Multi-layered Bayesian Neural Networks for Simulation and Prediction Stress-Strain Time Series

Agnieszka Krok

Tadeusz Kościuszko Cracow University of Technology, Cracow, Poland

Abstract—The aim of the paper is to investigate the differences as far as the numerical accuracy is concerned between feed-forward layered Artificial Neural Networks (ANN) learned by means of Kalman filtering (KF) and ANN learned by means of the evidence procedure for Bayesian technique. The stress-strain experimental time series for concrete hysteresis loops obtained by the experiment of cyclic loading is presented as considered example.

Keywords—Bayesian Neural Networks, Kalman filtering.

1. Introduction

Kalman filtering and Bayesian learning methods are based on the same assumption of modeling neural networks as the combination of random variables. In both cases, ANNs are layered, feed forward, learned by supervised method with a teacher. Learning set and testing set are considered. Learning process is based on methods known from probability theory and statistical analysis: Kalman filtering and Bayes theorem. The aim of the paper is to make a comparison of the two approaches from the same family techniques.

As far as the network architecture, the most common multi-layer perception ANN were considered. Two hidden layers were used for their ability to model nonlinear functions, according to the universal approximation theorem [1]. For a comparison purpose the same architecture of ANNs were considered. It results in the same number of ANN weights to be found. The Mean Squared Error (MSE) for learning and testing set was considered as the measure of learning efficiency. In addition, the qualitative criteria was examined. The shape of modeled time series is calculated by ANNs according to the experimental data. Possibilities of easy designing the network shape (number of neurons in each layer), the number of parameters that control the process of selection model and the time for implementing both methods were also verified.

2. Motivation and Related Background

Bayesian Neural Networks (BNN) are constructed as layered, feed forward networks learned by supervised methods that involves Bayes theorem [2]. The following four steps are considered:

1. make predictions including error bars for new input data;

2. estimate the weight parameters and their uncertainties.

3. estimate the weight decay parameters and their uncertainties;

4. repeat steps 2–3 with different initial conditions and different network architectures. Select the architecture and w-minimum with highest evidence. Optionally select a committee to reflect the uncertainty on this level [3].

The BBN were recently used for the case problems including a regression, a classification, and an inverse problem. The Internet traffic classification [4], modeling protein family [5], concrete quality estimation problem [6], assessment of lean manufacturing effect on business performance [7], medicine diagnoses finding [8], forecasting performances over the weekly sales of a Finance Magazine [9], image skin segmentation [10], classification of file system activities [11], analyzing weather data [12], classifying segmented outdoor images [13], were analyzed.

The traditional approach to the hysteresis modeling assumes using differential equation models that involves the parameters that are specific to the modeled material: Jiles–Atherton model [14], Ylinen's Model [15], Takács model [16], Prandtl–Ishlinskii model [17]. In most cases, the models are in the form of piece-wise functions different for the particular branches of the hysteresis [15], [18].

Also soft methods was considered: neural networks in the form of multi-layer perceptions, learned by the back-propagation algorithm for supervised training [19], [20], or the Levenberg-Marquardt algorithm [21]–[23] .

3. Kalman Filtering as the ANN Learning Technique

The KF as a method was adapted to ANN nonlinear models [24], as the learning technique and developed exces-

sively using selected nodes learning and as far as pruning the ANN is concerned [25].

The basic KF learning method – Node Decoupled Extended Kalman Filter (NDEKF) consists of process equation and measurement equation. After modifications they may be adopted to learn standard Multi-layered ANN [26]:

$$\mathbf{w}_{k+1}^i = \mathbf{w}_k^i + \omega_k^i, \tag{1}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{w}_k, \mathbf{x}_k) + v_k, \tag{2}$$

where: k – discrete pseudo-time parameter, i – the number of neuron in ANN- w_{k+1}^i for $i = 1, 2, \dots, W$ – state vector corresponding to the set of synaptic weights and biases, $\mathbf{h}$ – non-linear vector-function of input-output relation, $\mathbf{x}/\mathbf{y}$ – input/output vectors, ω_k^i, v_k – Gaussian process and measurement noises with mean and covariance matrices defined by:

$$\mathbf{E}(v_k) = \mathbf{E}(\omega_k^i) = 0, \tag{3}$$

$$\mathbf{E}(\omega_k^i * \omega_l^{iT}) = \mathbf{Q}_k^i \delta_{k,l}, \tag{4}$$

$$\mathbf{E}(v_k v_l^T) = \mathbf{R}_k \delta_{k,l}, \tag{5}$$

where: $\delta_{k,l} = 1$ for $k = l$, $\delta_{k,l} = 0$ for k not equal l.

The NDEKF algorithm assumes splitting state vector into groups. The single group was assigned to single neuron (nodes $i = 1, 2, \dots, N$). Similar to all teacher based learning techniques the change of $\mathbf{w}^i$ is made during the presentation of each k-th learning pattern:

$$\mathbf{K}_k^i = \mathbf{P}_k \mathbf{H}_k \Big[\sum_{j=1}^{g} (\mathbf{H}^j)_k^T \mathbf{P}_k^j \mathbf{H}_k^j + \mathbf{R}_k \Big]^{-1}, \tag{6}$$

$$\mathbf{w}_{k+1}^i = t \mathbf{w}_k^i + \mathbf{K}_k^i \xi_k, \tag{7}$$

$$\mathbf{P}_{k+1}^i = (\mathbf{I} - \mathbf{K}_k^i (\mathbf{H}_k^i)^T) \mathbf{P}_k^i + \mathbf{Q}_k^i, \tag{8}$$

where: $\mathbf{K}_k^i$ – Kalman gain matrix, $\mathbf{P}_k^i$ – approximate error covariance matrix, g – the number of ANN nodes (neurons), $\xi_k = \mathbf{y}_k - \widehat{\mathbf{y}}_k$ – error vector, with the target vector $\mathbf{y}_k$ for the k-th presentation of a training pattern, $\widehat{\mathbf{y}}_k$ – output vector given by ANN.

$\mathbf{H}$ is the matrix of current linearization of Eq. (2)

$$\mathbf{H}_k^i = \frac{\partial \mathbf{h}}{\partial \mathbf{w}^i}. \tag{9}$$

The considered parameters for the Gaussian noise adopted are e.g. in the form:

$$\mathbf{Q}_k^i = \alpha_1 \cdot e^{\frac{s-1}{\beta_1}} \cdot I, \tag{10}$$

$$\mathbf{R}_k = \alpha_2 \cdot e^{\frac{s-1}{\beta_2}} \cdot I, \tag{11}$$

where: I – identity matrix which dimension depends on the state vector dimension in ANN, s – the number of learning epoch, and α_1, α_2, β_1, β_2 are real numbers.

4. Bayesian Neural Networks

The ANN is formulated as [27]:

$$t = y(\mathbf{x}; \mathbf{w}) + \varepsilon, \tag{12}$$

where y is the non-linear vector function of input-output relation, ε – noise incorporated to the model, $\mathbf{w}$ – vector of ANN weights interpreted as the random variables, t is the target output variable interpreted as a random variable. Next, the

$$p(\mathbf{w}) \tag{13}$$

is the prior broad probability distribution of the $\mathbf{w}$, and representing little knowledge about values of $\mathbf{w}$:

$$p(\mathbf{w}|D) = \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)} \tag{14}$$

is the posterior probability distribution of the $\mathbf{w}$. It representing knowledge about values of $\mathbf{w}$ after data set D is presented to the network, $p(D|\mathbf{w})$ is the data set likelihood.

$$p(t|\mathbf{x}^*, D) = \int p(t|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|D)\mathrm{d}\mathbf{w} \tag{15}$$

is the predicted distribution of the ANN output y for the particular input vector $\mathbf{x}^*$;

$$E(t|\mathbf{x}^*, D) = \int t p(t|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|D)\mathrm{d}\mathbf{w} \tag{16}$$

is the point prediction of the ANN output t for the particular input vector $\mathbf{x}^*$. The requirement for small values of $\mathbf{w}$ suggests a Gaussian prior distribution the the ANN weights

$$p(\mathbf{w}) = \frac{1}{Z_W(\alpha)} \cdot e^{\frac{-\alpha\|\mathbf{w}\|^2}{2}}, \tag{17}$$

where α represents the inverse variance of the distribution of $\mathbf{w}$ and

$$\alpha = \frac{1}{D^2(\mathbf{w})}. \tag{18}$$

$Z_W(\alpha)$ is the normalization constant $Z_W(\alpha) = \left(\frac{2\pi}{\alpha}\right)^{\frac{W}{2}}$ where W is the number of ANN weights.

It is assumed the target data is given by the Gaussian distribution with zero mean and the constant inverse variance β, so the data set likelihood $p(D|\mathbf{w})$ is

$$p(D|\mathbf{w}) = \frac{1}{Z_D(\beta)} \cdot e^{\frac{-\beta\sum_{n=1}^{N}\|t(\mathbf{x}^n;\mathbf{w})-t^n\|^2}{2}}, \tag{19}$$

where β represents the inverse variance of the ε distribution defined as:

$$\beta = \frac{1}{D^2(\varepsilon)}. \tag{20}$$

The $Z_D(\alpha)$ is the normalization constant given by $Z_D(\alpha) = \left(\frac{2\pi}{\beta}\right)^{\frac{N}{2}}$, where N is the number of data point in D. Then

assuming α, β are random variables with their own probability distributions:

$$p(t|\mathbf{x}^*,D) = \iint p(t|\mathbf{x}^*,\mathbf{w},\beta)p(\mathbf{w}|\alpha,\beta,D)p(\alpha,\beta|D)\mathrm{d}\alpha\mathrm{d}\beta\,, \tag{21}$$

$$p(t|\mathbf{x}^*,\mathbf{w},\beta) = N\left(t|t(\mathbf{x}^*,\mathbf{w}),\beta^{-1}\right), \tag{22}$$

$$\ln p(\mathbf{w}|\alpha,\beta,D) = p(D|\mathbf{w})\,. \tag{23}$$

5. Evidence Procedure for Bayesian Neural Networks

The evidence procedure was used as an iterative algorithm for determining optimal weights and hyper parameters during Bayesian learning of the ANN [28].

Presented method is based on the approximating the hyper parameters posterior distribution with its value at the most probable (MP) values

$$p(\mathbf{w}|D) \sim \int p(y|\mathbf{w},\beta_{MP})p(w|\alpha_{MP},\beta_{MP},D)\mathrm{d}w\,. \tag{24}$$

To find the MP values of α and β one have to find the maximum of:

$$p(\alpha,\beta|D) = \frac{p(D|\alpha,\beta)p(\alpha,\beta)}{p(D)}\,. \tag{25}$$

In the further calculation $p(\alpha,\beta)$ is assumed to be uniform and ignored. Maximizing $p(D|\alpha,\beta)$ equals finding the maximum of:

$$p(D|\alpha,\beta) = \int p(D|\mathbf{w},\beta)p(\mathbf{w}|\alpha)\mathrm{d}\mathbf{w} \tag{26}$$

$$p(D|\alpha,\beta) = \frac{1}{Z_D(\beta)}\frac{1}{Z_W(\alpha)}\int \mathrm{e}^{-S(\mathbf{w})}\mathrm{d}\mathbf{w}\,, \tag{27}$$

where:

$$S(\mathbf{w}) = \frac{\beta}{2}\sum_{i=1}^{N}(y(\mathbf{x}^n;\mathbf{w})-t^n)^2 + \frac{\alpha}{2}\sum_{i=1}^{W}\mathbf{w}_i^2 = \beta E_D + \alpha E_W \tag{28}$$

is the misfit function. The t^n and $y(\mathbf{x}^n;\mathbf{w})$ are the target and computed output values for n-th pattern scaled to the interval $0\dots1$, $\mathbf{w} = w_i,\dots,w_W$ is the vector of ANN weights. By computing the logarithm of the Eq. (28) and the partial derivative with respect to α one can obtain:

$$\alpha = \frac{W - \sum_{i=1}^{W}\frac{\alpha}{\lambda_i+\alpha}}{2E_W(w_{MP})} = \frac{\gamma}{2E_W(w_{MP})}\,, \tag{29}$$

where $\mathbf{w} = \mathbf{w}_{MP}$, λ_i is the i-th eigenvalue of the Hessian matrix H:

$$H = \nabla\nabla E_D\,, \tag{30}$$

$$\gamma = \sum_{i=1}^{W}\frac{\lambda_i}{\lambda_i+\alpha}\,. \tag{31}$$

This implicit solution is used for the iterative procedure: after setting initial values of α that is used to find $\mathbf{w}_{MP}$ and $S_W(\mathbf{w}_{MP})$ the α is re-estimated according to [6]:

$$\alpha = \frac{\gamma}{2E_D(\mathbf{w}_{MP})}\,, \tag{32}$$

where $\mathbf{w} = \mathbf{w}_{MP}$.

By computing the logarithm of the objective function and the partial derivative with respect to β one can obtain:

$$\beta = \frac{N-\gamma}{2E_D(\mathbf{w}_{MP})}\,. \tag{33}$$

The procedure scheme can be written in following steps:

1. Choose the initial values of hyper parameters α and β, initial ANN weights drown from prior distribution given by α,

2. Train the ANN with Scaled Conjugate Gradients Algorithm (SCGA) [1], to minimize negative log probability of weight posterior probability misfit function $S(\mathbf{w})$, where $N = L$ is the number of learning patterns to find $\mathbf{w}_{MP}$,

3. Hyper parameters re-estimate:

$$\alpha(new) = \frac{\gamma}{2E_W(\mathbf{w}_{MP})}\,, \tag{34}$$

and

$$\beta(new) = \frac{N-\gamma}{2E_D(\mathbf{w}_{MP})}\,, \tag{35}$$

4. Update the log evidence

$$p(D|\alpha(new),\beta(new),\gamma)\,, \tag{36}$$

5. Repeat steps 2–4 until convergence.

Number of training cycles is the steps number during SCGA performance, number of inner loops is the number of updating $\alpha \Rightarrow \alpha(new)$, $\beta \Rightarrow \beta(new)$, number of outer loops is the number of repeating the $\mathbf{w}$ re-estimation.

6. Experimental Results for Simulation and Prediction of Steel Hysteresis Loops

6.1. Experimental Data

Many time series for simulation and prediction stress-strain relation was considered for steel and concrete. In this paper, the one specific numerical result would be presented. The main tendency and numerical accuracy during modeling the rest of the data was similar. All the tested examples may be found in [29].

Presented data set was the result of uniaxial low cyclic tension-compression test for stainless steel AISI 316L [24], see Fig 1.
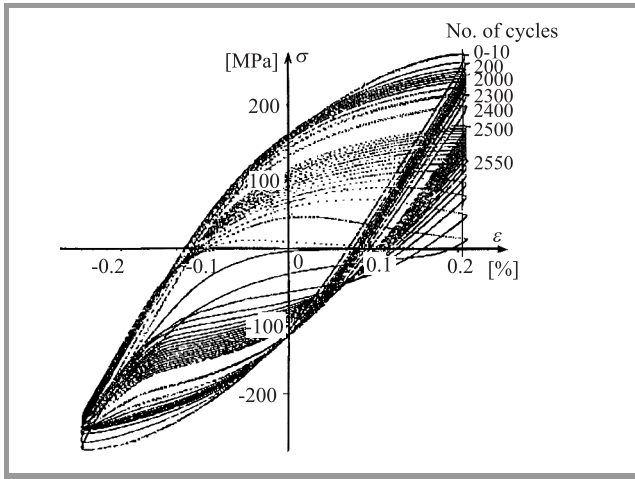
Fig. 1. Experimental data on σ-ε plane.
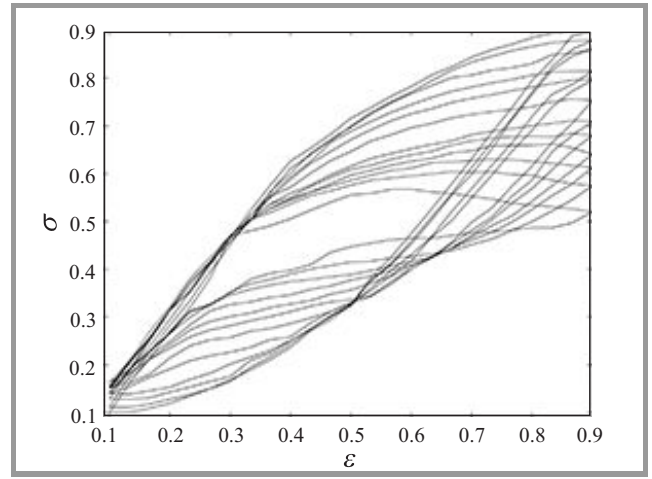

Fig. 2. Data on σ-ε plane for ANN learning and testing.

The aim of the conducted neural analysis was to simulate first part of experiment and to predict the phase before the material damage. Presented time series is based on the inner processes inside the material and is self-dependent because each next state of the material depends on all the previous states during experiment. During learning stage, time series simulation was performed, and whole testing stage time series prediction was made.

The twelve representative loops were selected for the neural computation. The loops were selected to the constant maximal stain value inclination. The first and second loop selected for neural analysis were taken from range of 0–2000 experimental loops, three next loops from range of 2000–24000 experimental loops, the remaining 7 loops from range, were the changes in stress and stein values were the largest. Each loop was discretized on 49 $(\sigma(k), \varepsilon(k))$ points for:

$$\varepsilon(k) = 0.2 - (k-1) \cdot \Delta_1 \varepsilon, \tag{37}$$

for $k = 1, 2, \ldots, 25$ and

$$\varepsilon(k) = -0.26 + (k-25) \cdot \Delta_2 \varepsilon, \tag{38}$$

for $k = 26, \ldots, 49$ with

$$\Delta_1 \varepsilon(k) = 0.2/25 = 0.008\%,$$
$$\Delta_2 \varepsilon(k) = 0.26/25 = 0.0104\%. \tag{39}$$

Adopted discretization results in $P = 12 \cdot 49 = 588$ data points for learning and testing. Given data ware scaled to the interval $0.1 \ldots 0.9$ for the ANN processing, see Fig. 2. The first nine loops containing $L = 949$ patterns for the $k = 1, \ldots, 441$ were used for the learning and $T = 588 - 441 = 147 = 3 \cdot 49$ patterns form final loops for $k = 442, \ldots, 588$ for testing.

The input vector **x** consists of scaled marker of current pattern $k/587$, scaled marker of current pattern number inside each loop separately $\mathrm{mod}(k,49)/49$ and the previous σ value given by ANN, marked $\sigma_{ANN}(k-1)$ [29]:

$$\mathbf{x}(k) = \left[\sigma_{ANN}(k-1), k/587, \quad \mathrm{mod}\,(k,49)/49 \right]. \tag{40}$$

The output vector for k-th input takes the form $\sigma(k)$.

7. Comparison of Neural Networks Learned by Bayesian Evidence Procedure Accuracy with KF Models

The basic KF model is simple, that makes is easy to implement. However, the model does not have many parameters to exploit. One may split ANN differently (layers not nodes) [26], or use different parameters for the Gaussian noise, α_1, α_2, β_1, β_2 in Eqs. (10)–(11). Also different noise models instead of Eqs. (10)–(11) may be adopted. First possibility enlarges excessively the model dimensions given by Eqs. (1)–(5) makes the model very time consuming. This second option does not influence the computational results much, see [29]. The possibility of automation of setting the network structure without the stopping learning process is very valuable. One may start from the large network and switch off the some of the network nodes during learning (pruning). The author's model development proved that ANN learned by KF may be successfully designed by pruning, and the approximate error covariance P, matrix may be used to more accurate learning, see Eq. (8) [29], [30]. KF learning technique was stated to be very promising tool as far as time series simulation and prediction [29], [31]–[34].

In comparison to KF, Bayesian learning technique is much more complicated to implement but have many more free parameters to change to adjust the model. It allows the better flexibility, but incorporates the problem of searching the parameter space for a suboptimal solution. For example the changing of characteristics of hiper parameter distribution significantly influence the model, see Eqs. (19)–(36). During pruning process the model given by Eqs. (24)–(36) have to be reformulated and there is a need to compute prior for sparsely connected ANN [35]. The method also depends on the SCGA performance that should be implemented and used correctly, whereas the in KF no additional high-level tool is needed.

The simulation for Bayesian learning process was made using modified Netlab Software [1]. Simulation for KF

was made for software developed fully by author in Matlab environment. The same ANN architecture was considered, and the same networks input vectors were used, see Eq. (40).

For Bayesian learning the initial prior hyper parameter $\alpha = 0.01$, initial noise hyper parameter $\beta = 50$, number of training cycles in inner loop 500, number of inner loops 3, number of outer loops 3 was found as the suboptimal solution for the given data set. The results of ANN simulation and prediction are presented in Figs. 3 and 4.
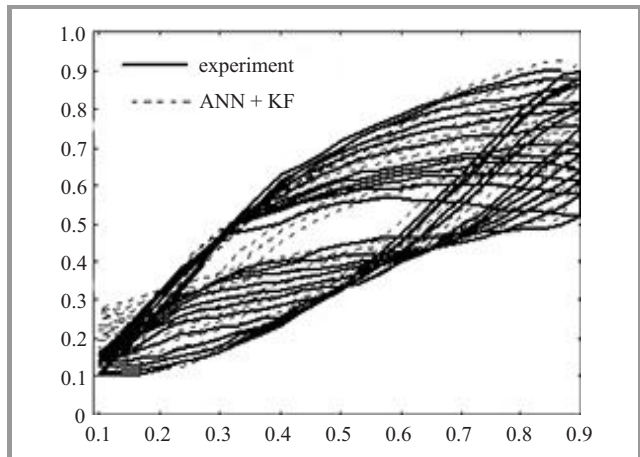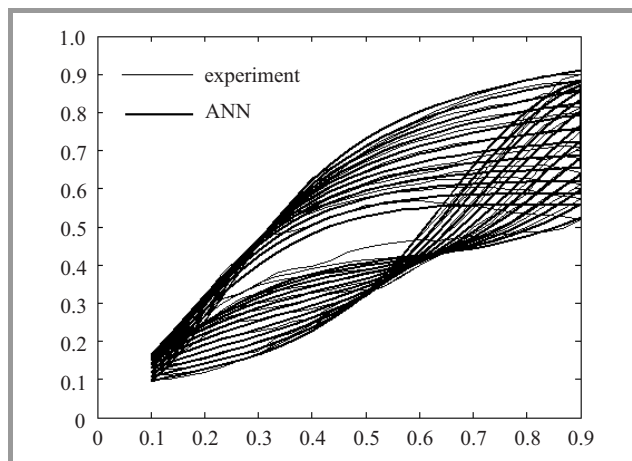

Fig. 3. Experimental vs. simulated by KF neural network hysteresis loops, 9 first loops for learning and remaining for testing on σ-ε plane.
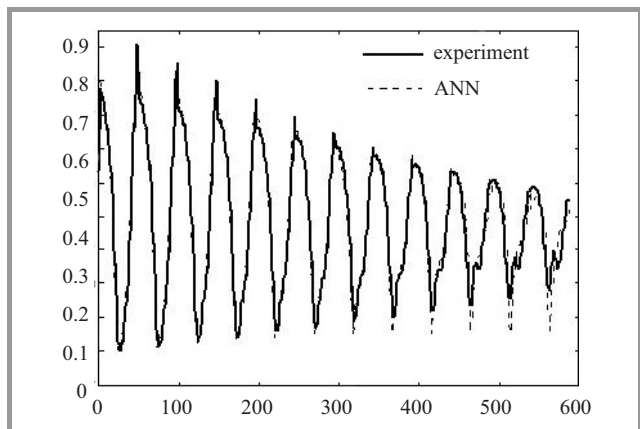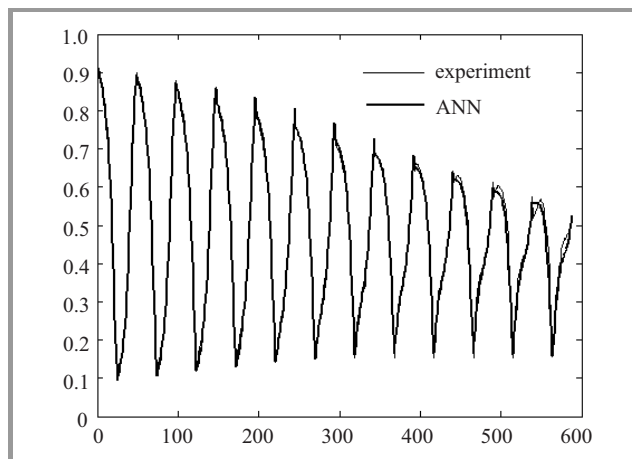

Fig. 4. Experimental vs. simulated by KF neural network hysteresis loops, 9 first loops for learning and remaining for testing on σ – no. of pattern plane.

For KF learning method with 1000 epochs and $\alpha_1 = 0.001$, $\alpha_2 = 7$, $\beta_1 = 50$, $\beta_2 = 50$ was adopted. The results of ANN simulation and prediction are presented in Figs. 5 and 6.

The presented method of ANNs learning enables simulation of the hysteresis loops with a very high accuracy using ANN of small number of parameters (first hidden layer 6 nodes, second hidden layer 6 nodes). ANN predicts the behavior of the considered material during the final step of loading and unloading properly.

Fig. 5. Experimental vs. simulated by Bayesian neural network hysteresis loops, 9 first loops for learning and remaining for testing on σ-ε plane.

Fig. 6. Experimental vs. simulated by Bayesian neural network hysteresis loops, 9 first loops for learning and remaining for testing on σ – no. of pattern plane.

Comparison with the results obtained by modeling with damage mechanics was made.

In [36] two theoretical models for stress-strain relation for the considered material were proposed. They were based on uniaxial nonlinear elasto-plastic Ylinen model [37]. The relationship between considered quantities had to be separated for two phases, two damage parameters was needed to obtain the proposed models. The models took into consideration the discrete process of opening and close of the cracks, see Fig. 7 for model A, and continuous process of opening and close of the cracks, see Fig. 8 for model B.

Proposed model A is inconsistent with the experiment as far as continuity of first order derivative of $\partial\sigma/\partial\varepsilon$ is considered. Both models are incorrect concerning negative values of stress strain close to their minimal values.

Simulating the stress-strain relation, using damage mechanics equations, require the values of material constants to be properly chosen for the theoretical model and calibration
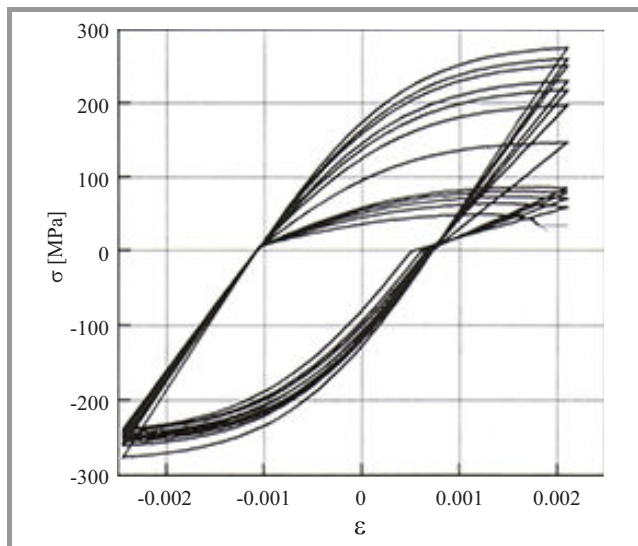
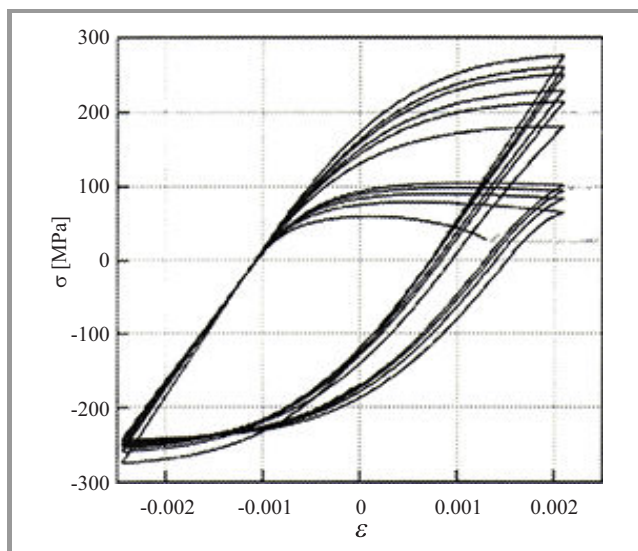**Fig. 7.** Hysteresis loops for model A, σ-ε plane.

Fig. 8. Hysteresis loops for model B, σ-ε plane.

as far as model parameters are considered. During ANN modeling none of this component is necessary. The model is based only on the experimental data and simple markers of the experiment phase. Information about mechanical effects is independent of arbitrarily chosen mechanical model and imposed model parameters.

KF and Bayesian ANN modeling incorporates no prior knowledge about mechanical model.

8. Conclusions

The evidence procedure for the Bayesian Neural Networks enables hysteresis loops simulation with a very high accuracy as far as results quality is concerned. The model fits the data better than model based on KF and it is superior to known mechanical models. Presented Bayesian model has three basic parameters to set: Bayesian learning initial

prior hyper parameter α, initial noise hyper parameter β, and number of training cycles in inner loop k.

Presented KF model has five parameters. The coefficients for Gaussian noise incorporated into the model (four values), and number of epochs of learning.

The influence of the Bayes model parameters is meaningful. It enables the model to adjust to the data better, but makes searching for optimal parameter set more demanding. For some set of parameters model calibrated to the data is incorrect or significantly worse.

The influence of the KF model parameters is hardly a significant. Different parameter setting leads to slightly longer teaching. The most significant parameter is the number of epochs of learning.

For both models, there is also the need for searching initial weight space. For different initial weight sets the results of simulation and prediction differs slightly.

The automatic setting of neural network shape, i.e. number of neurons in each hidden layer, by pruning procedure during learning process is much easier to implement in KF model. To adjust both models to different kind of data the distributions incorporated into models may be changed. However, any change in Bayesian theoretical model has more severe consequences into computational process, because all Eqs. (12)–(36) have to be changed.

Acknowledgements

References

[1] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice Hall, 1999.

[2] T. Bayes, "An Essay towards solving a problem in the doctrine of chances", *Philosoph. Trans. of the Royal Society of London*, vol. 53, pp. 370–418, 1763.

[3] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[4] T. Auld, A. W. Moore, and S. F. Gull, "Bayesian neural networks for Internet traffic classification", *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 223–239, 2007.

[5] D. MacKay, "Bayesian neural networks and density networks", *Nuclear Instrum. and Methods in Phys. Res. Section A*, vol. 354, no. 1, pp. 73–80, 1995.

[6] J. Lampinen and A. Vehtari," Bayesian approach for neural networks and case studies", *Neural Netw.*, vol. 14, no. 3, pp. 257–274, 2001.

[7] G. Büyüközkan, G. Kayakutlu, and I. S. Karakadılar, "Assessment of lean manufacturing effect on business performance using Bayesian Belief Networks", *Expert Syst. with Applications*, vol. 42, no. 19, pp. 6539–6551, 2015.

[8] I. Kononenko, "Bayesian neural networks", *Biolog. Cybernetics*, vol. 61, pp. 361–370, 1989.

[9] O. Kocadağlı and B. Aşıkgil, "Nonlinear time series forecasting with Bayesian neural networks", *Expert Syst. with Applications*, vol. 41, no. 15, pp. 6596–6610, 2014.

[10] A. A. Zaidan *et al.*, "Image skin segmentation based on multi-agent learning Bayesian and neural network", *Engin. Appl. of Artif. Intell.*, vol. 32, pp. 136–150, 2014.

[11] M. N. A. Khan, "Performance analysis of Bayesian networks and neural networks in classification of file system activities", *Comp. & Secur.*, vol. 31, no. 4, pp. 391–401, 2012.

[12] H. S. Hippert and J. W. Taylor, "An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting", *Neural Netw.*, vol. 23, no. 3, pp. 386–395, 2010.

[13] Ch. K. I. Williams and F. Vivarellia, "Comparing Bayesian neural network algorithms for classifying segmented outdoor images", *Neural Netw.*, vol. 14, no. 4–5, pp. 427–437, 2001.

[14] M. Hamimid, S. M. Mimoune, and M. Feliachi, "Minor hysteresis loops model based on exponential parameters scaling of the modified Jiles–Atherton model", *Physica B: Condensed Matter*, vol. 407, no. 13, pp. 2438–2441, 2012.

[15] A. Ganczarski and L. Barwacz, "Concept of continuous damage deactivation in modelling of low cycle fatigue", *in Proc. Int. Conf. on Fracture ICF11 2005*, Turin, Italy, 2005.

[16] K. Chwastek, "Modelling hysteresis loops in thick steel sheet with the dynamic Takács model", *Physica B: Condensed Matter*, vol. 407, no. 17, pp. 3632–3634, 2012.

[17] M. Al Janaideh, "A time-dependent stop operator for modeling a class of singular hysteresis loops in a piezoceramic actuator", *Physica B: Condensed Matter*, vol. 413, pp. 100–104, 2013.

[18] A. P. S. Baghel, A. Gupta, K. Chwastek, and S. V. Kulkarni, "Comprehensive modelling of dynamic hysteresis loops in the rolling and transverse directions for transformer laminations", *Physica B: Condensed Matter*, vol. 462, pp. 86–92, 2015.

[19] I. Kucuk, "Prediction of hysteresis loop in magnetic cores using neural network and genetic algorithm", *J. Magnetism and Magnetic Materials*, vol. 305, no. 2, pp. 423–427, 2006.

[20] R. Dong, Y. Tan, H. Chen, and Y. Xie, "A neural networks based model for rate-dependent hysteresis for piezoceramic actuators", *Sensors and Actuators A: Physical*, vol. 143, no. 2, pp. 370–376, 2008.

[21] A. Nouicer, E. Nouicer, and F. Mouloudc, "A neural network for incorporating the thermal effect on the magnetic hysteresis of the 3F3 material using the Jiles–Atherton model", *J. Magnetism and Magnetic Materials*, vol. 373, pp. 240–243, 2015.

[22] V. Wolfs and P. Willems, "Development of discharge-stage curves affected by hysteresis using time varying models, model trees and neural networks", *Environ. Model. & Softw.*, vol. 55, pp. 107–119, 2014.

[23] X. Zhang, Y. Tan, and M. Su, "Modeling of hysteresis in piezoelectric actuators using neural networks", *Mechan. Syst. and Sig. Process.*, vol. 23, no. 8, pp. 2699–2711, 2009.

[24] R. E. Kalman, "A new approach to linear filtering and prediction problems", *Trans. ASME – J. of Basic Engin.*, vol. 82, series D, pp. 35-45, 1960.

[25] L. Prechelt, "Adaptive parameter pruning in neural networks", ICSI Tech. Rep. TR-95-009, Berkeley, CA, USA, 1995.

[26] S. Haykin, Ed., *Kalman Filtering and Neural Networks*. New York: Wiley, 2001.

[27] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[28] R. M. Neal, "Bayesian Learning for Neural Networks", *Lecture Notes in Statistics*, vol. 118. Springer, 1996.

[29] A. Krok, "Analysis of mechanics of structures and material problems applying artificial neural networks learnt by means of Kalman fltering", Ph.D. thesis, Institute of Computer Mhetods in Civil Engineering, Cracov University of Techology, 2007 (in Polish).

[30] A. Krok, "The development of Kalman filter learning technique for artificial neural networks", *J. Telecom. Inform. Technol.*, no. 4, pp. 16–21, 2013.

[31] A. Krok and Z. Waszczyszyn, "Kalman filtering for neural prediction of response spectra from mining tremors", in *Proc. Int. Conf. Artif. Intell. AI-METH 2004*, Gliwice, Poland, 2004.

[32] A. Krok and Z. Waszczyszyn, "Simulation of building loops for a superconductor using neural networks with Kalman filtering", *Comp. Assisted Mechanics and Engin. Sci.*, vol. 13, pp. 575–582, 2006.

[33] A. Krok and Z. Waszczyszyn, "Kalman filtering for neural prediction of response spectra from mining tremors", *Computers and Structures*, vol. 85, no. 15–16, pp. 1257–1263, 2007.

[34] A. Krok, "Simulation of concrete hysteresis loops using gaussian processes calibrated by genetic algorithm", in *Proc. Int. Symp. IPM on Inverse Problems in Mechanics of Struc. Mater.*, Rzeszów-Łańcut, Poland, 2009.

[35] H. H. Thodberg, "Ace of bayes: application of neural networks with pruning", Tech. Rep. 1132E, The Danish Meat Research Institute, 1993.

[36] A. Ganczarski and L. Barwacz, "Low cycle fatigue based on unilateral damage ewolution", *Int. J. Damage Mech.*, vol. 16, no. 2, pp. 159–177, 2007.

[37] A. Ylinen, "A method of determining the buckling stress and the required cross-sectional area for centrally loaded straight columns in the elastic and inelastic range", *Int. Assoc. for Bridges and Struct. Engin.*, vol. 16, pp. 529–550, 1956.

[38] I. T. Nabney, *Netlab: Algorithms for Pattren Recognition*. Springer, 2002.

[39] H. H. Thodberg, "Review of Bayesian neural networks with application to near infrared spectroscopy", *IEEE Trans. Neural Netw.*, vol. 7, no. 1, pp. 56–72, 1996.

[40] Y. S. Othmana *et al.*, "Frequency based hysteresis compensation for piezoelectric tube scanner using Artificial Neural Networks", *Procedia Engin.*, vol. 41, pp. 757–763, 2012.

Agnieszka Krok received her M.Sc. in the field of stochastic processes at the Jagiellonian University, Cracow, Poland and Ph.D. degree in the field of neural networks at Tadeusz Kosciuszko Cracow University of Technology, Poland, in 2003 and 2007, respectively. From 2009 she is an Assistant Professor at Faculty of Physics, Mathematics and Computer Science, Tadeusz Kościuszko Cracow University of Technology. Her main scientific and didactic interests are focused mainly on artificial intelligence: artificial neural networks, genetic algorithms, and additionally on parallel processing and cryptography.
E-mail: agneskrok@gmail.com
Faculty of Physics, Mathematics
and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland