

Data traffic filtration of automated non-destructive testing based on an artificial neural network

Alisa Kosach^{1,*}, and Evgeny Kovshov¹

¹JSC «NIKIMT-Atomstroy», Moscow, Russian Federation

Abstract. The principle of constructing a universal software and hardware platform for the collection, processing and storage of data obtained as a result of non-destructive testing of products is considered. The platform also includes a neural network module for filtering data traffic, which has a wide potential for application: from image and multidimensional signal processing to data classification. The system of remote leakage control, built on the basis of a universal digital platform with artificial neural network, is described.

1 Introduction

Modern industrial enterprises are striving for a full-scale transition to digital production with the ubiquitous use of the industrial Internet of things and the management of the life cycle of products. Creation of flexible production systems, in particular, systems of non-destructive testing allows to significantly reducing the influence of the human factor on the results of control, to increase the reliability of the system and to ensure the accuracy of analysis of large unstructured data.

According to [1] all the problems solved by man from the standpoint of neuroinformation technologies can be conditionally classified into two large groups. The first group contains problems that have a known and definite set of conditions, on the basis of which it is necessary to obtain a clear, accurate, unambiguous answer on the known and definite algorithm. The second group includes tasks where it is not possible to take into account all the actual conditions on which the answer depends, but only an approximate set of the most important conditions can be identified. Since some of the conditions are not taken into account, the answer is inaccurate, approximate, and the algorithm for finding the answer can not be written accurately.

In solving the problems of the first group, you can use traditional application software based on an algorithm with a limited set of parameters that allows you to solve the problem.

To solve the problems of the second group, the use of neurotechnologies justifies itself in all respects, under the following conditions: first, the presence of a universal type of architecture and a single universal learning algorithm (no need to develop them for each type of tasks) and secondly, the presence of examples (training samples), on the basis of which neural networks are trained.

Currently, there is a wide range of the most popular among mathematicians and programmers of neural

networks such as: multilayer perceptron, Kohonen self-organizing maps, recurrent neural networks, convolutional neural network. All of them have their own distinctive features in topology, activation functions and methods of training, and of course in the mechanisms of software implementation. However, the limitation of the development of neural network algorithms is the high computational cost of their computer implementation. For the successful solution of this problem is used the organization of parallel (e.g. CUDA - Compute Unified Device Architecture) and distributed computing (e.g. SOA - Service-Oriented Architecture), in particular, on specialized hardware.

This article discusses a hardware and software platform consisting of a set of web services and hardware solutions for managing devices, collecting, processing and storing data. Software and hardware solutions based on artificial neural networks of various architectures and topologies are an important and universal part of the above mentioned platform, which can be used in the development of various automated industrial systems for most types of non-destructive testing, in particular, control, where the data is used in digital form: ultrasonic, radiographic and leakage testing.

2 Firmware

2.1. Architecture

The proposed intellectual platform is based on the service-oriented architecture of software solutions [2]. Due to its philosophy, SOA - is a modular approach to the development of application software, based on the use of distributed, loosely coupled standardized services. Services implement various application solutions and can be reused and combined.

When developing any information systems, modularity facilitates the software scaling and enables

* Corresponding author: KosachAA@atomrus.ru

the best use of the information system functions, while some services are still in development. With the growth of volumes of data and calculations, the construction of information systems based on SOA becomes the most preferable. In addition, SOA allows the use of external services and web services (for example, located in a private or public cloud).

2.2 Practical application

A model of an intelligent platform for an automated leakage control system was developed and tested with a mass-spectrometric method. The model included the collection and processing of data from technological devices of the lower level, the analysis of product validity and the automatic formation of technological documentation [3].

The intelligent platform is built on the basis of a service-oriented architecture, which makes it flexible and easily scalable, allows you to use virtual resources for calculations and storage of various information. To store the data received in the process of monitoring, DBMS with open source code PostgreSQL was used. The use of predefined software technologies [4] allows analysing and filtering data in the main software tools of e.g. Microsoft Office or its open source analogue. The automated control system for leakage control was developed in the open integrated programming environment NetBeans with using a pair of Java/C++ programming languages.

The application of the above solutions, based on automation of control and evaluation of the product's usefulness, allows shortening the time of the control. Due to the data storage in the DBMS, it is possible at any time to visualize the obtained dependencies and re-evaluate. The automation of the leakage control procedure excludes the influence of the human factor and increases the safety of the personnel involved.

3 Neural network

Tasks related to non-destructive testing almost always have several solutions and very often the so called "fuzzy" nature of the answer coincides with the way the result is issued by neural networks. In this regard, neural networks are used for classification in the field of computer diagnostics under non-destructive testing. Generation of optimal (adequate to solve the problem) neural network includes a sequence of main steps: the choice of the topology of the network, the choice of the activation function of the neurons of the hidden layer, the choice of the method of training and, in fact, network training.

To treat the data obtained in the process of leak testing, a neural network module was developed that performs the functions of normalization and data purification. The neuronet has the following topology (fig. 1): input neurons (the number varies depending on the data type), two hidden layers of two neurons and one bias neuron, output neuron that outputs the result of data classification (1 – "true" data, 0 – "false" data). The

neural network is trained on the basis of a test sample using the algorithm for back propagation of the error [5]. Try to ensure that lines are no thinner than 0.25 point.

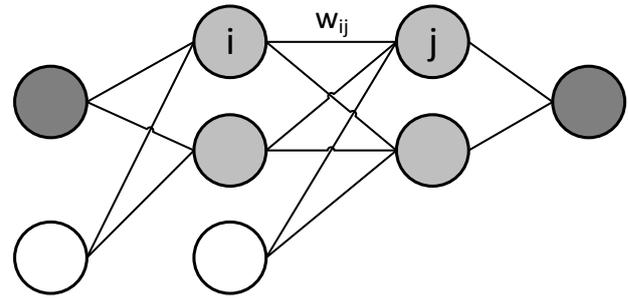


Fig. 1. Topology of the neuronet.

The bias neurons were placed in the neural network to shift the activation function of the neuron and, accordingly, to reduce the number of epochs required for its training. It should be noted that with an increase in the number of neurons in the hidden layers of the network, their weight connections in the learning process tend to zero, and additional neurons are redundant. Thus, to solve this problem of data traffic filtering, the topology of a neural network of this kind is the most appropriate.

When it teaching, the task is to minimize the error function:

$$E = \frac{1}{2} \sum_{j=1}^k (y_j - t_j)^2 \quad (1)$$

where y_j is the real value j of the network output; t_j is the desired value j of the network output; k is the number of outputs of the network. The minimum of the error function is determined by the stochastic gradient descent method. The activation function is:

$$f(x_j) = \frac{1}{1 + e^{-x_j}} \quad (2)$$

where x is the weighted sum at the input of the neuron:

$$x_j = \sum_{i=1}^k w_{ij} y_i \quad (3)$$

where w_{ij} are the weights of the connection between and neurons; y_i is the output of the previous layer neuron i .

At the beginning of training, the weights of the neuronal connections are initiated randomly in the interval $[-0.1; 0,1]$. Next, a training sample with known values at the outputs arrives at the neural network. For each sampling vector, the error value is calculated and an adjustment is made to the weights of the neuron connections according to the formulas:

$$w_{ij} = w_{ij-1} + \Delta w_{ij} \quad (4)$$

where w_{ij-1} is the weight of the connection between i and j neurons.

$$\Delta w_{ij} = -\eta \delta_j y_i + \mu \Delta w_{ij-1} \quad (5)$$

where δ_j is the error of the neuron in the layer j ; Δw_{ij-1} is the correction to the weight of the neuron connections made for the previous operation.

$$\delta_j = f'(x_j) \sum_{n=1}^k \delta_n w_{jn} \quad (6)$$

$$\delta_{out} = f'(x_{out})(t_{out} - y_{out}) \quad (7)$$

where δ_n is the error of the neuron in layer n ; n is the layer of neurons that stands after layer j ; out is the layer of output neurons.

The coefficients η and μ are introduced to control the training speed and the successful passage of the local minimum of the error function, respectively.

The main idea of this method is to spread the error signals from the outputs of the network to its inputs, that is, in the direction opposite to the direct propagation of signals in the normal mode of operation. After the repeated passage of the training sample (about 5000 epochs), the neural network is ready to process and clean incoming data.

Neural networks can be effectively applied in digital platform not only for data classification, but also for multidimensional digital signal recognition and image processing.

With the increase in the number of neurons, the size of the training sample and the complexity of the NDT tasks (for example, for image and sound recognition and processing etc.), when implementing a pilot industrial project of a digital platform for non-destructive testing of products and materials discusses to use specialized software libraries allow to effectively solve and implement applied software solutions as well as the Java Neural Network Framework [6].

For a full-scale implementation of the possibility of a parallel system for computing using a graphics card, the algorithm of configuration and direct operation with the neural network should be appropriately converted. Obviously, in this case, in addition to the computing resources of the central processor unit (CPU), CUDA – compatible graphics processor is used, for example, NVIDIA [7], implementing the architecture of parallel

computing CUDA [8, 9], which provides a specialized programming interface for calculations that are not directly related to raster and/or vector graphics.

Currently, the NVIDIA GPU Cloud (NGC) provides an easy access to the full catalogue of software for deep learning and high performance computing, optimized for the GPU [10]. The NGC repository includes containers with leading deep learning frameworks for neural networks, which are configured, tested, certified, and serviced by NVIDIA.

References

1. V.V. Parubets, O.G. Berestneva, D.V. Devyatykh, *Bulletin of the Tomsk Polytechnic University*, **320** (2012)
2. N. Bieberstein, S. Bose, M. Fiammante, *Service-Oriented Architecture (SOA) Compass: Business Value, Planning, and Enterprise Roadmap* (IBM Press, 2005)
3. A. Kosach, E. Kovshov, *Contemporary Engineering Sciences*, **10**, 367 (2017)
4. Microsoft Download Center. URL: www.microsoft.com
5. S. Osovskiy, *Neural networks for information processing* (Finance and Statistics, 2002)
6. Java Neural Network Framework. URL: <http://neuroph.sourceforge.net/index.html>
7. NVIDIA Developer. <https://developer.nvidia.com/>
8. CUDA Toolkit Documentation v. 9.1.85. URL: <https://docs.nvidia.com/cuda/>
9. NVIDIA Accelerated Computing. Deep learning. <https://developer.nvidia.com/deep-learning>
10. NVIDIA GPU Cloud. <https://www.nvidia.ru/gpu-cloud/>