Neural Networks 23 (2010) 386–395

Contents lists available at ScienceDirect

Neural Networks

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

An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting

Henrique S. Hippert[a,*], James W. Taylor[b]

[a] Universidade Federal de Juiz de Fora, Brazil

[b] Saïd Business School, University of Oxford, UK

ARTICLE INFO

Article history:
Received 7 September 2008
Received in revised form 12 August 2009
Accepted 25 November 2009

Keywords:
Bayesian neural networks
Load forecasting
Input selection
Bayesian model selection

ABSTRACT

Artificial neural networks have frequently been proposed for electricity load forecasting because of their capabilities for the nonlinear modelling of large multivariate data sets. Modelling with neural networks is not an easy task though; two of the main challenges are defining the appropriate level of model complexity, and choosing the input variables. This paper evaluates techniques for automatic neural network modelling within a Bayesian framework, as applied to six samples containing daily load and weather data for four different countries. We analyse input selection as carried out by the Bayesian 'automatic relevance determination', and the usefulness of the Bayesian 'evidence' for the selection of the best structure (in terms of number of neurones), as compared to methods based on cross-validation.

1. Introduction

Neural networks (NNs) have frequently been proposed for short-term load forecasting (STLF), because of their capabilities for nonlinear modelling of large multivariate datasets.

The family of NN models known as *multilayer perceptrons* (MLPs) are probably the most frequently used, since they have been shown to be *universal approximators* of functions (Haykin, 1999), and can be used to model the function that relates the electric load to its exogenous variables.

Modelling with MLPs is not an easy task though; two of the main challenges are defining the appropriate level of model complexity, and choosing the input variables. The complexity of a NN is dictated, in the first instance, by its architecture; for NNs with one hidden layer with sigmoid neurones, and an output layer with linear neurones (the sort of NN most frequently used for STLF), the complexity depends mostly on the number of hidden neurones. However, the architecture is but one aspect of the problem, since the size of the weights (their absolute values) must also be taken into account. If the weights are small, the activation functions of the neurones will be operating in the central part of their ranges, which is practically linear. A large NN, in this case, would be no more complex than a linear regression model.

In order to control the complexity of a NN, one has to work on these two aspects. First, one should choose the right architecture (the appropriate number of neurones). This may be done by changing the number of neurones step by step, until an optimum value is found. The initial model may be either a very large one, from which neurones are progressively removed ('pruning' algorithms), or a small one, to which neurones are progressively added ('growing' algorithms). The choice is based on the principle of parsimony (a model should be as complex as necessary for a given task, but not more), and it requires some measure of NN complexity to be evaluated at each step. Second, one should control the size of the NN weights by using 'regularisation' techniques. These techniques add a term to the NN cost function that penalises for large weights, which ensures that the training algorithm will lead to NN weights that are as small as possible (Haykin, 1999).

Another difficult task in NN modelling (as in all nonlinear modelling) is the selection of the subset of variables to be used as inputs. Some metrics for the influence of each variable on the output are needed; a comparison of methods to evaluate this influence is done by Papadokonstantakis, Lygeros, and Jacobsson (2006), who consider three groups of methods: (a) those that take only the data into account, before the NN modelling (statistical methods such as PCA, etc.); (b) those that affect the training (such as ARD, discussed below); and (c) those applied on the trained NN. For this last group, two kinds of metrics of input relevance are considered — the ones that measure the input's 'predictive importance' (i.e., the increase in the generalisation error when an input is omitted), and the ones that measure its 'causal importance' (the change in the output caused by changes in the input) (Lampinen & Vehtari, 2001). The predictive importance may be empirically evaluated on an independent sample; the causal importance is usually measured by analytical means, such as the second derivatives of the error, information theory measures, etc.

[*] Corresponding address: Depto. de Estatistica / ICE - UFJF, Campus Universitario, 36036 900 - Juiz de Fora, MG, Brazil. Tel.: +55 32 3229 3306.
E-mail addresses: henrique.hippert@ufjf.edu.br (H.S. Hippert), james.taylor@sbs.ox.ac.uk (J.W. Taylor).

0893-6080/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2009.11.016

Besides choosing architecture and inputs, the NN designer also has to tune several parameters, depending on the algorithm used, such as regularisation coefficients, momentum rate, learning rate, etc. A common way of supporting these choices and tunings is to do trial-and-error simulations on a 'validation' sample, distinct from the training and the testing ones; this procedure is called 'cross-validation' (CV).

Cross-validation is an empirical technique that can be put to many uses – comparing different architectures or input sets, avoiding overfitting, tuning training parameters, etc. However, it also has its problems. On the theoretical side, Cataltepe, Abu-Mostafa, and Magdon-Ismail (1999) showed that there is no guarantee that the model selected by CV is indeed the best one. Intuitively, this is easy to understand; given an infinite number of models, it is always possible to find one that overfits both the training and the validation set, and proves to be useless out-of-sample. In real world applications, however, the number of models to be compared is usually small, and this problem does not happen. On the practical side, CV has some limitations too. First, the estimates of generalisation ability obtained on a CV sample are noisy. A different CV sample may lead to different conclusions; to overcome this, it is advisable to use several CV samples and average their results, using methods such as *k-fold*, *leave-one-out*, or *bootstrap* (Efron & Tibshirani, 1993; Lendasse, Wertz, & Verleysen, 2003).

For forecasting applications, however, this is not possible, since the CV sample must always consist of data that are more recent than the training data, but older than the data used for out-of-sample testing; this reduces the choices of possible CV samples. Also, CV may only be used to tune one discrete variable at a time. If it is necessary to optimise n parameters, and each one can assume m distinct values, it will be necessary to run the simulation m^n times, and the computational cost might easily become prohibitive.

Among the methods that have been proposed to deal with these difficulties in NN modelling, this paper focuses on the Bayesian approach. In this method, any variables of interest (weights, regularisation coefficients, number of neurones, relevance of inputs, NN outputs, etc.) are modelled by random variables, for which prior distributions are assumed; after the data have been collected, posterior distributions are derived, by means of the theorem of Bayes. In principle, there are several advantages to this approach, in comparison to more traditional ones based on CV: it is possible to obtain probability distributions for the variables of interest, and not only point estimates (which allows the researcher to quantify the uncertainty by means of confidence intervals); all available data can be used for training (since there is no need to reserve data for a CV sample); the relevance of the inputs can be assessed after the training (by a technique called *Automatic Relevance Determination*, discussed below); and the optimum number of neurones can be found (by comparing the Bayesian 'evidence' of the models). Reviews of the Bayesian approach to NNs can be found in Lampinen and Vehtari (2001), Penny and Roberts (1999), Thodberg (1996) and Titterington (2004).

The application of Bayesian theory to neural networks was started by Buntine and Weigend (1991). Mackay (1992a, 1992b) introduced the 'evidence approximation' framework, which is based on a Gaussian approximation for the posterior distribution of the network weights. This framework simplified the mathematical treatment, and allowed the derivation of expressions to estimate the most probable values of the hyperparameters, and the most probable model. Other authors avoided the approximation by integrating the posterior with a numerical technique known as Markov Chain Monte Carlo (MCMC) (Barber & Bishop, 1997; Lampinen & Vehtari, 2001; Muller & Rios Insua, 1998), with hybrids between MCMC and genetic algorithms (Chua & Go, 2003; Liang, 2005), or with techniques that use neither Gaussian approximations or MCMC: the variational method (Titterington, 2004), and the Bayesian conjugate prior method (Rossi & Vila, 2006; Vila, Wagner, & Neveu, 2000). Research on Mackay's evidence approximation approach was carried further by several authors, but the conclusions about the usefulness of this technique were conflicting. Thodberg (1996) was overall favourable, but Lampinen and Vehtari (2001) and Penny and Roberts (1999) tended to be sceptical, particularly for problems where the number of training patterns was small with respect to the number of weights. Recently, however, the evidence approximation framework was applied to STLF by Lauret, Fock, Randrianarivony, and Manicom-Ramsamy (2008) and Silva and Ferreira (2007), with reportedly good results.

STLF is an essential task in the daily operation of electric power systems, both for technical and financial reasons. Forecasts with lead times ranging from a few hours to one week ahead are needed to support the decisions of the system operators and market agents, in performing tasks such as load dispatching, scheduling of generation, energy trading, and purchasing of fuel. Accurate forecasts have been shown to lead not only to more security in the operation but also to considerable cost savings (Bunn, 2000). All this has increased the demand for improved forecasting methods, and a great deal of research has been devoted to this area.

In this paper we evaluate the application of Mackay's evidence approximation framework to STLF, using six samples. Samples I and II are series of hourly values for loads and for five weather variables, from England and Wales; Samples III and IV are series of hourly loads and temperatures, from a utility in Rio de Janeiro. In order to confirm the findings from those data, we then repeat the study on two more samples, available online, which have already been used by several other researchers: one series of hourly loads and temperatures for a North American utility (Sample V); and one series of hourly loads and temperatures for a utility in Slovakia (Sample VI). We compare the results of the Bayesian methods to those of methods based on CV, and to a naïve method that serves as a benchmark.

The paper is organised as follows. Section 2, presents a short overview of the Bayesian approach to NN modelling; Section 3 describes the routines used in the studies; Section 4 presents the datasets and discusses the results; and Section 5 is the conclusion.

2. The Bayesian approach to neural network modelling – An overview

This section provides a short introduction to the Bayesian approach to NN modelling, summarised from Bishop (1995). The idea is to introduce the concepts of *evidence* and of *automatic relevance determination*, on which the analyses in this study are based.

Bayesian NN modelling starts by postulating that the weights $\mathbf{w}$ of a NN are random variables, and assuming a prior distribution $p(\mathbf{w})$ for them. After a sample has been observed, a posterior distribution $p(\mathbf{w}|D)$ is calculated by means of Bayes theorem,

$$p(\mathbf{w}|D) = \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)} \tag{1}$$

where $\mathbf{w}$ is the weight vector and $D = \{t\}^n$ is the set of target vectors. The probability $p(D|\mathbf{w})$ is the *likelihood*, usually called the *evidence* in this context. Expression (1) should have all the probabilities conditioned on the input data X; however, since the NNs do not model X, and the conditionings affect both sides of the equation, this can be omitted from the notation.

2.1. Evaluating the posterior distribution of weights

To ensure smooth mappings, exponential models are usually assumed both for the prior distribution and for the noise added to