8th. World Congress on Computational Mechanics (WCCM8)
5th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008)
June 30 –July 5, 2008
Venice, Italy

BAYESIAN NEURAL NETWORKS IN THE REGRESSION ANALYSIS OF STRUCTURAL MECHANICS PROBLEMS

***Zenon Waszczyszyn[1,2], Marek Słoński[2], Bartosz Miller[1] and Grzegorz Piątkowski[1]**

[1] Rzeszow University of Technology
ul. W. Pola 2, 35-959 Rzeszow, Poland
zewasz, bmiller, pgrzes@prz.edu.pl

[2] Cracow University of Technology
ul. Warszawska 24, 31-155 Krakow, Poland
zenwa, mslonski@L5.pk.edu.pl

Key Words: *Bayesian Neural Networks, Standard Neural Networks, Regression.*

ABSTRACT

The Bayesian inference and feed-forward neural networks are adequate tools in the regression analysis, defined as a mapping of input onto output variables. Joining Bayesian background and application of Standard Neural Networks (SNNs) gives the Bayesian Neural Networks (BNNs). BNNs were initiated at the beginning of 1990s, especially due to MacKay and Neal (cf. references in [1]) who stated that the development of computer hard- and software was a good base to consume a great potential of the Bayesian approaches.

BNNs are probabilistic networks. In the paper we discuss a special case of BNNs with deterministic input $\mathbf{x} \in R^D$ and a scalar random output $y(\mathbf{x};\mathbf{w})$, where $\mathbf{w}$ is a random vector of weights. BNNs learning and predictions are based on the Bayes' theorem:

$$p(\mathbf{w}|\mathbf{X},\mathbf{t},\alpha,\beta) = \frac{p(\mathbf{t}|\mathbf{X},\mathbf{w},\beta)p(\mathbf{w}|\alpha)}{p(\mathbf{t}|\mathbf{X},\alpha,\beta)}, \tag{1}$$

which can be verbalized as *posterior = likelihood $\times$ prior / evidence*. The evidence is an integral in which the integrand is taken over all values of the vector $\mathbf{w} = \{w_i\}_{i=1}^W$:

$$p(\mathbf{t}|\mathbf{X},\alpha,\beta) = \int_{R^W} p(\mathbf{t}|\mathbf{X},\mathbf{w},\beta)p(\mathbf{w}|\alpha)d\mathbf{w}. \tag{2}$$

In (1) and (2) the following notation is used: p – probability distribution; $\mathcal{D} = \{\mathbf{X},\mathbf{t}\}$ – dataset with input and output sets $\mathbf{X} = \{\mathbf{x}^n\}_{n=1}^N$, $\mathbf{t} = \{\mathbf{t}^n\}_{n=1}^N$; $\alpha,\beta = 1/\sigma^2$ – hyperparameters.

The generalized error function corresponding to (1) and (2) is defined in the form:

$$F(\mathbf{w}) = \beta E_{\mathcal{D}}(\mathbf{w}) + \alpha E_W(\mathbf{w}) = \frac{\beta}{2}\sum_{n=1}^{N}\left\{t^n - y(\mathbf{x}^n;\mathbf{w})\right\}^2 + \frac{\alpha}{2}\sum_{i=1}^{W} w_i^2, \tag{3}$$

where the term $\alpha E_W(\mathbf{w})$ plays a role of regularization function.

Different Bayesian approaches, semi-probabilistic and true probabilistic, can be formulated applying Eqs (1-3). Four of them, which use the weight vector $\mathbf{w}$, are listed in Table 1. The fifth approach,

Table 1: ANN framework, learning quantities and prediction functions

ANN and Bayesian frameworks	Learned quantities	Prediction functions
1. SNN or BNN-ML	$\mathbf{w}_{\text{ML}}$	$y(\mathbf{x}^{\text{N+1}}; \mathbf{w}_{\text{ML}})$
2. BNN-MAP	$p(\mathbf{w}\|\mathbf{X}, \mathbf{t}, \alpha_{\text{in}}, \sigma^2_{\text{in}})$	$p(t^{\text{N+1}}\|\mathbf{x}^{\text{N+1}}; \mathbf{w}_{\text{MAP}}, \sigma^2_{\text{in}})$
3. S-BNN (Simple BNN)	$p(\mathbf{w}\|\mathbf{X}, \mathbf{t}, \alpha_{\text{eff}}, \sigma^2_{\text{eff}})$	$p(t^{\text{N+1}}\|\mathbf{x}^{\text{N+1}}, \mathbf{t}, \alpha_{\text{eff}}, \sigma^2_{\text{eff}})$
4. T-BNN (True BNN)	$p(\mathbf{w}\|\mathbf{X}, \mathbf{t}, \alpha, \sigma^2)$	$p(t^{\text{N+1}}\|\mathbf{x}^{\text{N+1}}, \mathbf{t}, \alpha, \sigma^2)$
5. GP (Gaussian Process)	$C_{\text{N}}^{\text{-1}}(\mathbf{X})$, σ^2, $\boldsymbol{\theta}$	$p(t^{\text{N+1}}\|\mathbf{x}^{\text{N+1}}, \mathbf{t}, C_{\text{N+1}}^{\text{-1}}, \boldsymbol{\theta}, \sigma^2)$ $= \mathcal{N}(t^{\text{N+1}}\|m_{\text{N+1}}(\mathbf{x}^{\text{N+1}}, C_{\text{N+1}}^{\text{-1}}, \mathbf{t}, \boldsymbol{\theta}), \sigma^2)$

called the Gaussian process method, is based on the covariance matrix of input data $C_N^{-1}(\mathbf{X})$, where $\mathbf{X} = \{\mathbf{x}^n\}_{n=1}^N$. Another set of parameters $\boldsymbol{\theta} = \{\theta_j\}_{j=1}^J$ is applied for the kernel functions formulation.

The ML (Maximum Likelihood) method corresponds to the computation of min $E_{\mathcal{D}}(\mathbf{w})$, i.e. it leads to the formulation of SNNs. The MAP (Maximum A Posteriori) method is based on the introduction of regularization function $E_W(\mathbf{w})$. This method needs to fix in advanced the deterministic hyperparameters α_{in} and $\beta_{\text{in}} = 1/\sigma^2_{\text{in}}$. In simple BNN optimal values of hyperparameters α_{eff} and β_{eff} are computed in the iterative way. In True BNN the hyperparameters are assumed to be random variables.

The MML (Maximum of Marginal Likelihood) is computed for $\ln p(\mathbf{t}|\mathbf{X}, \alpha, \beta)$, i.e. for the $\ln$ of evidence (3). MML is used to the estimation of optimal values of hyperparameters or design parameters of NNs criterion.

In the paper the methods listed in Table 1 are applied to the analysis of four problems: 1) prediction of response spectra for paraseismic excitations [2], 2) estimation of HPC concretes [2], 3) identification of loads affected yielding of an elastic-plastic beam [4], 4) identification of mass attachment to steel and aluminium strips [5].

It was proved that the MAP-BNN and Simple BNN give quite satisfactory accuracy to the solutions of the discussed problems. The BNN, and especially GP, are based on new no iterative paradigms instead of the error minimization paradigm explored in SSNs. Simple-BNN and True-BNN are supported on the principle of parameter marginalization and GP uses the input covariance matrix.

The financial support of the Polish Grant MNiSW No. 506 1814 33 is highly appreciated.

REFERENCES

[1] C.M. Bishop. *Pattern Recognition and Machine Learning*, Springer, 2006.

[2] Z. Waszczyszyn and M. Słoński. "Bayesian neural networks for prediction of response spectra". *Foundations of Civil and Environmental Engineering*, Publ. House of Poznan Univ. of Technology, Poland No. 7, 343–361, 2006.

[3] M. Słoński. "Strength prediction using Bayesian neural networks". *Computer Assisted Mech. Eng. Sci.*, Vol. **14**, 345-352, 2007.

[4] B. Miller, G. Piątkowski and L. Ziemiański. "Beam yielding load identification by neural networks". *Computer Assisted Mech. Eng. Sci.*, Vol. **6**, 449-468, 1999.

[5] G. Piątkowski and L. Ziemiański. "The solution of an inverse problem in plates by means of artificial neural networks". In: L. Rutkowski et al. (Eds), *Artificial Intelligence and Soft Computing ICAISC 2004*, LNAI 3070, 1087-1092, Springer, 2004.