Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 52 (2020) 181–186

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

5th International Conference on System-Integrated Intelligence

Identification of dynamic loads on structural component with artificial neural networks

Osman Altun[a]*, Danyang Zhang[a], Renan Siqueira[a], Philipp Wolniak[a], Iryna Mozgova[a], Roland Lachmayer[a]

[a]*Leibniz University Hannover, Institute of Product Development, An der Universität 1, Garbsen, 30823, Germany*

* Corresponding author. Tel.: +49-511-762-3333; fax: +49-511-762-3333. *E-mail address:* altun@ipeg.uni-hannover.de

Abstract

Enhancing structural components by implementing sensors offers great potential regarding condition monitoring for lifetime analysis, predictive maintenance and automatic adaptation to environmental conditions. This article describes an approach to determining the operational forces applied to the front suspension arm of a car using strain gauges. Since suspension arms are components with free-form surfaces, an analytical calculation of applied forces by means of measured strains is not feasible. Hence, artificial neural networks are applied to approximate the functional relationship. The results reveal how artificial neural networks can be applied to identify load conditions on structural components and, therefore, deliver essential data for condition monitoring.

Keywords: Condition monitoring; load Identification; sensor integration; smart components; artificial neural networks.

Nomenclature

ANN	Artificial neural network
BP	Back propagation
b_{ij}	Threshold of i-th layer pointing to j-th layer
GA	Genetic algorithm
LM	Levenberg-Marquardt
Logsig	Log-sigmoid function
MSE	Mean square error
Q_O	Strain data collected by the upper sensor
Q_R	Strain data collected by the right sensor
Q_U	Strain data collected by the lower sensor
Q_L	Strain data collected by the left sensor
R^2	Decisive factor
RBF	Radial basis function
ReLU	Rectified Linear Unit
T_i	The i-th actual output
Tansig	Tanh-sigmoid function
Trainlm	Training function according to Levenberg-Marquardt
w_{ij}	Weight of i-th layer pointing to j-th layer
Δ_i	Error between i-th predicted and i-th actual output

1. Introduction

Condition monitoring of components for lifetime analysis, predictive maintenance, automatic adaptation to environmental conditions and the initiation of further actions is a current topic in research and development of intelligent systems. To realize condition monitoring for components in a system, these have to be equipped with sensors for data collection. Additionally, data processing, analysis and knowledge generation are required [1][2]. However, automotive components with

10.1016/j.promfg.2020.11.032

integrated sensors are not state of the art, but they offer a great potential with regard to increasing reliability and cost reduction. In addition, information from the use phase of a product can be used for the development of product generations. In the technical inheritance paradigm, these benefits from data feedback are discussed in detail [3].

Since a suspension arm is a component with free-form surfaces, an analytical calculation of the combined applied forces by means of measured strains is not feasible. Hence, artificial neural networks (ANN) are applied to approximate the functional relationship. In a first step, FE-simulations were carried out to identify suitable sensor positions. For this purpose, various load cases acting on a suspension arm during operation were computed. The most critical points were then defined as sensors positions (Figure 1(a)). Next, in order to physically test the operating conditions, a flexible multi-axial dynamic test bench for high loads based on a delta-robot configuration was used (Figure 1(b)). This test bench has been developed especially for the testing of smart products and is well suited with regard to the flexibility of testable components [4]. The suspension arm was equipped with four strain gauges in different directions and, based on the results of the design of experiment, subjected to dynamic loads. In the following, the recorded strain data of the component were used as input data in different artificial neural networks (RBF, BP and Elman). Predefined sinusoidal force curves were used as output data to train and test the neural network. These two types of data sets are first discretized and different input/output combinations are slightly modified for certain operating points.

The sinusoidal load was applied in x and y-direction according to Figure 2 and Figure 4. In reality, no sinusoidal loads would occur on the suspension arm. This is an idealization in this first step of the investigation. However, it is clear that both the force in the x-direction and in the y-direction is not constant while driving. In the x-direction, the road conditions define the load curve and in the y-direction the acceleration and braking operations.

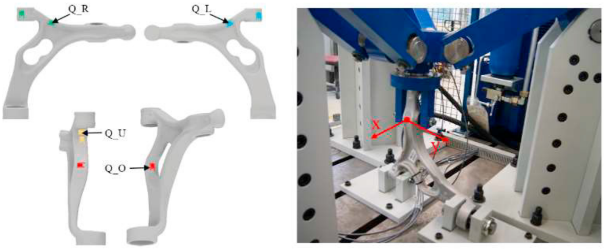

Fig. 1. (a) Sensors positions; (b) Suspension arm in the dynamic multi-axial test bench.

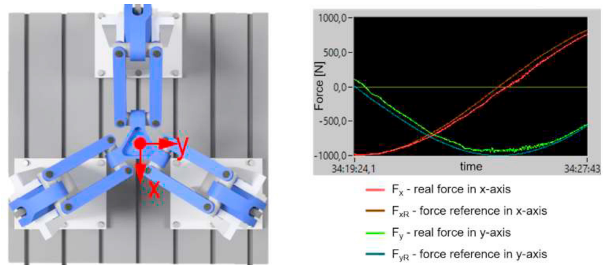

Fig. 2. Load application and directions.

So the aim of this work is to develop an ANN for load identification based on experimental data for the specific application of the used suspension arm, which in future applications in a car should be able to identify the existing loads in unknown real environments.

2. State of the art regarding application areas of artificial neural networks

After nearly half a century of development, artificial neural network theory has achieved widespread success in many research areas such as pattern recognition, automatic control, signal processing and assisted decision making. As one of the main directions for the research and development of ANNs, the application of ANN models and optimization algorithms has received increasing attention [5][6]. As one of the most widely used ANN, BP-ANN (Back propagation-ANN) is often used in pattern recognition and classification, system simulation, intelligent fault diagnosis, image processing, and function simulation due to its advantages such as simple structure, stable working state, and easy hardware implementation and optimal prediction [7][8][9]. In addition, there is RBF-ANN (Radial basis function-ANN). This model has simple training and fast learning convergence, and can approximate any non-linear function. Therefore, it has been widely used in the fields of time series analysis, pattern recognition, non-linear control, and graphic processing. Elman-ANN adds a support layer to the hidden layer of the feedforward network as a one-step delay operator to achieve the purpose of memory, so that the system has the ability to adapt to time-varying characteristics and can directly reflect the characteristics of dynamic process systems [10][11]. In order to optimize ANNs, many optimization algorithms have been proposed. As an intelligent global search algorithm, the genetic algorithm (GA) simulates the rules of survival and survival of the fittest, uses the parameters of the ANN, and searches parallel along multiple routes without falling into the trap of local superiority. The global optimum can be found in many local optimizations. [12][13]

3. Development process of the artificial neural network

Establishing a satisfactory load model has become the key to the technique of strain measurement of load. However, due to the complex structural design of automobile structural parts, multiple force transmission paths, and structural non-linearity caused by large loads, multiple linear regression cannot properly solve the errors caused by nonlinear problems [12]. Therefore, the ANN provides a new method for establishing the load model. This work compares different ANNs with regard to the chosen structural component and subsequently optimizes the parameters of the ANN by using the GA.

3.1. Procedure for identifying a suitable artificial neural network

In this paper, the following process is used to identify the right ANN for identifying loads on automobile structural parts by the usage of strain gauges. There is a need to simulate the non-linear mapping relationship between strain and load through the collected data. Therefore, in this paper BP-ANN, RBF-ANN and Elman-ANN, which are suitable for nonlinear data prediction, will be compared. Subsequently the optimal suitable network for this study has to be selected and further optimization must be carried out, including parameter optimization and weights optimization through the GA.

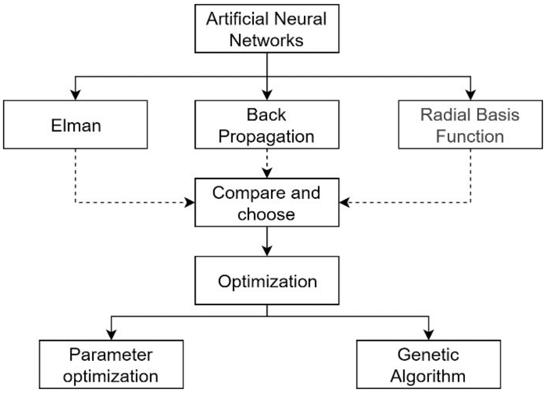

Fig. 3. Research plan.

3.2. Comparison of artificial neural networks

The four data sets measured experimentally by the strain gauges are used as input variables, and the forces in the *x* and *y*-directions are used as output variables. All data have been randomized and 80% of them have been used as the training set and 20% as the test set. The experimental collected data are sufficient, there are enough training data sets while also having enough test data sets (about 65000 data series). Because the sample data are often large and the range is scattered, in order to ensure that the network converges during the training process, and to avoid large network prediction errors due to the large difference in the magnitude of the input and output data, first the data have to be normalized before training [14]. After the ANN is trained, it is simulated and verified. The simulation verification results of each network are shown in Figure 4.
Because of the accurate measurement of *Fx* related data, they are regular, so the differences shown in the three different ANNs are not obvious. The decisive factors of *Fx* in BP-ANN, Elman-ANN and RBF-ANN are respectively 0.99564, 0.98278 and 0.99134. The performance of *Fy* in the three different ANNs is more different, and it is consistent with the trend of *Fx*. It can be seen from the prediction results that the decisive factors of *Fy* in the BP-ANN are 0.99564 and 0.96614, respectively, which are higher than the decisive factor of the RBF-ANN and the Elman-ANN. The errors of BP-ANN and Elman-ANN are generally smaller than those of RBF-ANN, but there are several special cases that exceed 30%. The training speed of the three ANNs are also different. In order to achieve the desired training goal, the RBF-ANN requires more neurons than the BP-ANN, so the training speed is also slower. The training of the BP-ANN is usually completed in one minute, while the average training time of the RBF-ANN is ten minutes. For the Elman-ANN, an additional layer in the network structure is used as a delay operator, so that the system has the ability to adapt to time-varying characteristics and can dynamically reflect the characteristics of dynamic systems. [15]

However, based on the load prediction through the strain method, the operating speed of the Elman-ANN is slower than that of the BP-ANN, it takes an average of twenty-one minutes. Generally speaking, the BP-ANN has good fitting degree, high prediction accuracy, small relative error, the most stable prediction result and is faster.

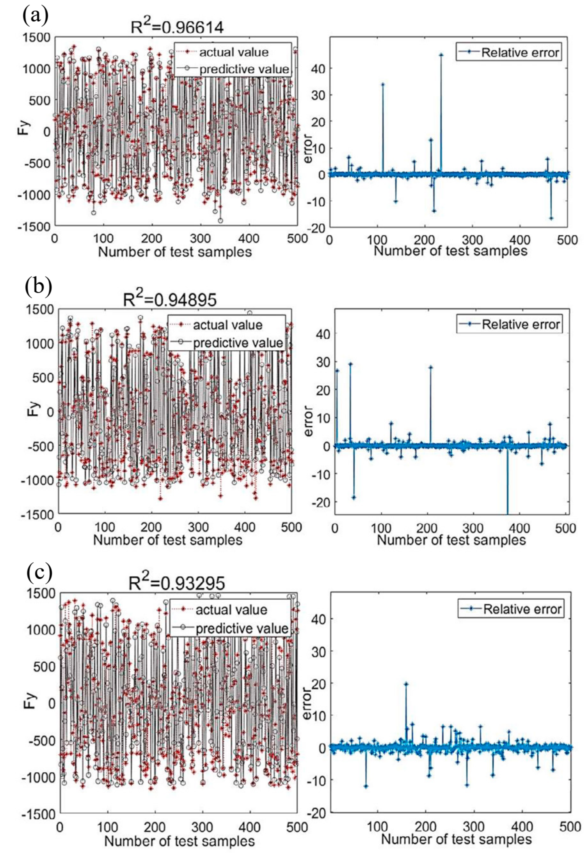

Fig. 4. Prediction results of three different ANN (a) BP; (b) Elman; (c) RBF.

RBF-ANN parameter adjustment is simple, the topology is compact, the convergence speed is fast and there is no local minimum problem, but it is not suitable for the problem of a large number of training samples [9][10]. Elman-ANN adds a bearer layer to the hidden layer, so that the system is adapt to the time-varying characteristics and enhance the global stability of the network, but the structure is more complicated [11][15]. BP-ANN has a simple topology, a high degree of nonlinearity and strong generalization ability, but it is relatively easy to fall into the local optima problem [7]. When solving the function approximation problem with the same accuracy requirements and training samples discussed in this article, the number of hidden layer neurons required by the RBF-ANN is higher than that of BP-ANN, making the structure of the RBF-ANN too large, resulting in an increased training time and no guarantee of accuracy. Elman-ANN also has the same problems as RBF-ANN due to its own computing characteristics and complex structure. BP-ANN neural network shows higher classification accuracy and convergence speed than RBF-ANN and Elman-ANN. Using GA for optimization avoids the local optima problem and enables the BP-ANN to obtain the global optima solution. [8][12][15]

3.3. Optimization and flowchart of back propagation artificial neural network

After comparing a first application of the three ANNs the faster and more accurate BP-ANN has been chosen and the optimization starts. The BP-ANN can contain one or more hidden layers. One disadvantage of single-layer perceptrons is that they can only classify linearly separable data sets. For this article, each non-linear output is determined by four non-linear inputs. It is a non-linear regression problem and uses an ANN with hidden layers. The hidden layer can abstract the features of the input data to a higher dimension, so that these originally non-linear features can be better linearized. Multiple hidden layers are actually multi-level abstractions of the input features, and the goal is to better divide the data of different features linearly. If there are enough neurons in a single hidden layer, the non-linear activation function can fit any function [16]. Therefore, an ANN with three layers, the input layer, the output layer and the hidden layer has been chosen. However, it has been theoretically proven that a network with a single hidden layer can achieve arbitrary non-linear mapping by appropriately increasing the number of neuronal nodes. Thus, this work uses a single hidden layer feedforward BP-ANN, which has strong non-linear mapping capabilities. This network consists of an input layer, a hidden layer and an output layer. The topology of the BP-ANN is shown in Figure 5. As mentioned in section 3.2 four strain data sets (*Q_O*, *Q_U*, *Q_L*, *Q_R*) are defined as the input and two data sets of the loads (*Fx*, *Fy*) as output. Accordingly, there are four neurons in the input layer and two neurons in the output layer. The number of neurons in the input layer, output layer and in the hidden layer jointly determine the number of weights and thresholds. As shown in Figure 5, the weights and thresholds in the BP-ANN have an important role.

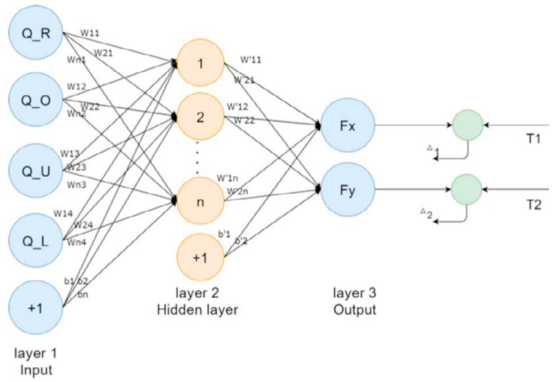

Fig. 5. The topology of the BP-ANN.

They are directly related to the accuracy of network prediction. So the number of hidden layer neurons is directly related to the convergence speed and accuracy of the ANN. For other parameters of the input layer and the output layer, the format and quality of the input and output have been varied. While ensuring that the input data are accurate and complete, due to the large data range, the data have been normalized to increase the convergence speed of the ANN and reduce the training time. Because the BP-ANN uses serial search, there is the possibility to fall into the local optima problem. The parallelism of the GA can make it easier to converge to the global best for the weights and thresholds. Each individual in the population contains the networks ownership and threshold. The individual calculates the individual fitness value through the fitness function. The GA is applied through the selection, crossover and mutation operations to find the optimal fitness value corresponding to the individual [17]. The BP-ANN uses GAs to get the optimal individual to assign the initial weight and threshold of the network, that is, to obtain the optimal weight and threshold. The optimization process of BP-ANN is shown in Figure 6.

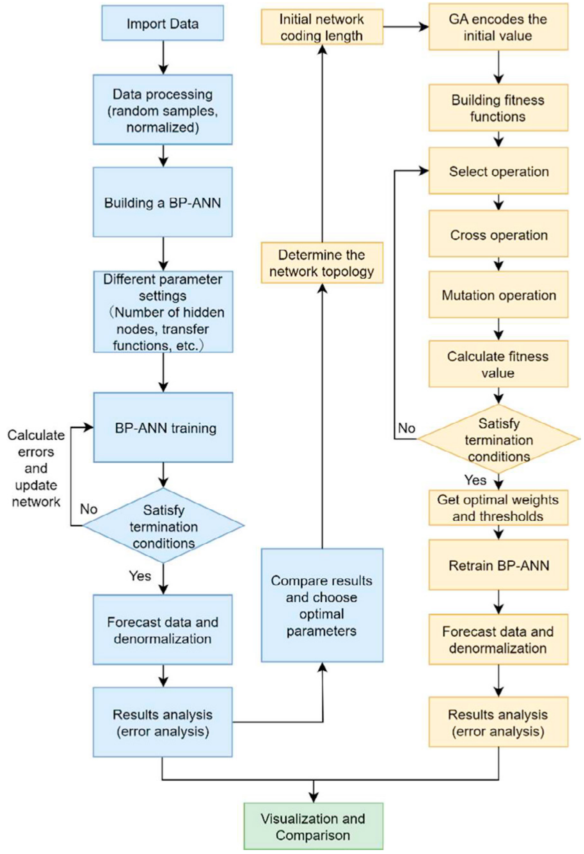

Fig. 6. The optimization process of BP-ANN.

3.3.1 Process of building back propagation artificial neural network and parameter optimization

The first thing to do is the blue highlighted part of the process in Figure 6, which is the parameter optimization of the BP-ANN. It mainly includes the selection and optimization of parameters such as the number of nodes in the hidden layer, transfer function, training function, and expected error value.

The number of hidden layer nodes has an impact on the performance of BP-ANN. Generally, a larger number of hidden layer nodes can bring better performance, but it may lead to excessive training time or overfitting. Here the number is estimated by using the following empirical formula [18][19]:

$$c = \sqrt{m+n} + a \quad (1)$$

$$c = 2\sqrt{(m+2)/n} \quad (2)$$

Among them, c is the number of hidden layer nodes, m and n are the number of neurons in the output and the input layer, and a is a constant between [0, 10]. After several trainings and tests of the BP-ANN, the optimal number of hidden layer nodes was finally determined to be six.

Subsequently there is a need to choose an appropriate transfer function and training method. The value of the sample itself has positive and negative values, so it is normalized to [-1, 1]. Matlab commonly used transfer functions are *logsig* (log-sigmoid), *tansig* (tanh-sigmoid) and *ReLU* (Rectified Linear Unit). Only the range of the *tansig* function is [-1, 1], the other two ranges from 0 to 1. In addition to the method of standard steepest descent, the BP-ANN has several improved training algorithms. The choice of training algorithm is related to the problem itself and the number of training samples. In general, for function approximation networks, the Levenberg-Marquardt (LM) algorithm has the fastest convergence speed and the mean square error is small. The Bayesian function has been used to modify the LM algorithm to make the networks generalization ability better and avoid overfitting. The disadvantages are a slow training speed and a large mean square error (MSE). As one of the termination conditions, limiting the maximum number of times that MSE of the verification sample does not fall, can avoid overfitting, so the training function is selected as *TrainLM*. It is also needed to set other training termination conditions, such as the maximum number of cycles, the expected MSE value etc., and satisfy any of them to end the training.

3.3.2 Process of genetic algorithm optimization with regard to weight optimization

The topology of the BP-ANN is defined by the parameters that have been determined before. From this, the GA is used to optimize the ANN to obtain the optimal weights and thresholds. The BP-ANN is retrained with the optimal value to obtain a new network. The elements of GA to optimize BP ANN include population initialization, fitness function, selection operation, cross operation and mutation operation.

As one of the main control parameters of the GA, the size of the population directly affects the final result and computational efficiency of the GA. The size of the group is too small, making the search space limited, which may lead to immature convergence. The size of the group is too large, and the number of fitness evaluations increases, which directly affects the evaluation efficiency of the algorithm. For a specific optimization problem, no theoretical result can tell us the specific population size. However, according to a large number of literature reports, the population size in practical applications is generally between ten and one hundred. In general, this value can fit the requirements of the problem [17]. The design of the fitness function is directly related to the selection operation in the GA. The calculation formula of the individual fitness value F is [17]:

$$F = k\left(\sum_{i=1}^{n} abs(y_i - o_i)\right) \quad (3)$$

Among them, n is the number of output nodes of the network, y is the expected output of the i-th node of the BP ANN, o is the predicted output of the i-th node, and k is a coefficient. There are many methods of GA selection operation, such as roulette and tournament. In this case, roulette is selected. This is a strategy based on fitness ratio selection. The selection probability of each individual i is p_i [17]:

$$f_i = k/F_i \quad (4)$$

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j} \quad (5)$$

In the formula, F_i is the fitness value of the individual i. Since the smaller the fitness value, the better, so the inverse of the fitness value is calculated before the individual selection. N is the number of individuals in the population and k is the coefficient. Because individuals use real number coding, the crossover operation method uses the real number crossover method. The k-th chromosome a_k and the l-th chromosome a_l have the following cross-operation method at the j position:

$$a_{kj} = a_{kj}(1-b) + a_{lj}b \quad (6)$$

$$a_{lj} = a_{lj}(1-b) + a_{kj}b \quad (7)$$

In these formula b is a random number between [0, 1]. In mutation operation we use the non-uniform mutation in the GA toolbox in Matlab. The non-uniform mutation changes one of the parameters of the parent based on a non-uniform probability distribution, selects the j-th gene of the i-th individual to mutate. After the GA optimization is over, we can get new prediction results, as shown in Figure 7.

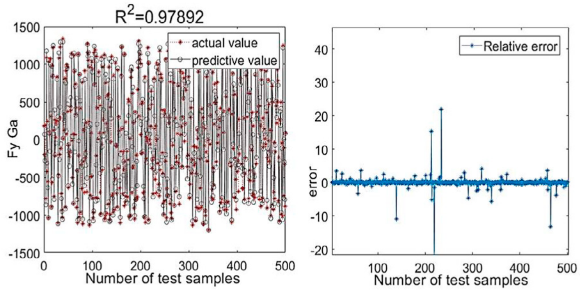

Fig. 7. Prediction results of BP-ANN after GA.

Because the amount of raw data is sufficient and relatively regular, there is not much room for improvement in the results of optimization. However, comparing with Figure 4, it can be seen that the BP-ANN optimized by the GA is more accurate in prediction, and the relative error is also improved. Although the prediction accuracy is very high. Figure 8 shows that the MSE of the validation and test sets not only does not increase, but is lower than the training set. This shows that the BP-ANN which is obtained does not appear to overfit in the training set. The best validation performance is at epoch 427, and at this time MSE is 0.0030465. In the following 6 epochs, the MSE of the test set does not continue to decline, and one of the termination conditions is satisfied, so the training ends.

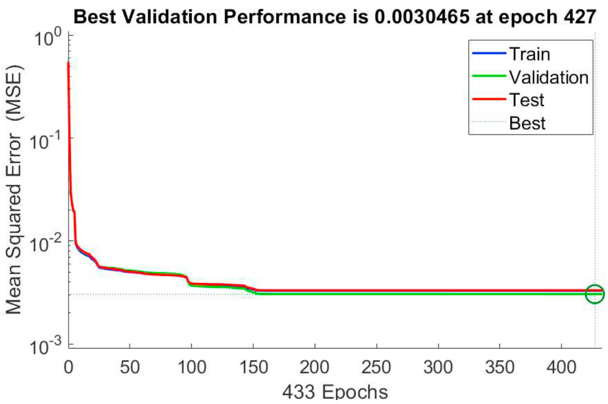

Fig. 8. Mean square error of each data set.

4. Conclusion and future work

The aim of this work was to evaluate the possibility of using ANNs to determine an analytically undetectable relationship between load and strain on structural components. For this purpose, a suspension arm has been equipped with strain gauges and subjected to load cases in a dynamic test bench, similar to the operating condition. The strain information was used as input and the force information as output data for three different ANNs. From first non-optimized networks, the one with the highest decisive factor and the lowest relative error was selected. This has been a BP-ANN. In a further step this network has been optimized with respect to its topology using Matlab and the GA. The final architecture of the BP-ANN provided a decisive factor of 0.979 and a maximum relative error of approximately 20%. The integration of sensor technology on structural components of a vehicle and generation of knowledge from sensor data is a method that is not state of the art. This article provides a flow chart for the development of a BP-ANN for structural components, which can be used to make load predictions. In future research work the authors will deal with further optimization measures. Data preparation plays a major role in this context. In the context of this work, it has not been checked whether there might have been double correlation to different force information due to missing decimal places of the strain information. Deleting this data or realizing a higher resolution of the sensor system could further improve the results. Furthermore, it is still an open question how far the developed BP-ANN can be used in real operating environments. For this purpose, in further steps the sensor technology and the ANN will be integrated into a car and the prediction accuracy of the ANN will be tested.

References

[1] Mozgova I, Yanchevskyi I, Gerasymenko M, Lachmayer R. Mobile automated diagnostics of stress state and residual life prediction for a component under intensive random dynamic loads, Procedia Manufacturing; 2018. Vol. 24, p. 210–215 DOI:10.1016/j.promfg.2018.06.037

[2] Mozgova I, Yanchevskyi I, Lachmayer R. Prediction of the Residual Life of a Component under Intensive Random Dynamic Loading within the Scope of Technical Inheritance, 15th International Design Conference; 21-24 May 2018, Dubrovnik, Croatia, p. 1643-1650.

[3] Lachmayer R, Mozgova I, Gottwald P. Formulation of Paradigm of Technical Inheritance, Proceedings of the 20th International Conference on Engineering Design (ICED15), Milan, Italy; 27.-30.07.2015

[4] Siqueira R, Altun O, Gembarski P, Lachmayer R. A delta-robot-based test bench for validation of smart products, 15th International Conference Dynamical Systems - Theory and Applications; 02-05 December 2019, Lodz, Poland

[5] Da-qi Z. The Research Progress and Prospects of Artificial Neural Networks[J]. Journal of Southern Yangtze University; 2004.

[6] Hammerstrom D. Working with neural networks. IEEE Spectrum, vol. 30, no. 7; 1993. p. 46-53.

[7] Liqun H. Artificial Neural Network Theory, Design and Application. Beijing: Chemical Industry Press; 2007.

[8] Goh ATC. Back-propagation neural networks for modeling complex systems. Artificial Intelligence in Engineering, Vol. 9, Iss. 3; 1995.

[9] Chen F. Back-propagation neural networks for nonlinear self-tuning adaptive control. IEEE Control Systems Magazine;1990. Vol. 10, no. 3, p. 44-48.

[10] Seshagiri S, Khalil HK. Output feedback control of nonlinear systems using RBF neural networks. IEEE Transactions on Neural Networks; 2000. Vol. 11, no. 1, p. 69-79.

[11] Koskela T, Lehtokangas M, Saarinen J, Kaski K. Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks. Tampere University of Technology. Electronics Laboratory. FIN-33101 Tampere, Finland.

[12] Shi-yong W, Tian-zhi W, Biao J, Xing-ping L. Optimization research of GA based on numerical calculation method. Computer Engineering and Design; 2009.

[13] Ding S, Su C, Yu J. An optimizing BP neural network algorithm based on GA; 2011. Artif Intell Rev 36, p. 153–162.

[14] Qu-wie Y, Cheng-hui S, Hua Z. Brief Analysis of Load Measurement of Wings on Large Commercial Jet. China Aviation Academic Conference; 2007.

[15] Lina R. Research on Medium-term Power Load Forecasting Model Based on Elman Neural Network. Lan Zhou: Lanzhou University of Technology; 2007

[16] Cybenko G, Approximation by Superpositions of a Sigmoidal Function. Math. Control Signals Systems; 1989. p.303-314.

[17] Rongmin Z, Yanfeng L. Pipe network optimization theory and technology: GA and Neural Network. Zheng Zhou: Yellow River Water Conservancy Press; 2002.

[18] Stathakiss D. How many hidden layers and nodes, International Journal of Remote Sensing; 2009. Vol. 30, No. 8, p. 2133–2147.

[19] Ming M. Matlab neural network principle and practical examples[M]. Beijing: Tsinghua University Press; 2013.