Nonlinear Models for Short-time Load Forecasting

Philippe Lauret, Mathieu David, Didier Calogine

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Energy Procedia 14 (2012) 1404 – 1409

Energy Procedia

ICAEE 2011

Nonlinear Models for Short-time Load Forecasting

Philippe Lauret[a*], Mathieu David, Didier Calogine

PIMENT – Université de La Réunion, 15 avenue René Cassin, BP 97715 Saint Denis – Ile de La Réunion -France

Abstract

Short Term Load Forecasting (STLF) is essential for planning the day-to-day operation of an electric power system. As this forecasting leads to increased security operation's conditions and economic cost savings, numerous techniques have been used to improve the STLF. We propose in this paper the comparison of two nonlinear regression techniques namely Gaussian Process (GP) regression models and Neural Network (NN) models. While the Bayesian approach to NN modelling offers significant advantages over the classical NN learning methods, it will be shown that the use of GP regression models will improve the performances of the forecasting. The proposed techniques are applied to real load data.

Keywords: Load modelling, load forecasting, neural network, Gaussian process, Bayesian inference;

1. Introduction

Short Term Load Forecasting (STLF) is essential for planning the day-to-day operation of an electric power system [1]. Accurate forecasts of the system load on an hour by hour basis from one day to a week ahead help the system operator to accomplish a variety of tasks like economic scheduling of generating capacity, scheduling of fuel purchases, etc… In particular, the forecasting of the peak demand is important as the generation capacity of an electric utility must meet this requirement. As this forecasting leads to increased security operation's conditions and economic cost savings, numerous techniques have been used to improve the STLF [2]. Among these techniques, the use of Neural Networks (NNs) is particularly predominant in the load forecasting field [2]. Indeed, the availability of historical load data on the utility

[a] Corresponding author. Tel.: +262262938127; fax: +262262938665.
E-mail address: philippe.lauret@univ-reunion.fr.

doi:10.1016/j.egypro.2011.12.1109

databases and the fact that NNs are data-driven approaches capable of performing a nonlinear mapping between a set of input and output variables make this modelling tool very attractive. However, as stated by ([2], [3], NNs are such flexible models that the task of designing a NN for a particular application is far from easy. This statement stems from the fact that NNs are able to approximate any continuous function at an arbitrary accuracy, provided the number of hidden neurons is sufficient [4]. However, this ability has a downside that such close approximation can become approximation to the noise. As a consequence, the model yields solutions that generalize poorly when new data are presented. In the NN community, this problem is called overfitting and may come about because the NN model is too complex. In other words, conventional NN learning methods must be improved. For this purpose, we propose first a probabilistic interpretation of the NN learning by using Bayesian techniques. MacKay [5] originally developed Bayesian methods for NNs. The Bayesian approach to modelling offers significant advantages over the classical NN learning process. Among others, one can cite the automatic matching of complexity of the NN model to the problem being solved. However, many approximations must be done in the Bayesian NN context. Conversely, GPs [6] are powerful methods for regression where most of the computations are analytically tractable. In this survey, the two modeling techniques (NNs and GPs) are applied to real-load data. The data were provided by EDF, the French electricity utility.

Nomenclature

x_1	Week-end or Holiday flag for day d-1
x_2	hour
x_3	Yesterday's actual temperature at this hour
x_4	Yesterday's actual global solar irradiance at this hour
x_5	Yesterday's actual load at this hour
x_6	Week-end or Holiday flag for day d
x_7	Actual temperature at this hour
x_8	Actual global solar irradiance at this hour
x_9	Actual load at this hour
x_{10}	Temperature forecast for the next day at this hour

2. Model description and context of study

In order to assess the feasibility of the proposed approaches, we designed a NN model and a GP model whose goal is to forecast the next day's load at the same hour. Actually, the model constitutes an hourly module that consists in determining the nonlinear relationship between each hour's load profile with past load and weather readings for the same hour. This hourly module is part of a global forecaster (that yields the complete load profile for the next day) obtained by combining the 24 hourly modules. A bibliographic survey ([2], [7]) helps us to a priori retain the input variables x_1 to x_{10} given in the nomenclature. The data were collected from a micro-region of the south of Reunion Island (21.06 S, 55.36 E) in 2001. The database contains up to 1074 hourly records.

3. Neural network: a parametric approach to STLF

3.1. Classical NN

The most popular form of NN is the so-called Multi-Layer Perceptron (MLP) structure. The MLP structure consists of an input layer, one or several hidden layers and an output layer. The input layer gathers the model's inputs vector **x** while the output layer yields the model's output vector **y**. In our case, **y** is reduced to the scalar *y* which is the corresponding forecast of the next day's load at the same hour. Fig. 1 represents a one hidden layer MLP.

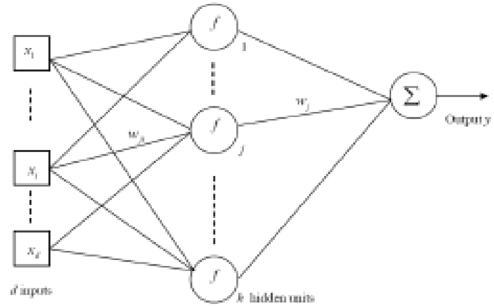

Fig. 1. Sketch of a MLP with d inputs and h hidden units, in our case, d=10. The output y is the next day's load at the same hour

The hidden layer is characterized by several nonlinear units (or neurons). The nonlinear function *f* is usually the tangent hyperbolic function

$$f(x) = (e^x - e^{-x})/(e^x + e^{-x}).$$

Therefore, an NN with d inputs, h hidden neurons and a single linear output unit defines a nonlinear parameterized mapping from an input $\mathbf{x}$ to an output y given by the following relationship $y = y(\mathbf{x}; \mathbf{w})$.The NN parameters w_{ji} and w_j, aggregated in the parameter vector **w**, govern the nonlinear mapping. The NN parameters **w** are estimated during a phase called the training or learning phase. The second phase, called the generalization phase, consists in evaluating the ability of the NN to generalize, that is to say, to give correct predictions when it is confronted with new input examples. Careful attention must be taken during the learning process as the NN model will overfit easily (i.e. will give poor predictions on unseen data) if the model is too complex. One means to controlling this complexity is to make use of Bayesian inference.

3.2. Bayesian NN

The Bayesian approach considers a probability density function (pdf) over weight space. This pdf represents the degrees of belief taken by the different values of the weight vector. This pdf is set initially to some prior distribution and converted into a posterior distribution once the data have been observed through the use of Bayes' rule

$p(a|b) = p(b|a)p(a)/p(b)$,

which can be also formulated as $posterior = likelihood$ x $prior/marginal\ likelihood$. So, instead of the single 'best' set of weights computed by the classical approach of maximum likelihood (through minimization of an error function), Bayesian methods yield a complete distribution for the NN parameters. This posterior weight distribution can then be used to infer predictions of the network for new values of the input variables. In such a supervised parametric learning, Bayesian inference is based on the ingredients depicted in Table 1.

Table 1. Bayesian NN inference

Item	Formulation	Remarks
Data	$D = \{\mathbf{x}_i, t_i\}_{i=1}^{n}$ and $D^* = \{\mathbf{x}_i^*, t_i^*\}_{i=1}^{n^*}$	Training set (n samples) and test set (n^* samples)
Model	$t_i = y(\mathbf{x}_i; \mathbf{w}) + \varepsilon_i$ The formulation of the Gaussian noise can also be given as: $\varepsilon_i \sim N(0, \sigma^2)$ i.e. ε_i follows a Gaussian distribution with mean zero and variance σ^2 .	We assume that the i^{th} target variable t_i is given by some deterministic function of input vector **x** with added independent Gaussian noise ε_i .
Likelihood	$p(\mathbf{t}|\mathbf{x}, \mathbf{w}) = \left(\frac{1}{2\pi\sigma^2}\right) \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(t_i - y(x_i; \mathbf{w}))\right)$	Since we chose a Gaussian noise, the likelihood is also Gaussian.
Prior	$p(\mathbf{w})$	
Posterior	$p(\mathbf{w}|\mathbf{x}, \mathbf{t}) = p(\mathbf{t}|\mathbf{x}, \mathbf{w})p(\mathbf{w})/p(\mathbf{t}|\mathbf{x})$	By applying Bayes' rule
Predictions	$p(y^*|x^*, \mathbf{x}, \mathbf{t}) = \int p(y^*|\mathbf{w}, x^*)\, p(\mathbf{w}|\mathbf{x}, \mathbf{t}) d\mathbf{w}$	Make predictions for a test case x^* by averaging over all parameter values weighted by their posterior probability.

It is important to note that in a multi-parameter context such as NN learning (the dimension of **w** is large), the evaluation of integrals over weight space cannot be analytically performed. MacKay [5] proposed a specific numerical approximation scheme called the evidence framework in order to overcome this problem. For more details about the implementation of Bayesian NN, the interested reader should also refer to [8].

4. Gaussian process: A non parametric approach to STLF

GPs are a relatively recent development in non-linear modeling [6]. GPs are well suited to regression problem as the inference can be done analytically as opposed with NN models where approximations are needed in order to evaluate the integrals. An alternative way of reaching identical (or even better) results to the previous section is possible by considering inference directly in function space (whereas, regarding NN, inference is taking place in the weight space). A GP is a generalization of a multivariate Gaussian distribution to infinitely many variables. A multivariate Gaussian distribution is fully specified by a mean vector μ and covariance matrix Σ e.g. $\mathbf{f} = (f_1, f_2, f_3, \ldots, f_n) \sim N(\mu, \Sigma)$. Similarly, a GP is fully specified by a mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x},\mathbf{x}')$ e.g $f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x},\mathbf{x}'))$.The covariance function specifies the covariance between pairs of random variables namely

$$\text{cov}\left(f(\mathbf{x}_p), f(\mathbf{x}_q)\right) = \text{cov}(\mathbf{x}_p, \mathbf{x}_q) = exp\left(-\tfrac{1}{2}\left|\mathbf{x}_p - \mathbf{x}_q\right|^2\right).$$

One can see that the covariance is almost unity between variables whose corresponding inputs are very close, and decreases as their distance in the input space increases. The specification of a covariance function implies a distribution over functions. Indeed, one can draw samples from the distribution of functions by first, calculating the covariance matrix K (whose elements are given by $k(x_p, x_q) = \text{cov}(x_p, x_q)$ and second, by generating a random vector $\mathbf{f} \sim N(0, K)$. One can then plot the generated values as a function of the inputs. Table 2 gives an overview of the different calculations.

Table 2. GP calculations

Item	Formulation	Remarks
Data	$D = \{\mathbf{x}_i, t\}_{i=1}^{n}$ the column vector inputs for all n cases can are aggregated in the so-called $d \times n$ design matrix X and the targets are collected in the vector $\mathbf{t}$, so we can write $D = \{X, \mathbf{t}\}$. Similarly, we have $D^* = \{X^*; \mathbf{t}^*\}$	Training set and test set
Predictions	$\mathbf{y}^* \vert X^*, X, \mathbf{t} \sim N\begin{pmatrix} \bar{y}^* = K(X^*, X)[K(X,X) + \sigma^2 I]^{-1}\mathbf{t}, \\ \text{cov}(\mathbf{y}^*) = K(X^*, X^*) - K(X^*, X)[K(X,X) + \sigma^2 I]^{-1} K(X, X^*) \end{pmatrix}$	Gaussian predictive distribution over the test set
Covariance matrix	$K(X, X^*)$ If there are n training points and n^* test points then $K(X, X^*)$ represents the $n \times n^*$ matrix of the covariances evaluated at all pairs of training and test points. The same procedure is used to calculate $K(X, X)$, $K(X^*, X^*)$and $K(X, X^*)$	

5. Results

We chose contiguous range of data for training and testing i.e. n=693 hourly values were used for the NN and GP training. The rest of the data were used for testing ($n*$ =381 samples). The performance of the models was assessed by reporting their MAPE and RMSE errors. In order to illustrate the overfitting problem and to highlight the benefits of the Bayesian approach, we chose deliberately to model the electric load with a (relatively large) classical NN of 32 hidden units. Table 3 lists the performance of the models obtained on the training and test sets. As one can see, the fit is very good on the training set but the performance degrades on the test set (line 1 of Table 3). This type of results is a sign of overfitting. Here, the model is too complex and one has to employ techniques, which could decrease this complexity. Therefore, we took the same NN structure but used the Bayesian approach to forecasting the electric load.

Table 3. Results

Model	RMSE Training set (MW)	MAPE Training set (%)	RMSE Test set (MW)	MAPE Test set (%)	CPU time (s)
Classical NN	3.27	1.08	21.34	6.95	410.2
Bayesian NN	6.69	1.97	13.49	4.27	1099.2
GP model	9.58	2.92	10.34	3.22	184.7

Table 3 (line 2) shows clearly the improvement brought by the Bayesian method. However, the Bayesian approach is computationally demanding as the numerical framework needs more iterations to find the optimal complexity. Finally, the GP model (line 3 of Table 3) led to improved performance. In addition of being more straightforward to build, the GP model is computationally less demanding. Fig. 2 shows the predictions of the GP model on the test set.

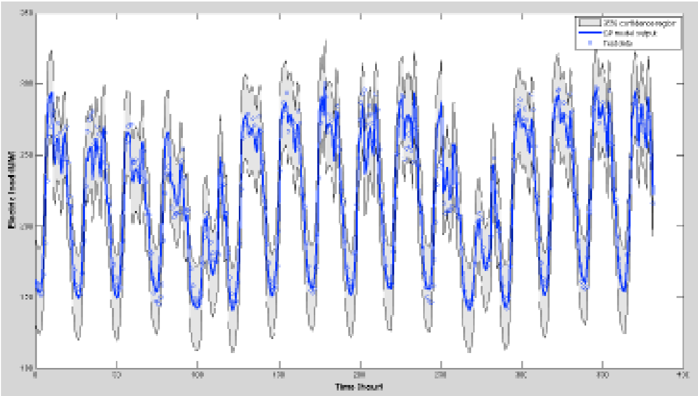

Fig. 2. Predictions of the GP model. The solid line indicates an estimation of the mean of $\mathbf{y}^*$. 95% confidence intervals are shaded

6. Conclusion

In this work, we have proposed a comparison of different approaches to STLF. It has been shown that, unlike the traditional NN techniques, the Bayesian NN method is able to deal quite efficiently with model complexity (and therefore with the problem of overfitting). The main drawback of the Bayesian NN approach stems from the fact that many approximations must be done in order to evaluate numerically the integrals over the weight space. Conversely, GPs offer another point of view by making Bayesian inference in the function space and therefore make the computations tractable. Future work will be devoted to the design of robust models those which, in particular, will take into account uncertainties in the model's inputs.

References

[1] Senjyu T, Takara H, Uezato K, Funabashi T. One-hour-ahead load forecasting using neural network. *IEEE transactions on power systems* 2002;**17**:113-118.

[2] Hippert HS, Pedreira CE, Souza RC. Neural networks for short-term load forecasting: a review and evaluation. *IEEE transactions on power systems* 2001;**16**:45-55.

[3] Zhang G, Patuwo BE, Hu MY. Forecasting with neural networks. *International journal of forecasting* 1998;**14**:35-62.

[4] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. *Neural Networks* 1989;**2**:359-366.

[5] MacKay DJC. A practical Bayesian framework for back-propagation networks. *Neural computation* 1992;**4**: 448-472.

[6] Rasmussen CE, Williams C. *Gaussian Processes for Machine Learning.* MIT Press; 2006.

[7] Yuan L, Fine TL. Neural-network design for small training sets of high dimension. *IEEE transactions on neural networks* 1998;**9**:267-280.

[8] Lauret P et al.. Bayesian neural network approach to short time load forecasting. *Energy Conversion and Management*, 2008;**49**:1156-1166.