Transportation Research Part C 17 (2009) 498–509

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Bayesian committee of neural networks to predict travel times with confidence intervals

C.P.IJ. van Hinsbergen [a,b,*], J.W.C. van Lint [a,1], H.J. van Zuylen [a,b,1]

[a] Department of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Stevinweg 1, P.O. Box 5048, 2600 GA, Delft, The Netherlands
[b] TRAIL Research School, The Netherlands

ARTICLE INFO

Article history:
Received 29 April 2008
Received in revised form 26 January 2009
Accepted 14 April 2009

Keywords:
Neural network
Bayesian inference
Committee
Travel time prediction
Confidence interval
Evidence framework

ABSTRACT

Short-term prediction of travel time is one of the central topics in current transportation research and practice. Among the more successful travel time prediction approaches are neural networks and combined prediction models (a 'committee'). However, both approaches have disadvantages. Usually many candidate neural networks are trained and the best performing one is selected. However, it is difficult and arbitrary to select the optimal network. In committee approaches a principled and mathematically sound framework to combine travel time predictions is lacking. This paper overcomes the drawbacks of both approaches by combining neural networks in a committee using Bayesian inference theory. An 'evidence' factor can be calculated for each model, which can be used as a stopping criterion during training, and as a tool to select and combine different neural networks. Along with higher prediction accuracy, this approach allows for accurate estimation of confidence intervals for the predictions. When comparing the committee predictions to single neural network predictions on the A12 motorway in the Netherlands it is concluded that the approach indeed leads to improved travel time prediction accuracy.

1. Introduction

The widely acknowledge potential of traffic information to alleviate congestion and to decrease negative environmental and societal side effects has led to a surge of research into reliable and accurate traffic and travel time prediction models in the past few decades (van Lint et al., 2005).

Among the most applied types of traffic prediction models are ARIMA-like time series approaches (Nihan, 1980; Lee and Fambro, 1999), Kalman filtering (Okutani and Stephanedes, 1984; Yang, 2005), linear weighted regression (Zhong et al., 2005; Nikovski et al., 2005), nearest neighbor techniques (Clark, 2003; Smith and Demetsky, 1996), neural networks (van Lint et al., 2005; Dougherty and Cobbett, 1997; Mark et al., 2004; Zhang, 2000; Innamaa, 2005; Dharia and Adeli, 2003) and so-called *committee* or *ensemble* approaches, in which multiple model-predictions are combined (Petridis et al., 2001; Kuchipudi and Chien, 2003; Zheng et al., 2006). The last two approaches, neural networks and committees, have shown a high accuracy for prediction of traffic conditions (van Hinsbergen et al., 2007). However, these two approaches exhibit some imperfections when applied in real-time applications, as will be shown in Sections 1.1 and 1.2.

* Corresponding author. Address: Department of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology, Stevinweg 1, P.O. Box 5048, 2600 GA, Delft, The Netherlands. Tel.: +31 15 2784066; fax: +31 15 2783179.
E-mail addresses: c.p.i.j.vanhinsbergen@tudelft.nl (C.P.IJ. van Hinsbergen), j.w.c.vanlint@tudelft.nl (J.W.C. van Lint), h.j.vanzuylen@tudelft.nl (H.J. van Zuylen).
[1] Tel.: +31 15 2781681; fax: +31 15 278 3179.

0968-090X/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.trc.2009.04.007

One valuable and objective piece of traffic information is the *travel time*. Real-time travel time predictions can be used in dynamic traffic management applications and in commercial applications, such as pre-trip planning or en-route navigation. This paper presents a neural network-based committee approach as an alternative for online travel time prediction.

1.1. Committees of prediction models

One way of improving prediction accuracy and reliability is to combine multiple prediction models in a *committee*, where the outcomes are a weighted combination of the outcomes of its members. It has been shown (Bishop, 1995) that a committee cannot produce an increase in the expected error, even if just the average of the predictions is taken and the weights are not optimized.

Previous attempts to combine traffic prediction models typically use the errors the models make in the previous time intervals (Petridis et al., 2001; Kuchipudi and Chien, 2003; Zheng et al., 2006). However, when applied to predicting *travel time*, one major complication occurs: it takes time (in fact the travel time) for the actual trip to be realized and consequently for a travel time to become available. Therefore, in most practical situations the actual travel time is not available within one discrete time step, especially in congested situations where accurate travel time prediction is most valuable. Using the error in the previous intervals to combine travel time prediction models must thus be considered a theoretical exercise and inapplicable to most real-time applications (Van Lint, 2008).

In (van Hinsbergen and van Lint, 2008) an alternative committee approach using Bayesian inference theory was applied to the travel time prediction problem. In this theory, a model's prediction as well as the *probability* that a model predicts the travel time correctly (the *evidence* for a model) is used. The relative probabilities of the models are then used to select the committee members. This approach does not involve evaluating the prediction error of the last prediction(s) made, which makes it appropriate for online applications. In (van Hinsbergen and van Lint, 2008) it is demonstrated that prediction accuracy can be improved using this approach.

1.2. Artificial neural networks

It is common practice in the application of (artificial) neural networks for travel time prediction to train many different candidate networks and then to select the best, based on the performance on an independent validation set, to make predictions. Although this might intuitively make sense, there are a number of serious drawbacks to this approach. In the first place, this implies that much effort involved in training networks is wasted. More seriously, the fact that one neural network model outperforms all other models on one particular validation data set does not guarantee that this neural network model indeed contains the "optimal" weights and structure, nor that this model has the best generalization capabilities. This completely depends on the statistical properties of the training and validation set (e.g. the amount of noise in the data), the complexity of the problem at hand and most importantly on the degree to which the training and validation set are representative for the true underlying process which is modeled. The network performing best on the validation set may therefore not be the one with the best performance on new data (Bishop, 1995).

These drawbacks can be overcome by combining all (or a representative selection of) trained neural network models in a committee. The Bayesian framework that is applied in (van Hinsbergen and van Lint, 2008) can be used for this purpose. The theory of Bayesian inference to train and combine a committee of feed-forward neural networks has been described in (Bishop, 1995; MacKay, 1992b, 1995) and has been applied in various fields of study (MacKay, 1994; Thodberg, 1993; Chua and Goh, 2003; Lisboa et al., 2003; Baesens et al., 2002). To the authors' knowledge this approach has not yet been applied to travel time prediction or traffic prediction in general.

1.3. Objective of this study

In this study the abovementioned Bayesian approach for neural network based travel time prediction will be used and its workings will be demonstrated on real data from the A12 motorway in The Netherlands. In this approach two *intrinsic* and informative quantities are calculated, which allow for real time model comparison and combination, First, during training, the so-called model-*evidence* is calculated, which ranks the models on the basis of the fit on the training data taking into account the degree of over-fitting (inducing variance) or under-fitting (resulting in bias). Second, in actual operation the approach also allows the analyst to estimate errors (error bars) on each prediction, which indicate the degree in which the currently presented input pattern matches with the input patterns "seen" during training. The committee approach is compared to individual neural networks to show that the committee provides a more accurate prediction of travel times and has better generalization performance.

As traffic systems are highly dynamic, it is expected that in order to make highly accurate travel time predictions, neural networks that are able incorporate these dynamics, such as feed-forward neural networks with multiple layers, recurrent neural networks or state-space neural networks (van Lint et al., 2005), are needed. However, to maintain focus on the workings and powerful properties of the Bayesian framework, relatively simple feed-forward neural networks are used in this study; the principles applied in this study can also be applied to more complex neural networks structures in future studies.

2. Methodology

In this section first the general approach to Bayesian model fitting is presented. Subsequently, the construction of a committee of neural networks and the derivation of error bars on each committee member's predictions are discussed.

2.1. Feed-forward neural networks for travel time prediction

Fig. 1 shows a typical feed-forward neural network topology with an input layer, a hidden layer and an output layer. The input layer consists of d input elements, the hidden layer of M hidden nodes and the output layer of c outputs.

2.1.1. Mathematical description of a neural network

An output y_k, $k = (1, \ldots, c)$ can be described by the following equations:

$$y_k(\mathbf{x}) = f_2\left(\sum_{j=1}^{M+1} w_{kj} z_j\right)$$
$$z_j = f_1\left(\sum_{i=1}^{d+1} w_{ji} x_i\right) \tag{1}$$

where w_{ji} and w_{kj} are called *weights* which are adjustable and whose values need to be estimated from data. The *bias weights* (*biases*) are represented by an extra node in a layer to the left (the grey nodes in Fig. 1) which have a constant output of 1, so $x_{d+1} = 1$ and $z_{M+1} = 1$. The functions f_1 and f_2 are called *activation functions* and apply transformations to the weighted sum of the output of the units to the left. Common forms of the activation of the hidden nodes are the *logistic sigmoid* and the *hyperbolic tangent* functions. In practice, the latter is found to give rise to faster convergence (Bishop, 1995). A linear activation function is commonly used for the output units.

$$f_1(a) = \tanh(a)$$
$$f_2(a) = a \tag{2}$$

The weights w and biases θ together form a weight vector $\mathbf{w}$ with a total of W weights (parameters). The input vector $\mathbf{x}^n \equiv (x_1^n, \ldots, x_d^n)$ is drawn from a data set $X \equiv (\mathbf{x}^1, \ldots, \mathbf{x}^N)$ of N data points. The output values of the network $\mathbf{y}(\mathbf{x}^n) \equiv (y_1(\mathbf{x}^n), \ldots, y_c(\mathbf{x}^n))$ can be compared to target values $\mathbf{t}^n \equiv (t_1^n, \ldots, t_c^n)$, drawn from a target data set $D \equiv (\mathbf{t}^1, \ldots, \mathbf{t}^N)$. Only networks with a single output, $c = 1$, are considered in this study, so the index k will be dropped.

2.1.2. Neural network training (model fitting)

The values of the weight vector $\mathbf{w}$ of the network need to be learned from data, which is usually referred to as neural network training. Typically this learning mechanism is based on a maximum likelihood approach, equivalent to the minimization of an error function such as the sum of squared error:

$$E_D = \frac{1}{2}\sum_{n=1}^{N}(y(\mathbf{x}^n; \mathbf{w}) - t^n)^2 \tag{3}$$

Preferably, a regularizer term is added to (3) to avoid overfitting of the networks to the training data. A commonly used regularizer is the *partitioned weight decay* error term which has empirically been found to improve network generalization (Krogh and Herts, 1995) and is invariant to transformations to the input or output data (Bishop, 1995). Let us briefly explain this regularizer. Define V groups of weights $\mathbf{w}_v$, e.g. one for each layer and one for the biases, and define the regularizer by:

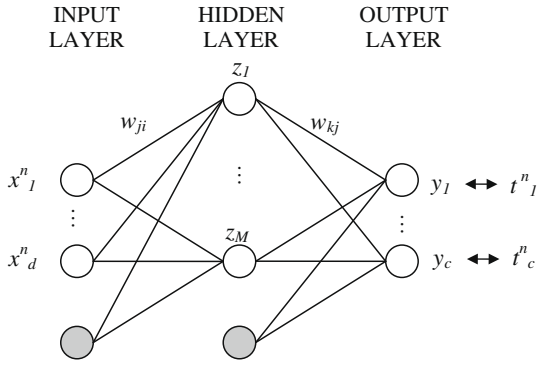

Fig. 1. A neural network with d input elements, one hidden layer with j hidden nodes and c outputs, where the biases are represented as an extra node.

$$E_W = \sum_v \alpha_v E_{W,v}$$
$$E_{W,v} = \frac{1}{2} \sum_{w \in \mathbf{w}_v} w^2 \tag{4}$$

where the parameters α_v control the extent to which the regularizer influences the solution. The regularized performance (error) function then becomes

$$E(\mathbf{w}) = E_D + E_W \tag{5}$$

The minimum of this performance function can be found by regular *back-propagation* or one of its many variations such as gradient descent (Rumelhart et al., 1986) or the (scaled) conjugate gradient algorithm (Press et al., 2007; Williams, 1991; Møller, 1993; Johansson et al., 1991). In the current study the scaled version of the latter algorithm is used.

In the conjugate gradient algorithm, a series of search directions $\mathbf{d}_j$ through weight space is constructed using the negative gradient $-\mathbf{g} = -\nabla E(\mathbf{w})$, which can be found by back propagating the errors (Hecht-Nielsen, 1989). A new search direction is set to always be *conjugate to* or *non-interfering with* all previous search directions, which ensures fast convergence to a minimum. After having found a search direction, the length of the step is determined using the Hessian $\mathbf{A} = \nabla\nabla E(\mathbf{w})$, which be exactly evaluated by a back propagation approach (Bishop, 1992). In the scaled version of the conjugate gradient algorithm, a Levenberg–Marquardt technique is added to ensure that the quadratic error approximation that is used in the approach is valid for the step under consideration.

However, instead of using maximum likelihood techniques, neural network training can be viewed from a Bayesian inference perspective (Bishop, 1995; MacKay, 1995). This has some major advantages in the application of the neural networks. First, error bars can be assigned to the predictions of a network. Second, an automatic procedure for weighing the two error parts E_D and E_W of the error function can be derived; the values of these weights can be inferred simultaneously from the training data without the need of a separate validation data set. Since all data is used for training, better models will result. Third, the *evidence* measure emerging from the Bayesian analysis can be used as an early stopping criterion in the training procedure. Finally, different networks can be selected and combined in a committee approach using this evidence measure.

2.2. Bayesian trained neural networks for travel time prediction

From a Bayesian inference perspective, the parameters in a neural network (or any model for that matter) should not be conceived as single values, but as a *distribution* of values representing various degrees of belief. The goal is then to find the posterior probability distribution for the weights after observing the dataset D, denoted by $p(\mathbf{w}|D)$.

2.2.1. Neural network training formulated as Bayesian inference

This posterior can be found using Bayes' theorem:

$$p(\mathbf{w}|D) = \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)} \tag{6}$$

where $p(D)$ is the normalization factor, $p(D|\mathbf{w})$ represents a noise model on the target data and corresponds to the likelihood function, and $p(\mathbf{w})$ is the prior probability of the weights (Bishop, 1995).

Although many forms of the prior and the likelihood function are possible, often Gaussian forms are chosen to simplify further analyses:

$$p(\mathbf{w}) = \frac{1}{Z_W(\boldsymbol{\alpha})} \exp\left(-\sum_v \alpha_v E_{W,v}\right)$$
$$p(D|\mathbf{w}) = \frac{1}{Z_D(\beta)} \exp\left(-\frac{\beta}{2} \sum_{n=1}^{N} (y(\mathbf{x}^n;\mathbf{w}) - t^n)^2\right) \tag{7}$$

where $Z_W = \int \exp(-\Sigma_v \alpha_v E_{w,v}) \mathrm{d}\mathbf{w}$ and $Z_D = \int \exp(-\beta E_D) \mathrm{d}D$ are normalizing constants and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_V)$ and β are called *hyperparameters* as they control the distributions of other parameters, the weights $\mathbf{w}$ of the network. The prior has zero mean and variances $1/\alpha_v$ for every group of weights, the likelihood function has zero mean and variance $1/\beta$. It can be seen that the exponents in Eq. (7) take the form of the error functions E_W and E_D already introduced in Eq. (3). Substituting Eqs. (7) in (6) results in an expression for the posterior:

$$p(\mathbf{w}|D) = \frac{1}{Z_S(\boldsymbol{\alpha}, \beta)} \exp(-E(\mathbf{w})) \tag{8}$$
$$E(\mathbf{w}) = \beta E_D + \sum_v \alpha_v E_{W,v} \tag{9}$$

where $Z_S(\boldsymbol{\alpha}, \beta) = \int \exp(-E(\mathbf{w})) \mathrm{d}\mathbf{w}$ is a normalizing constant. Consider now the maximum of the posterior distribution, $\mathbf{w}_{MP}$ (the most probable value of the weight vector). This can be found by minimizing the negative logarithm of (8), which is

equivalent to minimizing Eq. (9). Since this equation is similar to Eq. (5) (except for an overall multiplicative factor), the maximum of the posterior $p(\mathbf{w}|D)$ can be found by simple and well-established back-propagation techniques (see Section 2.1).

2.2.2. Approximation of the posterior distribution of the weights

Although the most probable values for the weights (the "peak" of the posterior distribution) can be found using normal back-propagation, the entire posterior distribution needs to be evaluated to generate for example error bars on the predictions or to construct a committee of networks, as will be shown later. A complication here is that the normalizing coefficient $Z_S(\boldsymbol{\alpha}, \beta)$ of Eq. (8) in most cases cannot be evaluated analytically. Therefore, the posterior needs to be approximated, for example by a Taylor expansion (MacKay, 1992b), which results in the posterior

$$p(\mathbf{w}|D) = \frac{1}{Z_S(\boldsymbol{\alpha}, \beta)} \exp\left(-E(\mathbf{w}_{MP}) - \frac{1}{2}\Delta\mathbf{w}^T\mathbf{A}\Delta\mathbf{w}\right) \tag{10}$$

where the normalizing constant $Z_S(\boldsymbol{\alpha}, \beta) = \int \exp(-E(\mathbf{w}_{MP}) - \frac{1}{2}\Delta\mathbf{w}^T\mathbf{A}\Delta\mathbf{w})d\mathbf{w}$ and where $\mathbf{A}$ is the Hessian given by

$$\mathbf{A} = \nabla\nabla E(\mathbf{w}) = \beta\nabla\nabla E_D + \sum_v \alpha_v \mathbf{I}_v \tag{11}$$

where E_D is the error function of Eq. (3) and $\mathbf{I}_v$ is a matrix with all elements zero except for the elements $I_{ii} = 1$ where i corresponds to a weight from a group v. This estimation of the posterior distribution of the weights can be used to construct error bars and to create a committee of networks.

2.2.3. Approximation of the posterior distribution of the hyperparameters

In order to evaluate Eq. (9), the values (distributions) of the hyperparameters β and $\boldsymbol{\alpha}$ in Eq. (9) need to be found. These can be approximated by the same Bayesian inference framework that is used to approximate the posterior distributions of the weights. The posterior distribution of $\boldsymbol{\alpha}$ and β given the data D is given by:

$$p(\boldsymbol{\alpha}, \beta|D) = \frac{p(D|\boldsymbol{\alpha}, \beta)p(\boldsymbol{\alpha}, \beta)}{p(D)} \tag{12}$$

It can be shown (Gull, 1989; MacKay, 1992a; Bishop, 1995) that this posterior is maximized with the following values for $\boldsymbol{\alpha}$ and β:

$$\begin{aligned} \alpha_v^{MP} &= \frac{\gamma_v}{2E_{W,v}} \\ \beta^{MP} &= \frac{N - \gamma}{2E_D} \end{aligned} \tag{13}$$

where $\gamma = \sum_v \gamma_v$ is the so-called number of well-determined parameters, the elements of which are given by:

$$\gamma_v = \sum_j \left\{ \frac{\eta_j}{\eta_j + \alpha_v} (\mathbf{V}^T\mathbf{I}_v\mathbf{V})_{jj} \right\} \tag{14}$$

where η_j is the jth eigenvalue of the Hessian $\mathbf{A}$, $\mathbf{V}$ is the matrix of eigenvectors of the Hessian $\mathbf{A}$ and $\mathbf{I}_v$ was defined when explaining Eq. (11). In this summation negative eigenvalues are omitted (Thodberg, 1993).

In practice, the optimal values for $\boldsymbol{\alpha}$ and β as well as the optimal weight vector $\mathbf{w}_{MP}$ need to be found simultaneously. A simple approach is to use a standard iterative training algorithm (i.e. the scaled conjugate gradient algorithm) to find $\mathbf{w}_{MP}$ while periodically re-estimating the values of $\boldsymbol{\alpha}$ and β using (13).

The initial values of the hyperparameters depend on the typical values of the input (e.g. speeds, flows) and outputs (e.g. travel times). The data are transformed to ensure that all of the input and target variables are of order unity, in which case it is expected that the network weights also are of order unity, and thus the hyperparameters can be initialized to one. If the variables are treated as independent, they can be transformed by

$$\tilde{x}_i^n = \frac{x_i^n - \bar{x}_i}{\sigma_i} \tag{15}$$

where $\bar{x}_i$ is the mean of the variable and σ_i is the standard deviation.

2.3. The evidence framework for committees of neural networks

In the next sections the Bayesian evidence framework for neural network training and model comparison will be discussed.

2.3.1. Calculating the evidence for a single neural network

Consider a certain neural network i with a set of assumptions H_i, such as the number of layers and the number of hidden units. The posterior probability of this model given the training data D, $P(H_i|D)$, can be determined using Bayes rule:

$$P(H_i|D) = \frac{p(D|H_i)P(H_i)}{p(D)} \tag{16}$$

where $P(H_i)$ is the prior probability of model H_i and $p(D|H_i)$ is called the *evidence* for H_i. The evidence is a measure which intuitively and consistently combines a model's ability to fit the data with its complexity (Mackay, 1992a). It naturally embodies *Occam's Razor*, which states to prefer a simpler model over a more complex one given it predicts the data sufficiently well and can be used to for example compare different models after they are trained. The evidence equals the denominator of (12) if the prior $P(H_i)$ is taken equal for all models and the conditional dependence on the model H_i are made explicit. Therefore, the evidence can be found using

$$p(D|H_i) = \int\int p(D|\boldsymbol{\alpha}, \beta, H_i) p(\boldsymbol{\alpha}, \beta|H_i) d\boldsymbol{\alpha} d\beta \tag{17}$$

If the same Gaussian approximation introduced in deriving Eq. (12) is assumed and the symmetries of neural networks with equal structures but different initial weights, corresponding to for example exchanges of weights or 'sign-flip' symmetries, are accounted for, the following logarithm of the evidence for a two-layer neural network model H_i emerges:

$$\begin{aligned} \ln p(D|H_i) = &-\sum_v \alpha_v^{MP} E_{W,v}^{MP} - \beta^{MP} E_D^{MP} - \frac{1}{2}\ln|\mathbf{A}| + \sum_v \frac{W_v}{2}\ln\alpha_v^{MP} + \frac{N}{2}\ln\beta^{MP} + \ln M! + 2\ln M \\ &+ \frac{1}{2}\sum_v \ln\left(\frac{2}{\gamma_v}\right) + \frac{1}{2}\ln\left(\frac{2}{N-\gamma}\right) \end{aligned} \tag{18}$$

where terms which are equal for all models H_i are omitted, as only the relative values of the log evidence of the different models are of interest as will be shown later. For the exact derivation of this equation, the reader is referred to (Bishop, 1995; Thodberg, 1993).

As the determinant of the Hessian **A** in Eq. (18) is a product of the eigenvalues it is sensitive to errors in small values of the eigenvalues. Therefore, eigenvalues smaller than a certain cutoff value ε should be excluded when determining $|\mathbf{A}|$ to avoid numerical problems (Bishop, 1995).

2.3.2. Using the evidence as a stopping criterion

The evidence can be used as a stopping criterion or as a guide for pruning, due to its abilities to balance between model fit and model complexity (Thodberg, 1993). In this study, the development of the evidence is monitored during training. It is found by looking at many examples that the evidence flattens around the point when there is little to be gained in the *generalization performance*.

Fig. 2 shows an example of this behavior for a case of predicting travel times where a dataset of 59 days was randomly split in two parts: 80% was assigned to a training set and the remaining 20% was used as a set to test the generalization performance. The log evidence, calculated using Eq. (18), of a network with 12 inputs and 15 hidden nodes hardly increases after epoch 100. Around the same time, the error of the network on the test set (the unseen dataset) does not decrease anymore, although the training error (the dataset with which the networks are trained) does decrease if training is continued. The training can therefore be stopped once the increase in the evidence falls below a certain threshold value ς.

2.3.3. Constructing a committee on the basis of the evidence

The evidence that was derived in Section 2.3 can also be used to select promising networks and to construct a *committee*. In a committee, the predictions of multiple models are combined. It has been shown that committees can lead to improved

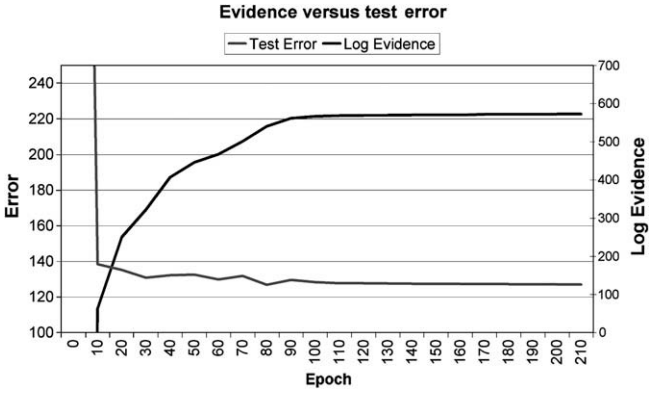

Fig. 2. Evidence versus test error during training of a network with 12 inputs and 15 hidden units; the evidence can be used as a stopping criterion as the test error and evidence start to flatten around the same point.

generalization (Bishop, 1995; Thodberg, 1993). In this study, neural networks with different structures and different weight distributions are combined.

Consider a generalized committee given by a weighted combination of predictions of its L members of the form (Perrone, 1994):

$$y_{GEN}(\mathbf{x}) = \sum_{i=1}^{L} q_i y_i(\mathbf{x}) \tag{19}$$

The best L committee members may be selected based on their evidence. Different types of weights q_i are possible, but in this study a simple average over all committee members is considered (Thodberg, 1993; MacKay, 1994):

$$y_{GEN}(\mathbf{x}) = \frac{1}{L}\sum_{i=1}^{L} y_i(\mathbf{x}) \tag{20}$$

2.4. Error bars on each committee member's predictions

If it is assumed that the output distribution arises from Gaussian noise on the output variables, that the distributions on the weights are Gaussian, and that the posterior distribution of the weights is sufficiently narrow so that it can be approximated by its linear expansion around $\mathbf{w}_{MP}$, then the output distribution of a single neural network is given by $N(y_{MP}, \sigma_t)$ where y_{MP} is the output of the network with the parameters set to $\mathbf{w}_{MP}$, and the standard deviation σ_t can be found by (Bishop, 1995):

$$\sigma_t^2 = \sigma_D^2 + \sigma_W^2 = \frac{1}{\beta} + \mathbf{k}^T\mathbf{A}^{-1}\mathbf{k} \tag{21}$$

where $\mathbf{A}$ is the Hessian and $\mathbf{k}$ is defined by:

$$\mathbf{k} \equiv \nabla_{\mathbf{w}} y|_{\mathbf{w}_{MP}} \tag{22}$$

This standard deviation (21) has two contributions: the first term reflects the spread (the uncertainty) in the target data, whereas the second term reflects the width of the posterior distribution of (and thus the uncertainty in) the network weights. The standard deviation can be used to construct error bars, for example 95% confidence intervals (twice the standard deviation) for the predictions.

A third and additional source of output variance is in the spread of the predictions between members of a committee. If the committee members' predictions are combined using the simple average of Eq. (20), it can be shown that the combined error bar for a prediction becomes (Thodberg, 1993):

$$\sigma_{total}^2 = \bar{\sigma}_D^2 + \bar{\sigma}_W^2 + \sigma_C^2 \tag{23}$$

where $\bar{\sigma}_D^2$ is the average over all $\sigma_{i,D}^2$, $\bar{\sigma}_W^2$ the average over all $\sigma_{i,W}^2$ and σ_C^2 is the committee variance (the disagreement among the networks) given by:

$$\sigma_C^2 = \frac{1}{L}\sum_{i=1}^{L}(y_{GEN}(\mathbf{x}) - y_i(\mathbf{x}))^2 \tag{24}$$

In the next section all ingredients discussed so far are summarized and presented in a step-by-step description of the Bayesian committee approach.

2.5. Step-by-step procedure: Bayesian committee of neural networks

To summarize all key concepts, below a step-by-step procedure is presented for making the committee predictions.

1. Construct many different networks with different numbers of hidden units and with different initial weight values.
2. For a model, draw initial weight values for the hyperparameters from their priors.
3. Train the networks by the scaled conjugate gradient algorithm.
4. Every step of the algorithm, re-estimate values for $\boldsymbol{\alpha}$ and β using (13) and (14).
5. Calculate the evidences for each network every few epochs. If the increase in the evidence relative to the previous epoch it was calculated falls below a certain threshold ς, stop, otherwise go to step 3 and repeat the procedures.
6. After all networks are trained, choose a selection of the better networks on the basis of their final evidences and construct a committee using (19).
7. Combine the error bars using (23) and draw 95% confidence intervals by adding and subtracting twice the standard deviation from the committee predictions.

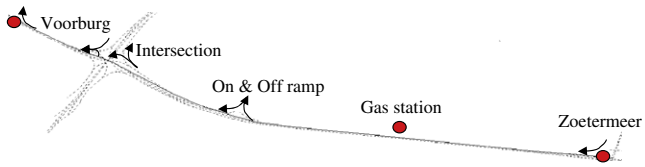

Fig. 3. The A12 motorway from Zoetermeer to The Hague.

3. Experiment

The theory of a committee of neural networks to predict travel times is applied to an 8.5 km (5.3 mi) long route of the A12 motorway in The Netherlands, from an on ramp (Zoetermeer) to an off ramp (Voorburg) (see Fig. 3). On this route, 84 neural networks with the number of hidden nodes varying from 3 to 14 and with different initial weight values were trained, after which the networks with the highest evidence were selected and combined. To investigate the effects of early stopping discussed in Section 2.3.2, the networks were also trained using a fixed number of 400 epochs, using the same structure and initial weight values as when trained with early stopping.

3.1. Data

At both the on ramp and the off ramp license plate cameras are placed that record each vehicles' license plate. Individual travel times based on matches of license plates were available for 95 days in the winter and spring of 2007. The data were filtered for outliers, which were a considerable number, mainly due to the fact that only four characters out of six are recorded due to privacy legislations. After filtering the data and inspecting them visually, the travel times of the vehicles leaving in the same 5-min time period were averaged. A total of 47 peak periods of about 3.5 h each were selected from the data set. These peak periods were randomly split over two subsets: 37 peak periods with which the networks were initially trained and 10 peak periods on which the performance of the individual networks and of the committee was validated.

As input to the neural networks, 12 double loop detectors, evenly spread over the route, are available, reporting speeds and flows every minute. The speed data are available in 1 min arithmetic mean speeds of all vehicles that are recorded (i.e. time mean speeds). Due to the inherent bias in time mean speeds when used as a proxy for space mean speeds, the speeds were corrected to space mean speeds using an estimate for the variance of the speeds in the 1 min interval described in (van Lint, 2004; van Hinsbergen and van Lint, 2008).

3.2. Parameters

Initial values for the hyperparameters (Section 2.2.3) were set to $\alpha_v = 1$ for all v and $\beta = 1$. The cutoff value when calculating the determinant of the Hessian (Section 2.3) was set to $\varepsilon = 10^{-10}$. The early stopping criterion (Section 2.3.2) was set to 1%, where the evidence was evaluated every 10 epochs.

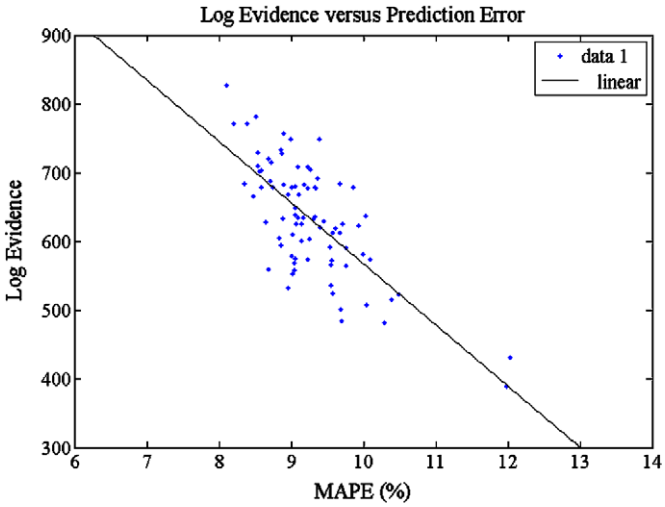

Fig. 4. The log evidence versus MAPE on the test set for 84 different neural networks shows a negative trend.

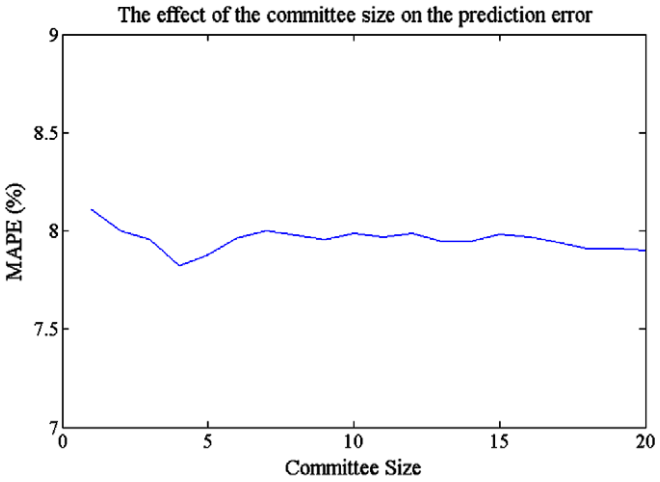

Fig. 5. The MAPE versus the committee size shows an optimal size of 4.

Table 1
The performance of the individual networks compared to the committee prediction.

Predictor	Log evidence	MAPE (%)
#1	827.3	8.11
#2	781.0	8.51
#3	771.3	8.39
#4	771.0	8.89
Committee	–	7.82

Table 2
The effect of early stopping on training and on the prediction results for 84 networks.

Stopping criterion	Total training time (min)	Mean number of epochs	Optimal committee size	Committee MAPE
<1% evidence increase	475	134	4	7.82%
400 epochs	1415	400	21	7.72%

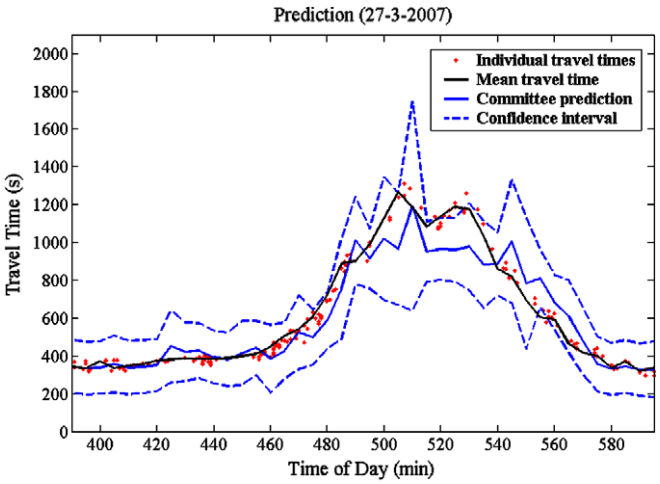

Fig. 6. Prediction of travel time with confidence intervals. In congested situations, the error bars are larger than in free flow situations.

4. Results

Fig. 4 shows the log evidence versus the test errors. A negative trend can be seen from this graph: the lower the error, the higher the log evidence; the fitted linear line has an R^2 of 0.52. As the R^2 deviates significantly from zero, the graph shows that the evidence is informative about the accuracy of the predictions on a new data set, although the correlation does show imperfections. Fig. 5 shows the effect of varying the size of the committee on the prediction error of the combined models. It shows that the optimal size is 4 for this case, and that after that point there is no gain in increasing the size of the committee.

Table 1 shows the Mean Absolute Percentage Error (MAPE) of the committee of four networks compared to the four individual networks' predictions on the test set. It can be seen that the committee leads to a small gain in accuracy: a relative decrease of almost 4% in the error, compared to selecting the single network with the highest evidence, is achieved by retaining multiple networks and combining their predictions.

Table 2 presents the effects of the early stopping criterion discussed in Section 2.3.2 on the training time for all 84 neural networks and the mean committee prediction error. The number of epochs when stopping early varied between 50 and 300, with a mean of 134 epochs. It can be seen that the total training time of the networks is significantly lower when using the early stopping criterion, at the cost of only a small decrease in performance.

Fig. 6 shows a particular day where the error bars are plotted together with the committee predictions. The error bars are larger in the peak of the day, where the predictions are indeed deviating more from the actual travel times because in these conditions the travel time shows more variance. It was found that 97.4% of the actual travel times fell within the calculated 95% committee confidence intervals. However, the confidence intervals are found to be too pessimistic on occasions, where the first factor of Eq. (21), $1/\beta$, appears to be dominant. This is due to the fact that the error term E_D is relatively large for all networks, as they show oscillating behavior around the actual travel times in some peak periods of the training days. This can be explained by the fact that relatively simple neural network architectures, which are not capable of capturing all traffic dynamics, are chosen in this study, as was already noted in the introduction (Section 1.3).

5. Discussion

As is shown in Fig. 4, the evidence is found to be informative on the generalization performance of the neural networks. It can therefore be used to select high performance neural networks from all models that are trained, without having to split the training set in two and using a part to test the generalization performance. The evidence framework provides a convenient and simple way to select high performance networks, leaving all training data to be used to train the networks. The correlation between the evidence and the test error does show imperfections, as is reported by other authors as well (Bishop, 1995; Thodberg, 1993; MacKay, 1992b). Apart from the fact that the calculation of the evidence involves several simplifications and assumptions, (MacKay, 1992b) notes that a poor correlation between evidence and generalization error may be an indicator for the limitations of the models. The neural networks used in this study all have one hidden layer and use only one time period of flows and speeds as input; in other words, the predictive power of these networks is limited due to their relatively simple input structures. It is expected that if some of these limitations are overcome, for example by using recurrent or state space networks (van Lint et al., 2005), the correlation between the evidence and the generalization error will become stronger. Furthermore, the test error is measured on a finite data set and therefore is a noisy quantity, causing part of the scatter in Fig. 4. It is expected that the correlation becomes stronger when the networks are tested on a larger data set.

The error of the committee is 0.3–1.0% lower to that of the individual neural network with the highest evidence. This means that the effort in training many candidate networks is not lost, but can be used to improve predictions. Besides of this being positive for the modeler, the gain in prediction accuracy will benefit the road user, as they will have more accurate information available about the travel time they will experience. This may be beneficial to alleviate congestion and to decrease negative effects on the environment and the society.

The confidence interval provides a convenient way to inform the road user about the uncertainty of the predictions. It is desirable to avoid giving the road user a false sense of certainty when in fact the travel time proves hard to be predicted (by the selected prediction models). The users' trust of the information is an important factor for the impact of ATIS applications (Kantowitz et al., 1997), as providing inaccurate traffic information causes drivers to distrust the information and the possible beneficial effects of ATIS to decrease. The estimation of the error bars appeared to be too pessimistic on occasions, due to oscillating behavior of the models causing relatively large errors on training data. When more powerful models are used, the data error term E_D is expected to decrease, and as $\beta \sim 1/E_D$, from Eq. (21) it follows that the confidence intervals will decrease as a result.

6. Conclusion

In this study two successful approaches to traffic prediction have been fused: combined prediction and neural networks. The Bayesian framework for neural networks, which is applied to traffic prediction for the first time, introduces a way of dealing with noisy input data when training neural networks and naturally leads to confidence intervals for the predictions. A new stopping criterion using the evidence factor calculated for each neural network was introduced in the paper.

Furthermore, the evidence proved to be useful as a measure to select high performance networks and to form a committee of travel time predictors.

The predictions of the committee with the selected high-evidence networks proved to be more accurate than those of the individual networks. This leaves the modeler with a procedure to construct a more accurate prediction with very little additional effort, but more importantly, the end user with more accurate information. Together with the error bars that follow from the Bayesian analysis, the end user does not only receive more accurate traffic information, but also receives information on the reliability of the information and of the traffic conditions. This leads to more useful information for commercial as well as dynamic traffic management applications.

Future research will focus on the application of the theory on other traffic variables, such as traffic flow, which can serve as an input to Dynamic Traffic Assignment (DTA) models. The DTA models can then be used to predict traffic conditions on entire road networks or as a dynamic traffic management tool.

If the Bayesian learning of the network weights is applied to neural networks with more powerful structures, such as recurrent networks, or with different optimization strategies, such as Bayesian pruning, and if the analysis is applied to larger training and test sets, it is expected that the evidence becomes more informative and the error bars become more accurate.

Acknowledgements

This research was sponsored by the Advanced Traffic MOnitoring (ATMO) project under the Transumo (TRANsition SUstainable MObility) program. For more information, please visit www.atmo.tudelft.nl. Vialis Traffic BV (www.vialis.nl) is thanked for delivering the datasets.

References

Baesens, B., Viaene, S., Van den Poel, D., Vanthienen, J., Dedene, G., 2002. Bayesian neural network learning for repeat purchase modelling in direct marketing. European Journal of Operational Research 138, 191–211.

Bishop, C.M., 1992. Exact calculation of the Hessian matrix for the multilayer perceptron. Neural Computation 4, 494–501.

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford University Press, Oxford.

Chua, C.G., Goh, A.T.C., 2003. A hybrid Bayesian back-propagation neural network approach to multivariate modelling. International Journal for Numerical and Analytical Methods in Geomechanics 27, 651–667.

Clark, S., 2003. Traffic prediction using multivariate nonparametric regression. Journal of Transportation Engineering 129, 161–168.

Dharia, A., Adeli, H., 2003. Neural network model for rapid forecasting of freeway link travel time. Engineering Applications of Artificial Intelligence 16, 607–613.

Dougherty, M.S., Cobbett, M.R., 1997. Short-term inter-urban traffic forecasts using neural networks. International Journal of Forecasting 13, 21–31.

Gull, S.F., 1989. Developments in Maximum Entropy Data Analysis. Kluwer, Dordrecht.

Hecht-nielsen, R., 1989. Theory of the backpropagation neural network. In: International Joint Conference on Neural Networks, Washington, DC, USA.

Innamaa, S., 2005. Short-term prediction of travel time using neural networks on an interurban highway. Transportation 32, 649–669.

Johansson, E.M., Dowla, F.U., Goodman, D.M., 1991. Backpropagation learning for multilayer feed-forward neural networks using the conjugate gradient method. International Journal of Neural Systems 2, 291–301.

Kantowitz, B.H., Hanowski, R.J., Kantowitz, S.C., 1997. Driver acceptance of unreliable traffic information in familiar and unfamiliar settings. Human Factors 39, 164–176.

Krogh, A., Herts, J.A., 1995. A simple weight decay can improve generalization. Advances in Neural Information Processing Systems 4, 950–957.

Kuchipudi, C.M., Chien, S.I.J., 2003. Development of a hybrid model for dynamic travel-time prediction. Transportation Research Record 1855, 22–31.

Lee, S., Fambro, D.B., 1999. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. Transportation Research Record 1678, 179–188.

Lisboa, P.J.G., Wong, H., Harris, P., Swindell, R., 2003. A Bayesian neural network approach for modelling censored data with an application to prognosis after surgery for breast cancer. Artificial Intelligence in Medicine 28, 1–25.

Mackay, D.J.C., 1992a. Bayesian interpolation. Neural Computation 4, 415–447.

Mackay, D.J.C., 1992b. A practical Bayesian framework for backpropagation networks. Neural Computation 4, 448–472.

Mackay, D.J.C., 1994. Bayesian non-linear modelling for the prediction competition. ASHRAE Transactions 100, 1053–1062.

Mackay, D.J.C., 1995. Probable networks and plausible predictions – a review of practical Bayesian methods for supervised neural networks. Network: Computation in Neural Systems 6, 469–505.

Mark, C.D., Sadek, A.W., Rizzo, D., 2004. Predicting experienced travel time with neural networks: a PARAMICS simulation study. In: The 7th International IEEE Conference on Intelligent Transportation Systems, Washington, DC, USA.

Møller, M., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6, 525–533.

Nihan, N.L., 1980. Use of the box and Jenkins time series technique in traffic forecasting. Transportation 9, 125–143.

Nikovski, D., Nishiuma, N., Goto, Y., Kumazawa, H., 2005. Univariate short-term prediction of road travel times. In: 8th International IEEE Conference on Intelligent Transportation Systems, Vienna, Austria.

Okutani, I., Stephanedes, Y.J., 1984. Dynamic prediction of traffic volume through Kalman filtering theory. Transportation Research Part B 18, 1–11.

Perrone, M.P., 1994. General averaging results for convex optimization. Connectionists Models Summer School, Hillsdale, NJ, USA.

Petridis, V., Kehagias, A., Petrou, L., Bakirtzis, S., Kiartzis, S., Panagiotou, H., Maslaris, N., 2001. A Bayesian multiple models combination method for time series prediction. Journal of Intelligent and Robotic Systems 31, 69–89.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 2007. Numerical Recipes: The Art of Scientific Computing. Cambridge University Press, New York, NY, USA.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning Internal Representations by Error Propagation. MIT Press, Cambridge, MA, USA.

Smith, B.L., Demetsky, M.J., 1996. Multiple-interval freeway traffic flow forecasting. Transportation Research Record 1554, 136–141.

Thodberg, H.H., 1993. Ace of Bayes: Application of Neural Networks with Pruning. The Danish Meat Research Institute, Roskilde.

Van Hinsbergen, C.P.I., Van Lint, J.W.C., 2008. Bayesian combination of travel time prediction models. 87th Annual Meeting of the Transportation Research Board, Washington DC, USA.

Van Hinsbergen, C.P.I., Van Lint, J.W.C., Sanders, F.M., 2007. Short term traffic prediction models. In: Proceedings of the 14th ITS World Congress, Beijing, China.

Van Lint, J.W.C., 2004. Reliable Travel Time Prediction for Freeways. Delft University of Technology, Delft, The Netherlands.

Van Lint, J.W.C., 2008. Online learning solutions for freeway travel time prediction. IEEE Transactions in Intelligent Transportation Systems 9, 38–47.
Van Lint, J.W.C., Hoogendoorn, S.P., Van Zuylen, H.J., 2005. Accurate travel time prediction with state-space neural networks under missing data. Transportation Research Part C: Emerging Technologies 13, 347–369.
Williams, P.M., 1991. A Marquardt Algorithm for Choosing the Step-Size in Backpropagation Learning with Conjugate Gradients. University of Sussex, Sussex.
Yang, J.-S., 2005. Travel time prediction using the GPS test vehicle and Kalman filtering techniques. In: American Control Conference, Portland, Oregon, USA.
Zhang, H.M., 2000. Recursive prediction of traffic conditions with neural network models. Journal of Transportation Engineering 126, 472–481.
Zheng, W., Lee, D., Shi, Q., 2006. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. Journal of Transportation Engineering 132, 114–121.
Zhong, M., Sharma, S., Lingras, P., 2005. Refining genetically designed models for improved traffic prediction on rural roads. Transportation Planning and Technology 28, 213–236.