Node Relevance Determination

Neil D. Lawrence
Microsoft Research, Cambridge, U.K.*
neil@thelawrences.net

Abstract

Hierarchical Bayesian inference in parameterised models offers an approach for controlling complexity. In this paper we utilise a novel prior for the leaning of a model's structure. We call the prior *node relevance determination*. It is applicable in a range of models including sigmoid belief networks and Boltzmann machines. We demonstrate how the approach may be applied to determine structure in a multi-layer perceptron.

1 Introduction

Bayesian inference provides one approach to optimising model complexity. In *maximum likelihood learning* we find a particular parameterisation, $\hat{\boldsymbol{\theta}}$, for our model, $\mathcal{M}$, from the set of all possible parameterisations, $\boldsymbol{\theta}$, through maximising the log likelihood of the data:

$$\ln p(D|\boldsymbol{\theta}, \mathcal{M}) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\boldsymbol{\theta}, \mathcal{M}). \tag{1}$$

Here the data-set, D, has been assumed to be composed of N independent observations $\mathbf{x}_n$. In *Bayesian learning* an inference process replaces this optimisation. Rather than considering point estimates, $\hat{\boldsymbol{\theta}}$, of the parameters we treat them as stochastic variables. We then infer the posterior distribution of the parameters given the data. To determine this posterior we are also required to define a *prior* distribution over the parameters, $p(\boldsymbol{\theta})$. Once we have selected a prior we marginalise the parameters and obtain the model likelihood

$$p(D|\mathcal{M}) = \int p(D|\boldsymbol{\theta}, \mathcal{M}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{2}$$

This model likelihood can then be made use of in model selection.

When the parameters are continuous, a common choice for the prior is a spherical, zero mean, Gaussian distribution. More complex priors are also possible, the parameters may be placed into G vectors, $\boldsymbol{\theta}_g$, each of which is associated with a separate hyper-parameter α_g: $p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \prod_{g=1}^{G} \left(\frac{\alpha_g^{K_g}}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{\boldsymbol{\alpha}_g}{2}\boldsymbol{\theta}_g^{\mathrm{T}}\boldsymbol{\theta}_g\right)$. Here K_g is the number of parameters in group g. In its most flexible form such a

*This work was completed while the author was at The Computer Laboratory, Cambridge University, Cambridge, U.K.

prior might contain a hyper-parameter for every parameter [7]. Normally, however, the groups will be larger. In the context of neural networks, for example, the weights may be grouped according to the role they play in the network, e.g. ARD priors [5, 4].

2 Node Relevance Determination

In this paper we present a novel prior which takes the grouping of weights a stage further. We consider a prior which places each parameter in two groups and utilise this prior to optimise model structure.

We will apply our prior to a two-layer feed-forward neural network with I input nodes, H hidden nodes and a single output node. The network function may therefore be written as $f(\mathbf{x}, \mathbf{w}) = \sum_{h=1}^{H} v_h g(\mathbf{u}_h^{\mathrm{T}} \mathbf{x})$, where $\mathbf{w} = \{\mathbf{u}_1 \dots \mathbf{u}_H, \mathbf{v}\}$ is a vector representing the parameters or 'weights' of the network. The input to hidden weights are represented by a matrix, $\mathbf{U}$, of H vectors $\mathbf{u}_h$, each vector being the weights that 'fan-in' to hidden unit h. $\mathbf{v}$ is the vector of the hidden to output weights, consisting of H elements v_h. We account for 'biases' by considering additional input and hidden nodes whose values are taken to be one at all times. The activation function $g(\cdot)$ is often taken to be a hyperbolic tangent, for reasons of tractability though we use an alternative, the cumulative Gaussian distribution function[1].

We model the data-set, $D = \{\mathbf{x}_n, t_n\}_{n=1}^{N}$, as being derived from a underlying true function $y(\mathbf{x})$ with Gaussian noise added. This leads us to consider a likelihood function of the form:

$$p(D|\mathbf{w}, \beta) = \left(\frac{\beta}{2\pi}\right)^{N/2} \exp\left(-\frac{\beta}{2} \sum_{n=1}^{N} (t_n - f(\mathbf{x}_n, \mathbf{w}))^2\right), \tag{3}$$

where β is a parameter governing the inverse noise variance. We implement the node relevance determination prior by associating a hyper-parameter with each node in the network. We split the hyper-parameters into three sub-groups: $\boldsymbol{\alpha}^{(I)}$, $\boldsymbol{\alpha}^{(H)}$ and $\alpha^{(O)}$, the sub-groups contain the hyper parameters associated with the input nodes, hidden nodes and output node respectively.

Our prior then takes the form

$$\begin{aligned} p(\mathbf{w}|\boldsymbol{\alpha}^{(I)}, \boldsymbol{\alpha}^{(H)}, \alpha^{(O)}) &= \prod_{i=1}^{I} \prod_{h=1}^{H} \left(\frac{\alpha_i^{(I)} \alpha_h^{(H)}}{2\pi}\right)^{\frac{1}{2}} \exp\left\{-\frac{1}{2} \alpha_i^{(I)} \alpha_h^{(H)} u_{ih}^2\right\} \\ &\quad \times \prod_{i=1}^{H} \left(\frac{\alpha_i^{(H)} \alpha^{(O)}}{2\pi}\right)^{\frac{1}{2}} \exp\left\{-\frac{1}{2} \alpha_i^{(H)} \alpha^{(O)} v_i^2\right\}, \end{aligned} \tag{4}$$

where u_{ih} and v_i are elements of the matrix $\mathbf{U}$ and the vector $\mathbf{v}$. We term this prior node relevance determination (NRD) because its objective is to determine the relevance of each node in the model.

[1] $g(x) = \sqrt{\frac{2}{\pi}} \int_0^x \exp(-t^2)\, dt$

We may treat the hyper-parameters with a second level of Bayesian inference for which we utilise the following hyper-prior: $p(\boldsymbol{\alpha}) = \prod_{i=1}^{I+H+1} \text{gam}(\alpha_i|a,b)$. where $\text{gam}(\cdot)$ is the gamma distribution[2]. Note that exact Bayesian inference in this model is intractable. We therefore turn to variational methods to make progress.

2.1 The Variational Approach

Consider the bound on the likelihood obtained through the introduction of a variational distribution $q(\mathbf{w}, \boldsymbol{\alpha}, \beta)$,

$$\ln p(D) \geq \int q(\mathbf{w}, \boldsymbol{\alpha}, \beta) \ln \frac{p(D, \mathbf{w}, \boldsymbol{\alpha}, \beta)}{q(\mathbf{w}, \boldsymbol{\alpha}, \beta)} \; d\mathbf{w} \; d\boldsymbol{\alpha} \; d\beta. \tag{5}$$

We now assume that the variational distribution factorises, $q_w(\mathbf{w}, \boldsymbol{\alpha}, \beta) = q_w(\mathbf{w}) q_\beta(\beta) q_{\alpha^{(I)}}(\boldsymbol{\alpha}^{(I)}) q_{\alpha^{(H)}}(\boldsymbol{\alpha}^{(H)}) q_{\alpha^{(O)}}(\boldsymbol{\alpha}^{(O)})$. As we are constraining our investigations to the treatment of the distribution, q_α, we only consider the *free-form optimisation* of that distribution [3, 1] leading to $q_\alpha(\boldsymbol{\alpha}) \propto \exp \langle \ln p(\mathbf{w}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}) \rangle_{q_w}$, where we have utilised $\langle \cdot \rangle_q$ to represent an expectation with respect to the distribution q. The free-form optimisation gives

$$q_{\alpha^{(H)}}(\boldsymbol{\alpha}^{(H)}) = \prod_{i=1}^{H} \Gamma(\alpha_i^{(H)} | \tilde{a}_\alpha^{(H)}, \tilde{b}_{\alpha^{(H)}i}). \tag{6}$$

Similar forms for the input and output hyper-parameters may also be obtained. The parameters of these q-distributions are found as

$$\tilde{a}_\alpha^{(I)} = a_\alpha^{(I)} + \frac{H}{2}, \qquad \tilde{b}_{\alpha^{(I)}i} = b_\alpha^{(I)} + \sum_{h=1}^{H} \frac{\left\langle \alpha_h^{(H)} \right\rangle \left\langle u_{ih}^2 \right\rangle}{2}$$

$$\tilde{a}_\alpha^{(H)} = a_\alpha^{(H)} + \frac{I+2}{2}, \qquad \tilde{b}_{\alpha^{(H)}h} = b_\alpha^{(H)} + \sum_{i=0}^{I} \frac{\left\langle \alpha_i^{(I)} \right\rangle \left\langle u_{ih}^2 \right\rangle}{2} + \frac{\left\langle \alpha^{(O)} \right\rangle \left\langle v_h^2 \right\rangle}{2}$$

$$\tilde{a}_\alpha^{(O)} = a_\alpha^{(O)} + \frac{H+1}{2}, \qquad \tilde{b}_{\alpha^{(O)}} = b_\alpha^{(O)} + \sum_{h=0}^{H} \frac{\left\langle \alpha_h^{(H)} \right\rangle \left\langle v_h^2 \right\rangle}{2}.$$

In the above equation the variable averages, $\langle \cdot \rangle$, are over their respective distributions. In the case of the likelihood function we defined for the regression neural network we may also calculate bound (5) (see [2]). This enables us to monitor convergence of the optimisation and additionally to perform model comparison.

[2] $\text{gam}(\tau|a,b) = \frac{b^a}{\Gamma(a)} \tau^{a-1} \exp(-b\tau)$.

2.2 Expectation-Maximisation Structure Optimisation

Optimisation of the lower bound on $p(D|\mathcal{M})$ with respect to the q-distributions can be viewed, in the context of the EM-algorithm, as an approximate expectation step. This expectation step can then be followed by a maximisation step which maximises the lower bound on $p(D|\mathcal{M})$ with respect to the structure of the model $\mathcal{M}$. This could involve the removal of individual weights, but here we consider only node removal. We use the following heuristic to select a node to remove. If we wish to remove a node in the hidden layer we first compute bound (5). We then compute the *effective number of parameters*[3] for each hidden node, $\gamma_h = \sum_i \frac{\left\langle \alpha_h^{(H)} \right\rangle \left\langle \alpha_i^{(I)} \right\rangle}{\left\langle u_{ih}^2 \right\rangle - \langle u_{ih} \rangle^2} + \frac{\left\langle \alpha_h^{(H)} \right\rangle \left\langle \alpha^{(O)} \right\rangle}{\left\langle v_h^2 \right\rangle - \langle v_h \rangle^2}$ (see [4]). We remove the node with the lowest effective number of parameters and re-evaluate the bound. If it has increased we continue training with the new structure; otherwise we replace the node. The same process can be undertaken for the input nodes.

3 Results

In all the experiments the variational distribution governing the parameters $\mathbf{w}$ was chosen to be a diagonal covariance Gaussian for its ease of implementation. Gamma priors were placed over the parameter governing noise variance, β. The posterior distribution of which was then determined by a variational free-form optimisation.

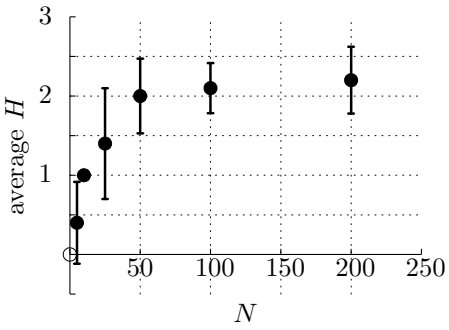

Figure 1: The average determined number of hidden nodes vs. number of data for the toy problem. The error bars show the standard deviation of the number of hidden nodes.

3.1 Toy Problem

To determine the effectiveness of the node-removal prior, we first studied a simple problem involving samples from a sine wave. We took N values from the function $0.4\sin(2\pi x)$. The x value was sampled from a uniform distribution over the interval[4] (0, 1). We added Gaussian noise of variance 0.0025 to the funciton output. Using this data a regression neural network with five hidden nodes was trained using the node relevance prior. We chose very broad hyper-priors by setting $a_\alpha^{(I)} = a_\alpha^{(H)} = a_\alpha^{(O)} = \sqrt{3 \times 10^{-4}}$ and $b_\alpha^{(I)} = b_\alpha^{(H)} = b_\alpha^{(O)} = 1 \times 10^{-3}$. A quasi-newton optimiser was used to optimise q_w. Optimisation was followed by an update of the posterior of β and $\boldsymbol{\alpha}$. We then attempted to optimise structure, in the manner described in the previous

[3] Also known as the *number of well determined parameters.*

[4] This is one period of oscillation.

Table 1: Performance of different priors on the data-sets.

Prior type	Sunspot	Tecator
Single	0.190 $(12 \times 8 \times 1)$	0.549 $(10 \times 8 \times 1)$
Grouped	0.194 $(12 \times 8 \times 1)$	0.540 $(10 \times 8 \times 1)$
ARD	0.153 $(3 \times 8 \times 1)$	0.539 $(10 \times 8 \times 1)$
NRD	0.163 $(5 \times 5 \times 1)$	0.532 $(10 \times 2 \times 1)$

section, by cycling through these operations ten times. The experiment was repeated for $N = 5$, 10, 25, 50, 100, 200. Ten networks were trained for N using a different set of samples from the function. The results are summarised in Figure 1, where the average number of determined hidden nodes is plotted against the number of data-points.

3.2 Real Data

Our first real world data-set involves the annual average[5] of sunspots from 1700 to 1920. This time series has served as a benchmark in the statistical literature [8]. The number of hidden nodes was initially taken to be eight and the input window was chosen arbitrarily to be 12, i. e. we modelled $x_n = f(x_{n-1}, \ldots x_{n-12})$. Training and test set selection was as in [9]. We also assessed the performance of our approach on the Tecator data-set[6]. This benchmark was first used by Thodberg [6] to demonstrate the benefits of an evidence approximation based Bayesian approach compared to the early stopping technique[7].

The optimisations, for both data-sets, were undertaken in a similar manner to those of the toy-problem. The NRD network with the highest model likelihood on the Tecator data used two hidden nodes and all the input nodes. For the sunspot data, the NRD network with the highest model likelihood used three hidden nodes and five input nodes. The input nodes used to predict[8] x_n were $x_{n-1}, x_{n-2}, x_{n-5}, x_{n-7}$ and x_{n-8}. For the sunspot data networks were initialised with $I = 12$ and $H = 8$, for the Tecator data $I = 10$ and $H = 8$. Table 1 summarises the results obtained. Alongside each result is the structure of the networks obtained in the form $I \times H \times 1$. The sunspot results quote the normalised mean squared error, however for the Tecator results we follow Thodberg [6] in our use of the standard error of prediction to enable comparisons. The priors we tried are named 'single' which considers only one hyper-parameter, 'grouped' which groups the weights according as input-hidden layer, hidden biases, hidden-output layer and output biases; ARD which further groups the

[5]The data are daily, monthly and annually reported by the Royal Observatory of Belgium and can be found at `http://www.oma.be/KSB-ORB/SIDC/sidc_txt.html`.

[6]The data are recorded on a Tecator Infratec Food and Feed Analyser working in the wavelength range 850 - 1050 nm by the Near Infrared Transmission (NIT) principle.

[7]The data-set is available from `http://temper.stat.cmu.edu/datasets/Tecator`

[8]Note that in time series prediction it is normal to try and select optimal windows of inputs. Normally a window of size W would include the inputs from $n-1$ to $n-W$.

input-hidden layer weights according to the input node with which they are associated and the NRD prior described above.

4 Discussion

We have introduced a novel form of prior for determining the relevance of individual nodes or variables in the network and showed how it may be used to determine structure automatically.

When implemented with the noisy sine data, higher complexity (in the form of more hidden nodes) was utilised by the algorithm as more data-points were presented to the model. This behaviour is in line with our expectations. Model complexity is able to increase as more information is provided.

In the benchmark data-sets we studied, the performance of the NRD prior was comparable with that of other widely used approaches. However, the NRD prior was able to discover more compact representations of the data.

References

[1] N. D. Lawrence. *Variational Inference in Probabilistic Models*. PhD thesis, Computer Laboratory, University of Cambridge, New Museums Site, Pembroke Street, Cambridge, CB2 3QG, U.K., 2000. Available from `http://www.thelawrences.net/neil`.

[2] N. D. Lawrence and M. Azzouzi. A variational Bayesian committee of neural networks. Available from `http://www.thelawrences.net/neil`, 1999.

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

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

[5] R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, 1996. Lecture Notes in Statistics 118.

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

[7] M. E. Tipping. The relevance vector machine. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 652–658, Cambridge, MA, 2000. MIT Press.

[8] H. Tong. *Non-linear Time Series: a Dynamical System Approach*, volume 6 of *Oxford Statistical Science Series*. Clarendon Press, Oxford, 1995.

[9] A. S. Weigend, B. A. Huberman, and D. E. Rumelhart. Predicting sunspots and exchange rates with connectionist networks. In S. Eubank and M. Casdagli, editors, *Proceedings of the 1990 NATO Workshop on Nonlinear Modeling and Forcasting, Santa Fe, New Mexico*. Addison-Wesley, 1990.