Bayesian MLP Neural Networks for Image Analysis

Aki Vehtari [a] and Jouko Lampinen [a]

[a] *Laboratory of Computational Engineering, Helsinki University of Technology, P.O.Box 9400, FIN-02015, HUT, Finland*

Abstract

We demonstrate the advantages of using Bayesian multi layer perceptron (MLP) neural networks for image analysis. The Bayesian approach provides consistent way to do inference by combining the evidence from the data to prior knowledge from the problem. A practical problem with MLPs is to select the correct complexity for the model, i.e., the right number of hidden units or correct regularization parameters. The Bayesian approach offers efficient tools for avoiding overfitting even with very complex models, and facilitates estimation of the confidence intervals of the results. In this contribution we review the Bayesian methods for MLPs and present comparison results from two case studies. In the first case, MLPs were used to solve the inverse problem in electrical impedance tomography. The Bayesian MLP provided consistently better results than other methods. In the second case, the goal was to locate trunks of trees in forest scenes. With Bayesian MLP it was possible to use large number of potentially useful features and prior for determining the relevance of the features automatically.

Key words: Multi Layer Perceptron, Bayesian Statistics, Markov Chain Monte Carlo, Tomography , Classification

1. Introduction

A universal task in many areas of image analysis is to infer some needed piece of information from measurements that only partly determine the information. For example in classification and segmentation of image regions the set of precomputed features are often insufficient for uniquely separating the classes.

Recently Bayesian approaches have shown considerable potential in such problems. In the Bayesian approach prior information from the problem is combined to the evidence from the data, giving the posterior probability of the solutions. Predictions are made by integrating over this posterior distribution. In case of insufficient data the prior dominates the solution, and the effect of the prior diminishes with increased evidence from the data. In one of the pioneering works by Geman et.al. [5], Bayesian approach was developed for image restoration. This work also introduced the Gibbs sampling technique for computing posterior distributions.

In classification and non-linear function approximation, multi layer perceptron (MLP) neural networks have become very popular in recent years. With MLPs the main difficulty is in controlling the complexity of the model. Another problem of standard MLP models is the lack of tools for analyzing the results (confidence intervals, like 10 % and 90 % quantiles, etc.). Bayesian methods have become a viable alternative to the older error minimization based (ML or MAP) approaches [1 ,8 ,10]. The main advantages of Bayesian MLPs are:

- Automatic complexity control: Values of regularization coefficients can be selected using only the training data, without the need to use separate training and validation data.
- Possibility to use prior information and hierarchical models for the hyperparameters.
- Predictive distributions for outputs.

In this contribution we demonstrate the advantages of Bayesian MLPs in two case problems. In section 3 we give a review of the Bayesian methods for MLPs. In section 4 we report results on using Bayesian MLPs for image reconstruction in electrical impedance tomography. In section 5 we present results comparing Bayesian MLPs and other classification methods for classification of objects in forest scenes.

2. Multi Layer Perceptron

We concentrate here to one hidden layer MLPs with hyperbolic tangent (tanh) activation function. However Bayesian methods can be used for other types of neural networks, like RBF networks, too. Basic MLP model with k outputs is

$$f_k(\mathbf{x}, \mathbf{w}) = w_{k0} + \sum_{j=1}^{m} w_{kj} \tanh\left(w_{j0} + \sum_{i=1}^{d} w_{ji} x_i\right), \quad (1)$$

where $\mathbf{x}$ is a d-dimensional input vector, $\mathbf{w}$ denotes the weights, and indices i and j correspond to hidden and output units, respectively.

MLP is often considered as a generic semiparametric model, which means that the effective number of parameters may be less than the number of available parameters. Effective number of parameters determines the complexity of the model. For small weights the network mapping is almost linear and has low effective complexity, since the central region of sigmoidal activation function can be approximated by linear transformation. Traditionally the complexity of MLP has been controlled with early stopping or weight decay [1].

In early stopping weights are initialized to very small values. Part of the training data is used to train the MLP and the other part is used to monitor the validation error. Iterative optimization algorithms used for minimizing the training error gradually take parameters in use. Training is stopped when the validation error begins to increase. Since training is stopped before a minimum of the training error, the effective number of parameters remains less than the number of available parameters.

The basic early stopping is rather inefficient, as it is very sensitive to the initial conditions of the weights and only part of the available data is used to train the model. These limitations can easily be alleviated by using a committee of early stopping MLPs, with different partitioning of the data to training and stopping sets for each MLP. When used with caution MLP early stopping committee is good baseline method for neural networks.

In weight decay penalizing term is added to the error function. Using, e.g., sum of squares of weights the weights are encouraged to be small. In practice each layer in an MLP should have different regularization parameter [1], giving the penalty term

$$\alpha_1 \sum_{j,i} w_{ji}^2 + \alpha_2 \sum_{j,k} w_{kj}^2. \quad (2)$$

Problem is how to select good values for α_i. Traditionally this has been done with cross validation (CV). Since CV gives noisy estimate for error, it does not guarantee that good values for α_i can be found. Also it easily becomes computationally prohibitive as computational expenses grow exponentially with number of parameters to be selected.

3. Bayesian Learning for MLP

Bayesian methods use probability to quantify uncertainty in inferences and the result of Bayesian learning is a probability distribution expressing our beliefs regarding how likely the different predictions are. Bayesian paradigm offers consistent way to do inference using models with even very large number of parameters. See, e.g., [4] for good introduction to Bayesian methods.

3.1. Bayesian Learning

Consider a regression or classification problem involving the prediction of a noisy vector $\mathbf{y}$ of target variables given the value of a vector $\mathbf{x}$ of input variables.

The process of Bayesian learning is started by defining a model, $\mathcal{M}$, and prior distribution $p(\theta)$ for the model parameters θ. Prior distribution expresses our initial beliefs about parameter values, before any data has been observed. After observing new data $D = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}$, prior distribution is updated to the posterior distribution using Bayes' rule

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \propto L(\theta|D)p(\theta), \tag{3}$$

where the likelihood function $L(\theta|D)$ gives the probability of the observed data as function of the unknown model parameters.

To predict the new output $\mathbf{y}^{(n+1)}$ for the new input $\mathbf{x}^{(n+1)}$, predictive distribution is obtained by integrating the predictions of the model with respect to the posterior distribution of the model parameters

$$p(\mathbf{y}^{(n+1)}|\mathbf{x}^{(n+1)}, D) = \int p(\mathbf{y}^{(n+1)}|\mathbf{x}^{(n+1)}, \theta)p(\theta|D)\mathrm{d}\theta. \tag{4}$$

This is the same as taking the average prediction of all the models weighted by their goodness.

3.2. Models

A statistical model is defined by with its likelihood function. If we assume that the n data points $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ are exchangeable we get

$$L(\theta|D) = \prod_{i=1}^{n} p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \theta). \tag{5}$$

The term $p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \theta)$ in Eq. (5) depends on our problem. In regression problems, it is generally assumed that the distribution of the target data can be described by a deterministic function of inputs, corrupted by additive Gaussian noise of a constant variance. Probability density for a target y_j is then

$$p(y_j|\mathbf{x}, \mathbf{w}, \sigma) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp(-\frac{(y_j - f_j(\mathbf{x}, \mathbf{w}))^2}{2\sigma_j^2}), \tag{6}$$

where σ_j^2 is the noise variance for the target. See [10] for t-distribution and per-case normal noise variance models. For a two class classification (logistic regression) model, the probability that a binary-valued target, y_j, has the value 1 is

$$p(y_j = 1|\mathbf{x}, \mathbf{w}) = [1 + \exp(-f_j(\mathbf{x}, \mathbf{w}))]^{-1} \tag{7}$$

and for many class classification (softmax) model, the probability that a class target, y, has value j is

$$p(y = j|\mathbf{x}, \mathbf{w}) = \frac{\exp(f_j(\mathbf{x}, \mathbf{w}))}{\sum_k \exp(f_k(\mathbf{x}, \mathbf{w}))}. \tag{8}$$

In Eqs. (6), (7) and (8) function $f(\mathbf{x}, \mathbf{w})$ is in this case an MLP. Traditionally in many methods one of the problems has been to find a good topology for the MLP. In Bayesian approach we could use infinite number of hidden units. We do not need to restrict the size of the MLP based on the size of the training set, but in practice, we will have to use finite number of hidden units due to computational limits [10].

3.3. Priors

Next, we have to define the prior information about our model parameters, before any data has been seen. Usual prior is that the model has some unknown complexity but the model is not constant nor extremely flexible. To express this prior belief we set hierarchical model specification.

Parameters $\mathbf{w}$ define the model $f(\mathbf{x}, \mathbf{w})$. As discussed in section 2, complexity of the MLP can be controlled by controlling the size of the weights $\mathbf{w}$. This can be achieved by using, e.g., Gaussian prior distribution for weights $\mathbf{w}$ given hyperparameter α

$$p(\mathbf{w}|\alpha) = (2\pi)^{-m/2}\alpha^{m/2} \exp(-\alpha \sum_{i=1}^{m} w_i^2/2). \tag{9}$$

This prior states that smaller weights are more probable, but how much more is determined by the value of the hyperparameter α. Since we do not know the correct value for the hyperparameter α, we set a vague hyperprior $p(\alpha)$ expressing our belief that complexity controlled by α is unknown but the model is not constant nor extremely flexible. A convenient form for this hyperprior is vague Gamma distribution with mean μ and shape parameter a

$$p(\alpha) \sim \mathrm{Gamma}(\mu, a) \propto \alpha^{a/2-1} \exp(-\alpha a/2\mu). (10)$$

In order to have prior for weights which is invariant under the linear transformations of data, separate priors (each having its own hyperparameters α_i) for different weight groups in each layer of a MLP are used [10].

In MLPs, the weights from less important inputs are typically smaller than weights from more important inputs [1] . Prior belief that some inputs are likely to be more relevant than others can be implemented by using different priors for weight groups from each input, and hierarchical hyperpriors for these priors. The posteriors for hyperparameters should then adjust according to relevance of the inputs. This prior is called Automatic Relevance Determination (ARD) [9 ,10 ,11].

For regression models we need prior for noise variance σ in Eq. (6), which is often specified in terms of corresponding precision, $\tau = \sigma^{-2}$. As for α, our prior information is usually quite vague, stating that noise variance σ is not zero nor extremely large. This prior can be expressed with vague Gamma-distribution with mean μ and shape parameter a

$$p(\tau) \sim \mathrm{Gamma}(\mu, a) \propto \tau^{a/2-1} \exp(-\tau a/2\mu). (11)$$

3.4. Prediction

After defining the model and prior information, we combine the evidence from the data to get the posterior distribution for the parameters

$$p(\mathbf{w}, \alpha, \tau | D) \propto L(\mathbf{w}, \alpha, \tau | D) p(\mathbf{w}, \alpha, \tau). \qquad (12)$$

Predictive distribution for new data is then obtained by integrating over this posterior distribution

$$p(\mathbf{y}^{(n+1)} | \mathbf{x}^{(n+1)}, D) = \int p(\mathbf{y}^{(n+1)} | \mathbf{x}^{(n+1)}, \mathbf{w}, \alpha, \tau) p(\mathbf{w}, \alpha, \tau | D) \mathrm{d}\mathbf{w}\alpha\tau. (13)$$

We can also evaluate expectations of various functions with respect to the posterior distribution for parameters. For example in regression we may evaluate the expectation for a component of $\mathbf{y}^{(n+1)}$

$$\hat{\mathbf{y}}_k^{(n+1)} = \int f_k(\mathbf{x}^{(n+1)}, \mathbf{w}) p(\mathbf{w}, \alpha, \tau | D) \mathrm{d}\mathbf{w}\alpha\tau, \ (14)$$

which corresponds to the best guess with squared error loss.

The posterior distribution for the parameters $p(\mathbf{w}, \alpha, \tau | D)$ is typically very complex, with many modes. Evaluating the integral of Eq. (14) is therefore a difficult task. The integral can be approximated with parametric approximation as in [8] or with numerical approximation as described in next section.

3.5. Markov chain Monte Carlo method

Neal has introduced implementation of Bayesian learning for MLPs in which the difficult integration of Eq. (14) is performed using Markov chain Monte Carlo (MCMC) methods [10]. In [6] there is a good introduction to basic MCMC methods and many applications in statistical data analysis.

The integral of Eq. (14) is the expectation of function $f_k(\mathbf{x}^{(n+1)}, \mathbf{w})$ with respect to the posterior distribution of the parameters. This and other expectations can be approximated by Monte Carlo method, using a sample of values $\mathbf{w}^{(t)}$ drawn from the posterior distribution of parameters

$$\hat{\mathbf{y}}_k^{(n+1)} \approx \frac{1}{N} \sum_{t=1}^{N} f_k(\mathbf{x}^{(n+1)}, \mathbf{w}^{(t)}). \qquad (15)$$

Note that samples from the posterior distribution are drawn during the “learning phase” and predictions for new data can be calculated quickly using the same samples and Eq. (15).

In the MCMC, samples are generated using a Markov chain that has the desired posterior distribution as its stationary distribution. Difficult part is to create Markov chain which converges rapidly

[1] Note that in the non-linear network the effect of an input may be small even if the weights from it are large and vice verse, but in general the size of the weights roughly reflects the relevance of the input.

and in which states visited after convergence are not highly dependent.

Neal has used the hybrid Monte Carlo (HMC) algorithm [3] for parameters and Gibbs sampling [5] for hyperparameters. HMC is an elaborate Monte Carlo method, which makes efficient use of gradient information to reduce random walk behavior. The gradient indicates in which direction one should go to find states with high probability. Use of Gibbs sampling for hyperparameters helps to minimize the amount of tuning that is needed to obtain good performance in HMC.

When the amount of data increases, the evidence from the data causes the probability mass to concentrate to the smaller area and we need less samples from the posterior distribution. Also less samples are needed to evaluate the mean of the predictive distribution than the tail-quantiles like, 10% and 90% quantiles. So depending on the problem 10–200 samples may be enough for practical purposes (given that samples are not too highly dependent).

In our examples [2] (sections 4, 5) we have used Flexible Bayesian Modeling (FBM) software [3], which implements the methods described in [10].

4. Case I: Inverse Problem in Electrical Impedance Tomography

In this section we report results on using Bayesian MLPs for solving the ill-posed inverse problem in electrical impedance tomography (EIT). The full report of the proposed approach is presented in [7].

The aim in EIT is to recover the internal structure of an object from surface measurements. Number of electrodes are attached to the surface of the object and current patterns are injected from through the electrodes and the resulting potentials are measured. The inverse problem in EIT, estimating the conductivity distribution from the surface potentials, is known to be severely ill-posed [12].

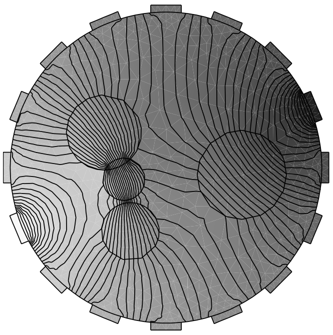

Figure 1. Example of the EIT measurement. The simulated bubble formation is bounded by the circles. The current is injected from the electrode with the lightest color and the opposite electrode is grounded. The gray level and the contour curves show the resulting potential field.

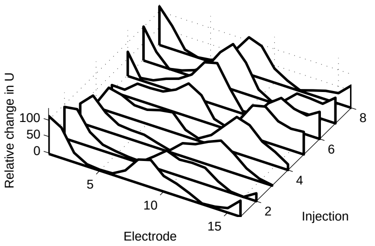

Figure 2. Relative changes in potentials compared to homogeneous background. The eight curves correspond to injections from eight different electrodes.

Fig. 1 shows a simulated example of the EIT problem. The circles in the image represent gas bubbles floating in liquid. The conductance of the gas is much lower than that of the liquid, producing the equipotential curves shown in the figure. Fig. 2 shows the resulting potential signals, from which the image is to be recovered.

In [7] we proposed a novel feedforward solution for the reconstruction problem. The approach is based on computing the principal component decomposition for the potential signals and the eigenimages of the bubble distribution from the autocorrelation model of the bubbles. The input to the MLP is the projection of the potential signals to the first principal components, and the MLP gives the coefficients for reconstructing the image as weighted sum of the eigenimages.

[2] The data and the specific modelling parameters are available from the first author upon request.

[3] `<URL:http://www.cs.toronto.edu/~radford/fbm.software.html>`

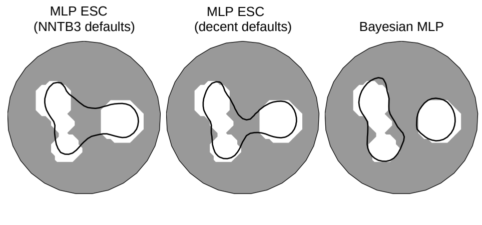

Figure 4. Example of the image reconstruction with Bayesian MLP and early stopping committees. See text for explanation of the models.

The reconstruction was based on 20 principal components of the 128 dimensional potential signal and 30 eigenimages with resolution 41×41 pixels. The training data consisted of 500 simulated bubble formations with one to ten overlapping circular bubbles in each image. To compute the reconstructions MLPs containing 30 hidden units were used. MLP models tested were

MLP ESC (NNTB3 defaults) : Early stopping committee of 20 MLPs, with different division of data to training and stopping sets for each member. The weights were initialized with the Matlab Neural Network Toolbox 3.0 default procedure (Nguyen-Widrow algorithm).

MLP ESC (decent defaults) : Similar committee to the previous, but the weights were initialized to near zero weights to guarantee that the mapping is smooth in the beginning.

MLP ESC (mlp-bgd-1) : Early stopping committee used in [11] for benchmarks.

Bayesian MLP : Bayesian MLP with FBM-software, using vague priors, noise model with t_4-distribution, and MCMC-run specifications similar as used in [11]. 20 networks from the posterior distribution of network parameters were used.

Fig. 3 shows examples of the bubble images reconstructed with Bayesian MLP. Fig. 4 shows the goodness of the image reconstructions with different MLP models for one example image. Table 1 shows the quality of the image reconstructions with different MLP models, measured by error in the void fraction and percentage of erroneous pixels in the segmentation.

An important goal in the studied process tomography application was to estimate the void fraction, which is the proportion of gas and liquid in the image. With the proposed approach such goal variables can be estimated directly without explicit reconstruction of the image. Table 2 shows the relative absolute errors in estimating the void fraction directly from the projections of the potential signals.

Table 1
Errors in reconstructing the bubble shape and estimating the void fraction from the reconstructed images. See text for explanation of the different models.

Method	Classification error %	Error in void fraction %
MLP ESC (NNTB3 def)	4.7	16.2
MLP ESC (decent def)	4.5	15.7
Bayesian MLP	3.8	6.0

Table 2
Relative errors in estimating the void fraction directly. See text for explanation of the different models. Error mean and 90% interval estimated from 4 runs with different random seeds.

Method	Relative test error %
MLP ESC (NNTB3 defaults)	8.6±1.2
MLP ESC (mlp-bgd-1)	6.42±0.04
MLP ESC (decent defaults)	4.10±0.03
Bayesian MLP	3.16±0.02

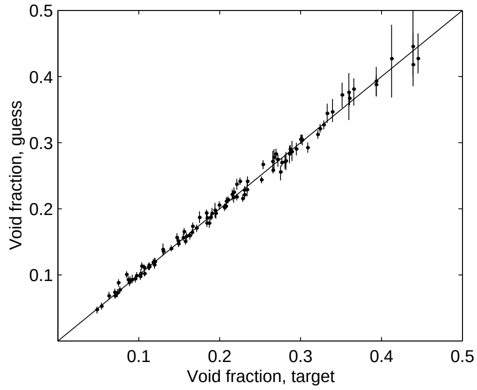

Figure 5. Scatterplot of the void fraction estimate with 10% and 90% quantiles.

Fig. 5 shows the scatter plot of the void fraction versus the estimate by the Bayesian MLP. The 10% and 90% quantiles are computed directly from the posterior distribution of the model output.

See [7] for results for effect of additive Gaussian noise to the performance of the method.

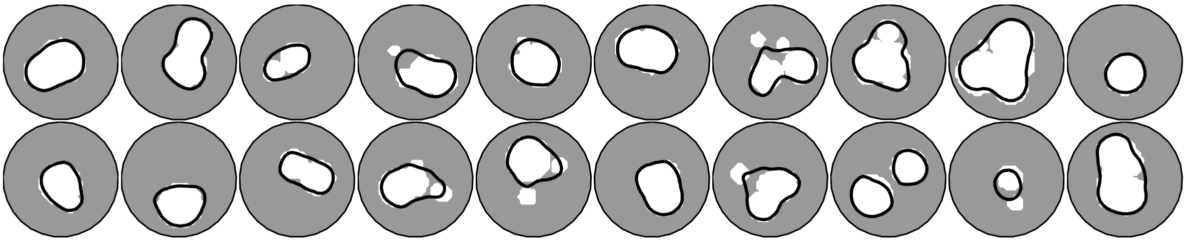

Figure 3. Examples of bubble formations reconstructed with Bayesian MLP. The white blobs show the actual simulated bubbles and the black lines show the contours of the reconstructed bubbles.

5. Case II: Forest Scene Analysis

In this section we report results of using Bayesian MLP for classification of forest scenes, to accurately recognize and locate the trees from any background. Potential applications include forest inventory (estimation of the volume and growth rate of the trees) and autonomous forest harvester (navigation and tree manipulation tasks).

Forest scene classification task is demanding due to the texture richness of the trees, occlusions of the forest scene objects and diverse lighting conditions under operation. This makes it difficult to determine which are optimal image features for the classification. A natural way to proceed is to extract many different types of potentially suitable features.

In [13] we extracted total of 84 statistical and Gabor features over different sized windows at each spectral channel. Due to great number of features used, many classifier methods would suffer from the curse of dimensionality, but Bayesian MLP manage well in high dimensional problems.

Total of 48 images were collected by using an ordinary digital camera in varying weather conditions. The labeling of the image data was done by hand via identifying many types of tree and background image blocks with different textures and lighting conditions. In this study only pines were considered.

To estimate classification errors of different methods we used eight folded cross-validation (CV) error estimate, i.e., 42 of 48 pictures were used for training and the six left out for error evaluation, and this scheme was repeated eight times. In addition to 20 hidden unit MLP models *MLP ESC* and *Bayesian MLP* (see section 4), the models tested were:

Table 3
CV error estimates for forest scene classification. See text for explanation of the different models.

	Error %, all 84 features	Error %, 16–20 pca features
KNN LOOCV	20	16
CART	30	23
MLP ESC	13	15
Bayesian MLP	12	13
Bayesian MLP +ARD	11	13

KNN LOOCV : K-nearest-neighbor, where K is chosen by leave-one-out cross-validation.

CART : Classification And Regression Tree [2].

Bayesian MLP +ARD : Same as *Bayesian MLP* plus using Automatic Relevance Determination prior.

We also tested Principal Component Analysis (PCA) for dimension reduction. With PCA we selected first components describing 99% of variance in training data, which were the first 16 to 20 principal components depending on training set.

CV error estimates are collected in Table 3. Fig. 6 shows example image classified with different methods.

6. Summary discussion

Above case problems in image analysis illustrate the advantages of using Bayesian MLPs. The approach contains automatic complexity control as the Bayesian inference techniques allow the values of regularization coefficients to be selected using only the training data, without the need to

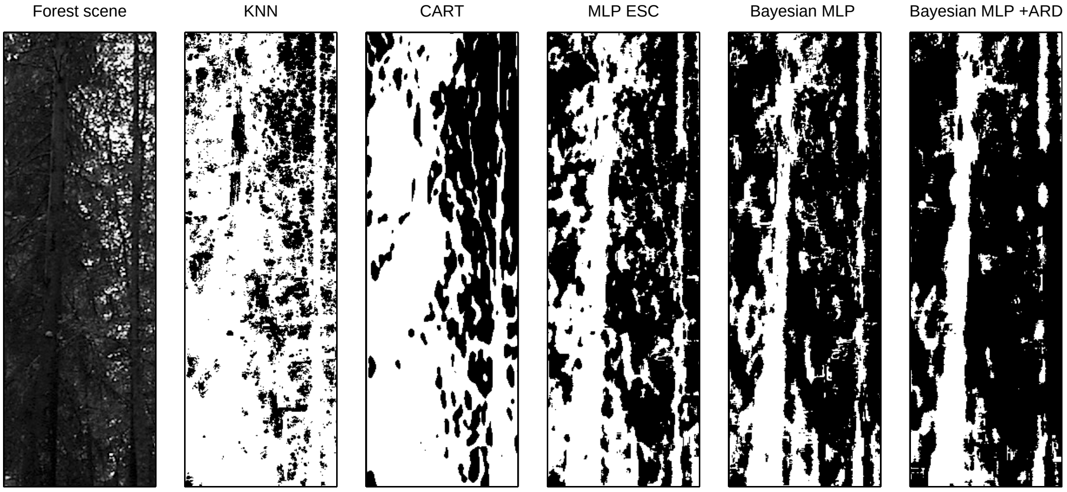

Figure 6. Examples of classified forest scene. See text for explanation of the different models.

use separate training and validation data. We can use large number of inputs and there is no need to search for minimal set of sufficient inputs. It is possible to use prior information, like ARD. The Bayesian approach gives the predictive distributions for outputs, which can be used to estimate reliability of the predictions.

Acknowledgments

This study was partly funded by TEKES Grant 40888/97 (Project *PROMISE, Applications of Probabilistic Modeling and Search*).

References

[1] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[2] Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. *Classification and regression trees.* Chapman & Hall, 1984.

[3] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195:216–222, 1987.

[4] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald R. Rubin. *Bayesian Data Analysis.* Chapman & Hall, 1995.

[5] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[6] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice.* Chapman & Hall, 1996.

[7] Jouko Lampinen, Aki Vehtari, and Kimmo Leinonen. Using Bayesian neural network to solve the inverse problem in electrical impedance tomography. In B. K. Ersboll and P. Johansen, editors, *Proceedings of 11th Scandinavian Conference on Image Analysis SCIA'99*, pages 87–93, Kangerlussuaq, Greenland, 1999.

[8] David J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[9] David J. C. MacKay. Bayesian non-linear modelling for the prediction competition. In *ASHRAE Transactions, V.100, Pt.2*, pages 1053–1062, Atlanta Georgia, 1994. ASHRAE.

[10] Radford M. Neal. *Bayesian Learning for Neural Networks.* Springer-Verlag, 1996.

[11] Radford M. Neal. Assessing relevance determination methods using DELVE. In Christopher M. Bishop, editor, *Neural Networks and Machine Learning*, pages 97–129. Springer-Verlag, 1998.

[12] M. Vauhkonen, J. P. Kaipio, E. Somersalo, and P. A. Karjalainen. Electrical impedance tomography with basis constraints. *Inverse Problems*, 13(2):523–530, 1997.

[13] Aki Vehtari, Jukka Heikkonen, Jouko Lampinen, and Jouni Juujärvi. Using Bayesian neural networks to classify forest scenes. In David P. Casasent, editor, *Proceedings of SPIE 3522*, pages 66–73. SPIE, 1998.