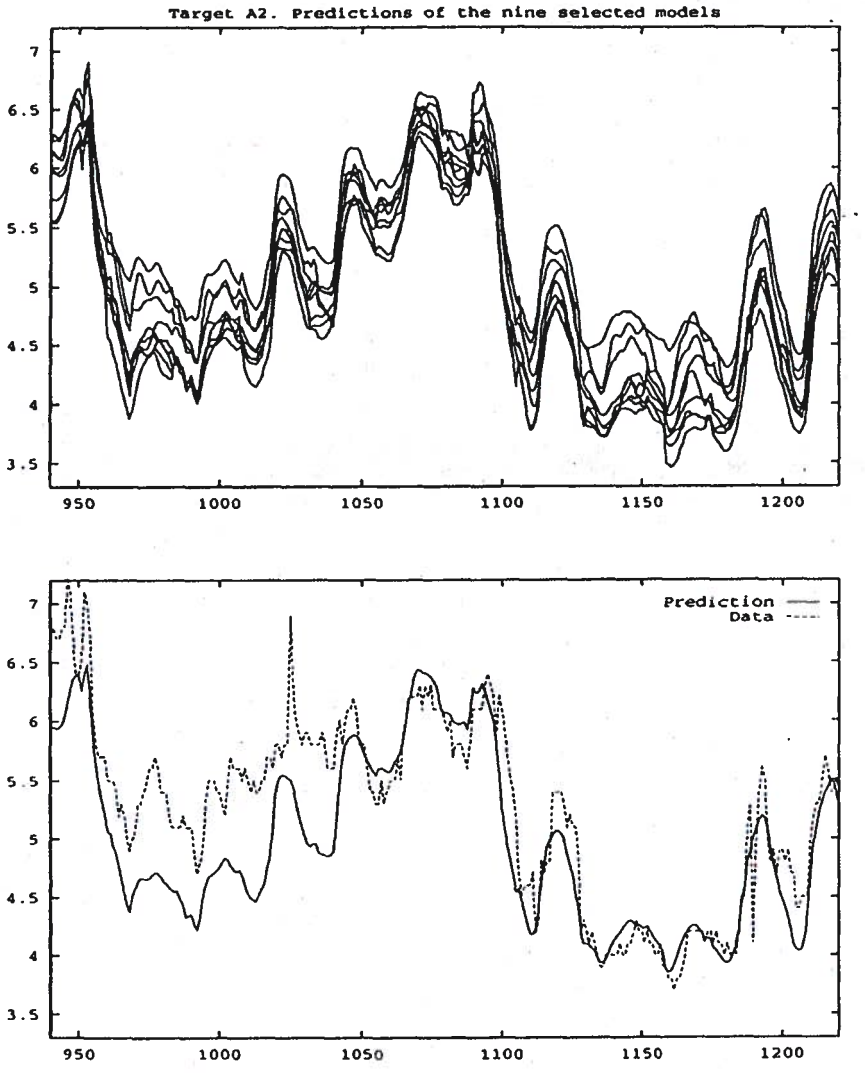

Figure 5 *Target A2—detail from test period. This figure shows detail from Figure 2 and illustrates the use of a "committee" of nine equally weighted models to make predictions. The diversity of the different models' predictions emphasizes the importance of elucidating the uncertainty in one's predictions. The x-axis is the time in hours from the start of the testing period. The prediction (lower graph) is the mean of the functions produced by the nine models (upper graph).*

this procedure was intended to mimic the Bayesian predictions $P(t|D) = \int P(t|D, \mathcal{H})P(\mathcal{H}|D)\, d\mathcal{H}$. The size of the committee was chosen so as to minimize the validation error of the mean predictions. This method of selecting committee size has also been described under the name *stacked generalization* (Breiman 1992). In all cases, a committee was found that performed significantly better on the validation set than any individual model.

The predictions and residuals are shown in Figures 1 through 3. There are local trends in the testing data that the models were unable to predict. Such trends were presumably overfitted in the training set. Clearly, a model incorporating local correlations among residuals is called for. Such a model would not perform much better by the competition criteria, but its on-line predictive performance would be greatly enhanced.

In the competition rules, it was suggested that scatter plots of the model predictions versus temperature should be made. The scatter plot for problem A3 is particularly interesting. Target A3 showed a strong correlation with temperature in the training set (dots in Figure 6b). When the models' predictions for the testing set were examined, I was surprised to find that, for target A3, a significantly offset correlation was predicted ("+"s in Figure 6a). This change in correlation turned out to be correct ("+"s in Figure 6b). This indicates that these nonlinear models controlled with Bayesian methods discovered nontrivial underlying structure in the data. Most other entrants' predictions for target A3 showed a large bias; presumably none of their models extracted the same structure from the data.

In the models used for problem A3, the values of the parameters $\{\alpha_c, \gamma_c\}$ have been examined; these give at least a qualitative indication of the inferred "relevance" of the inputs. For prediction of the hot water consumption, the time of year and the current temperature were the most relevant variables. Also highly relevant were the holiday indicator, the time of day, the current solar flux and wind speed, and the moving average of the temperature over the last 144 hours. The current humidity was not relevant, but the moving average of the humidity over 72 hours was. The solar flux was relevant on a time scale of 24 hours. None of the 2.5-hour filtered inputs seemed especially relevant.

How Much Did ARD Help?

An indication of the utility of the ARD prior was obtained by taking the *final* weights of the networks in the optimal committees as a starting point and training them further using the standard model's regularizer (i.e., just three regularization constants). The dotted lines in Figure 7 show the validation error of these networks before and after adaptation. As a control, the solid lines show what happened to the validation error when the same networks were used as a starting point for continued optimization under the ARD model. The validation error is a noisy performance measure, but the trend is clear: the standard models suffer an increase in error of between 5% and 30% because of overfitting by the parameters of the less relevant inputs; the ARD models, on the other hand, do not overfit with continued training. In some cases, the validation errors for the ARD model change with continued training because the restarting procedure set the α_i to default values, which displaced the model parameters into a new optimum.

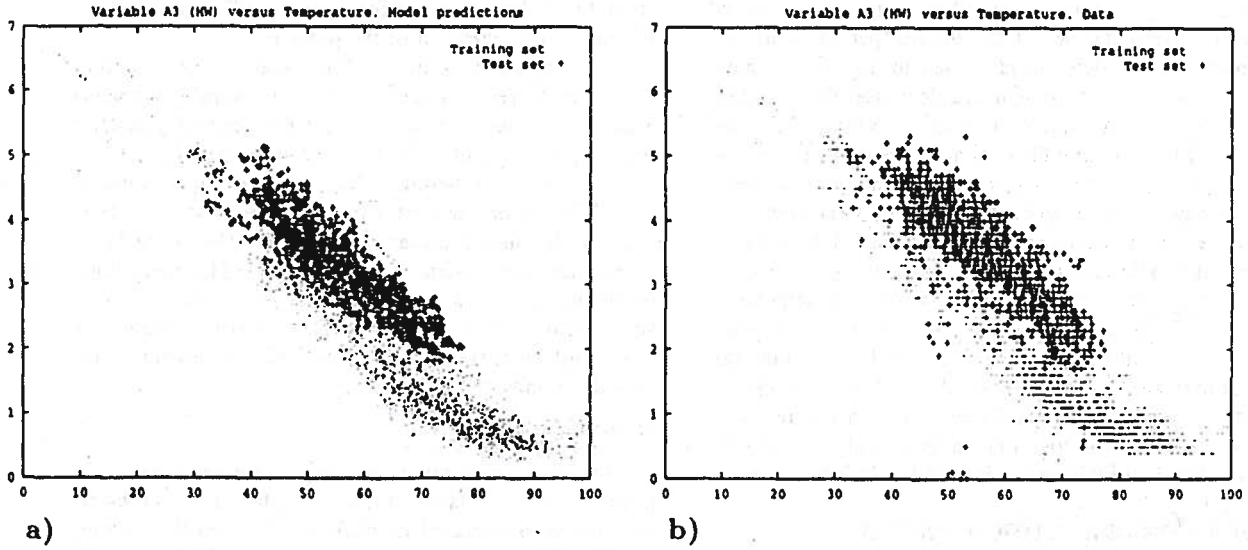

Figure 6 *Predictions for target A3 (hot water) versus temperature. (a) Model predictions. This graph shows that the author's model predicted a substantially different correlation between target A3 and temperature (+) from that shown in the training set (·). (b) Data. This predicted offset was correct. Units: hot water (10^6 Btu) versus temperature (°F).*

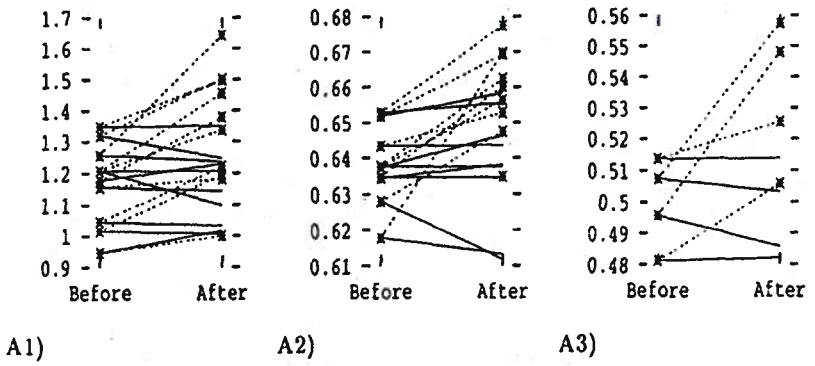

Figure 7 *Change in validation error when the ARD prior is suspended. The solid lines without stars show the performance of ARD models. The dotted lines with stars show the models with ARD suspended. In most cases, these standard ("ARD off") models get significantly worse.*

On the competition test data, the performance difference between these two sets of models is not so pronounced because the residuals are dominated by other effects. Maybe the greatest contribution of the ARD method to this problem was that it guided the choice of input variables to include large time delays.

After the competition, it was revealed that the building in this study was a large university engineering center in Texas. Some of the glitches in the data were caused by the bursting of water pipes during a frost—a rare event apparently not anticipated by Texas architects. The holiday period for staff ended on January 1, but the student population did not return to the building for a couple of weeks. This may account for the significant bias error in the predictions of electricity use (Figure 1). Another factor that changed between the training period and the test period is that the computer science department moved to another building. This also caused a reduction in electricity use. The reduction in electricity consumption may also account for some fraction of the biases in the cold and/or hot water supplies: one might expect less cooling water to be used, or more heating water, to make up the missing energy. The observed average electrical power deficit (according to the author's model) of 50 kilowatts corresponds to an expected decrease in cold water or an increase in hot water consumption of 0.17×10^6 Btu (assuming that the cold and hot water figures measure the actual energy delivered to the building). This is only about a fifth of the overall shift in correlation between hot water and temperature shown in Figure 6b. In fact, relative to the author's models, both cold water and hot water showed an increase of about 0.2×10^6 Btu.

PREDICTION COMPETITION: PART B

The data for part B consisted of 3,344 measurements of four input variables at hourly intervals during daylight hours over about 300 days. Quasi-random chunks of this data set had been extracted to serve as a test set of 900. The other 2,444 examples were accompanied by a single target variable. The physical source of the data was measurements of solar flux from five outdoor devices. Four of the devices had a fixed attitude. The fifth, whose output was to be predicted, was driven by motors so that it pointed at the sun. The aim was to enable four cheap fixed devices to substitute for one expensive moving one. Clearly, information such as the day of the week and past history of the input variables was not expected to be relevant. However, I did not realise this, and I spent some time exploring different temporal preprocessings of the input. Satisfyingly, all time-delayed inputs and the time of the week were correctly found to be irrelevant by the ARD model, and these inputs were pruned from the final models used for making predictions—without physical comprehension of the problem.

The inputs used in the final models were the four sensor measurements and a five-dimensional continuous encoding of the time of day and the time of year. For training, one-third of the training set was selected at random and the remaining two-thirds were reserved as a validation set. This random selection of the training set was later regretted because it leaves a Poisson distribution of holes where there are no training data. This caused the predictions of the author's models to become unnecessarily poor on a small fraction of the testing data. As in part A, a committee of networks was formed. Each network had between 5 and 10 hidden units.

Results

Problem B was a much easier prediction problem. This is partly due to the fact that it was an interpolation problem, with test data extracted in small chunks from the training set. Typical residuals were less than 1% of the data range, and contrasts between different methods were not great. Most of the sum-squared error of the author's models' predictions is due to a few outliers.

TABLE 1
Performance of Different Methods on Test Sets

Problem A1	RMS	Mean	CV	MBE	$RMS_{90\%}$	$Mean_{90\%}$	RCV
ARD	64.7	50.3	10.3	8.1	54.1	42.2	11.1
ARD off	71.2	56.2	11.4	9.0	59.3	47.3	12.2
Entrant 6			11.8	10.5			
Median			16.9	-10.4			
Problem A2	**RMS**	**Mean**	**CV**	**MBE**	$RMS_{90\%}$	$Mean_{90\%}$	**RCV**
ARD	.642	-.314	13.0	-6.4	.415	-.296	11.2
ARD off	.668	-.367	13.5	-7.4	.451	-.349	12.2
Entrant 6			13.0	-5.9			
Median			14.8	-7.6			
Problem A3	**RMS**	**Mean**	**CV**	**MBE**	$RMS_{90\%}$	$Mean_{90\%}$	**RCV.**
ARD	.532	-.204	15.2	-5.8	.384	-.167	9.15
ARD off	.495	-.121	14.2	-3.5	.339	-.094	8.08
Entrant 6			30.6	-27.3			
Median			31.0	-27.0			
Problem B	**RMS**	**Mean**	**CV**	**MBE**	$RMS_{90\%}$	$Mean_{90\%}$	**RCV**
ARD	11.2	1.1	3.20	0.32	6.55	0.67	.710
Entrant 6			2.75	0.17			
Median			6.19	0.17			

Key:

My models:

- ARD — The predictions entered in the competition using the ARD model.
- ARD off — Predictions obtained using derived models with the standard regularizer.

Other entries:

- Entrant 6 — The entry which came 2nd by the competition's average CV score.
- Median — Median (by magnitude) of scores of all entries in competition.

Raw Performance measures:

- RMS — Root mean square residual.
- Mean — Mean residual.
- CV — Coefficient of variation (percentage). The competition performance measure.
- MBE — Mean Bias Error (percentage).

Robust Performance measures:

- $RMS_{90\%}$ — Root mean square of the smallest 90% of the residuals.
- $Mean_{90\%}$ — Mean of those residuals.
- RCV — $RMS_{90\%}$/(90% data range).

Normalizing constants:

Problem	Mean of test data	90% data range
A1	624.77	486.79
A2	4.933	3.7
A3	3.495	4.2
B	350.8	923

DISCUSSION

The ARD prior was a success because it made it possible to include a large number of inputs without fear of overfitting. Further work could be well spent on improving the noise model, which assumes that the residuals are Gaussian and uncorrelated from frame to frame. A better predictive model for the residuals shown in Figures 1 through 3 might represent the data as the sum of the neural net prediction and an unpredictable, but autocorrelated, additional disturbance. Also, a robust Bayesian noise model is needed that captures the concept of outliers.

In conclusion, the winning entry in this competition was created using the following data-modeling philosophy: use huge flexible models, including all possibilities that might be appropriate; control the flexibility of these models using sophisticated priors; and use Bayes as a helmsman to guide the search in this model space.

ACKNOWLEDGMENTS

I am grateful to Radford Neal for invaluable discussions. I thank the Hopfield group, Caltech, and the members of the radioastronomy lab at Cambridge for generous sharing of computer resources. This work was supported by a Royal Society research fellowship and by the Defense Research Agency, Malvern.

REFERENCES

Box, G.E.P., and G.C. Tiao. 1973. *Bayesian inference in statistical analysis*. New York: Addison-Wesley.

Breiman, L. 1992. Stacked regressions. Technical Report 367, Dept. of Statistics. Berkeley: University of California.

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

MacKay, D.J.C., and R.M. Neal. 1994. Automatic relevance determination for neural networks. Technical report in preparation. Cambridge, UK: Cambridge University.

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

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

Skilling, J. 1993. Bayesian numerical analysis. In *Physics and Probability*, W.T.G. Milonni, Jr., and P. Milonni, eds. Cambridge, UK: Cambridge University Press.