BAYESIAN NEURAL NETWORKS IN PREDICTION OF GEOMAGNETIC STORMS

GABRIELA ANDREJKOVÁ

Department of Computer Science, Faculty of Science
P. J. Šafárik University
Jesenná 5, 041 54 Košice, Slovakia
e-mail: andrejk@kosice.upjs.sk

Bayesian probability theory provides a framework for data modelling. In this framework it is possible to find models that are well-matched to the data, and to use these models to make nearly optimal predictions. In connection to neural networks and especially to a neural network learning the theory is interpreted as an inference of the most probable parameters for the model and the given training data. This article describes an application of the Bayesian probability theory to the physical problem "Prediction of Geomagnetic Storms".

Keywords: Bayesian probability theory; Neural Network; Geomagnetic Storm; Prediction.

1 Introduction

Neural networks continue to offer an attractive paradigm for the design and analysis of adaptive, intelligent systems for many applications in artificial intelligence [4], [5]. This is true for a number of reasons: for example, amenability to adaptation and learning, robustness in the presence of noise, potential for massively parallel computation.

Predictions of the hourly D_{st} index from the interplanetary magnetic field and solar plasma data, based on Artificial Neural Networks (ANN), were made and analysed by Lundstedt and Wintoft (1994) (feedforward networks) [7] and Andrejková et al. (1996, 1999) (recurrent networks, fuzzy neural networks) [1], [2]. Recent results have shown that it is possible to use dynamic neural networks for GMS prediction and modelling of the solar wind-magnetosphere coupling. In this study we are reporting preliminary results using a Bayesian neural network model.

There has been increased interest in combining artificial neural networks with Bayesian probability theory [3]. The Bayesian probability theory have been

proved to be very successful in a variety of applications, for example D. J. C. MacKay (1995), [8], [9], M. I. Schlessinger and V. Hlaváč, [13] and P. Müller and D. R. Insua, [10]. The effectiveness of the models representing nonlinear input-output relationships depends on the representation of the input-output space.

A designed neuro-Bayesian model will predict the occurrence of geomagnetic storms on the base of input parameters n, v, σ_{B_z} and B_z: n ... the plasma density of solar wind, v ... the bulk velocity of solar wind, B_z, σ_{B_z} ... z-component of the interplanetary magnetic field and its fluctuation.

To follow the changes of the geomagnetic field values we use the quantity D_{st} *index*. Its values are in interval $\pm$10nT during normal situation but during the geomagnetic storm they can decrease as much as some hundreds nT in a few hours.

In Section 2, we describe some basic definitions and properties of the Bayesian probability theory. In Section 3, we briefly describe the neural networks as a probabilistic models. Section 4 contains the starting point to the finding weights of neural networks. Some interesting results for GMS prediction are described in Section 5.

2 Bayesian probability theory

A Bayesian data-modeller's aim is to develop probabilistic model that is well matched to the data and makes optimal predictions using that model. Bayesian inference satisfies the likelihood principle: Inferences depend only on the probabilities assigned to the data that were received, not on properties of the data which might have occured.

We will use the following notation for *conditional* probabilities:

- $\Omega, \Omega \neq \emptyset$ - the space of elementary events;
- $\mathcal{H}$ - $\sigma - algebra$ of some nonempty subset of Ω (a model of computation),
- A, B - events, $P(A), P(B)$ - a probability of the events A, B,
- $(\Omega, \mathcal{H}, \mathcal{P})$ - a probability space,
- $P(A|B, \mathcal{H})$ is pronounced "the probability of A, given B and $\mathcal{H}$" and it explains the conditional probability;

- the statements B and $\mathcal{H}$ mean the conditional assumptions on which this measure of plausibility is based;

The Bayesian approach require:

- specifying a set of prior distributions for all of weights in the network (and variance of the error) and
- computing the posterior distributions for the weights using Bayes' Theorem.

Prior distribution is a probability distribution on the unknown parameter vector $\omega \in \Omega$ in the probability model, typically described by its density function $P(\omega)$ which encapsulates the available information about the unknown value of ω. In our case, the vector of weights $\mathbf{w}$ has not some know prior distribution and it means the prior distribution will be replaced by a reference prior function.

Posterior distribution is a probability distribution on the unknown parameter vector $\omega \in \Omega$ in the probability model, typically described by its density function $P(\omega|D)$, conditionally on the model, encapsulates the available information about the unknown value of ω, given the observed data D and the knowledge about ω, which the prior distribution $P(\omega)$ might contain. It is obtained by Bayes' Theorem.

Bayes' Theorem: Given data $D\{\mathbf{x}^{(m)}, \mathbf{y}^{(m)}\}$ generated by the probability model $\{P(D|A), A \in \Omega\}$ and a prior distribution $P(A)$, the posterior distribution of A is $P(A|D) \propto P(D|A) * P(A)$. The proportionality constant is $\{\int_\Omega P(D|A) * P(A)dA\}^{-1}$.

Two approaches have been tried in the finding of the posterior probability:

- to find the most probable parameters (weights) using methods similar to the conventional training and then approximate the distribution over weights using information available at this maximum.
- to use Monte Carlo method to sample from the distribution over weights. We applied the method and we use Markov chaines.

There are two rules of probability which can be used:

- The *product rule* relates to joint probability of A and $B, P(A, B|\mathcal{H})$ to the conditional probability:

$$P(A, B|\mathcal{H}) * P(B|\mathcal{H}) = P(A|B, \mathcal{H}) \tag{1}$$

- The *sum rule* relates the *marginal* probability distribution of A, $P(A|\mathcal{H})$, to the joint and conditional distributions:

$$P(A|\mathcal{H}) = \sum_B P(A, B|\mathcal{H}) = \sum_B P(A|B, \mathcal{H})P(B|\mathcal{H}) \tag{2}$$

Having specified the joint probability of all variables as in equation, we can use the rules of probability to evaluate how our beliefs and predictions should change when we get new information.

3 Neural Networks as probabilistic models

A supervised neural network is a non-linear parametrized mapping from an input $\mathbf{x}$ to an output $\hat{\mathbf{y}} = \mathbf{f}(\mathbf{x}, \mathbf{w}; \mathcal{A})$. The output is a continuous function of the parameters $\mathbf{w}$, which are called weights and $\mathcal{A}$ is an architecture of the network.

The network is trained in the classical way using a data set $D = \{\mathbf{x}^{(m)}, \mathbf{y}^{(m)}\}$ by the backpropagation algorithm. It means the following sum squared error is minimized

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_m \sum_i (y_i^{(m)} - f_i(\mathbf{x}^{(m)}; \mathbf{w})^2 \tag{3}$$

The weight decay is often to include to the objective function for the minimization. It means

$$M(\mathbf{w}) = \beta E_D + \alpha E_W, \tag{4}$$

where $E_W = \frac{1}{2} \sum_i w_i^2$.

The learning process above can have the following probabilistic interpretation. The error function is interpreted as minus the log likelihood for a noise model:

$$P(D|\mathbf{w}, \beta, \mathcal{H}) = \frac{1}{\mathcal{Z}_\mathcal{D}(\beta)} \exp\ (-\beta \mathrm{E}_\mathcal{D}) \tag{5}$$

parameter β here defines a noise level $\sigma_n^2 = \frac{1}{\beta}$.

$$P(\mathbf{w}|\alpha, \mathcal{H}) = \frac{1}{\mathcal{Z}_\mathcal{W}(\alpha)} \exp(-\alpha \mathrm{E}_\mathcal{W}) \tag{6}$$

where $\sigma_W^2 = \frac{1}{\alpha}$.

The function E corresponds to the deduction of parameters $\mathbf{w}$ according to data D. It means

$$P(\mathbf{w}|D,\alpha,\beta,\mathcal{H}) = \frac{\mathcal{P}(\mathcal{D}|\mathbf{w},\alpha,\beta,\mathcal{H}) * \mathcal{P}(\mathbf{w}|\alpha,\mathcal{H})}{\mathcal{P}(\mathcal{D},\alpha,\beta,\mathcal{H})} \tag{7}$$

Bayesian inference for modelling problems may be implemented by analytical methods, by Monte Carlo sampling, or by deterministic methods using Gaussian approximations.

4 Starting points to the application

We deal only with neural networks used for regression. Assuming a Gaussian noise model, the conditional distribution for the output vector given the input vector based on this mapping will be as follows:

$$P(\mathbf{y}|\mathbf{x},\mathbf{w}) = (2\pi\sigma^2)^{d/2} exp(-\frac{|\mathbf{y} - f(\mathbf{x},\mathbf{w})|^2}{2\sigma^2}) \tag{8}$$

where d is the dimension of the output vector and σ is the level of the noise in the outputs.

In the Bayesian approach to the statistical prediction, one does not use a single "best" vector of weights, but rather integrates the prediction from all possible weight vectors over the posterior weight distribution which combines the data with prior computed weights.

The best prediction for the given input from the testing data $\mathbf{x_{i+1}}$ can be expressed

$$\widehat{\mathbf{y}}_{n+1} = \int_{R^d} f(\mathbf{x_{n+1}},\mathbf{w}) P(\mathbf{w}|(\mathbf{x_1},\mathbf{y_1})\dots(\mathbf{x_n},\mathbf{y_n}))\, d\mathbf{w} \tag{9}$$

where d is the dimension of the weight vector.

Posterior probabilities of weight vectors are the following:

$$\begin{aligned} P(\mathbf{w}|(\mathbf{x_1},\mathbf{y_1}),\dots,(\mathbf{x_n},\mathbf{y_n})) &= \frac{P(\mathbf{w})\mathbf{P}((\mathbf{x_1},\mathbf{y_1}),\dots,(\mathbf{x_n},\mathbf{y_n})|\mathbf{w})}{P((\mathbf{x_1},\mathbf{y_1}),\dots,(\mathbf{x_n},\mathbf{y_n}))} \\ &= \frac{P(\mathbf{w})\mathbf{P}(\mathbf{y_1},\dots,\mathbf{y_n}|\mathbf{x_1},\dots,\mathbf{x_n},\mathbf{w})}{P(\mathbf{y_1},\dots,\mathbf{y_n}|\mathbf{x_1},\dots,\mathbf{x_n},\mathbf{w})} \quad (10) \\ &= \frac{P(\mathbf{w})\prod_{\mathbf{i=1}}^{\mathbf{n}}\mathbf{P}(\mathbf{y_i}|\mathbf{x_i},\mathbf{w})}{P(\mathbf{y_1},\dots,\mathbf{y_n}|\mathbf{x_1},\dots,\mathbf{x_n})} \quad (11) \end{aligned}$$

For the full formulation of Bayesian problem it is necessary to add the prior distribution of weights. One of the possibilities is the following:

$$P(\mathbf{w}) = (\mathbf{2}\pi\omega^{\mathbf{2}})^{-\frac{\mathbf{N}}{\mathbf{2}}} \exp\left(-\frac{|\mathbf{w}|^{\mathbf{2}}}{\mathbf{2}\omega^{\mathbf{2}}}\right) \tag{12}$$

To compute the previous integrals it is very time consuming problem. It is possible to use Metropolis algorithm and it is the base for Monte Carlo method that we used in the prediction of GMS. We used the Monte Carlo method for a construction of our models. The algorithm was applied according the construction of R. M. Neal [11].

5 Results of GMS predictions

We have to discuss various implementation issues which is necessary to do for the real prediction.

The data are available from the NASA "OMNI tape" and are distributed by National Space Science Data Center and WDC-A for Rockets&Satellites. In the period 1963 - 1999 at each hour are measured and saved the next quantities: B_z, σ_{B_z}, n, v and D_{st}.

Some data are not complete and we use liner interpolation to fill them but only in the case if the gap has less then 30 hours. The reconstructed data are used for a choose of the samples to the training set according to the following criteria: if the value D_{st} decreases at least 40 nT during two hours then the training sample (the storm) is created from the measured values 36 hours before the decreasing, 2 hours of the identification of decreasing and 108 hours after the decreasing. The file of the values have to fulfill requirement of completeness of measurements. It means 144 hours describe one event - GMS. One storm is used for the learning of the neural network by the moving of 8 hours window.

We have prepared the training data set two data testing sets A and B. To prepare the A and B sets we used the data from years $1980 - 1984$ and $1989 - 1999$ because we had the continued values of parameters n, v, B_z, σ_{B_z} and D_{st}. The prepared data were represented by a sequence of

$$\mathbf{p}^t = (n^t, v^t, B_z^t, \sigma_{B_z}^t, D_{st}^t),$$

where $\mathbf{p}^t$ can be applied as time series.

Table 1. Experimental Results

Data	*#Iteration*	*#Good Predictions*	*#Bad Predictions*	*Average Error*	*Success*
A	4000	49	87	2.78377	36,03 %
B	4000	84	52	1.86015	61,76 %
A	6000	62	74	1.90585	45,59 %
B	6000	101	35	0.48040	74,26 %
A	12000	76	60	1.19665	55,88 %
B	12000	113	23	0.23863	83,09 %
A	18000	86	50	0.77771	63,24 %
B	18000	109	27	0.23801	80,15 %

The software of M. Levický described in [6] was modified and used in the present application. The algorithm based on the works of Neal and McKay was written in Delphi 5.

The first computed results are in the following Table 1. The models are just tested. We present results computed with two data sets A and B. Prediction performance is measured by $\#Good\,Predictions$, $\#Bad\,Predictions$, $AverageError$ and $\%\,of\,Success$.

Total test samples in testing sets A and B: 272, the number of input neurons in the neural network: 32, the number of hidden neurons in the neural networks: 28, the number of output neurons: 1.

The computed results are interesting from the following points of view:

- With the higher number of iteration the average error decreases. It is one of criteria to the evaluation of the model.
- After 18000 iterations the success grows very slowly in the case of the testing data set A and decrease in the case B.
- Bayesian neural networks that we used in the prediction of geomagnetic storms look like very good model. They move the weight vector to the most probable part of the weight space.

References

1. Andrejková, G., Azorová, J., Kudela, K.: *Artificial Neural Networks in Prediction D_{st} Index.* Proceedings of the 1st Slovak Neural Network Symp., ELFA, Košice, 1996, pp. 51-59.
2. Andrejková, G., Tóth, H., K., Kudela, K.: *Fuzzy Neural Networks in the Prediction of Geomagnetic Storms.* Proceedings of "Artificial Intelligence in Solar-Terrestrial Physics", Publisher European Space Agency, Lund, 1997, p. 173-179.
3. Bernardo, J. M.: *Bayesian Reference Analysis*, A Postgraduate Tutorial Course, Facultat de Matematiques, Valencia, 1998.
4. Hassoun, M. H.: *Fundamentals of artificial neural networks*, MIT Press, Cambridge, 1995.
5. Hertz, J., Krogh, A., Palmer, R.G.: *Introduction to the theory of neural computation*, LN Vol. 1, Santa Fe Institute Studies in the science of complexity, Addison-Wesley. 1991.
6. M. Levický: *Neural Networks in the analysis and the document classification*, Diploma Thesis, P. J. Šafárik University, Košice, 2002.
7. Lundstedt, H., Wintoft, P.: *Prediction of geomagnetic storms from solar wind data with the use of a neural network.* Ann. Geophysicae 12, EGS-Springer-Verlag, 1994, p.19-24.
8. MacKay, D.J.C.: *Bayesian Methods for Neural Networks: Theory and Applications.* Neural Network Summer School, 1995.
9. MacKay, D.J.C.: *A practical Bayesian Framework for Backprop Networks.* Neural Computation 4, p. 448-472.
10. Müller, P., Insua, D.R.: *Issues in Bayesian Analysis of Neural Network Model.* Neural Computation 10, p. 749-770.
11. Neal, R. M.: *Probabilistic Inference Using Markov Chain Monte Carlo Methods.* Technical report CRG-TR-93-1, University of Toronto, 1993.
12. Neal, R. M.: *Bayesian Training of Backpropagation Networks by the Hybrid Monte Carlo Method* Technical report CRG-TR-92-1, University of Toronto, 1992.
13. Schlesinger, M. I., Hlaváč, V. : *Deset přednášek z teorie statistického a strukturního rozpoznávaní.* ČVUT, Praha 1999.