

Using Artificial Neural Networks in Solving Heat Conduction Problems

M. Heidari¹, M. Garshasbi^{2*}

¹Department of Engineering, Toyserkan Branch, Islamic Azad University, Toyserkan, Iran.

²School of Mathematics, Iran University of Science and Technology, Tehran, Iran.

Received September 2014; Revised February 2015; Accepted February 2015

Abstract — This paper solves an important class of inverse heat conduction problems by using a back propagation neural network (BPN) to identify the unknown boundary conditions. The feasibility of the proposed method is examined in a series of numerical simulations. The results show that the proposed neural network method successfully predicting the unknown parameters with acceptable error.

Keywords — Feed-forward neural network, Inverse heat conduction problems, Back propagation neural network.

1. INTRODUCTION

In forward heat conduction problems the heating characteristics, the boundary condition and the initial conditions of a body are known and are used to establish the internal temperature field. Conversely, in inverse heat conduction problems (IHCPs), experimental temperature measurement are taken at various points in the interior of a body and are used to estimate the unknown boundary conditions existing at the external surface. IHCPs are mathematically ill-posed in the sense that the existence, uniqueness and stability of their solutions cannot be assured (Beak, 1985). IHCPs are generally solved using some form of numerical technique. Classical approaches include space marching (Shin, 2000) the single future time step method, the function specification method, the regularization method and the trial function method (Beak, 1985). Since the 1970s, computer science and technology have advanced rapidly and hence contemporary researchers generally solve IHCPs using numerical methods such as the finite element method, the finite difference method (FDM) and Genetic algorithm (Raudensk, 1995).

The rapid development of artificial neural network technology in recent years has led to an entirely new approach for solution of IHCPs (Raudensk, 1995). Neural networks are artificial intelligence systems which mimic the biological processes of a human brain by using non-linear processing units to simulate the functions of biological neurons.

2. DESCRIPTION OF FEED-FORWARD NEURAL NETWORKS

Artificial neural networks are modeled from the human brain and neural systems, which are suitable tools to solve large-scale problems. There are many references in theory and applications, modeling, algorithms, design, architecture and mathematics of neural networks. Artificial neural networks consist of calculation units called neurons. Every neuron has some real-valued inputs. Each input is multiplied with the corresponding synaptic (neural) coefficient and the sum of all these products is added to a value called bias. Finally, activation (instigation) function affects this sum and determines the real-valued output of the neuron, feed forwardly. In most of applications, activation functions are hyperbolic tangent or logistic function in the form of $1/(1+e^{-x})$. Learning is the process of determining the optimal values for the synaptic weights. If this procedure works based on minimization of the error between the computed and the known desired output values, then it is called supervised learning. In contrast, there are no desired output values for the neuron in unsupervised learning. The main power of the neural computing arises from too many neurons that connect and adapt in order to form networks. The simplest topology for these networks is a group of neurons that are organized in one layer, called single layer network or perceptron. The architecture of multilayer neural networks is formed by a group of one layer networks. In multi-input–single-output feed forward multilayer neural networks, there is no recursion (feedback loop) and outputs of the neurons form the inputs for the next layer. Such networks are called multilayer perceptrons (McClelland, 1986), which contain at least three layers naming input, output and inner (hidden) layers (Fig. 1).

Accordingly, artificial neural networks can make a nonlinear mapping from the inputs to the outputs of the corresponding system (Hornik, 1990). This is suitable for analyzing the systems described by initial-boundary value problems that have no analytical solutions or their analytical solutions are not computable easily.

One of the applications of the multilayer perceptrons is the ability of global approximation for real-valued multi-variable

*Corresponding author's email: m_garshasbi@iust.ac.ir

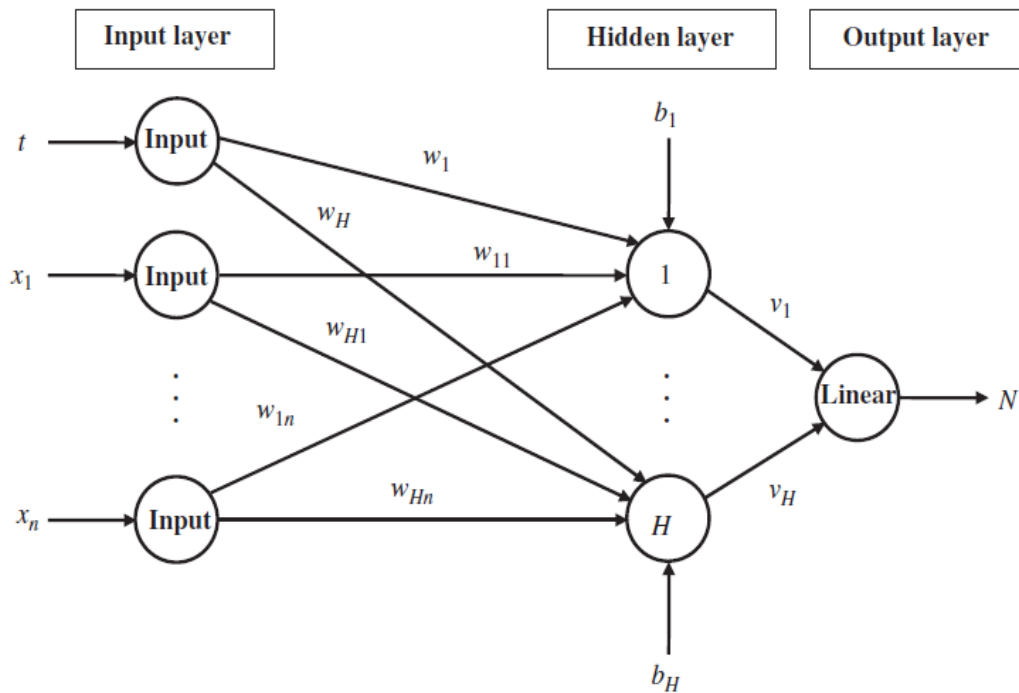


Figure 1. Structure of the general feed forward single hidden layer perceptron

functions in a closed analytical form. Namely such neural networks are universal approximators. They can be trained to approximate any Borel measurable function defined on a hypercube with any desired precision. The least conditions for the activation functions in the inner layers are nonlinearity and integrability. The approximation precision is related to the neurons in the hidden layers and not related to the number of the hidden layers (Shekari, 2009).

3. FORMULATION FOR INVERSE PROBLEMS

In this section an inverse heat conduction problem is considered as the following problem

$$\frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + g(x, t) = pc \frac{\partial T}{\partial t}, \quad 0 \leq x \leq L, \quad t \geq 0 \tag{1}$$

$$T(x, 0) = f(x), \quad 0 \leq x \leq L \tag{2}$$

$$T(0, t) = p(t), \quad \text{or} \quad -k \frac{\partial U}{\partial t} \Big|_{(0,t)} = \Phi(t), \quad t \geq 0 \tag{3}$$

$$T(L, t) = q(t), \quad \text{or} \quad -k \frac{\partial U}{\partial t} \Big|_{(L,t)} = \Psi(t), \quad t \geq 0 \tag{4}$$

where k denotes the heat coefficient, g is the source term and p and c represent the heat capacity coefficients. All these functions and parameters are considered to be known. The determination of the boundary function $q(t)$ is the goal of this work using neural network approach.

4. MAPPING NEURAL NETWORK ARCHITECTURE

A feed forward network with n -input units and m -output units can perform mapping from an n -dimensional cube \mathbb{R}^n to an m -dimensional cube \mathbb{R}^m . According to the original theorem proposed by Kolmogorov (Patterson, 1998): For all $n \geq 2$, and for any continuous real function g of n variables in the domain $[0, 1]^n$, $g : [0, 1]^n \rightarrow \mathbb{R}$, there exist $2n + 1$ continuous, monotonously increasing one-variable functions in $[0, 1]$, by which g can be reconstructed according to the following equation:

$$g(u) = \sum_{j=1}^{2n+1} \Psi_j \left(\sum_{i=1}^n \varphi_{ij}(u_i) \right) \tag{5}$$

where Ψ_j are continuous functions with one variable. Many authors have attempted to improve this theorem since its original

formulation, most commonly by replacing \varnothing_{ij} with $b_i \varnothing_{ij}$ where b_i is a constant. Therefore, Eq. (5) can be rewritten as

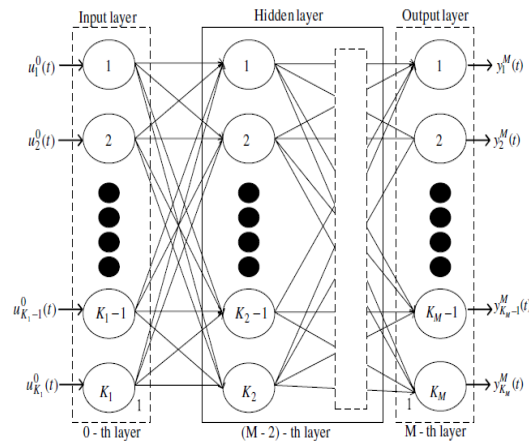


Figure 2: Structure of multilayered neural network.

$$g(u) = \sum_{j=1}^{2n+1} \Psi_j \left(\sum_{i=1}^n b_i \varphi_j(u_i) \right) \quad (6)$$

Eq. (6) corresponds to a three-layered feed forward network architecture with a single output. Based on the theoretical foundation above, and using a more complex proof process (Antognetti, 1991), the following theorem can be obtained:

Let $\varphi(u)$ be a non-constant, bounded and monotonously increasing continuous function. There exist an integer k and sets of real constants c_i , θ_i and w_{ij} , where $(i = 1, 2, \dots, k)$, $(j = 1, 2, \dots, k)$ such that expression:

$$\hat{g}(u_1, u_2, \dots, u_n) = \sum_{i=1}^k c_i \varphi \left(\sum_{j=1}^n w_{ij} u_j - \theta_i \right) \quad (7)$$

can be define to meet

$$\max |g(u_1, u_2, \dots, u_n) - \hat{g}(u_1, u_2, \dots, u_n)| < \varepsilon \quad (8)$$

where w_{ij} corresponds to the neuron weight. From the theoretical description above, for the condition $\varepsilon > 0$, there exists a three-layered feedforward network in which the activation function of the hidden layer is $\varphi(u)$. The activation function of the input layer is non-linear, while that of the output layer is linear, and the three-layered feed forward network given in Eq. (7) satisfies Eq. (8). The theorem presented above provides the foundation for applying the BPN network to the solution of IHCPs.

5. BACK PROPAGATION NEURAL NETWORK

Fig2 illustrate a general BPN network. As shown, this network is a feed-forward, fully connected hierarchical M-layered network consisting of an input layer, M-2 hidden layers and output layer. If the kth unit in the Mth layer is denoted by (M,K), the state variable u_k^M for this unit and its output signal y_k^M to the units in the next layer (M+1, k) can be written as follows:

$$u_k^M = \sum_j (w_{k,j}^{M,M-1} u_j^{M-1} + \theta_k^M) \quad (9)$$

$$y_k^M = \varphi_k^M(u_k^M) \quad (10)$$

Here, $w_{k,j}^{M,M-1}$ is the connection strength between units (M, k) and $(M-1, j)$, and θ_k^M and $\varphi_k^M(\bullet)$ are the bias and activation functions of unit (M, k) , respectively. The output signal y_k^M is transmitted to all units in the next M_{th} layer.

In the current study, the BPN network is used to solve various IHCPs. The input parameters, $u^0 = (u_1^0, u_2^0, \dots)$, to the neural network are the temperature data at specified points in the interior of the object of interest, while the outputs of the network, $y^M = (y_1^M, y_2^M, \dots)$, are parameters relating to the boundary conditions (Deng, 2005).

6. NUMERICAL EXPERIMENTS

To confirm the validity of the proposed method two test case problems are investigated. In all cases we used a neural network having one hidden layer.

Example1: The goal is estimating the boundary condition $q(t)$ by using the extra condition $T((n - 1) * h, t_j)$ about 100 data training:

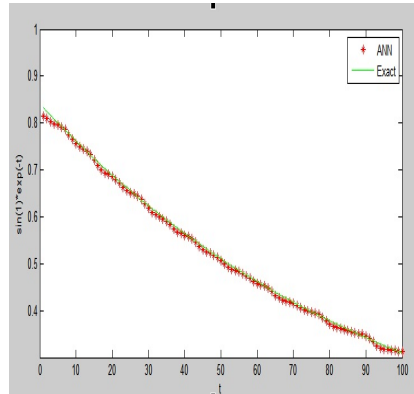


Figure 3: Comparison of exact an ANN for Example1.

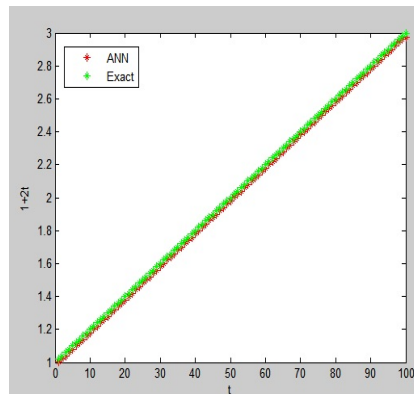


Figure 4: Comparison of exact an ANN for Example2.

Where $j = 1 / m, h = 1 / n, t_j = j * m$ and m, n are data training.

$$\begin{aligned} T_t(x, t) &= T_{xx}(x, t), \\ T(x, 0) &= \sin(x), \quad 0 \leq x \leq 1 \\ T(0, t) &= 0, \quad 0 \leq t \leq 1 \\ T(1, t) &= q(t), \quad 0 \leq t \leq 1 \end{aligned}$$

The exact solution is $T(x, t) = \sin(x)e^{-t}, q(t) = \sin(1)e^{-t}$. Figure 3 demonstrates a comparison between the exact and approximate solution for $q(t)$.

Example2: The goal is estimating the boundary condition $q(t)$ by using the extra condition $T((n - 1) * h, t_j)$ about 100 data training:

Where $j = 1 / m, h = 1 / n, t_j = j * m$ and m, n are data training.

$$\begin{aligned} T_t(x, t) &= T_{xx}(x, t), \\ T(x, 0) &= x^2, \quad 0 \leq x \leq 1 \\ T(0, t) &= 2t, \quad 0 \leq t \leq 1 \\ T(1, t) &= q(t), \quad 0 \leq t \leq 1 \end{aligned}$$

The exact solution is $T(x, t) = x^2 + 2t, q(t) = 1 + 2t$. Figure 4 demonstrates a comparison between the exact and approximate solution for $q(t)$.

CONCLUSION

In this work, the feed-forward neural networks are used to solve a class of inverse heat conduction problems. The neural
 1813-713X Copyright © 2014 ORSTW

network here allows us to achieve good approximation without wasting memory space and computational time and therefore reducing the complexity of problem, due to parallel structure of the network. The numerical result demonstrate the ability of the proposed method. This approach can be extended to solve many complex problems such as nonlinear inverse problems.

REFERENCES

1. Beck, J.V., Blackwell, B., and Clair, C.R.ST. (1985) . Inverse Heat Conduction Ill-posed Problems, *Wiley, New York*, 1-3.
2. Shin, M.S., and Lee, J.W. (2000). Prediction of the inner wall snap of an eroded furnace by the nonlinear inverse heat conduction technique, *JSME International Journal. Series B, Fluids and Thermal Engineering*, 43 (4):544-549.
3. Raudensky, M., Woodbury, K.A., Kral, J., and Brezina, T. (1995). Genetic algorithm in solution of inverse heat conduction problems, *Numerical Heat Transfer Fundamentals*, 28 (3): 293-306.
4. Raudensky, M., Horsky, J., and Krrjsa, J. (1995). Usage of neural network for coupled parameter and function specification inverse heat conduction problem, *International Communications in Heat and Mass Transfer*, 22 (5): 661-670
5. McClelland, J.L., and Romelhart, D.E. (1986). Distributed Processing Exploration in the Microstructure of cognition, *MIT Bradford Press, Cambridge MA*.
6. Hornik, K., Stinchcombe, M., and White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feed forward networks, *Neural Networks*, 3:551-560.
7. Shekari Beidokhti, R., and Malek, A. (2009). Solving Initial-boundary value problems for systems of partial differential equations using neural networks and optimization techniques, *Journal of Franklin Institute*, 346:900-901.
8. Deng, S., and Hwang, Y. (2005). Applying neural networks to the solution of forward and inverse heat conduction problems, *International Journal of Heat and Mass Transfer*, 49 (25-26):4732-4750.