sensors

MDPI

*Article*

# Probabilistic Damage Detection of a Steel Truss Bridge Model by Optimally Designed Bayesian Neural Network

**Tao Yin** [1,*] **and Hong-ping Zhu** [2]

[1] School of Civil Engineering, Wuhan University, Wuhan 430072, China
[2] School of Civil Engineering & Mechanics, Huazhong University of Science and Technology, Wuhan 430074, China; hpzhu@mail.hust.edu.cn
* Correspondence: tyin@whu.edu.cn

Received: 12 September 2018; Accepted: 6 October 2018; Published: 9 October 2018

**Abstract:** Excellent pattern matching capability makes artificial neural networks (ANNs) a very promising approach for vibration-based structural health monitoring (SHM). The proper design of the network architecture with the suitable complexity is vital to the ANN-based structural damage detection. In addition to the number of hidden neurons, the type of transfer function used in the hidden layer cannot be neglected for the ANN design. Neural network learning can be further presented in the framework of Bayesian statistics, but the issues of selection for the hidden layer transfer function with respect to the Bayesian neural network has not yet been reported in the literature. In addition, most of the research works in the literature for addressing the predictive distribution of neural network output is only for a single target variable, while multiple target variables are rarely involved. In the present paper, for the purpose of probabilistic structural damage detection, Bayesian neural networks with multiple target variables are optimally designed, and the selection of the number of neurons, and the transfer function in the hidden layer, are carried out simultaneously to achieve a neural network architecture with suitable complexity. Furthermore, the nonlinear network function can be approximately linear by assuming the posterior distribution of network parameters is a sufficiently narrow Gaussian, and then the input-dependent covariance matrix of the predictive distribution of network output can be obtained with the Gaussian assumption for the situation of multiple target variables. Structural damage detection is conducted for a steel truss bridge model to verify the proposed method through a set of numerical case studies.

**Keywords:** structural health monitoring; probabilistic damage detection; truss bridge; model class selection; Bayesian neural network

## 1. Introduction

Recent structural damage or component failure has raised public awareness of the need for improved infrastructure safety and maintenance. It is of great practical significance to regularly evaluate and assess the service status of civil infrastructure. There has been great interest in the development of structural health monitoring (SHM) methodology, based on vibration measurements in recent decades [1]. Structural damage generally leads to the reduction of local stiffness, which has an influence on the dynamic characteristic parameters, and changes of these parameters before and after damage are further utilized by the vibration-based structural damage detection methods to identify the damage location and related extent. Different methods have been developed in various aspects of this area so far, such as dynamic analysis methods [2–4], probabilistic methods [5–10], model reduction-based methods [11–13], artificial intelligence methods [14–16], modal identification methods [17–20], wavelet-based methods [21,22], optimal sensor configurations [23–25], and signal processing techniques [26], et al.

Among these methods, artificial neural networks (ANNs) have not been specifically developed for structural damage detection, but their pattern matching capability makes them very promising to be employed as a tool for this purpose. It is obvious that the complexity of the ANN models, including the number of hidden layers and the number of hidden neurons, impacts significantly on the training process of the ANN, as well as the performance of the trained ANN, especially for simulating a complicated functional relationship. If these numbers are too small, the obtained ANN might not be able to capture the true behavior of the training data. In contrast, if they are too large, the trained ANN may produce outputs that will fluctuate in the region between training data points. Thus, proper design of the structure of ANN models, with suitable complexity, is necessary to ensure the successful application of ANN-based structural damage detection methods. However, the ANN model is usually determined by experience or rule of thumb only, and very limited research works exist in the area of ANN-based damage detection, addressing the issue of designing ANN models properly [27–29]. Other than the number of hidden neurons, the type of transfer (activation) function utilized in the hidden layer is also a non-negligible factor with respect to the ANN design, since the nonlinearity of the transfer function has an effect on the generalization ability of neural networks of the trained ANN.

In the traditional ANN approach as mentioned above, the values of the network parameters, including weights and biases, are estimated from the training data set by minimizing the sum of squared errors, representing the error between the target variables and the output of the neural network. More importantly, neural network learning can also be presented in the framework of Bayesian inference. Starting from the early research works related to Bayesian neural network [30,31], there has been a growing interest in the application of Bayesian inference theory in the field of neural networks [32–37]. By noting the significance of the ANN design as mentioned previously, the selection of the number of neurons in the hidden layer for the Bayesian neural network has been addressed [28]; however, to the best of our knowledge, the issues of selection of the transfer function, with respect to the Bayesian neural network, has not yet been addressed in previous studies. In addition, one of the significant features of the Bayesian neural network is the capability of quantifying the error or uncertainty of the network output, by employing the Bayesian inference approach. Nevertheless, most of the research activities in the literature for addressing the predictive distribution of neural network output only deal with a single target variable [38–40], whereas the distribution over network output for multiple target variables is seldom mentioned.

In this paper, the Bayesian neural networks with multiple target variables is optimally designed for the purpose of probabilistic structural damage detection; additionally, the simultaneous selection of numbers of neurons and the corresponding transfer function in the hidden layer to obtain a neural network with suitable complexity is well addressed. Furthermore, by assuming that the posterior distribution of network parameters is approximated as a sufficiently narrow Gaussian, so that the nonlinear network function is approximately linear with respect to the network parameters over the region of parameter space, the input-dependent covariance matrix of predictive distribution, being approximated also as a Gaussian, can be obtained for the case of multiple target variables. The feasibility and validity of the proposed method is verified by the numerical case studies of a steel truss bridge model.

## 2. Theoretical Development

The multi-layer feedforward neural network is commonly employed for ANN-based structural damage detection in the literature [41], and it has been proved to be capable of approximating any functional relationship between inputs and outputs with only one hidden layer [42]. In the present paper, without loss of generality, the proposed method focuses on the design of the structure of the hidden layer for a single-hidden-layer feedforward Bayesian neural network, since a linear function is always utilized as the transfer function in the output layer. This thus involves the selection of the number of neurons in the hidden layer, as well as the corresponding transfer function of the neurons in the same layer. The Bayesian neural network investigated in this paper updates the weight and bias values according to Levenberg–Marquardt optimization routine through the Bayesian regularization procedure by minimizing a linear combination of squared errors and weights.

The set of model classes for the Bayesian neural network in the present paper are defined as $\mathbb{M}_1, \mathbb{M}_2, \cdots, \mathbb{M}_{N_M}$, representing the network model with the different number of neurons and various types of transfer functions in the hidden layer, and $N_M$ is the total number of model classes considered. Consider the problem of predicting multiple target variables $\mathbf{t} \in \mathbb{R}^{N_O \times 1}$ from a vector $\mathbf{x} \in \mathbb{R}^{N_I \times 1}$ of inputs by utilizing the Bayesian neural network approach. $N_I$ and $N_O$ are the number of neurons in the input and output layers. Let $\mathbb{D}_N = \{\{\mathbf{x}_1, \mathbf{t}_1\}, \{\mathbf{x}_2, \mathbf{t}_2\}, \cdots, \{\mathbf{x}_N, \mathbf{t}_N\}\}$ denote the input–output training data set which is generated from the finite-element analysis of target structure, and $N$ is the total number of training data sets. For the structural damage detection problem investigated in this paper, the changes of natural frequencies and normalized partial mode shapes corresponding to the measured degrees of freedom (DOF) for the first few modes before and after damage are utilized as pattern features for the damage, which correspond to the ANN input. The damaged members that are to be identified relate to the ANN output. Thus, the aim of the Bayesian neural network design is to select the most plausible class of models by using data $\mathbb{D}_N$ from $N_M$ prescribed classes of Bayesian neural network models to approximate the functional relationship defined by the input and output data set.

It is assumed that for the $j$th class of model $\mathbb{M}_j$, that the conditional distribution $p(\mathbf{t}|\mathbf{x}, \mathbf{w}_j; \mathbb{M}_j)$ is Gaussian with an $\mathbf{x}$-dependent mean given by the output of a neural network model $\mathbf{y}(\mathbf{x}, \ \mathbf{w}_j; \mathbb{M}_j)$. Also, the multiple target variables $\mathbf{t}$ are assumed to be independent conditional on the inputs $\mathbf{x}$ and network parameters $\mathbf{w}_j$ including weights and biases with shared noise precision parameter $\beta_j$, which represents the precision of the Gaussian noise. Then, the conditional distribution of the target values with respect to the $j$th class of models is given by [38]:

$$p(\mathbf{t}|\mathbf{x}, \ \mathbf{w}_j, \beta_j, \mathbb{M}_j) = \mathcal{N}\left(\mathbf{t}|\mathbf{y}(\mathbf{x}, \ \mathbf{w}_j; \mathbb{M}_j), \ \beta_j^{-1}\mathbf{I}_{N_O}\right) \tag{1}$$

where $\mathbf{I}_{N_O}$ is an identity matrix in dimension $N_O$.

Given the input−output training data set $\mathbb{D}_N$, one can construct the corresponding likelihood function as:

$$\begin{aligned} p(\mathbb{D}_N|\mathbf{w}_j, \beta_j, \mathbb{M}_j) = & \prod_{n=1}^{N} \mathcal{N}\left(\mathbf{t}_n|\mathbf{y}(\mathbf{x}_n, \ \mathbf{w}_j; \mathbb{M}_j), \ \beta_j^{-1}\mathbf{I}_{N_O}\right) = \\ & = \left(\frac{\beta_j}{2\pi}\right)^{NN_o/2} \exp\left(-\frac{\beta_j}{2}\sum_{n=1}^{N} \|\mathbf{y}(\mathbf{x}_n, \ \mathbf{w}_j; \mathbb{M}_j) - \mathbf{t}_n\|^2\right) \end{aligned} \tag{2}$$

where $\|\cdot\|$ represents the Euclidean norm.

Similarly, a prior distribution can be chosen to be Gaussian over the uncertain network parameters $\mathbf{w}_j$ as the following:

$$P(\mathbf{w}_j|\alpha_j, \mathbb{M}_j) = \mathcal{N}\left(\mathbf{w}_j|\mathbf{0}, \ \alpha_j^{-1}\mathbf{I}_{W_j}\right) = \left(\frac{\alpha_j}{2\pi}\right)^{W_j/2} \exp\left(-\frac{\alpha_j}{2}\|\mathbf{w}_j\|^2\right) \tag{3}$$

where $\mathbf{I}_{W_j}$ is a $W_j \times W_j$ identity matrix. $W_j = N_I + N_O + N_H$ is the dimension of the weight vector, and $N_H$ is the number of neurons in the hidden layer.

By following the Bayes' theorem, the posterior distribution of the network parameters $\mathbf{w}_j$ for the model class $\mathbb{M}_j$ is given by:

$$p(\mathbf{w}_j|\mathbb{D}_N, \alpha_j, \beta_j, \mathbb{M}_j) = \frac{p(\mathbb{D}_N|\mathbf{w}_j, \beta_j, \mathbb{M}_j)\, p(\mathbf{w}_j|\alpha_j, \mathbb{M}_j)}{p(\mathbb{D}_N|\alpha_j, \beta_j, \mathbb{M}_j)} \tag{4}$$

which, however, is non-Gaussian due to the consequence of the nonlinear dependence of the network function $\mathbf{y}(\mathbf{x},\ \mathbf{w}_j;\mathbb{M}_j)$ on the network parameters $\mathbf{w}_j$.

By utilizing the Laplace approximation, one can seek a Gaussian approximation to the posterior distribution $p(\mathbf{w}_j|\mathbb{D}_N,\alpha_j,\beta_j,\mathbb{M}_j)$ in Equation (4) at a (local) maximum of the posterior, say $\mathbf{w}_{j,\text{MP}}$, which can be obtained through the standard nonlinear optimization routine. As usual, it is convenient to maximize the logarithm of the posterior, which can be written as:

$$\ln\left[p(\mathbf{w}_j|\mathbb{D}_N,\alpha_j,\beta_j,\mathbb{M}_j)\right] \propto -\frac{\beta_j}{2}\sum_{n=1}^{N}\|\mathbf{y}(\mathbf{x}_n,\ \mathbf{w}_j;\mathbb{M}_j)-\mathbf{t}_n\|^2-\frac{\alpha_j}{2}\|\mathbf{w}_j\|^2 \tag{5}$$

which corresponds to a regularized sum-of-squares error function, while the two hyperparameters $\alpha_j$ and $\beta_j$ are fixed and known at this moment.

Then, from Equation (5), a local Gaussian approximation can be built in by evaluating the matrix of second derivatives of the negative log posterior distribution at the maximum of the posterior, and it is given by:

$$p(\mathbf{w}_j|\mathbb{D}_N,\alpha_j,\beta_j,\mathbb{M}_j)=\mathcal{N}\left(\mathbf{w}_j|\mathbf{w}_{j,\text{MP}},\ \mathbf{H}_j^{-1}(\mathbf{w}_{j,\text{MP}})\right) \tag{6}$$

where $\mathbf{H}_j(\mathbf{w}_{j,\text{MP}})=-\nabla\nabla\ln\left[p(\mathbf{w}_j|\mathbb{D}_N,\alpha_j,\beta_j,\mathbb{M}_j)\right]|_{\mathbf{w}_j=\mathbf{w}_{j,\text{MP}}}$ is the Hessian matrix evaluated at $\mathbf{w}_{j,\text{MP}}$.

By making use of the evidence framework, together with the Gaussian approximation to the posterior utilizing the Laplace approximation, the marginal likelihood or evidence for the two hyperparameters is obtained by integrating the network parameters as:

$$\begin{aligned} p(\mathbb{D}_N|\alpha_j,\beta_j,\mathbb{M}_j)=&\int p(\mathbb{D}_N|\mathbf{w}_j,\beta_j,\mathbb{M}_j)\,p(\mathbf{w}_j|\alpha_j,\mathbb{M}_j)\mathrm{d}\mathbf{w}_j \\ &\simeq p(\mathbb{D}_N|\mathbf{w}_{j,\text{MP}},\beta_j,\mathbb{M}_j)\,p(\mathbf{w}_{j,\text{MP}}|\alpha_j,\mathbb{M}_j)\,(2\pi)^{W_j/2}|\mathbf{H}_j(\mathbf{w}_{j,\text{MP}})|^{-1/2}\end{aligned} \tag{7}$$

In the evidence framework, the point estimates for hyperparameters $\alpha_j$ and $\beta_j$ can be obtained by maximizing $\ln\left[p(\mathbb{D}_N|\alpha_j,\beta_j,\mathbb{M}_j)\right]$ with respect to $\alpha_j$ and $\beta_j$ at the maximum of the posterior $\mathbf{w}_{j,\text{MP}}$ [38], respectively, as:

$$\frac{1}{\alpha_{j,\text{MP}}}=\frac{1}{\gamma_{j,\text{MP}}}\|\mathbf{w}_{j,\text{MP}}\|^2,\ \frac{1}{\beta_{j,\text{MP}}}=\frac{1}{NN_O-\gamma_{j,\text{MP}}}\sum_{n=1}^{N}\|\mathbf{y}(\mathbf{x}_n,\ \mathbf{w}_{j,\text{MP}};\mathbb{M}_j)-\mathbf{t}_n\|^2 \tag{8}$$

where $\gamma_{j,\text{MP}}=W_j-\alpha_{j,\text{MP}}\text{tr}\left(\mathbf{H}_j^{-1}(\mathbf{w}_{j,\text{MP}})\right)$ represents the effective number of parameters corresponding to the $j$th model class. Equation (8) presents a practical iterative procedure for estimating the hyperparameters during the training process of Bayesian neural network.

To select the most plausible model class among $N_M$ prescribed model classes for the Bayesian neural network, the probability of a model class $\mathbb{M}_j$ conditional on the given set of input-target training data $\mathbb{D}_N$ should be calculated. This can be obtained by following the Bayes' theorem as:

$$p(\mathbb{M}_j|\mathbb{D}_N)=\frac{p(\mathbb{D}_N|\mathbb{M}_j)\pi(\mathbb{M}_j)}{\sum_{j=1}^{N_c}p(\mathbb{D}_N|\mathbb{M}_j)\pi(\mathbb{M}_j)} \tag{9}$$

where the prior probability $\pi(\mathbb{M}_j)$ on the model class $\mathbb{M}_j$, for $j=1$ to $N_M$, satisfies $\sum\limits_{j=1}^{N_c}\pi(\mathbb{M}_j)=1$. As there is generally no prior information about each class of models for the purpose of structural damage detection, it is simply assumed hereafter that each individual model class possesses the same initial plausibility, i.e., $\pi(\mathbb{M}_j)=1/N_M$. The factor $p(\mathbb{D}_N|\mathbb{M}_j)$ is the most important term in Equation (9), and it is known as the evidence for the model class $\mathbb{M}_j$ giving the set of input–output training data $\mathbb{D}_N$. Generally, the class of models to be used is the one that maximizes the posterior probability $p(\mathbb{M}_j|\mathbb{D}_N)$, or equivalently, maximizes the model evidence $p(\mathbb{D}_N|\mathbb{M}_j)$ with respect to $\mathbb{M}_j$.

It is noted that instead of utilizing the usual Bayesian treatment for hyperparameters involving marginalization over all possible values, the model evidence $p(\mathbb{D}_N|\mathbb{M}_j)$ presented in Equation (9) can be approximated by substituting the values of hyperparameters $\alpha_{j,\mathrm{MP}}$ and $\beta_{j,\mathrm{MP}}$ obtained at the maximum of the posterior $\mathbf{w}_{j,\mathrm{MP}}$ from the iterative optimization procedure given in Equation (8) into the marginal likelihood in Equation (7), i.e.,

$$p(\mathbb{D}_N|\mathbb{M}_j) \simeq p(\mathbb{D}_N|\mathbf{w}_{j,\mathrm{MP}}, \beta_{j,\mathrm{MP}}, \mathbb{M}_j)\, p(\mathbf{w}_{j,\mathrm{MP}}|\alpha_{j,\mathrm{MP}}, \mathbb{M}_j)\,(2\pi)^{W_j/2}|\mathbf{H}_j(\mathbf{w}_{j,\mathrm{MP}})|^{-1/2} \tag{10}$$

It is important to note that the form of the evidence $p(\mathbb{D}_N|\mathbb{M}_j)$ given in Equation (10) is consistent with that given by the authors of [43], that is, the evidence consists of two terms, $p(\mathbb{D}_N|\mathbf{w}_{j,\mathrm{MP}}, \beta_{j,\mathrm{MP}}, \mathbb{M}_j)$ and $p(\mathbf{w}_{j,\mathrm{MP}}|\alpha_{j,\mathrm{MP}}, \mathbb{M}_j)(2\pi)^{W_j/2}|\mathbf{H}_j(\mathbf{w}_{j,\mathrm{MP}})|^{-1/2}$, namely, the likelihood factor and the Ockham factor. Specifically, the likelihood factor favors more complex model classes. Thus, it will be higher for those model classes making the probability of the data $\mathbb{D}_N$ higher, implying a better fit to the data. The Ockham factor, however, imposes a penalty against the complexity of the specified model class. The balance between these two factors allows one to select the most probable model class through a mathematically rigorous and robust way, which is just complex enough to fit the given data. In this study, the class of models to be selected is the one possessing the highest value of evidence, i.e., maximum posterior probability, among the entire set of model classes for the given set of train data. It is also noted that because the corresponding numerical values are usually very large, the logarithmic form of the evidence is taken in order to avoid the computational problem during the procedure of model class selection. This yields that:

$$\begin{aligned} \ln[p(\mathbb{D}_N|\mathbb{M}_j)] \simeq \tfrac{NN_O}{2}\ln\beta_{j,\mathrm{MP}} - \tfrac{\beta_{j,\mathrm{MP}}}{2}\sum_{n=1}^{N}\|\mathbf{y}(\mathbf{x}_n,\ \mathbf{w}_{j,\mathrm{MP}};\mathbb{M}_j) - \mathbf{t}_n\|^2 \\ +\tfrac{W_j}{2}\ln\alpha_{j,\mathrm{MP}} - \tfrac{\alpha_{j,\mathrm{MP}}}{2}\|\mathbf{w}_{j,\mathrm{MP}}\|^2 + \tfrac{W_j}{2}\ln(2\pi) - \tfrac{1}{2}\ln|\mathbf{H}_j(\mathbf{w}_{j,\mathrm{MP}})| \end{aligned} \tag{11}$$

where the sum of the terms in the first row gives the logarithm of the likelihood factor, and that in the second row represents the logarithmic expression of the Ockham factor.

In addition, it should be noted that if Equation (9) is applied for identifying the 'optimal' class of Bayesian neural network models by direct comparison of the conditional probability $p(\mathbb{M}_j|\mathbb{D}_N)$, the total number of model classes $N_M$ to be considered in the identification process should be specified at the beginning. If $N_M$ is too small, the 'optimal' model class might not be included in the study. However, if this number is too large, the computational consumption would be unaffordable. Thus, instead of directly comparing the all $N_M$ model classes and picking up the 'best' one, the number of neurons as well as the transfer function in the hidden layer is identified by following a computationally efficient algorithm [29]. Denote $N_T$ as the number of transfer functions involved for comparison, and the main procedure of the algorithm is summarized as following:

1. Initialize the index of the type of hidden layer transfer function $i = 1$ (outer loop).
2. Initialize the index of number of hidden neurons $k = 1$ (inner loop), and calculate the corresponding log evidence $\ln\left[p\left(\mathbb{D}_N|\mathbb{M}_{j(i,k)}\right)\right]$ in Equation (11), where $j$ depends on both $i$ and $k$.
3. Increase the index $k$ by 1, and calculate the log evidences $\ln\left[p\left(\mathbb{D}_N|\mathbb{M}_{j(i,k+1)}\right)\right]$, which is compared with $\ln\left[p\left(\mathbb{D}_N|\mathbb{M}_{j(i,k)}\right)\right]$; If $\ln\left[p\left(\mathbb{D}_N|\mathbb{M}_{j(i,k+1)}\right)\right] < \ln\left[p\left(\mathbb{D}_N|\mathbb{M}_{j(i,k)}\right)\right]$, then $\mathbb{M}_{j(i,k)}$ is the model class with 'optimal' number of hidden neurons as for the $i$th transfer function, and the inner loop will stop; Or, the algorithm will increase the index $k$ by 1 and repeat this step; Record the log evidence $\ln\left[p\left(\mathbb{D}_N|\mathbb{M}_{j(i,k_i)}\right)\right]$ related to the 'optimal' number of hidden neurons for the present transfer function.
4. Increase the index of transfer function $i$ by 1, repeat steps 2 and 3, and record the log evidence $\ln\left[p\left(\mathbb{D}_N|\mathbb{M}_{j(i+1,k_{i+1})}\right)\right]$ related to the 'optimal' number of hidden neurons for the $(i+1)$th

transfer function; Compare $\ln\left[p\left(\mathbb{D}_N|\mathbb{M}_{j(i+1,k_{i+1})}\right)\right]$ with $\ln\left[p\left(\mathbb{D}_N|\mathbb{M}_{j(i,k_i)}\right)\right]$, and record the larger one.

5. If $i < N_T$, set $i = i + 1$, and repeat step 4; Otherwise, if $i = N_T$, the whole algorithm will stop, and output the 'optimal' number of hidden neurons $k_{i^*}$ as well as the related hidden layer transfer function index $i^*$ corresponding to the 'optimal' model class $\mathbb{M}_{j(i^*,k_{i^*})}$ with recorded largest log evidence $\ln\left[p\left(\mathbb{D}_N|\mathbb{M}_{j(i^*,k_{i^*})}\right)\right]$.

On the other hand, it is assumed that the identified most plausible model class through the above design procedure is $\mathbb{M}_K$, and $K = j(i^*, k_{i^*})$. Then, the posterior of the network parameters given in Equation (6) can be employed to produce a distribution over the network outputs, and the predictive distribution is obtained by marginalizing with respect to this posterior distribution, given as:

$$p(\mathbf{t}|\mathbf{x}, \mathbb{D}_N, \mathbb{M}_K) = \int p(\mathbf{t}|\mathbf{x},\ \mathbf{w}_K, \beta_{K,\mathrm{MP}}, \mathbb{M}_K)p(\mathbf{w}_K|\mathbb{D}_N, \alpha_{K,\mathrm{MP}}, \beta_{K,\mathrm{MP}}, \mathbb{M}_K)\mathrm{d}\mathbf{w}_K \tag{12}$$

It is noted that this integration is analytically intractable due to the nonlinearity of the network function $\mathbf{y}(\mathbf{x},\ \mathbf{w}_K; \mathbb{M}_K)$ as a function of the network parameters $\mathbf{w}_K$. Under such circumstances, an approximation to evaluate this integral is generally required. To make progress, it is assumed that the covariance of posterior distribution of network parameters is small, so that the network function is approximately linear with respect to the parameters over the region of parameter space for which the posterior probability is significantly nonzero [38]. By making a Taylor series expansion of the network function around the maximum of the posterior $\mathbf{w}_{K,\mathrm{MP}}$ and retaining only the linear terms, the conditional distribution of the target values can be written as:

$$p(\mathbf{t}|\mathbf{x},\ \mathbf{w}_K, \beta_{K,\mathrm{MP}}, \mathbb{M}_K) \simeq \mathcal{N}\left(\mathbf{t}|\mathbf{y}(\mathbf{x},\ \mathbf{w}_{K,\mathrm{MP}}; \mathbb{M}_K) + \mathbf{J}_K(\mathbf{x}, \mathbf{w}_{K,\mathrm{MP}})(\mathbf{w}_K - \mathbf{w}_{K,\mathrm{MP}}),\ \beta_{K,\mathrm{MP}}^{-1}\mathbf{I}_{N_O}\right) \tag{13}$$

where the mean is a linear function of $\mathbf{w}_K$, and $\mathbf{J}_K(\mathbf{x}, \mathbf{w}_{K,\mathrm{MP}}) = \nabla\mathbf{y}(\mathbf{x},\ \mathbf{w}_K; \mathbb{M}_K)|_{\mathbf{w}_K=\mathbf{w}_{K,\mathrm{MP}}}$ is the $\mathbf{x}$-dependent Jacobian matrix of the vector-valued network function evaluated at the maximum of the posterior $\mathbf{w}_{K,\mathrm{MP}}$, and can be calculated by utilizing the finite difference approach.

Then, by noticing that the posterior distribution of weights is approximated as a sufficiently narrow Gaussian, one arrives at a Gaussian distribution over the outputs of the network as the following:

$$p(\mathbf{t}|\mathbf{x}, \mathbb{D}_N, \mathbb{M}_K) \simeq \mathcal{N}(\mathbf{t}|\mathbf{y}(\mathbf{x},\ \mathbf{w}_{K,\mathrm{MP}}; \mathbb{M}_K),\ \Sigma_K(\mathbf{x}, \mathbf{w}_{K,\mathrm{MP}})) \tag{14}$$

where the predictive distribution is approximated as a multivariate Gaussian, the mean of which is given by the network function $\mathbf{y}(\mathbf{x},\ \mathbf{w}_{K,\mathrm{MP}}; \mathbb{M}_K)$ with the network model parameters set to their maximum of the posterior -estimate. The corresponding $\mathbf{x}$-dependent covariance matrix is given by:

$$\Sigma_K(\mathbf{x}, \mathbf{w}_{K,\mathrm{MP}}) = \beta_{K,\mathrm{MP}}^{-1}\mathbf{I}_{N_O} + \mathbf{J}_K(\mathbf{x}, \mathbf{w}_{K,\mathrm{MP}})\mathbf{H}_K^{-1}(\mathbf{w}_{K,\mathrm{MP}})\mathbf{J}_K^{\mathrm{T}}(\mathbf{x}, \mathbf{w}_{K,\mathrm{MP}}) \tag{15}$$

It is apparent that the $\mathbf{x}$-dependent covariance matrix $\Sigma_K(\mathbf{x}, \mathbf{w}_{K,\mathrm{MP}})$ consists of two terms. The first one reflects the intrinsic noise on the target variable, whereas the second is an $\mathbf{x}$-dependent term expressing uncertainty, due to the uncertainty of the weights.

## 3. Case Studies

To verify the proposed methodology, numerical case studies are conducted in this section for a simply-supported steel truss bridge model, as shown in Figure 1. The space steel truss bridge, including 11 crossbeams, has a total length of 2.8 m, a width of 0.48 m and a height of 0.4 m, respectively. Except for the cross-section of the crossbeams being the I-steel type, the other components, including the upper and lower chords, and the diagonal and vertical rods, are all made of a pair of angle irons with the same size. Sectional and material properties of the truss bridge model are shown in Table 1. Each component of the truss bridge model is discretized into one beam finite element, and the total number of such elements for the entire finite element model is 85. The configuration of measurement points on the bridge deck is shown in Figure 2, and the ten measurement points, evenly located on the both sides of bridge deck, measure the vertical motion of the bridge structure. The first six mode shapes of the steel truss bridge deck in intact status are shown in Figure 3, including the first three vertical vibration and torsional modes.

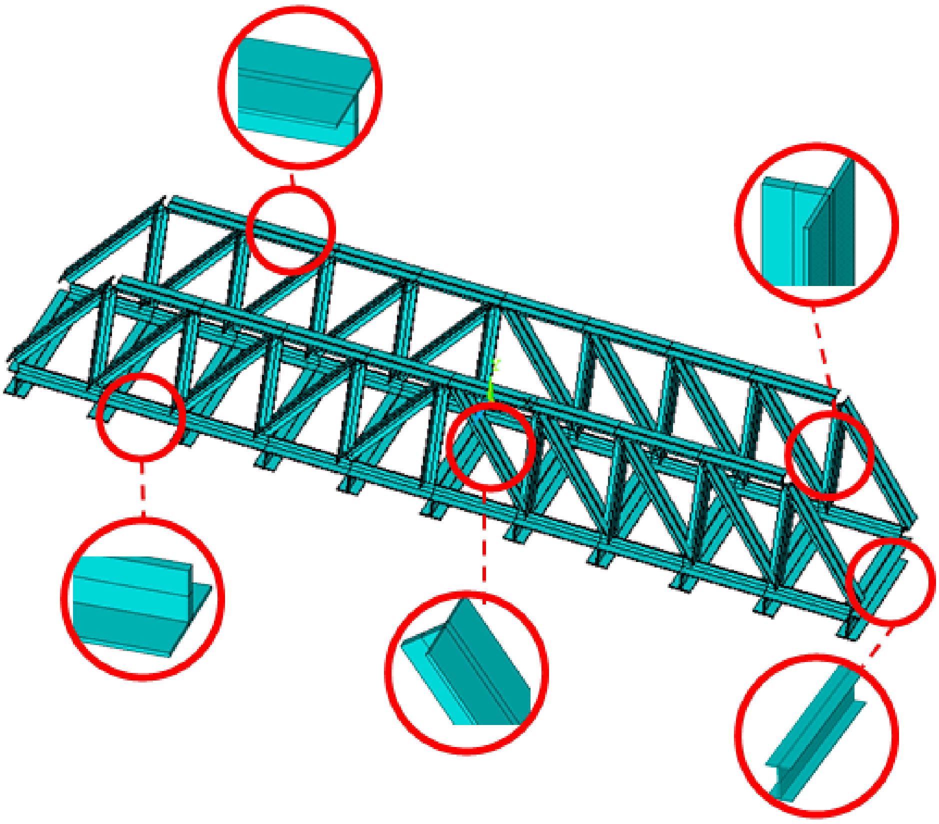

**Figure 1.** Steel truss bridge model.

**Table 1.** Sectional and material properties of the steel truss bridge model.

| Parameter Name | Values |
|---|---|
| Young's modulus | $2.06 \times 10^{11}$ N/m$^2$ |
| Mass density | $7.85 \times 10^{3}$ kg/m$^3$ |
| Cross-sectional area (Angle iron) | $0.47 \times 10^{-3}$ m$^2$ |
| Cross-sectional area (I-steel) | $1.57 \times 10^{-3}$ m$^2$ |
| Moment of inertia (Angle iron) | $0.11 \times 10^{-6}$ m$^4$ |
| Moment of inertia (I-steel) | $2.21 \times 10^{-6}$ m$^4$ |
| Bridge span | 2.8 m |
| Bridge width | 0.48 m |
| Bridge height | 0.4 m |

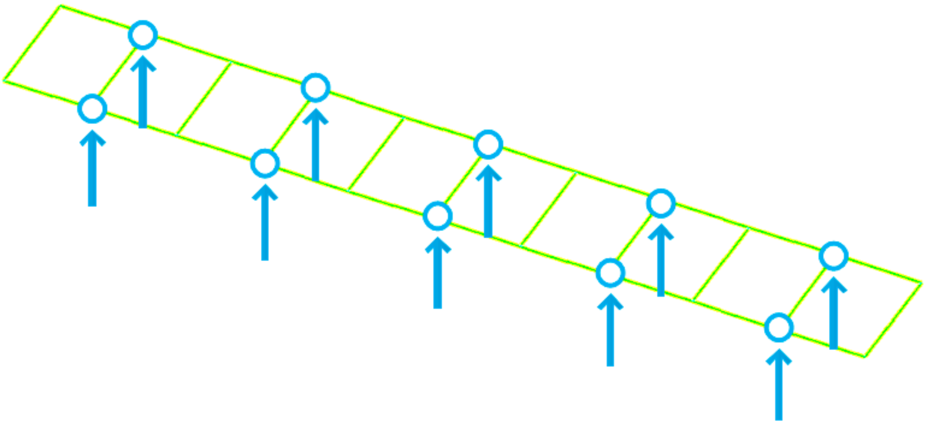

**Figure 2.** Configuration of ten measurement points on the bridge deck.

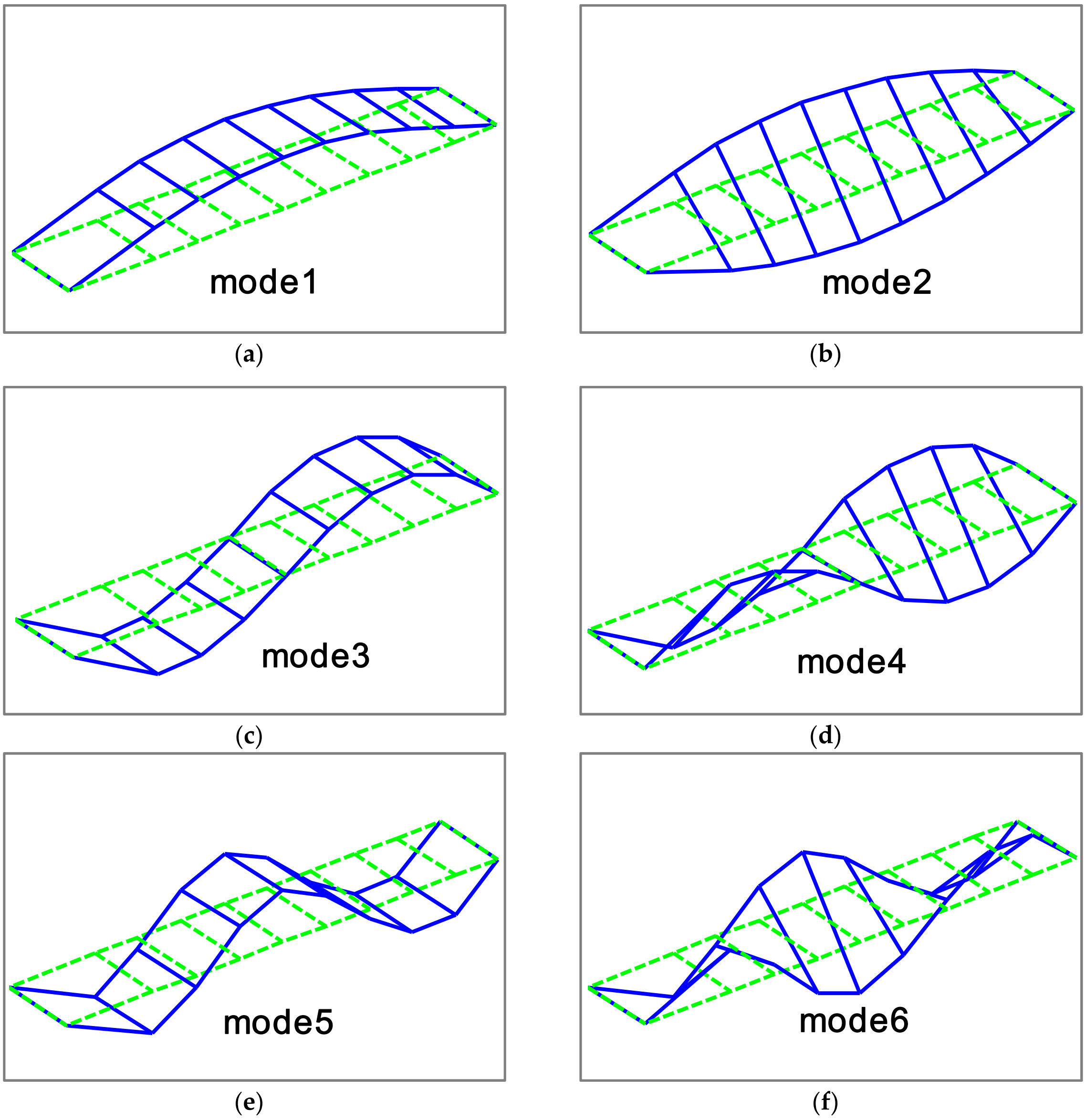

**Figure 3.** First six mode shapes: (**a**) Mode 1 (74.91 Hz); (**b**) Mode 2 (83.67 Hz); (**c**) Mode 3 (221.13 Hz); (**d**) Mode 4 (240.72 Hz); (**e**) Mode 5 (312.06 Hz); (**f**) Mode 6 (370.63 Hz).

The damage is considered on the lower chord of the steel truss bridge model by reducing the bending stiffness, which is achieved by reducing the Young's Modulus of the potentially damaged structural member with a scaling factor in the present study, and there is a total of 20 potential damage locations, which are denoted by E1 to E20, as shown in Figure 4, respectively. However, the excessive number of damaged locations to be identified will result in too many output neurons, which is not conducive to Bayesian neural network training and subsequent damage identification procedures. Thus, one can consider combining these 20 possible damage locations into 5 groups, represented by EG1 to EG5, respectively, and each group consists of 4 lower chord elements, as shown in Figure 4 and Table 2. In this way, the number of output neurons in the neural network is greatly reduced, from 20 to 5. For example, the element group EG2 includes four individual elements E3, E4, E13, and E14. In such circumstances, if any one or more of the four elements in the same element group are damaged, the damage is only reflected by the overall bending stiffness reduction of the element group EG2. In addition, considering the fact that in the initial stage of the damage development, the number of simultaneously damaged elements would be very few, it is reasonably assumed herein that there are, at most, two damaged elements at the same time for the truss bridge model. Furthermore, three damage extents of 0, 20%, and 40% with respect to the element group, respectively, are utilized to generate the training data for the Bayesian neural network. It is also noted that for each sample of training data, the differences of the natural frequencies and mode shapes before and after the damage for the steel truss bridge model is stacked into a vector and used as the network input. It is noted that although all modes can be calculated from the finite element model, only modal parameters of the first two modes are utilized for damage identification of the steel truss bridge model, in order to evaluate the proposed method with limited information to mimic reality, and the total number of input neurons is 22.

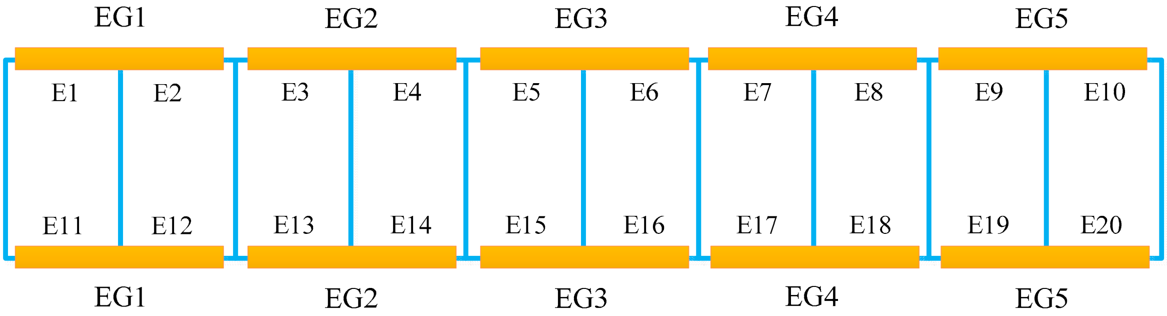

**Figure 4.** Configuration of element groups for generating training data.

**Table 2.** Definition of element groups for the steel truss bridge model.

| Element Groups | Number of Elements | Included Elements |
|---|---|---|
| EG1 | 4 | {E1, E2, E11, E12} |
| EG2 | 4 | {E3, E4, E13, E14} |
| EG3 | 4 | {E5, E6, E15, E16} |
| EG4 | 4 | {E7, E8, E17, E18} |
| EG5 | 4 | {E9, E10, E19, E20} |

This example considers two commonly used types of hidden layer transfer functions, i.e., tansig and satlins [44], labeled as TF1 and TF2 respectively. Figure 5 shows the iterative curve of the proposed Bayesian neural network design procedure. The abscissa represents the number of iterations, while the ordinate denotes the logarithm of evidence. This curve can be clearly divided into two segments, related to the hidden layer transfer functions TF1 and TF2, respectively. It is clear from the results corresponding to TF1 that, as the number of neurons in the hidden layer increases, the log-evidence value gradually increases. Specifically, as shown in Figure 5, when the number of hidden neurons is less than five, the increase of log evidence along with the number of hidden neurons is particularly significant, whereas after that, this increase becomes very gentle. It can thus be seen that, the number of neurons in the hidden layer has a great influence on the performance of the Bayesian neural network, especially when the number of hidden neurons is small; however, as the number increases to a certain extent, the influence becomes less obvious. This clearly indicates that the number of neurons in the hidden layer cannot be too small; otherwise, the Bayesian neural network might not work well. As the number of hidden neurons increases to 12, the log-evidence value reaches the maximum value with respect to the hidden layer transfer function TF1, and the continuous increase of the number of hidden neurons results in a decrease in the log evidence. Thus, the network design algorithm switches to the next candidate transfer function TF2. Similarly, with the increase of number of hidden neurons, the log evidence increases significantly when the number of hidden neurons is relatively small, and gradually reaches its local maximum value. This, however, is smaller than the largest log-evidence value found previously for TF1, and the algorithm stops and finally determines TF1 as the best transfer function under consideration, and the corresponding optimal number of hidden layer neurons is found to be 12 at the same time.

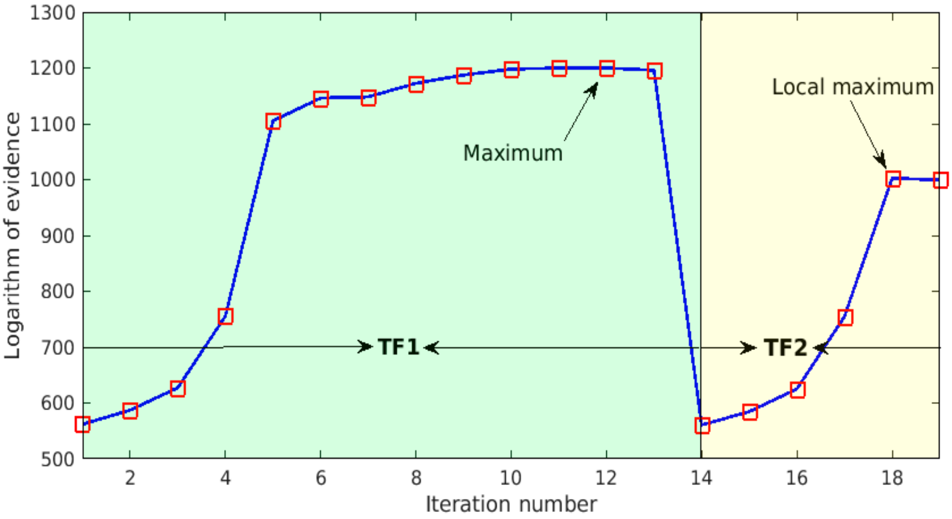

**Figure 5.** Iteration history of the model class selection procedure.

In order to interpret the principle of the proposed network design approach more clearly, taking the hidden layer transfer function TF1 as an example, Figure 6 shows the full picture of the log evidence, the likelihood factor, and the Ockham factor when the number of neurons in the hidden layer gradually increases from 1 to 20, with the number of input and output neurons remaining unchanged. It can clearly be seen from this figure that, as the number of neurons in the hidden layer gradually increases, the structure of the Bayesian neural network model becomes more complex, and network model is expected to better fit the training data, which is reflected by the increase of likelihood factor. At the same time, the absolute value of the Ockham factor gradually increases, which represents a gradual increase in penalty for the complexity of the network model, and the detailed numerical results are also presented in Table 3 for reference (the results corresponding to the 'optimal' numbers of hidden neurons for both TF1 and TF2 are shown in bold). Therefore, under the joint influence of these two factors, the Bayesian network model with the optimal number of hidden neural can be reasonably determined. In addition, it should be realized here that the network performance is sharply increased for a small number of hidden neurons, and then gradually decreased after the log evidence reaches its maximum value, but the magnitude of the decrease is not too obvious. This clearly indicates that the normal operation of the Bayesian neural network requires a sufficient number of hidden neurons, and it also shows that the network optimization design is particularly important in the case of fewer neurons in the hidden layer.

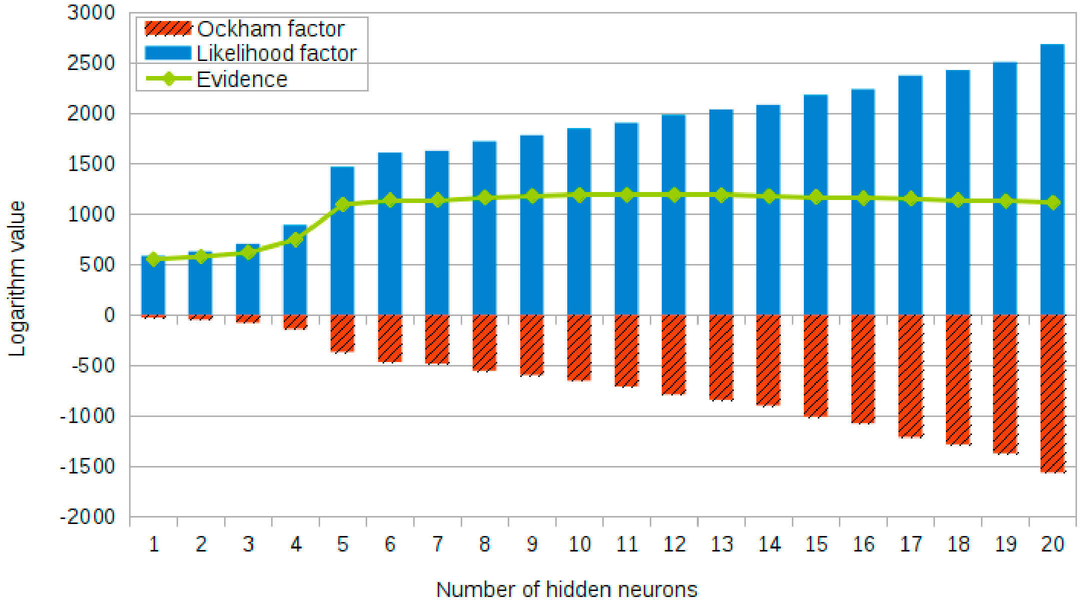

**Figure 6.** Logarithm value of likelihood factor, Ockham factor and evidence.

**Table 3.** The results of model class selection for the Bayesian neural network.

| **Transfer Functions** | **Number of Hidden Neurons $N_H$** | **Logarithm of** | | |
|---|---|---|---|---|
| | | **Evidence** | **Likelihood Factor** | **Ockham Factor** |
| TF1 (tansig) | 8 | 1172.42 | 1724.16 | −551.74 |
| | 9 | 1187.23 | 1780.93 | −593.70 |
| | 10 | 1197.86 | 1849.99 | −652.12 |
| | 11 | 1199.74 | 1905.59 | −705.85 |
| | **12** | **1199.93** | **1986.27** | **−786.34** |
| | 13 | 1195.72 | 2037.02 | −841.30 |
| | 14 | 1186.62 | 2083.90 | −897.28 |
| | 15 | 1178.15 | 2183.16 | −1005.01 |
| TF2 (satlins) | 1 | 560.74 | 588.01 | −27.27 |
| | 2 | 584.43 | 633.46 | −49.03 |
| | 3 | 625.17 | 705.89 | −80.73 |
| | 4 | 754.87 | 895.72 | −140.84 |
| | **5** | **1002.49** | **1265.83** | **−263.34** |
| | 6 | 999.30 | 1280.24 | −280.94 |
| | 7 | 995.60 | 1290.60 | −295.00 |
| | 8 | 993.11 | 1302.23 | −309.12 |

In this paper, three damage cases are considered for the steel truss bridge model. As shown in Figure 7 and Table 4, the damage is simulated by reducing bending stiffness of the corresponding individual element. Specifically, Case 1 considers the single damage situation, and the damage only occurs in element E3, which belongs to the 2nd element group EG2. Case 2 relates to a double-damage condition, and the damage exists simultaneously in E3 and E14. The last one is a multi-damage case, based on Case 2, where the simultaneous damage on E19 and E20 is further involved. Referring to the element group configuration as shown in Table 2 and Figure 4, both Case 1 and Case 2 represent damage that occurred in EG2, whereas the latter shows a relatively greater damage extent. At the same time, Case 3 assumes that EG2 and EG5 are simultaneously damaged. Without loss of generality, the modal parameters utilized for the training data of the neural network involve 1% Gaussian white noise, and the previously optimized Bayesian neural network is employed for the network training and subsequent probabilistic damage detection process. The prediction results of the network output are shown in Table 5. It can be seen from this table that the results of each damage case and the corresponding degree of uncertainty can be successfully identified. Comparing the damage identification results of Case 1 and Case 2, it is clearly seen that the uncertainty of identified damage is not much different under both cases, but the identified damage extent of EG2 for Case 2 is significantly larger than Case 1. This is consistent with the prescribed damage for each case as defined in Table 4, i.e., in Case 2, element group EG2 contains two damaged individual elements at the same time. For the last case, the proposed method also successfully identifies the actually damaged element groups EG2 and EG4, and the uncertainty associated with the damage identification results in such a case is also successfully quantified.

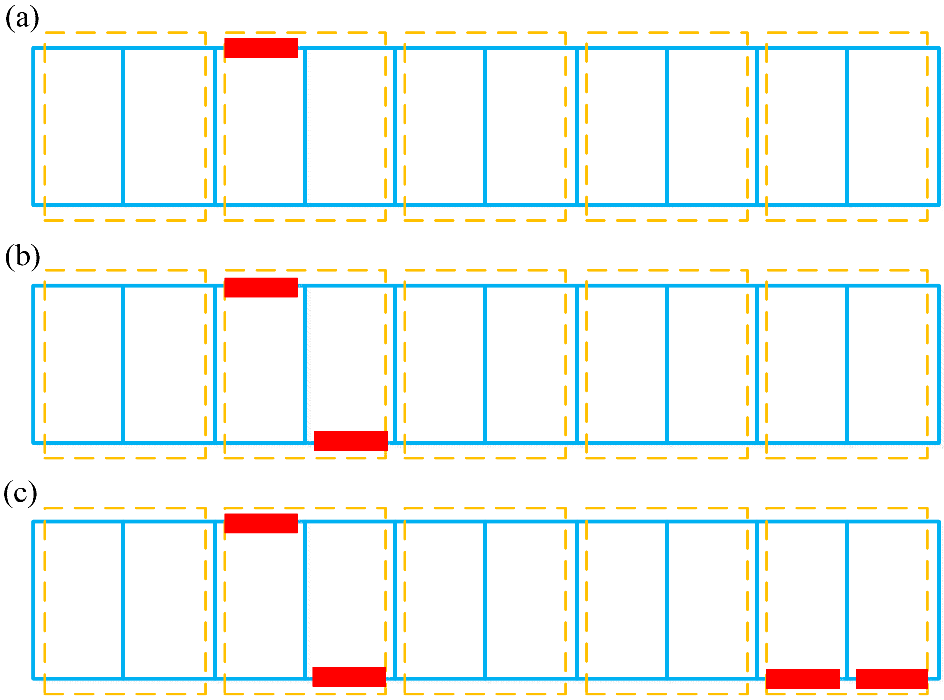

**Figure 7.** Damage configurations for the truss bridge model: (**a**) Case 1 (E3); (**b**) Case 2 (E3 and E14); (**c**) Case 3 (E3 and E14 and E19 and E20).

**Table 4.** Damage cases considered for the steel truss bridge model.

| Cases | Damaged Elements (Stiffness Reduction) | Element Groups |
|---|---|---|
| Case 1 | E3 (30%) | EG2 |
| Case 2 | E3 (30%) & E14 (50%) | EG2 |
| Case 3 | E3 (30%) & E14 (50%) & E19 (30%) & E20 (50%) | EG2 & EG5 |

**Table 5.** Prediction results of the Bayesian neural network.

| Network Output | Case 1 | | Case 2 | | Case 3 | |
|---|---|---|---|---|---|---|
| | Identified | STD Value | Identified | STD Value | Identified | STD Value |
| $t_1$ (EG1) | 0.0095 | 0.06 | 0.0052 | 0.06 | −0.0049 | 0.10 |
| $t_2$ (EG2) | **0.1408** | 0.14 | **0.2938** | 0.15 | **0.2689** | 0.29 |
| $t_3$ (EG3) | 0.0175 | 0.05 | 0.0161 | 0.06 | 0.0182 | 0.10 |
| $t_4$ (EG4) | 0.0523 | 0.10 | 0.0287 | 0.12 | −0.0812 | 0.20 |
| $t_5$ (EG5) | 0.0047 | 0.06 | 0.0066 | 0.07 | **0.4202** | 0.10 |

Additionally, the damage probabilities [45] for this truss bridge model are further quantified to seek a better interpretation of the damage detection results, the data of which are shown in Figure 8 for all three cases under consideration. It can be found for Case 1 that the 2nd element group of the truss bridge model has the greatest probability of damage, as indicated in Figure 8(a1−e1), since the corresponding probability value is somewhat more significant than others. But it is also realized that, as compared to the relatively small extent of damage in EG2, EG4 also possesses a certain degree of damage probability due to the symmetry of the steel truss bridge model. This might not be virtually revealed by directly inspecting the results presented in Table 5, and it will possibly cause some interference to the damage judgment for the present case. As for Case 2, there are also similar observations derived from Figure 8(a2–e2), i.e., EG2 is also the most likely damaged, as compared to the remaining element groups. The probability of damage in EG2 is much more significant than others, since it is a more serious damage configuration as compared to Case 1. For Case 3, it is very obvious from Figure 8(a3−e3) that both EG2 and EG5 have a relatively large possibility of damage, which coincides well with the actual damage configuration prescribed for this case.

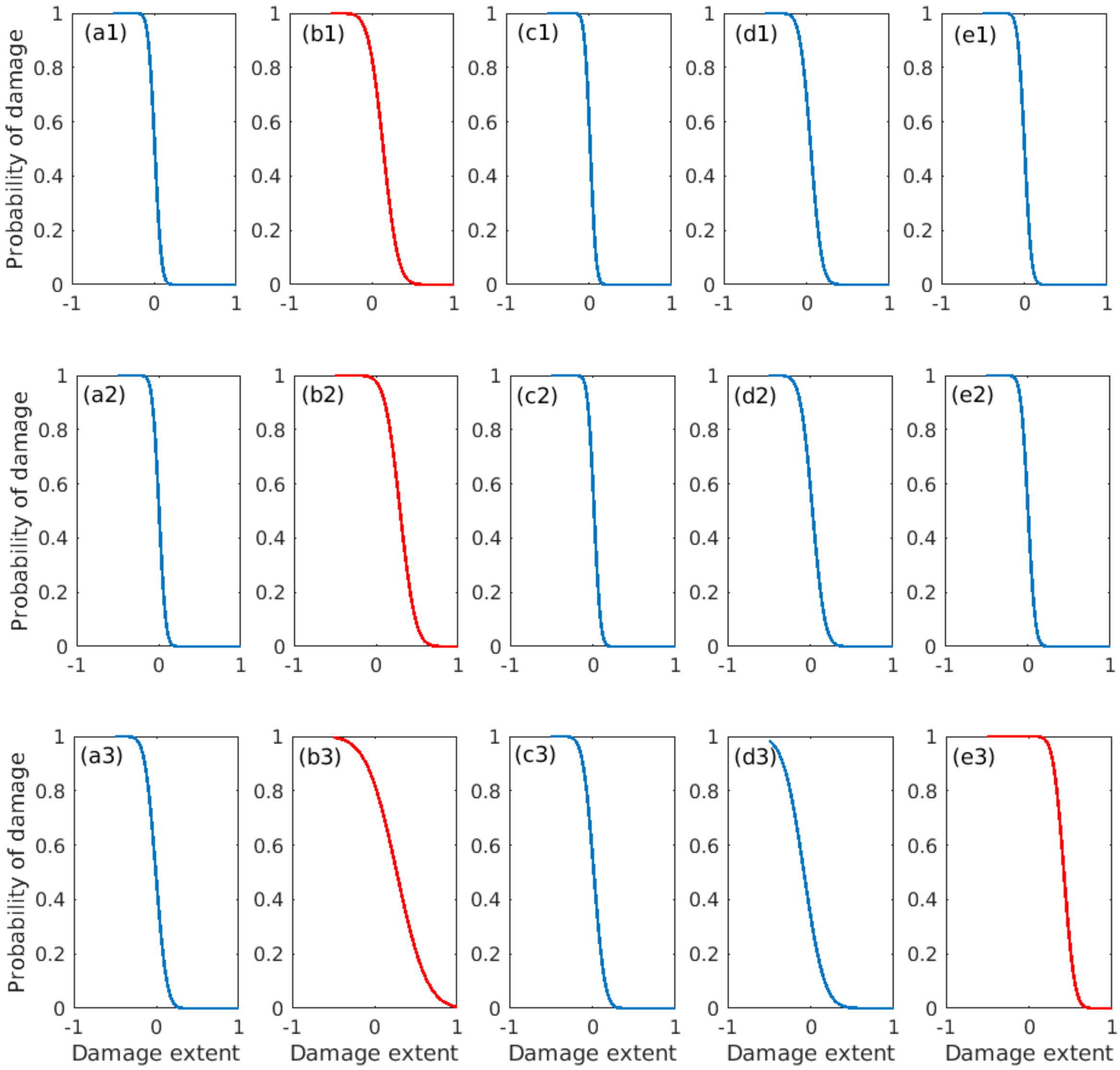

**Figure 8.** Probability of damage: (**a1**)−(**e1**) Case 1; (**a2**)−(**e2**) Case 2; (**a3**)−(**e3**) Case 3.

## 4. Conclusions

For the purpose of probabilistic damage detection based on vibration measurements, the artificial neural network embedded with Bayesian inference for multiple target variables is optimally designed through simultaneous selection of the number of hidden neurons and the type of transfer functions in the hidden layer to obtain a network model with suitable complexity. The validity of the proposed methodology is verified by the numerical case studies conducted for a steel truss bridge model. The obtained results clearly show that under the joint influence of the likelihood and Ockham factors, the proposed network design procedure allows one to determine the optimal architecture of Bayesian neural network model, including a suitable hidden layer transfer function, as well as the optimal number of hidden neurons at the same time via a mathematically rigorous and robust way, which is just complex enough to fit the training data. In addition, it is also revealed that for the present example, regardless of the specific form of the hidden layer transfer function, the normal operation of the Bayesian neural network requires a sufficient number of neurons in the hidden layer, and the network optimization design is particularly important in the situation of fewer neurons in the hidden layer. Furthermore, in addition to the successful identification of damage locations and relative extents, the multivariate predictive distribution of the Bayesian neural network output derived in this paper can be efficiently utilized for quantifying the uncertainty associated with the statistical damage identification results. It should be pointed out that both the Hessian matrix of the logarithm of the posterior and the input-dependent Jacobian matrix of the vector-valued network function evaluated at the maximum of the posterior of weight vector, are calculated by the finite difference approach, which is very time consuming, especially for the weight vector with high dimension. Thus, one of the directions of future work will focus on reducing such computational cost for the purpose of practice application. Furthermore, only numerical examples are employed in the present paper, and the proposed method will be further validated by the experimental case studies in the coming publications.

**Author Contributions:** Writing-Original Draft Preparation, T.Y.; Writing-Review & Editing, H.Z.

**Funding:** The authors gratefully acknowledge the financial support provided by the National Natural Science Foundation of China (Grants No. 51778506, 51838006), and a scholarship from the China Scholarship Council (File No. 201806275091) while the first author visiting the Center for Engineering Dynamics and Institute for Risk and Uncertainty in the University of Liverpool.

**Acknowledgments:** The authors would also like to thank the Editor and the anonymous reviewers for their constructive comments and valuable suggestions to improve the quality of the article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Qarib, H.; Adeli, H. Recent advances in health monitoring of civil structures. *Sci. Iran.* **2014**, *21*, 1733–1742.
2. Fei, Q.G.; Xu, Y.L.; Ng, C.L.; Wong, K.Y.; Chan, W.Y.; Man, K.L. Structural health monitoring oriented finite element model of Tsing Ma bridge tower. *Int. J. Struct. Stab. Dyn.* **2007**, *7*, 647–668. [CrossRef]
3. Yin, T.; Lam, H.F. Vibration analysis of a rectangular thin plate with two parallel cracks. *Adv. Struct. Eng.* **2010**, *13*, 741–753. [CrossRef]
4. Yin, T.; Lam, H.F. Dynamic analysis of finite-length circular cylindrical shells with a circumferential surface crack. *J. Eng. Mech.* **2013**, *139*, 1419–1434. [CrossRef]
5. Beck, J.L.; Au, S.K.; Vanik, M.W. Monitoring structural health using a probabilistic measure. *Comput. Aided Civ. Inf.* **2001**, *16*, 1–11. [CrossRef]
6. Huang, Y.; Beck, J.L.; Li, H. Hierarchical sparse Bayesian learning for structural damage detection: Theory, computation and application. *Struct. Saf.* **2017**, *64*, 37–53. [CrossRef]
7. Mu, H.Q.; Yuen, K.V. Novel outlier-resistant extended Kalman filter for robust online structural identification. *J. Eng. Mech.* **2014**, *141*, 04014100. [CrossRef]
8. Yin, T.; Lam, H.F.; Chow, H.M. A Bayesian probabilistic approach for crack characterization in plate structures. *Comput. Aided Civ. Inf.* **2010**, *25*, 375–386. [CrossRef]
9. Mu, H.Q.; Yuen, K.V. Ground motion prediction equation development by heterogeneous Bayesian learning. *Comput. Aided Civ. Inf.* **2016**, *31*, 761–776. [CrossRef]
10. Huang, Y.; Beck, J.L.; Wu, S.; Li, H. Robust Bayesian compressive sensing for signals in structural health monitoring. *Comput. Aided Civ. Inf.* **2014**, *29*, 160–179. [CrossRef]
11. Yin, T.; Lam, H.F.; Chow, H.M.; Zhu, H.P. Dynamic reduction-based structural damage detection of transmission tower utilizing ambient vibration data. *Eng. Struct.* **2009**, *31*, 2009–2019. [CrossRef]
12. Lam, H.F.; Yin, T. Dynamic reduction-based structural damage detection of transmission towers: Practical issues and experimental verification. *Eng. Struct.* **2011**, *33*, 1459–1478. [CrossRef]

13. Yin, T.; Jiang, Q.H.; Yuen, K.V. Vibration-based damage detection for structural connections using incomplete modal data by Bayesian approach and model reduction technique. *Eng. Struct.* **2017**, *132*, 260–277. [CrossRef]
14. Lam, H.F.; Yuen, K.V.; Beck, J.L. Structural health monitoring via measured Ritz vectors utilizing artificial neural networks. *Comput. Aided Civ. Infrastruct. Eng.* **2006**, *21*, 232–241. [CrossRef]
15. Green, P.L. Bayesian system identification of a nonlinear dynamical system using a novel variant of Simulated Annealing. *Mech. Syst. Signal. Proc.* **2015**, *52–53*, 133–146. [CrossRef]
16. Kourehli, S.S. Damage assessment in structures using incomplete modal data and artificial neural network. *Int. J. Struct. Stab. Dyn.* **2015**, *15*, 1450087. [CrossRef]
17. Au, S.K.; Zhang, F.L.; Ni, Y.C. Bayesian operational modal analysis: Theory, computation, practice. *Comput. Struct.* **2013**, *126*, 3–14. [CrossRef]
18. Yang, J.H.; Lam, H.F.; Hu, J. Ambient vibration test, modal identification and structural model updating following Bayesian framework. *Int. J. Struct. Stab. Dyn.* **2015**, *15*, 1540024. [CrossRef]
19. Zhang, F.L.; Au, S.K. Fundamental two-stage formulation for Bayesian system identification, Part II: Application to ambient vibration data. *Mech. Syst. Signal. Proc.* **2016**, *66*, 43–61. [CrossRef]
20. Zhang, F.L.; Xiong, H.B.; Shi, W.X.; Ou, X. Structural health monitoring of Shanghai Tower during different stages using a Bayesian approach. *Struct. Control Health Monit.* **2016**, *23*, 1366–1384. [CrossRef]
21. Zhu, B.; Leung, A.Y.T.; Wong, C.K.; Lu, W.Z. On-line health monitoring and damage detection of structures based on the wavelet transform. *Int. J. Struct. Stab. Dyn.* **2008**, *8*, 367–387. [CrossRef]
22. Alvandi, A.; Bastien, J.; Jolin, M. Bridge integrity assessment by continuous wavelet transforms. *Int. J. Struct. Stab. Dyn.* **2009**, *9*, 11–43. [CrossRef]
23. Papadimitriou, C.; Beck, J.L.; Au, S.K. Entropy-based optimal sensor location for structural model updating. *J. Vib. Control* **2000**, *6*, 781–800. [CrossRef]
24. Yi, T.H.; Li, H.N.; Gu, M.; Zhang, X.D. Sensor placement optimization in structural health monitoring using niching monkey algorithm. *Int. J. Struct. Stab. Dyn.* **2014**, *14*, 1440012. [CrossRef]
25. Yin, T.; Yuen, K.V.; Lam, H.F.; Zhu, H.P. Entropy-based optimal sensor placement for model identification of periodic structures endowed with bolted joints. *Comput. Aided Civ. Inf.* **2017**, *32*, 1007–1024. [CrossRef]
26. Qarib, H.; Adeli, H. A new adaptive algorithm for automated feature extraction in exponentially damped signals for health monitoring of smart structures. *Smart Mater. Struct.* **2015**, *24*, 125040. [CrossRef]
27. Yuen, K.V.; Lam, H.F. On the complexity of artificial neural networks for smart structures monitoring. *Eng. Struct.* **2006**, *28*, 977–984. [CrossRef]
28. Arangio, S.; Beck, J.L. Bayesian neural networks for bridge integrity assessment. *Struct. Control Health Monit.* **2012**, *19*, 3–21. [CrossRef]
29. Lam, H.F.; Ng, C.T. The selection of pattern features for structural damage detection using an extended Bayesian ANN algorithm. *Eng. Struct.* **2008**, *30*, 2762–2770. [CrossRef]
30. MacKay, D.J.C. A practical Bayesian framework for back-propagation networks. *Neural Comput.* **1992**, *4*, 448–472. [CrossRef]
31. Buntine, W.; Weigend, A. Bayesian back-propagation. *Complex Syst.* **1991**, *5*, 603–643.
32. MacKay, D.J.C. *Bayesian Methods for Backpropagation Networks. Model of Neural Networks III*; Chapter 6; Springer: Berlin, Germany, 1994; pp. 211–254.
33. Lampinen, J.; Vethari, A. Bayesian approach for neural networks—Review and case studies. *Neural Netw.* **2001**, *14*, 257–274. [CrossRef]
34. Barber, D. Bayesian Methods for Supervised Neural Networks. In *Handbook of Brain Theory and Neural Networks*; MIT Press: Cambridge, UK, 2002.
35. Neal, R.M. *Bayesian Learning for Neural Networks*; Lecture Notes in Statistics; Springer: Berlin, Germany, 1996; Volume 118.
36. Lee, H.K.H. *Bayesian Nonparametrics via Neural Networks*; ASA−SIAM Series on Statistics and Applied Probability; SIAM: Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2004.
37. Nabney, I.T. *NETLAB—Algorithms for Pattern Recognition*; Springer: Berlin, Germany, 2004.
38. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin, Germany, 2006.
39. Iruansi, O.; Guadagnini, M.; Pilakoutas, K.; Neocleous, K. Predicting the shear resistance of RC beams without shear reinforcement using a Bayesian neural network. *Int. J. Reliab. Saf.* **2012**, *6*, 82–109. [CrossRef]
40. Kocadağlı, O.; Aşıkgil, B. Nonlinear time series forecasting with Bayesian neural networks. *Expert Syst. Appl.* **2014**, *41*, 6596–6610. [CrossRef]

41. Sohn, H.; Farrar, C.R.; Hernez, F.M.; Czrnecki, J.J.; Shunk, D.D.; Stinemates, D.W.; Nadler, B.R. *A Review of Structural Health Monitoring Literature: 1996–2001*; Report No. LA-13976-MS; Los Alamos National Laboratory: Los Alamos, NM, USA, 2004.
42. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signal* **1989**, *2*, 303–314. [CrossRef]
43. Beck, J.L.; Yuen, K.V. Model selection using response measurement: A Bayesian probabilistic approach. *J. Eng. Mech.* **2004**, *130*, 192–203. [CrossRef]
44. *Neural Network Toolbox User's Guide*; Mathworks Inc.: Natick, MA, USA, 2016.
45. Yin, T.; Zhu, H.P.; Fu, S.J. Damage identification of periodically-supported structures following the Bayesian probabilistic approach. *Int. J. Struct. Stab. Dyn.* **2018**. [CrossRef]