NEURAL COMPUTING
NCRG
RESEARCH GROUP

Neural Computing Research Group
Dept of Computer Science & Applied Mathematics
Aston University
Birmingham B4 7ET
United Kingdom
Tel: +44 (0)121 333 4631
Fax: +44 (0)121 333 4586
http://www.ncrg.aston.ac.uk/

Using Bayesian neural networks to classify segmented images

Francesco Vivarelli
F.Vivarelli@aston.ac.uk

Christopher K. I. Williams
C.K.I.Williams@aston.ac.uk

Technical Report NCRG/97/007 April 21, 1997

To appear in: Proc. Fifth IEE International Conference on Artificial Neural Networks.

Abstract

We present results that compare the performance of neural networks trained with two Bayesian methods, (i) the Evidence Framework of MacKay (1992) and (ii) a Markov Chain Monte Carlo method due to Neal (1996) on a task of classifying segmented outdoor images. We also investigate the use of the Automatic Relevance Determination method for input feature selection.

1 Introduction

This work deals with the Bayesian training of neural networks for classifying regions of outdoor scenes. Outdoor scene analysis is usually carried out on images which have been segmented into regions. To carry out the task successfully it is important to classify each region not only using its own attributes (e.g. colour, shape, texture) but also to take account of the *context* of the other regions. For example, this means that a region surrounded by sky should probably not be classified as a vehicle. Taking account of context can be handled in two ways; either by searching for a consistent interpretation of the whole scene, or by taking account the local context in which a region finds itself. Examples of the whole-scene method are [Ohta 1985; McKeown, Harvey, and McDermott 1985]. The second approach is found in [Wright 1989; Wright, Mackeown, and Greenway 1995; Mackeown 1994], where segmented regions are classified using neural networks. This work follows the same approach as Wright [Wright 1989], but it compares and contrasts two implementations of Bayesian learning of neural networks: the evidence framework approximation (EF) and the Markov Chain Monte Carlo method (MCMC).

The layout of the paper is as follows: The next section introduces the image database used for the classification task and the extraction of features. The two implementations of Bayesian learning of neural networks are then introduced and applied to the training of a Multiple Logistic Regression network (MLR) and a Multi Layer Perceptron (MLP). We then present the results on the region classification task, the use of the Automatic Relevance Determination technique of MacKay and Neal [Neal, R. M. 1996] to deal with the issue of feature selection and a comparison of the EF and MCMC methods on training sets of various size.

2 The Database

The database employed consists of 96 coloured images extracted from the Sowerby Image Database of British Aerospace. All the scenes have been photographed using small-grain 35mm transparency film. Each image of the database has been digitised with a calibrated scanner, generating an high quality 24-bit colour representation of size 768 by 512 pixels.

The segmentation of the images into regions has been achieved by using an *ideal segmentation*. An ideally segmented image is composed by the set of regions which correspond to a predefined set of labels. It is the segmentation that would be obtained from drawing *by hand* boundaries around each object appearing in the images. It was implemented by Mackeown [Mackeown 1994] by running a segmentation algorithm (based on a region growing method), merging the over-segmented adjacent regions which had received the same label, and splitting the under-segmented regions containing more than one object. Such an ideal segmentation permits the investigation of classification performance without being affected by inaccuracies due to the segmentation.

The numerical description of the database is realised by computing a feature vector which encodes the characteristics of each region. The features chosen are divided in two parts, the internal and the contextual features. The internal features describe the interior characteristics of a given region such as colour, topology, size, position, shape and texture. The contextual features describe the relationships between region and the regions around it. Following [Wright, Mackeown, and Greenway 1995], the contextual features are defined as the relative size, intensity and position of a region with respect to the four largest regions surrounding it. Because of the fixed length of the contextual features, if the number of surrounding regions is less than four there are some features which are not defined: in this case the undefined contextual features are set to the average of the defined components. Further details of the features used can be found in [Vivarelli 1996].

All of the features have been rescaled using a linear normalising transformation, obtaining rescaled features which have mean 0 and variance 1.

The 96 images have been divided randomly in two independent sets: a training set and a testing

		Training set		Testing set	
	Number of regions	5832		1505	
	Class	Region (%)	Area (%)	Region (%)	Area (%)
1	Clouds	7.90	8.18	6.38	11.03
2	Vegetation	27.35	35.37	26.25	33.19
3	Road Marking	1.68	0.20	1.79	0.17
4	Road Surface	15.29	39.30	13.02	40.40
5	Road Border	9.88	9.32	8.04	5.57
6	Building	20.82	4.20	24.98	5.19
7	Bounding object	7.13	2.08	6.45	2.17
8	Road sign	0.77	0.05	0.93	0.15
9	Telegraph Pole	2.45	0.26	2.79	0.23
10	Illumination Shadow	2.52	0.40	2.13	0.34
11	Mobile Object	4.20	0.65	7.24	1.56

Table 1: The table shows the labels used and the composition of the training and testing sets. For both data sets the first column gives the percentage of each class computed from the number of regions of the class in the data set against the total number of regions; the second column reports the percentage of each class computed from the total area of that class against the total area of all the regions in the data set.

set. Each region of the two sets has been hand labelled in one of the eleven categories. Table (1) shows a list of the 11 categories and the composition of the two sets.

3 Bayesian training of neural networks

For the classification of segmented images, neural networks with one (MLR) or two layers (MLP) of adaptable weights have been used. The activation function of the kth output unit is the softmax function $y_k^n = \exp a_k^n / \sum_{l=1}^{c} \exp a_l^n$, where $c = 11$ is the total number of classes, and a_k^n is the activation of the unit given the input vector $\mathbf{x}^n$. The activation function of the jth hidden unit of the MLP is the sigmoid function $z_j^n = \tanh a_j^n$.

For Bayesian training of neural networks it is natural to describe the prior specification of the networks in a two-level hierarchy. The first level involves a probability distribution over the synaptic weights $p(\mathbf{w}|\boldsymbol{\alpha})$ of a neural network given the vector of *hyperparameters* $\boldsymbol{\alpha}$. The second level of the hierarchical description is concerned with the probability distribution of the hyperparameters $p(\boldsymbol{\alpha})$.

In order to detect the relevant components of the input vector one hyperparameter can be associated with each group of weights which connects one input unit to all of the units in the next layer. This is the Automatic Relevance Determination (ARD) method of MacKay and Neal [Neal, R. M. 1996]. Two further hyperparameters control the distribution of the hidden-to-output weights and the biases of the output units of the MLP. This is shown in Fig. 1. The hyperparameter α_g controls the size of the group of weights g through a Gaussian prior distribution with 0 mean and standard deviation $\sigma_g = \sqrt{1/\alpha_g}$. After the training phase, the knowledge about the distribution of the network parameters is given by the Bayes' theorem

$$p(\mathbf{w}|\boldsymbol{\alpha}, \mathcal{D}) = \frac{p(\mathcal{D}|\mathbf{w})\, p(\mathbf{w}|\boldsymbol{\alpha})}{p(\mathcal{D}|\boldsymbol{\alpha})}, \tag{1}$$

where $\mathcal{D}$ is the data of the training set, $p(\mathcal{D}|\mathbf{w})$ is the likelihood and the denominator is the normalising factor $p(\mathcal{D}|\boldsymbol{\alpha}) = \int p(\mathcal{D}|\mathbf{w})\, p(\mathbf{w}|\boldsymbol{\alpha})\, d\mathbf{w}$.

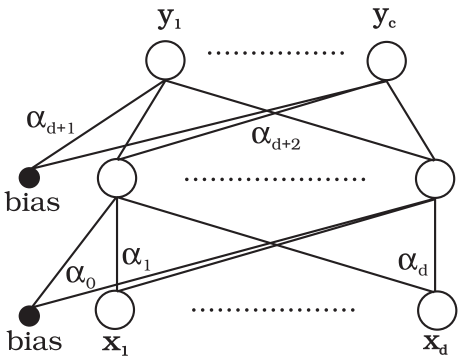

Figure 1: Graphical representation of the MLP with Automatic Relevance Determination. α_0 controls the biases of the hidden units; the hyperparameters $(\alpha_1, \ldots \alpha_d)$ control the groups of synaptic weights connecting each input to the hidden layer; α_{d+1} controls the biases of the output layer and α_{d+2} controls the weights connecting the hidden to the output units.

Bayes' theorem also expresses the probability distribution of $\boldsymbol{\alpha}$ given the data $\mathcal{D}$:

$$p(\boldsymbol{\alpha}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\alpha})\, p(\boldsymbol{\alpha})}{p(\mathcal{D})}, \tag{2}$$

where $p(\boldsymbol{\alpha})$ is the prior distribution. The factor $p(\mathcal{D}|\boldsymbol{\alpha})$ is called the evidence for $\boldsymbol{\alpha}$.

Bayesian prediction for a new input $\mathbf{x}$ is given by the *marginalisation* of the classification performed by the model $p(C_k|\mathbf{x}, \mathbf{w}, \boldsymbol{\alpha})$ with respect to the posterior distribution of the parameters:

$$p(C_k|\mathbf{x}, \mathcal{D}) = \int p(C_k|\mathbf{x}, \mathbf{w}, \boldsymbol{\alpha})\, p(\mathbf{w}, \boldsymbol{\alpha}|\mathcal{D})\, d\mathbf{w} d\boldsymbol{\alpha}. \tag{3}$$

The second term of the integrand represents the degree of belief in the values of the parameters $\mathbf{w}$ and $\boldsymbol{\alpha}$ given the data $\mathcal{D}$ and can be expressed as

$$p(\mathbf{w}, \boldsymbol{\alpha}|\mathcal{D}) = p(\mathbf{w}|\boldsymbol{\alpha}, \mathcal{D})\, p(\boldsymbol{\alpha}|\mathcal{D}). \tag{4}$$

There are two methods which are able to determine the posterior distribution (4) and they are based upon different approaches: the evidence framework and the Markov Chain Monte Carlo method.

3.1 The evidence framework

The EF (based on work by Gull) has been discussed by MacKay [MacKay 992a; MacKay 992b] and is similar to the *type II maximum likelihood* method.

The EF computes an approximation to equation (4) by assuming that the posterior probability of the hyperparameters $p(\boldsymbol{\alpha}|\mathcal{D})$ is sharply peaked around its maximum $\boldsymbol{\alpha}^{mp}$. To reflect the lack of knowledge about the best value of $\boldsymbol{\alpha}$, the hyper-prior $p(\boldsymbol{\alpha})$ is chosen as a constant function on a logarithmic scale. Thus the value of $\boldsymbol{\alpha}$ maximising the posterior $p(\boldsymbol{\alpha}|\mathcal{D})$ can be found maximising the evidence $p(\mathcal{D}|\boldsymbol{\alpha})$. Integrating over the parameter $\mathbf{w}$

$$p(\mathcal{D}|\boldsymbol{\alpha}) = \int p(\mathcal{D}|\mathbf{w})\, p(\mathbf{w}|\boldsymbol{\alpha})\, d\mathbf{w} \tag{5}$$

and approximating the integrand as a Gaussian centred around $\mathbf{w}^{mp}$, it is possible to maximise $p(\mathcal{D}|\boldsymbol{\alpha})$ with respect $\boldsymbol{\alpha}$. The mean of the Gaussian $\mathbf{w}^{mp}$ is the point of the weight space maximising the posterior $p(\mathbf{w}|\boldsymbol{\alpha}, \mathcal{D})$.

The components of the optimal values of $\boldsymbol{\alpha}^{mp}$ are given by

$$\alpha_g^{mp} = \gamma_g / \sum_{i \in g} (w_i^{mp})^2 \tag{6}$$

with $\gamma_g = W_g - \alpha_g \mathrm{Tr}_g \left(\nabla\nabla G + \boldsymbol{\alpha}\mathbf{I}\right)^{-1}$ and where W_g is the total number of weights controlled by α_g^{mp}, Tr_g is the trace of the matrix computed with respect to the components of $\mathbf{w}$ belonging to the group g and G is the negative logarithm of the penalized likelihood: $G = -\ln p\left(\mathcal{D}|\mathbf{w}\right) - \ln p\left(\mathbf{w}|\boldsymbol{\alpha}\right)$. Since the hyperparameters are scale factors for the weights, their uncertainty is usually represented on a logarithmic scale. The standard deviation of the distribution of the values of $\log \alpha_g$ is $\sigma_{\log \alpha_g|\mathcal{D}} = \sqrt{2/\gamma_g}$ (see [Bishop 1995]).

The EF proceeds by a two-step iterative procedure by computing the value $\mathbf{w}^{mp}$ maximising the penalised likelihood while periodically re-estimating the hyperparameter $\boldsymbol{\alpha}^{mp}$. In our experiments the first step has been achieved by optimising the penalised likelihood with a scaled conjugate algorithm for 70 iterations (for the MLP) or 50 iterations (for the MLR). The re-estimation of the hyperparameters $\boldsymbol{\alpha}^{mp}$ was carried out 10 times.

Some problems may arise during the implementation of EF. Since it is based on an approximation of the posterior around a local minimum, EF supposes that the Hessian matrix is positive definite at $\mathbf{w}^{mp}$. Sometimes this is not the case because the optimisation of the weights has not fully reached a local minimum. In this case some of the eigenvalues of can be negative, introducing instability in the implementation. A useful trick is to set the negative eigenvalues in $\nabla\nabla G$ to 0. Another problem is due to the fact that EF is based upon the computation of a Hessian matrix; for large networks the amount of storage required for this matrix is considerable, as its size grows as the square of the number of weights.

3.2 The Markov Chain Monte Carlo method

The second method of implementing the Bayesian learning of neural networks is by Markov Chain Monte Carlo [Neal, R. M. 1996]. An MCMC algorithm constructs a Markov chain whose equilibrium distribution is the desired probability density $p\left(\mathbf{w}, \alpha|\mathcal{D}\right)$. Although samples from the chain are not independent, they can be used for computing the necessary integrals.

The implementation of the algorithm[1] used in this work was written by R. Neal and is described in [Neal, R. M. 1996]. A brief outline is given below.

The posterior distribution of the network parameters is sampled using the *Hybrid Monte Carlo* method [Duane, Kennedy, Pendleton, and Roweth 1987]. This merges the Metropolis algorithm with a dynamical simulation. As it avoids random walks it performs better than a simple Metropolis algorithm.

In the Hybrid Monte Carlo method each network variable has an associated fictitious momentum, thereby creating a dynamical system which is described in phase space by a set of coordinates $(\mathbf{q}, \mathbf{p})$; for neural networks, the vector position $\mathbf{q}$ is interpreted as the network weights while $\mathbf{p}$ is the associated momenta. This system is described by an Hamiltonian function H given by the sum of the kinetic and the potential energies (denoted by $K\left(\mathbf{p}\right)$ and $V\left(\mathbf{q}\right)$ respectively). The kinetic energy is a function of the momentum vector $K\left(\mathbf{p}\right) = \mathbf{p}^T\mathbf{p}/2$. The potential energy is a function of the position $\mathbf{q}$. The Hybrid Monte Carlo method samples from the canonical distribution for $\mathbf{q}$ and $\mathbf{p}$ defined as $p\left(\mathbf{q}, \mathbf{p}\right) = \exp\left(-H\left(\mathbf{q}, \mathbf{p}\right)\right) = \exp\left(-\left(K\left(\mathbf{p}\right) + V\left(\mathbf{q}\right)\right)\right)$. Provided that $V\left(\mathbf{q}\right) = V\left(\mathbf{w}\right) = -\ln p\left(\mathbf{w}|\mathcal{D}, \boldsymbol{\alpha}\right)$, a set of values of $\mathbf{q}$ (whose posterior probability distribution is $p\left(\mathbf{q}\right) = p\left(\mathbf{w}|\mathcal{D}, \boldsymbol{\alpha}\right)$ is obtained by sampling from $p\left(\mathbf{q}, \mathbf{p}\right)$ and ignoring $\mathbf{p}$.

Sampling from the joint distribution $p\left(\mathbf{q}, \mathbf{p}\right)$ is achieved by generating new points in the phase space with constant value of H and then by doing a Gibbs sampling of the momentum $\mathbf{p}$ to change

[1] The source code of the implementation of Bayesian learning of neural networks is available at the ftp address: `ftp://ftp.cs.utoronto.ca/pub/radford/`.

the value of H. The *leap-frog* method is used to approximate the Hamilton's first order differential equations. At the end of a chain of L leapfrog steps the state of the system can be accepted or rejected depending upon the average value of the energy H over a window of states. For all the MCMC simulations reported below, the number of leap-frog steps L is 100, the window size is composed by 10 states and the step size correction factor (for the approximation of the Hamilton's equations) is 0.3.

The prior distributions for each group of weights is a 0 mean Gaussian whose precision (α) is specified by a vague Gamma prior. We also used scaling on the hidden-to-output weight as recommended in [Neal, R. M. 1996] so that the prior variances of the activations of the softmax units do not grow with the number of hidden units[2]. The sampling phase of the MCMC simulations has been run for 200 iterations, and the first third of these have been discarded try to let the simulation reach the equilibrium distribution [Gelman, Carlin, Stern, and Rubin 1995]. Note that in general it is very difficult to know when a MCMC simulation has reached equilibrium, see [Cowles and Carlin 1996]. During the simulations, the rejection rates were around 1%.

4 Results

MLR and MLP networks have been trained using the two Bayesian techniques. Two different approaches have been followed: In the first one the distribution of the input weights is controlled by one hyperparameter, regardless the input unit those weights are connected with. This leads to the specification 2HYP and 4HYP in Table 2. The second approach implements ARD and is achieved by associating one hyperparameter to the group of weights which connects one input unit to all of the units in the next layer. From the values of the hyperparameters which control the groups of weights connecting the input units to the next layer, it is possible to determine the inputs which are more relevant than others. A set of weights associated with a very small α will likely have a large norm, since the variance of their distribution will be large; the weights controlled by such a Gaussian are quite spread out around 0 and therefore the input unit connected to those weights is relevant for the classification of the pattern. Conversely weights associated with a large hyperparameter value will likely have a small norm, since their distribution will be peaked around 0. Inputs that have small weights associated with them are thus determined to be irrelevant.

The simulations were done with a MLP network with 30 hidden units. This is a relatively large number of hidden units, and was chosen with a view to observing a difference between the EF and MCMC runs. However, the results in Table 2 show that although the best results were obtained with a MLP trained with the MCMC method with ARD, the differences between the different methods are not statistically significant when the full training set is used.

To further investigate any differences between the EF and MCMC methods we conducted experiments using smaller training sets. We successively subdivided the original training set to obtain two data sets of half size (2916 examples), four of size 1458 and eight of size 729. Below this ten data sets each for the sizes 365, 182, 91 and 46 were used. For both the MCMC and EF methods all networks were initialized with the same randomly chosen weights. The test set performance of both methods is shown for each training set in Figure 2. As expected an improvement in performance is observed as the size of the training set is increased. We have computed the differences between the EF and MCMC test performances for each training set. Note that this is a paired comparison, i.e. the differences bewteen the EF and MCMC methods are computed on *the same* training set. Using a 2-sided t-test we find that the differences are statistically significant at the $p \geq 0.95$ level for training set sizes 1458, 729, 182, 91 and 46. As the number of examples in the smaller training sets is much smaller than the number of weights in the network it is unlikely that the Gaussian approximation used in the EF is a good one, and hence the effect observed that the EF performs less well for small data sets is not unexpected. We also note another disadvantage of the EF for small data sets, namely that a large part of the computational effort is taken up with the inversion of the Hessian; this means that the MCMC method we have used provides its results

[2]The actual prior specification is `net-spec mlp30-log 35 30 11 / - 0.05:0.5:0.5 0.05:0.5 - x0.05:0.5 - 0.05:0.5`.

Specifications	Region based accuracy %	Area based accuracy %
MLR, ARD, EF	65.2 ± 2.4	88.2 ± 14.8
MLR, ARD, MCMC	67.6 ± 2.4	91.4 ± 12.9
MLR, 2HYP, EF	65.9 ± 2.1	88.6 ± 12.5
MLR, 2HYP, MCMC	66.9 ± 2.4	88.4 ± 14.7
MLP, ARD, EF	67.2 ± 2.4	90.2 ± 13.6
MLP, ARD, MCMC	69.0 ± 2.3	90.3 ± 13.6
MLP, 4HYP, EF	67.2 ± 2.4	88.5 ± 14.7
MLP, 4HYP, MCMC	68.9 ± 2.3	89.1 ± 14.3

Table 2: The Table shows the overall accuracies of classification achieved by the networks on the test set. The region based accuracy is computed by the ratio of the number of regions correctly classified and the total number of regions of the test set. Similarly the area based accuracy is defined as the ratio between the overall size of regions correctly classified and the total amount of area of the test set. The uncertainty associated with each accuracy is the 95% confidence interval and it is computed from the overall standard error.

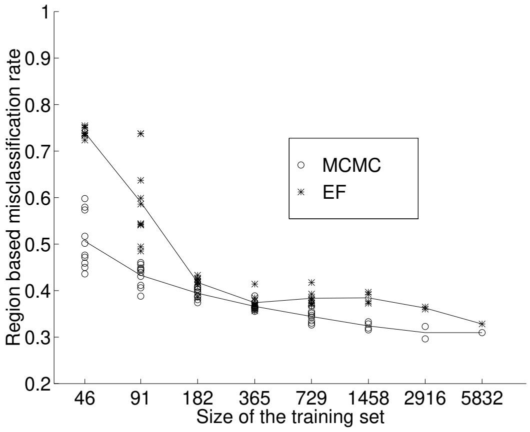

Figure 2: Plot of the learning curves for the EF and MCMC methods obtained by reducing the number of data points in the training sets.

using less CPU time for small data sets. Further simulations are required to check if similar results are obtained from other starting points in weight-space.

The values of the hyperparameters determined by ARD for the MLP network using the full training set with both MCMC and EF approaches are shown in Figure 3. The ARD parameter values for the MLR models trained with the EF and MCMC methods were broadly similar to the MLP results.

The two training methods give quite similar results: almost all of the hyperparameters determined by MCMC and EF overlap within the 95% confidence intervals. The plots show that the features describing the colour, size and the y position of the regions in training the classifier (inputs 1, 2, 3, 7 and 9) are the most relevant; the hyperparameters corresponding to those features have a small values for both the MCMC and the EF methods. The lesser relevance of the topological features and the x position of the regions is also recognised by both of the methods.

There is some disagreement between the EF and MCMC methods over the relevance of the shape descriptors (inputs $10 - 16$), although we note that the MCMC method in particular yields large error bars. Of the contextual features (inputs $20 - 35$), features $24 - 27$ describing the intensity ratio between one region and the four surrounding it appear the most useful.

All of the experiments have shown that the coordinates (x, y) of the centroids have a different

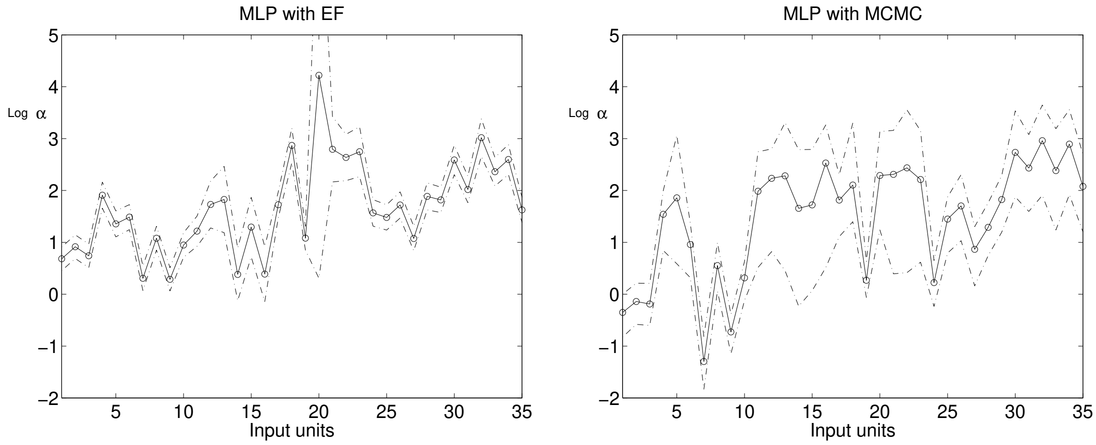

Figure 3: Plots of the hyperparameters determined by EF (left) and MCMC (right) for the MLP with ARD. The labels of the input units are displayed on the x axis. The features are the mean intensity (input 1), hue angle (2, 3), topological descriptors $(4-6)$, size of the region (7), coordinates of the centroid (8,9), shape $(10-16)$ and texture $(17-19)$ descriptors, and contextual features describing the intensity ratios $(20-23)$, size ratios $(24-27)$, and the relative positions $(28-35)$ of the four largest surrounding regions. The logarithm (base 10) of α_i, $i = 1, \ldots 35$ are reported on the y-axis. The dotted lines show the error bars.

weight in training the networks. Because images are taken with the y axis closely aligned to the vertical, the classification of the regions is unlikely to depend upon the x coordinate of the centroid: for example, regions representing cars appear in the data base in many positions along the x axis, whereas their y coordinates are ranged in a well defined interval. A similar consideration applies for the contextual features $28-35$, where a different relevance is accorded to the x and the y offsets of the relative position.

We note that the determination of the irrelevance of some of the features does not mean that those attributes are *absolutely* irrelevant. It may simply mean that the features chosen have not properly encoded the information about a region.

5 Discussion

In this paper we have compared the Evidence Framework and a MCMC method for training neural networks on the task of classifying segmented outdoor images. This is the first paper we are aware of that has carried out such a comparison. Our results suggest that on this task using large amounts of traning data the EF and MCMC performance is similar, but that the MCMC method seems superior on smaller-sized training sets. We have also used the ARD method to evaluate the relevance of different input variables.

6 Acknowledgments

The authors thank British Aerospace Sowerby Research Centre for making available the Sowerby Image Database, Dr. Andy Wright for his assistance with this research and Drs. Neill Campbell and William Mackeown for some helpful information about the data base.

This research forms part of the "Validation and Verification of Neural Network Systems" project funded jointly by EPSRC (GR/K 51792) and British Aerospace.

References

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

Cowles, M. K. and B. P. Carlin (1996). Markov Chain Monte Carlo Convergence Diagnostics: A Comparitive Review. *J. American Statistical Assn.* **91**(434), 883–904.

Duane, S., A. D. Kennedy, B. J. Pendleton, and D. Roweth (1987). Hybrid Monte Carlo. *Physics Letters B* **195**(2), 216–222.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (1995). *Bayesian Data Analysis.* Chapman and Hall.

MacKay, D. J. C. (1992a). Bayesian interpolation. *Neural Computation* **4**(3), 415–447.

MacKay, D. J. C. (1992b). The evidence framework applied to classification networks. *Neural Computation* **4**(5), 720–736.

Mackeown, W. P. J. (1994). *A Labelled Image Database and its Application to Outdoor Scene Analysis.* Ph. D. thesis, University of Bristol, England.

McKeown, D. M., W. A. Harvey, and J. McDermott (1985, Sept.). Rule-based interpretation of aerial imagery. *IEEE Trans. Pattern Analysis and Machine Intelligence* **7**(5).

Neal, R. M. (1996). *Bayesian Learning for Neural Netowrks.* Springer. Lecture Notes in Statistics 118.

Ohta, Y. (1985). *Knowledge based interpretation of Outdoor Natural Scenes.* London: Pittmann.

Vivarelli, F. (1996). The use of neural networks in classifying segmented images. M. Sc. Thesis, University of Aston, England.

Wright, W. A. (1989). Image labelling with a neural network. In *Proceedings* 5^{th} *Alvey Vision Conference*, Reading, UK, pp. 227–232. Sheffield University Press.

Wright, W. A., W. P. J. Mackeown, and P. Greenway (1995). The use of neural networks for region labelling and scene understanding. In J. G. Taylor (Ed.), *Neural Networks*, pp. 165–192. Alfred Waller.