Structural Engineering and Mechanics, Vol. 18, No. 4 (2004) 441-459
DOI: http://dx.doi.org/10.12989/sem.2004.18.4.441

Prediction of contact lengths between an elastic layer and two elastic circular punches with neural networks

Talat Şükrü Özşahin†, Ahmet Birinci† and A. Osman Çakıroğlu‡

Karadeniz Technical University, Civil Engineering Department, 61080, Trabzon, Turkey

(Received January 9, 2004, Accepted June 16, 2004)

Abstract. This paper explores the potential use of neural networks (NNs) in the field of contact mechanics. A neural network model is developed for predicting, with sufficient approximation, the contact lengths between the elastic layer and two elastic circular punches. A backpropagation neural network of three layers is employed. First contact problem is solved according to the theory of elasticity with integral transformation technique, and then the results are used to train the neural network. The effectiveness of different neural network configurations is investigated. Effect of parameters such as load factor, elastic punch radii and flexibilities that influence the contact lengths is also explored. The results of the theoretical solution and the outputs generated from the neural network are compared. Results indicate that NN predicted the contact length with high accuracy. It is also demonstrated that NN is an excellent method that can reduce time consumed.

Key words: contact length; elasticity; elastic layer; elastic punch; neural network.

1. Introduction

The human brain is the most sophisticated biological neural network which is often much more efficient, adaptable and tolerant than conventional computers in the fields of recognition, control and learning. Neural networks are intended to mimic the behavior of biological learning and decision making process without trying to be biologically realistic in detail. Artificial neural networks represent simplified models of human brain.

There are two principal functions of artificial NNs. One is the input-output mapping or feature extraction. The other is pattern association or generalization. The mapping of input and output patterns is estimated or learned by the NN with a representative sample of input and output patterns. The generalization of the NN is an output pattern in response to an input pattern based on the network memories that function like the human brain. Therefore, a NN can learn patterns from a sample data set and determine the class of new data based on previous knowledge. NNs derive its computing power through its massively parallel distributed structure. Important characteristic that artificial NN share with biological neural system is fault tolerance. NNs can be designed to be

† Doctor
‡ Professor

insensitive to small damage to the network and the network can be retrained in cases of significant damage.

Because of technology's enormous potential, an increased number of researchers have concentrated their efforts on neural networks. Several articles have summarized neural networks and their potential. Many impressive results have been produced. Several authors have used NNs in engineering applications.

Anderson *et al.* (1997) performed a series of tests and results have been used to train an artificial neural network (ANN) to predict bi-linear moment-rotation characteristics for minor-axis beam-to-column connections that govern restraint to the columns against buckling. A method to predict 28-day compressive strength of concrete by using multi-layer feed-forward neural networks was proposed based on the inadequacy of present methods dealing with multiple variable and nonlinear problems by Hong-Guang and Ji-Zong (2000). The authors collected the data from extensive and detailed laboratory work. They chose 14 test parameters (inputs) to be the factors that influence the concrete strength (output). Another study on concrete strength prediction by means of neural network is done by Lai and Serra (1997). They developed a model for predicting the compressive strength of cement conglomerates. Seibi and Al-Alawi (1997) explored the potential use of artificial neural networks in the field of fracture mechanics. They used an ANN model to predict the fracture toughness K_c of an aluminum alloy based on experimental data and developed relationships between K_c and crack geometry, specimen dimensions and operating temperature. A NN approach is developed to determine the crack opening load from differential displacement signal curves by Kang and Song (1998). In order to examine the measurement accuracy and precision of the NN method, they performed computer simulation for various combinations of crack opening levels and signal-to-noise ratios. Chandrashekhara *et al.* (1998) developed a method to determine the contact force on laminated composite plates subjected to low velocity impact using the FEM and a NN. They trained the NN using the contact force and strain histories obtained from finite element simulation results.

The effectiveness of different neural network configurations for estimating contact force is investigated. Goh (1994) presented a neural network to predict the friction capacity of piles in clays. The neural network was trained with field data of actual case records. Shahin *et al.* (2000) explored the possibility of using NNs to predict the settlement of shallow foundations on cohesionless soils. In this work data records were used for modeling. The results of the ANN were compared with three of the most commonly used traditional methods. Ural and Saka (1998) used NNs to analyse liquefaction. A NN model for detection of natural periods of vibrations of prefabricated, medium height buildings proposed by Kuźniar *et al.* (2000). They also used NN technique to simulate dynamic response at selected floor of one of the analysed buildings subjected to seismic loading induced by explosives in a nearby quarry. Both the training and testing patterns were formulated on the basis of measurements performed on actual structures. Waszczyszyn and Bartczak (2002) used NNs for predicting the buckling loads for axially compressed cylindrical shells with initial geometrical imperfections. Study based on the measured imperfections and tests on laboratory shell specimens gathered in the Imperfection Data Bank at the DELFT University of Technology. Lee and Han (2002) developed an efficient NN based models for the generation of artificial earthquake and response spectra. They proposed five NN based models for replacing traditional processes. They showed that the procedure using NN based models is applicable to generate artificial earthquakes and response spectra.

In this study, a NN approach is proposed dealing with the solution of frictionless contact problem of an elastic layer loaded by means of two circular elastic punches. A method is presented to determine contact lengths automatically using a NN that is capable of learning by training. First, contact problem is solved according to the theory of elasticity with integral transformation technique, then the data set obtained from the first solution is used to train ANN.

2. Contact problem

A way of choosing the patterns representing the characteristics of the structure which are to be used as the input to neural network is one of the most important subjects in this approach. The patterns for NN were prepared according to the theory of elasticity with integral transformation technique.

General methods for the solution to the classical contact problems are given by Galin (1961) and Ufliand (1965). Many researchers also made important contributions to the solution of special contact problems with improvements in computer technology and developments in new solution techniques. Contact problems have applications in a variety of structures of practical interests. Railways, roads, foundations, tanks, silos, rolling mills, balls and rolling bearings are some application areas of contact problems in engineering mechanics.

A long layer has been one of the most widely studied problems in contact mechanics. The importance of the problem lies in the fact that its geometry approximates a very common structural component. The elastic layer which is resting on an elastic or rigid foundation is considered in (Erdogan and Ratwani 1974, Civelek and Erdogan 1975, Geçit and Gökpınar 1985, Dempsey *et al.* 1990, Jaffar 1993). The examples for the contact problems which the load is transmitted to the elastic layer by the rigid or elastic punches can be found in (Civelek and Erdogan 1974, Adams and Bogy 1976, Artan and Omurtag 2000, Lan *et al.* 1996). In (Geçit 1986), friction isn't taken into account. The separation between contact surfaces are examined in (Birinci and Erdol 2003, Comez *et al.* 2003, Birinci and Erdol 2001, Çakıroğlu *et al.* 2001).

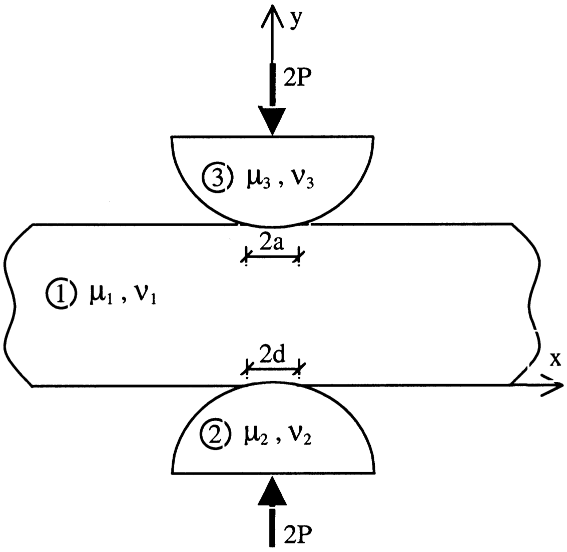


Fig. 1 Elastic layer loaded by means of two elastic circular punches

A homogeneous, isotropic, elastic layer of thickness h loaded by means of two circular elastic punches is considered according to the theory of elasticity (Fig. 1). The magnitude of the load applied to the elastic layer by each punch is $2P$. Although the elastic layer lies through the region $-\infty < x < \infty$, because of a plane of symmetry, the calculations are done in the region $0 \le x < \infty$.

The contact problem is solved under the assumptions that the contact along the interfaces are frictionless and only compressive normal tractions can be transmitted across the contact surfaces, and the local radii of the curvatures R_1 and R_2 of the elastic punches are sufficiently large compared to the length of contacts so that in expressing the surface displacements of the punches in the contact areas in terms of the contact pressures $P_1(x)$ and $P_2(x)$, the mediums 2 and 3 may be approximated by elastic half-planes (Civelek and Erdogan 1974).

Displacements of each elastic media expressed as the Fourier sine and Fourier cosine transforms of the unknown functions. Necessary derivatives of these functions are substituted into Navier equations and an ordinary differential equation system is obtained. This equation system is solved and general expressions for displacements and stresses are determined in terms of unknown constants. Unknown constants are determined from the following boundary conditions;

$$\tau_{xy_1}(x, 0) = 0, \qquad 0 \le x < \infty \tag{1}$$

$$\tau_{xy_2}(x, 0) = 0, \qquad 0 \le x < \infty \tag{2}$$

$$\tau_{xy_1}(x, h) = 0, \qquad 0 \le x < \infty \tag{3}$$

$$\tau_{xy_3}(x, h) = 0, \qquad 0 \le x < \infty \tag{4}$$

$$\sigma_{y_1}(x, h) = -P_1(x), \qquad 0 \le x < a \tag{5}$$

$$\sigma_{y_1}(x, 0) = -P_2(x), \qquad 0 \le x < d \tag{6}$$

$$\sigma_{y_1}(x, 0) = \sigma_{y_2}(x, 0), \qquad 0 \le x < \infty \tag{7}$$

$$\sigma_{y_1}(x, h) = \sigma_{y_3}(x, h), \qquad 0 \le x < \infty \tag{8}$$

$$\frac{\partial[v_1(x, h) - v_3(x, h)]}{\partial x} = f_1(x), \qquad 0 \le x < a \tag{9}$$

$$\frac{\partial[v_1(x, 0) - v_2(x, 0)]}{\partial x} = f_2(x), \qquad 0 \le x < d \tag{10}$$

where a and d are the half of the unknown contact lengths, unknown functions $P_1(x)$ and $P_2(x)$ are the contact pressures. Functions $f_1(x)$ and $f_2(x)$ define the derivatives of upper and lower circular punch profiles with radii R_1 and R_2 respectively.

$$f_1(x) = x(R_1^2 - x^2)^{-1/2} \tag{11}$$

$$f_2(x) = -x(R_2^2 - x^2)^{-1/2} \tag{12}$$

Displacement and stress expressions are substituted into boundary conditions (1)-(8) and unknown constants are determined in terms of functions $P_1(x)$ and $P_2(x)$.

After some simple manipulations, one may obtain the following pair of singular integral equations from boundary conditions (9) and (10).

$$-\frac{1}{\pi}\int_0^a P_1(t)dt\left[\left[-\frac{(1+\kappa_3)}{2\omega}-\frac{(1+\kappa_1)}{2}\right]\left[\frac{1}{t+x}-\frac{1}{t-x}\right]+\int_0^\infty k_1(x,t)d\alpha\right]$$

$$-\frac{1}{\pi}\int_0^d P_2(t)dt\int_0^\infty k_2(x,t)d\alpha = 2\mu_1 f_1(x) \tag{13}$$

$$-\frac{1}{\pi}\int_0^a P_1(t)dt\int_0^\infty k_3(x,t)d\alpha-\frac{1}{\pi}\int_0^d P_2(t)dt\left[\left[\frac{(1+\kappa_2)}{2\theta}+\right.\right.$$

$$\left.\left.\frac{(1+\kappa_1)}{2}\right]\left[\frac{1}{t+x}-\frac{1}{t-x}\right]+\int_0^\infty k_4(x,t)d\alpha\right] = 2\mu_1 f_2(x) \tag{14}$$

where $\omega = \mu_3/\mu_1$, $\theta = \mu_2/\mu_1$. $k_i(x,t)(i = 1, \ldots, 4)$ are the Fredholm kernels of singular integral equations and they are given in the Appendix. μ_i and κ_i $(i = 1, \ldots, 3)$ are the elastic constants, $\kappa_i = (3-4\gamma_i)$ for plane strain, $\kappa_i = (3-\gamma_i)/(1+\gamma_i)$ for plane stress, γ_i is the Poisson's ratio. γ_i is taken 0.2~0.3 in this study. The elastic layer is in equilibrium so one has

$$\int_0^a P_1(t)dt = P \tag{15}$$

$$\int_0^d P_2(t)dt = P \tag{16}$$

As $\alpha \to 0$, it is seen that kernels $k_i(x,t)(i = 1, \ldots, 4)$ behave as α^{-1}, therefore Fredholm kernels in Eqs. (13) and (14) involving the integrals of $k_i(x,t)(i = 1, \ldots, 4)$ are not bounded. On the other hand, assuming ε to be very small, kernels may be written in the following form.

$$\int_0^\infty k_i(x,t)d\alpha = \int_0^\varepsilon k_i(x,t)d\alpha + \int_\varepsilon^\infty k_i(x,t)d\alpha \tag{17}$$

and kernels $k_i(x,t)(i = 1, \ldots, 4)$ are replaced by their series expansions around $\alpha = 0$ in Eqs. (13) and (14) (Civelek and Erdogan 1974).

Index of the integral Eqs. (13) and (14) are –1, hence the solution of singular integral equations may be expressed as

$$g_i(s_i) = G_i(s_i)(1-s_i^2)^{1/2}, \quad (i = 1, 2) \tag{18}$$

Gauss-Chebyshev integration formulas in (Erdogan and Gupta 1972) allow Eqs. (13)-(16) to be replaced by the following algebraic equations

$$-\sum_{i=1}^{n}\frac{(1-s_{1_i}^{2})}{(2n+1)}G_1(s_{1_i})a\Bigg[\Bigg[-\frac{(1+\kappa_3)}{2\omega}-\frac{(1+\kappa_1)}{2}\Bigg]\Bigg[\frac{1}{a(s_{1_i}+w_{1_j})}-\frac{1}{a(s_{1_i}-w_{1_j})}\Bigg]+$$

$$\Bigg[\frac{3(1+\kappa_1)}{h^3}(a^2s_{1_i}^2)(aw_{1_j})\varepsilon-\frac{(1+\kappa_3)}{2\omega}(aw_{1_j})\varepsilon^2+O(\varepsilon^3)\Bigg]+\int_{\varepsilon}^{\infty}k_1(w_{1_j},s_{1_i})\frac{dz}{h}\Bigg]$$

$$-\sum_{i=1}^{n}\frac{(1-s_{2_i}^{2})}{(2n+1)}G_2(s_{2_i})d\Bigg[\Bigg[-\frac{3(1+\kappa_1)}{h^3}(d^2s_{2_i}^2)(aw_{1_j})\varepsilon+O(\varepsilon^3)\Bigg]+\int_{\varepsilon}^{\infty}k_2(w_{1_j},s_{2_i})\frac{dz}{h}\Bigg]$$

$$=2\frac{\mu_1}{P/h}\,f_1(w_{1_j}),\qquad (j=1,\ldots,n+1)\tag{19}$$

$$-\sum_{i=1}^{n}\frac{(1-s_{1_i}^{2})}{(2n+1)}G_1(s_{1_i})a\Bigg[\Bigg[\frac{3(1+\kappa_1)}{h^3}(a^2s_{1_i}^2)(dw_{2_j})\varepsilon+O(\varepsilon^3)\Bigg]+\int_{\varepsilon}^{\infty}k_3(w_{2_j},s_{1_i})\frac{dz}{h}\Bigg]-$$

$$\sum_{i=1}^{n}\frac{(1-s_{2_i}^{2})}{(2n+1)}G_2(s_{2_i})d\Bigg[\Bigg[\frac{(1+\kappa_1)}{2}+\frac{(1+\kappa_2)}{2\theta}\Bigg]\Bigg[\frac{1}{d(s_{2_i}+w_{2_j})}-\frac{1}{d(s_{2_i}-w_{2_j})}\Bigg]+$$

$$\Bigg[-\frac{3(1+\kappa_1)}{h^3}(d^2s_{2_i}^2)(dw_{2_j})\varepsilon+\frac{(1+\kappa_2)}{2\theta}(dw_{2_j})\varepsilon^2+O(\varepsilon^3)\Bigg]+\int_{\varepsilon}^{\infty}k_4(w_{2_j},s_{2_i})\frac{dz}{h}\Bigg]$$

$$=2\frac{\mu_1}{P/h}\,f_2(w_{2_j}),\qquad (j=1,\ldots,n+1)\tag{20}$$

$$\sum_{i=1}^{n}\pi\frac{(1-s_{1_i}^{2})}{(2n+1)}G_1(s_{1_i})\frac{a}{h}=1\tag{21}$$

$$\sum_{i=1}^{n}\pi\frac{(1-s_{2_i}^{2})}{(2n+1)}G_2(s_{2_i})\frac{d}{h}=1\tag{22}$$

where

$$s_{1_i}=s_{2_i}=\cos\left(\frac{\pi i}{2n+1}\right),\qquad (i=1,\ldots,n)\tag{23}$$

$$w_{1_j}=w_{2_j}=\left(\frac{\pi}{2}\left(\frac{2j-1}{2n+1}\right)\right),\qquad (j=1,\ldots,n+1)\tag{24}$$

$\frac{\mu_1}{P/h}$ is named as load factor. The $(n+1)^{\text{th}}$ equations in (19) and (20) are zero. So equations given by (19)-(22) constitute a system of (2n + 2) equations for (2n + 2) unknowns; $G_1(s_{1_i})$, $G_2(s_{2_i})$, $(i=1,\ldots,n)$, a and d. The problem is highly nonlinear in a and d, and these unknowns are determined from a time consuming interpolation scheme.

3. Neural network approach

The human brain is the most complex computing device known to man. The brain's powerful thinking, remembering and problem solving capabilities have inspired many scientists to attempt computer modeling of its operations. The brain consists of billions of neurons which are densely interconnected. Each neuron is a micro-processing unit built up of three parts; the cell body, the dentrites and the axon. The axon splits up and connects to dentrites of other neurons through junctions referred to as synapses. A neuron receives and combines signals from other neurons through the dentrites, and if the combined signal is strong enough, it causes neuron to fire producing an output signal. The output signal travels along the axons to other receiving neurons. The magnitude of the signal sent depends on the amount of chemicals released by the axon and received by the dentrites. The synaptic efficiency or strength is what is modified when the brain learns.

In an artificial neural network, the basic unit called the artificial neuron or the processing unit may have several input paths, corresponding to dentrites. The unit combines, usually by a simple summation, the weighted values of these input paths. The result is an internal activity level for the unit. The combined values are then modified by a transfer function. The output value of the transfer function is generally passed directly to the output path of the unit. The output path of a unit may be connected to the input path of other units through connection weights which correspond to the synaptic strength of the biological neural connections. Each connection has a corresponding weight where the signals on the input lines to a unit are modified or weighted prior to being summed. The summation function is a weighted summation. Fig. 2 shows the basics of an artificial neuron.

Units are usually organized into groups called layers. Basically, all artificial NNs have a similar structure of topology. Only two layers bound the network; The input layer consists of neurons that receive input from the external environment and the output layer where the response of the network to given input is produced. Layers other than these two are hidden layers.

As opposed to classical algorithm, a NN is not programmed but is taught. Learning is to modify the variable connection weights on the inputs of each processing element in order to achieve some desired results for a given set of inputs. There are two types of learning; supervised and unsupervised. In the case of supervised learning, the actual output of a NN is compared to the desired output. Weights are then adjusted by the network so that the next iteration will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until acceptable network accuracy is reached.

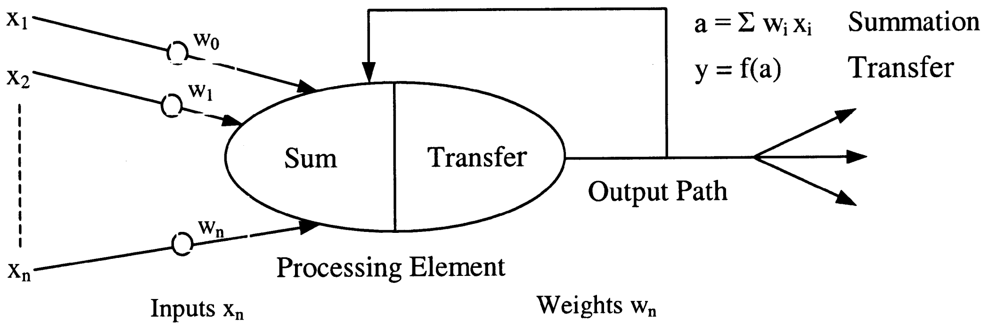


Fig. 2 Artificial neuron

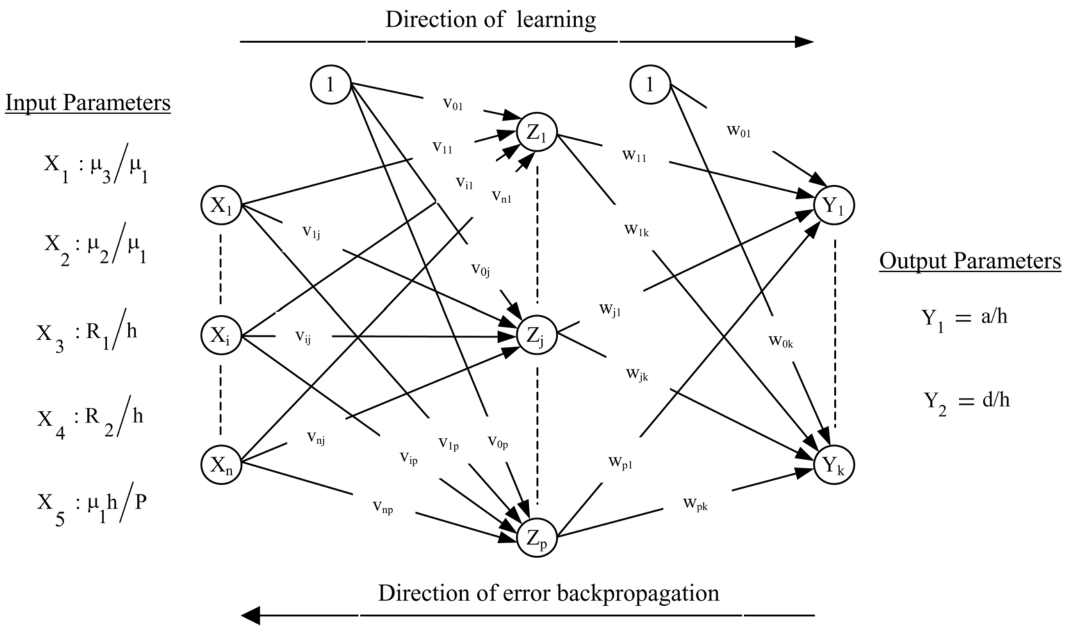


Fig. 3 The architecture of BP network model

The most popular supervised learning approach is multi layer perceptron (MLP) architecture. The MLP learns via a process called backpropagation (BP) shown in Fig. 3.

BP is a systematic method for training MLP. Basically the idea is to measure error across the output units and then to propagate the error backwards through the network making necessary weight changes at each layer of units, including hidden layers. The learning rule associated with this model is known as the generalized delta rule.

Training a network by backpropagation involves three stages; the feedforward of the input training pattern, the backpropagation of the associated error, and the adjustment of weights. The BP algorithm is given briefly as follows;

Each input unit $(X_i, i = 1, \ldots, n)$ receives input signal x_i and broadcasts this signal to all units in the hidden layer. Each hidden unit $(Z_j, j = 1, \ldots, p)$ sums its weighted input signals as

$$z_{-in_j} = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \tag{25}$$

where v_{ij} is the weight on connection from input layer unit X_i to hidden layer unit Z_j. v_{0j} indicates the bias on hidden layer unit j. Bias acts like a weight on a connection from a unit with a constant activation of 1. Then each hidden unit applies its activation function to compute its output signal $z_j = f(z_{-in_j})$. One of the most typical activation function which is also used in this study is the binary sigmoid function which has the range [0, 1] and is defined as

$$f(x) = 1/(1 + e^{-x}) \tag{26}$$

Hidden layer unit sends this signal to all units in the output layer. Each output unit $(Y_k, k = 1, \ldots, m)$ sums its weighted input signals in the following form

$$y_{-in_k} = w_{0k} + \sum_{j=1}^{p} z_j w_{jk} \tag{27}$$

where w_{jk} is the weight on connection from hidden layer unit Z_j to output layer unit Y_k. w_{0k} is the bias. Afterwards, output unit applies its activation function to compute its output signal $y_k = f(y_{-in_k})$. This process is called feedforward.

Each output unit $(Y_k, k = 1, \ldots, m)$ receives a target pattern $(t_k, k = 1, \ldots, m)$ corresponding to the input training pattern, and computes its error information term as follows,

$$\delta_k = (t_k - y_k) f'(y_{-in_k}) \tag{28}$$

Output units calculates its weight correction term using δ_k as

$$\Delta w_{jk} = \alpha \delta_k z_j \tag{29}$$

and it sends δ_k to units in the hidden layer. The quantity α is the learning rate and is used to dampen out predicted changes in weighting factors. Each hidden unit $(Z_j, j = 1, \ldots, p)$ sums its delta inputs as follows,

$$\delta_{-in_j} = \sum_{k=1}^{m} \delta_k w_{jk} \tag{30}$$

and multiplies δ_{-in_j} by the derivative of its activation function to calculate its error information term δ_j as

$$\delta_j = \delta_{-in_j} f'(z_{-in_j}) \tag{31}$$

Hidden unit uses δ_j to calculate its weight correction term in the following form

$$\Delta v_{ij} = \alpha \delta_j x_i \tag{32}$$

This process is called backpropagation.
Then each output unit $(Y_k, k = 1, \ldots, m)$ and hidden unit $(Z_j, j = 1, \ldots, p)$ updates their biases and weights $w_{jk}(j = 0, \ldots, p)$, $v_{ij}(i = 0, \ldots, n)$ with momentum as follows respectively.

$$w_{jk}(t+1) = w_{jk}(t) + \alpha \delta_k z_j + \eta [w_{jk}(t) - w_{jk}(t-1)] \tag{33}$$

$$v_{jk}(t+1) = v_{jk}(t) + \alpha \delta_k z_j + \eta [v_{jk}(t) - v_{jk}(t-1)] \tag{34}$$

where η is the momentum parameter. New weights for training step $(t + 1)$ are based on the weights at training steps t and $t - 1$. Training process stops when a given number of epochs elapse or when the error reaches an acceptable level or when the error stops improving. The most common output error to be minimized is known as the total sum squared error (TSSE) and given as

$$E = \sum E_P = \frac{1}{2} \sum_p \sum_k (t_k - y_k)^2 \tag{35}$$

where k is the number of elements in the output layer, and p is the input pattern number (Fausett 1994). The foregoing algorithm used in this study updates the weights after an epoch is presented. Epoch is one cycle through the entire set of training patterns.

4. Construction, teaching and testing of neural network

The main objective of the work is to investigate the ability of NNs in predicting contact lengths *a/h* and *d/h* automatically by using the BP paradigm. Data for the purpose of training were obtained from the theoretical solution previously summarized.

It is important to choose the proper network size. If the network is too small, it may not be able to represent the system adequately. On the other hand, if the network is too big, it becomes over-trained and may provide erroneous results for untrained patterns. In general, it is not straightforward to determine the best size of the networks for a given system. As shown in Fig. 3, a three-layer network is selected for the present study. Each layer is connected to the next but no connections exist between neurons on the same level. The number of neurons in the first and third layers, which contain input and output data respectively, are predetermined and depends on the problem at hand. There are 5 nodes in the input layer corresponding to the 5 variables. These variables are the following:

μ_2/μ_1 : Ratio of shear modulus of lower elastic punch to elastic layer.
μ_3/μ_1 : Ratio of shear modulus of upper elastic punch to elastic layer.
$\mu_1 h/P$: Load factor.
R_1/h : Ratio of radius of upper elastic punch to elastic layer height.
R_2/h : Ratio of radius of lower elastic punch to elastic layer height.

The input parameter ranges used in this study are presented in Table 1.

Various combinations of these parameters are used to obtain 400 different patterns from theoretical solution. They are split into the training, testing and validation patterns of the numbers $L = 237$, $T = 123$, $V = 40$, respectively. Input values and the desired outputs of the validation set are given in Table 2.

Data preprocessing is also known as data normalization. Raw data needs to be preprocessed into a range that can be accepted by the network. A sigmoidal transfer function is used within the network. Upper and lower limits of output from a sigmoid transfer function are 1 and 0 respectively. Scaling of the inputs to the range [0,1] greatly improves the learning speed, as these values fall in the region of the sigmoid transfer function. So each group of input and output values are normalized into range [0.1, 0.9] as,

$$\text{Normalised Value} = \left[\frac{\text{Raw Value} - \text{Minimum Value}}{\text{Maximum Value} - \text{Minimum Value}}\right] * (0.9 - 0.1) + 0.1 \tag{36}$$

The definition of network size is a compromise between generalization and convergence. Convergence is the capacity of the network to learn the pattern on the training set and generalization

Table 1 Input parameters used for training and testing in this study

μ_3/μ_1	μ_2/μ_1	R_1/h	R_2/h	$\frac{\mu_1}{P/h}$
0.52 1.65 2.8 5	0.52 1.65 2.8 5	10 100 1000	10 100 1000	100 250 500 1000 1500

Table 2 Validation patterns

	Input parameters					Desired outputs	
No	μ_2/μ_1	μ_3/μ_1	R_1/h	R_2/h	$\frac{\mu_1}{P/h}$	a/h	d/h
1	0.52	0.52	10	100	250	0.34965	0.91270
2	0.52	0.52	100	1000	500	0.81880	1.50450
3	0.52	0.52	1000	1000	250	3.26270	3.26270
4	0.52	1.65	10	10	1500	0.10814	0.14050
5	0.52	1.65	10	1000	1000	0.13370	1.10000
6	0.52	1.65	100	100	100	1.22697	1.38080
7	0.52	2.8	10	100	500	0.15952	0.69250
8	0.52	2.8	100	1000	100	1.44600	2.11400
9	0.52	2.8	1000	1000	1500	0.89580	1.17770
10	0.52	5	10	10	100	0.32102	0.51860
11	0.52	5	10	1000	250	0.21715	1.51949
12	0.52	5	100	1000	1500	0.27736	1.07203
13	1.65	1.65	10	100	1500	0.10064	0.30234
14	1.65	1.65	100	100	500	0.50150	0.50150
15	1.65	1.65	1000	1000	250	1.97430	1.97430
16	1.65	2.8	10	10	250	0.23870	0.25560
17	1.65	2.8	10	1000	500	0.17523	0.93401
18	1.65	2.8	100	1000	1000	0.38380	0.84900
19	1.65	5	10	10	500	0.16310	0.18400
20	1.65	5	100	100	1500	0.28707	0.32330
21	1.65	5	1000	1000	1000	0.95400	1.01970
22	2.8	2.8	10	100	100	0.34956	0.76440
23	2.8	2.8	10	1000	1500	0.08679	0.66130
24	2.8	2.8	100	100	500	0.45678	0.45678
25	2.8	5	10	10	1000	0.10556	0.11240
26	2.8	5	100	1000	250	0.65375	1.08910
27	2.8	5	1000	1000	100	2.64950	2.66470
28	5	5	10	100	1000	0.10573	0.31108
29	5	5	10	1000	500	0.15002	0.79310
30	5	5	100	100	100	0.83950	0.83950
31	0.52	1.65	100	10	1000	0.38417	0.17196
32	0.52	5	100	10	100	0.75260	0.53750
33	1.65	5	100	10	500	0.44620	0.18510
34	2.8	2.8	100	10	250	0.57090	0.22284
35	2.8	5	100	10	1500	0.26000	0.09213
36	0.52	0.52	1000	100	500	1.50450	0.81880
37	0.52	1.65	1000	100	1000	0.88650	0.53602
38	1.65	5	1000	100	100	1.47000	1.22100
39	2.8	5	1000	100	250	1.02900	0.68020
40	5	5	1000	100	1500	0.62981	0.26816

Table 3 Comparison of different neural networks

Number of hidden layer unit	α	η	Cycle	Epoch	Min Train error (%)	Min Test error (%)	Max Relative error (%) in validation set	Average Relative error (%) of validation set
10	1	1	5	500000	1.03588	1.15154	43.554	8.304
15	1	1	8	11085	0.48099	0.74471	21.578	6.489
20	1	1	10	443975	0.13828	0.58884	45.993	4.177
30	1	1	15	50410	0.05710	0.56878	14.581	3.179
40	1	1	20	22512	0.05538	0.56878	18.032	3.042
10	0.5	1	5	149092	1.03778	0.88653	67.083	10.858
15	**0.5**	**1**	**8**	**500000**	**0.21168**	**0.84270**	**5.2390**	**1.569**
20	0.5	1	10	75074	0.18074	0.63826	42.309	4.609
30	0.5	1	15	67109	0.06860	0.84406	26.596	4.075
40	0.5	1	20	145324	0.02364	0.59734	13.975	2.859
10	0.1	1	5	500000	0.80052	1.40933	39.582	8.664
15	0.1	1	8	500000	0.25988	0.86270	29.770	5.579
20	0.1	1	10	205569	0.19789	0.75797	27.755	5.114
30	0.1	1	15	375440	0.04634	0.63018	22.026	3.296
40	0.1	1	20	500000	0.02231	0.80483	16.478	3.141
10	1	0.5	5	500000	0.93049	1.28459	51.062	8.389
15	1	0.5	8	61854	0.33814	0.68933	29.706	6.019
20	1	0.5	10	12619	0.35253	0.81978	31.039	6.159
30	1	0.5	15	61854	0.04308	0.63956	14.654	3.158
40	1	0.5	20	41832	0.03460	0.75345	18.452	1.802
10	1	0.1	5	356231	0.72356	0.97216	33.319	7.232
15	1	0.1	8	168005	0.24018	0.77819	32.641	4.847
20	1	0.1	10	251463	0.12320	0.67559	20.770	3.458
30	1	0.1	15	132071	0.05104	0.67955	21.761	3.687
40	1	0.1	20	67348	0.03211	0.57893	11.947	2.903
10	0.1	0.1	5	71909	1.15616	1.33406	56.612	10.812
15	0.1	0.1	8	453265	0.26353	0.78401	22.114	4.492
20	0.1	0.1	10	500000	0.15294	0.73499	28.600	4.555
30	0.1	0.1	15	169615	0.15938	0.76725	26.482	4.326
40	0.1	0.1	20	500000	0.03792	0.54971	15.892	3.314
10	0.5	0.5	5	258533	1.14453	1.79287	40.186	7.907
15	0.5	0.5	8	187903	0.23228	0.79429	27.137	5.063
20	0.5	0.5	10	281288	0.15075	0.56211	25.216	4.743
30	0.5	0.5	15	81718	0.06993	0.72324	25.729	3.891
40	0.5	0.5	20	33680	0.11831	0.75472	27.631	3.831

is the capacity to respond correctly to new patterns. The idea is to implement the smallest network possible, so it is able to learn all patterns and at the same time provide good generalization. As for the number of hidden layer, it is well said that one hidden layer is sufficient for most usual applications, Thus only one hidden layer is used in this study. Determining the number of nodes to include in the hidden layer is not an exact science, so network is tested for different number of hidden layer nodes. As it is seen from Table 3, 10, 15, 20, 30 and 40 nodes are used in the hidden layer to find the optimum number of hidden layer nodes.

During the training process, all the training patterns are introduced to the network and corresponding outputs are obtained. Then the network error E is computed according to formula (35) and the increments of generalized weights are computed by formulas (33) and (34). The choice of initial weights will influence the net reaches a global minimum of the error and, if so, how quickly it converges. As mentioned earlier the update of the weight between two units depends on both the derivative of the upper unit's activation function and the activation of the lower unit. For this reason, it is important to avoid choices of initial weights that would make it likely that either activations or derivatives of activations are zero. In this study, the weights are initialized into random values between –0.5 and 0.5, a procedure commonly accepted.

Factors α and η in Eqs. (33) and (34) also influence the convergence. α, the learning rate, is the constant of proportionality of the generalized rule. The larger the value is, the greater the changes are in weights. Three different α values 0.1, 0.5 and 1 are introduced to network for training. η, the momentum term, which is used to smooth out the weight changes to prevent network training from oscillating is chosen as 0.1, 0.5 and 1. Different combinations of selected values of α and η are tried for good convergence of the neural network (Table 3).

The level of convergence in training is monitored using TSSE of training and testing patterns separately. It is seen from Fig. 4 that, as the TSSE of training patterns falls, the TSSE of testing patterns initially falls and then rises. The rise is due to the net overfitting where it simply learns the

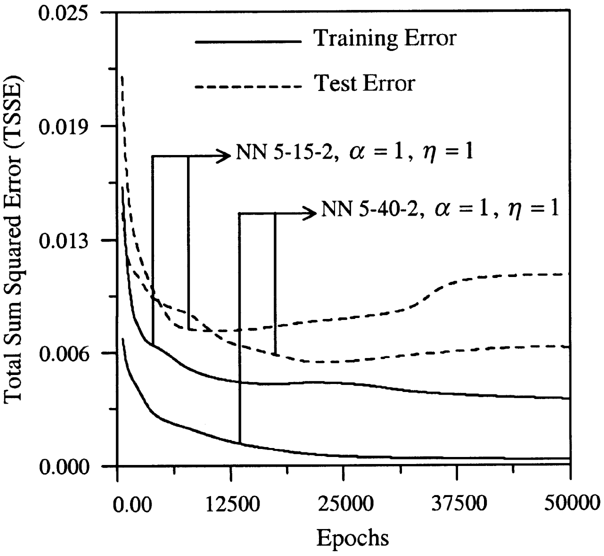


Fig. 4 Total sum squared error of training and testing patterns of two neural networks: NN 5-15-2 $\alpha = 1$ $\eta = 1$, NN 5-40-2 $\alpha = 1$ $\eta = 1$ with epochs

training pattern, but is no longer able to generalize to predict any new patterns. This can be avoided by making sure that training set is large enough, and by setting some stopping criteria for the training process. So each of the training patterns is used once first then 5 times in an epoch for each NN model. Finally NN is trained with a group of training patterns which are nearly 15 times the total weight number. 4th column (Cycle) in Table 3 shows how many times training set is repeated in an epoch to get a group of training patterns which are nearly 15 times the total weight number. The patterns are presented to each epoch in the same order.

After the learning set of data presented to the NN model, we stopped the learning process when the epochs reached 500000 and determined what epoch number gives the minimum TSSE of testing set given in 5th column of Table 3 for various NN alternatives.

5. Results and discussion

It is found that there is a trade off between the performance of a network and time consumed. Generally, the performance of a network is found to increase with the suitable increase in the number of samples, epochs (learning time) and the number of hidden layer nodes. Meanwhile, the increase of these parameters also increases consumed time.

It is seen from the Table 3 that although the smallest train error is obtained from the network 5-40-2 with $\alpha = 0.1$ and $\eta = 1$, the smallest test error is obtained from the network 5-40-2 with $\alpha = 0.1$ and $\eta = 0.1$, so it may be said that besides hidden layer nodes, epochs and number of samples, learning rate α and momentum term η influence network to provide good generalization too much. Network 5-40-2 with $\alpha = 1$ and $\eta = 0.5$ has quite a good generalization. Even if the maximum relative error in the validation set is 18.45%, relative errors of the other validation patterns are smaller than 10% and the average relative error of the validation set is 1.8%. Relative error listed in Table 3 is calculated as $e_{rel} = \left| 1 - \frac{O_{NN}}{O_{actual}} \right| *100$ where O_{NN} and O_{actual} are the computed and theoretically determined contact lengths, respectively. The smallest maximum relative error in validation set is obtained from the network 5-15-2 with $\alpha = 0.5$ and $\eta = 1$ as 5.23%. The average relative error of validation set in this case is 1.56%. It is the smallest network with the best generalization. Thus network 5-15-2 with $\alpha = 0.5$ and $\eta = 1$ is accepted as the optimum network model to find contact lengths *a/h* and *d/h*. Maximum relative error may be reduced if stopping criteria, epoch number, is increased.

Besides, conjugate gradient or scaled conjugate gradient methods may be used to reduce maximum relative error instead of generalized delta rule in learning. Also different network structures with one or more hidden layers or nodes with different learning rates and momentum terms may produce smaller error.

We used the validation set to evaluate the confidence in the performance of the trained network. 40 validation vectors are used to test the NN model. Fig. 5 is an expression of the learning results of network 5-15-2 with $\alpha = 0.5$ and $\eta = 1$, each point standing for a validation vector. The nearer the points gather around the diagonal, the better are the learning results. The relative errors of the points on the diagonal are zero.

Once trained well, the NN model is able to quickly estimate the corresponding output for any given input. Although time consumed for learning of network 5-15-2 with $\alpha = 0.5$ and $\eta = 1$ is

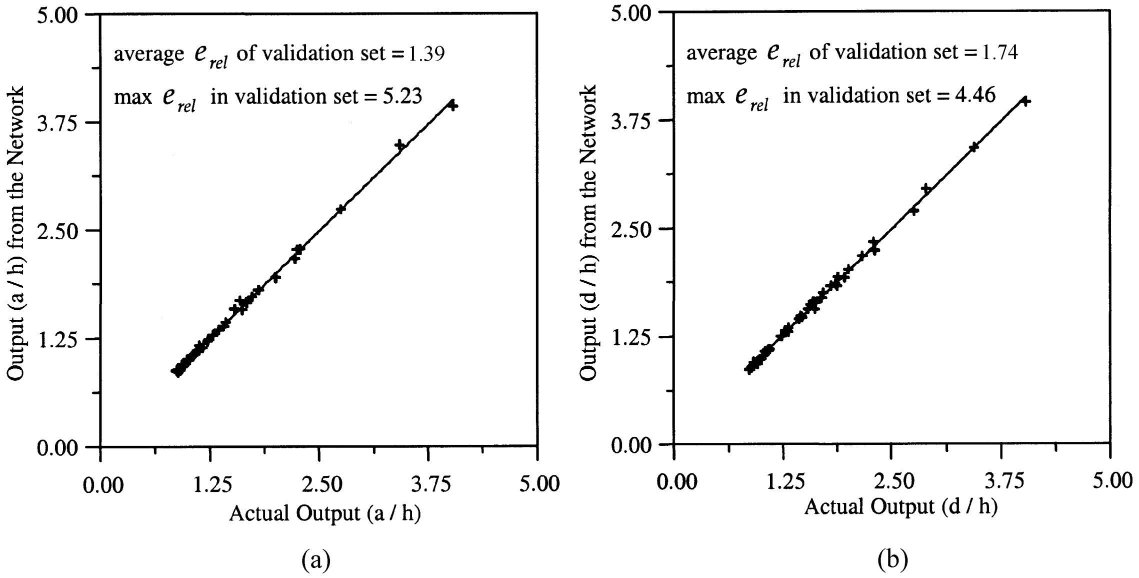


Fig. 5 Comparison of contact lengths obtained from the NN 5-15-2 with $\alpha = 0.5$, $\eta = 1$ and theoretical solution

Table 4 Effect of load factor $\mu_1 h/P$ on contact lengths generated by the neural network and from theoretical solution ($\mu_2/\mu_1 = 1.65$, $\mu_3/\mu_1 = 5$, $R_1/h = 100$, $R_2/h = 100$)

	a/h			*d/h*		
$\frac{\mu_1}{P/h}$	Actual Output	Output from the NN	$e_{rel} = \left\|1-\frac{O_{NN}}{O_{actual}}\right\|*100$	Actual Output	Output from the NN	$e_{rel} = \left\|1-\frac{O_{NN}}{O_{actual}}\right\|*100$
150	0.794702	0.783145	1.454	0.864697	0.881533	1.947
350	0.548151	0.528094	3.658	0.608997	0.615945	1.140
450	0.491243	0.475700	3.164	0.547638	0.556008	1.528
650	0.417897	0.406901	2.631	0.46800	0.472958	1.059
750	0.392199	0.38188	2.631	0.439902	0.441502	0.363
1250	0.311953	0.309156	0.896	0.350951	0.345152	1.652

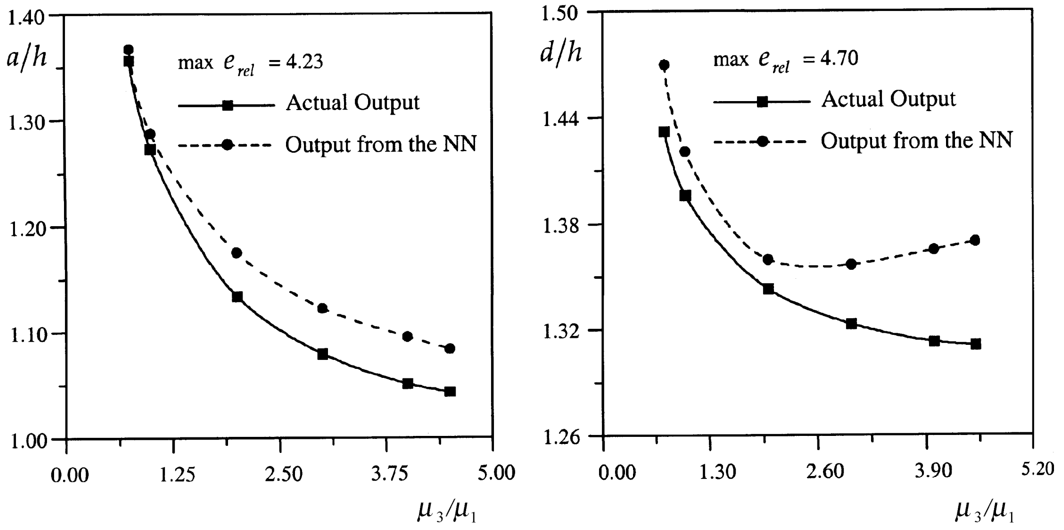
18000 seconds in a personal computer (PC), trained network responds any given input in 5 seconds in the same PC. The trained NN model is used to simulate the effect of some factors on the contact lengths *a/h* and *d/h*, and the results are compared with the theoretical solutions.

Table 4 shows the effect of load factor $\mu_1 h/P$ on contact lengths *a/h* and *d/h* for $\mu_2/\mu_1 = 1.65$ and $\mu_3/\mu_1 = 5$. Elastic punches radii are equal ($R_1/h = R_2/h = 100$). As load factor increases contact lengths *a/h* and *d/h* decrease. Load factor values are selected different from those in the test, train and validation sets. Maximum relative error is obtained as 3.65%.

Table 5 shows the changes in contact lengths *a/h* and *d/h* with elastic punches radii R_1/h and R_2/h. Flexibilities of the elastic punches are equal ($\mu_2/\mu_1 = \mu_3/\mu_1 = 2.8$). The load factor value is $\mu_1 h/P = 1000$. It is seen from the Table 5 that as elastic punch radius increases, while the other

Table 5 Effect of punches radii on contact lengths generated by the neural network and from the theoretical solution ($\mu_2/\mu_1 = 2.8$, $\mu_3/\mu_1 = 2.8$, $\mu_1 h/P = 1000$)

R/h		a/h			d/h		
		Actual Output	Output from the NN	$e_{rel} = \left\|1-\frac{O_{NN}}{O_{actual}}\right\|*100$	Actual Output	Output from the NN	$e_{rel} = \left\|1-\frac{O_{NN}}{O_{actual}}\right\|*100$
R_1/h	60	0.27359	0.23884	12.701	0.75220	0.75543	0.429
	75	0.30475	0.28011	8.084	0.75674	0.75853	0.236
	150	0.42299	0.46570	10.095	0.77630	0.77226	0.468
R_2/h	600	0.86410	0.90986	5.295	0.76429	0.84297	10.293
	750	0.88560	0.92476	4.422	0.82840	0.87047	5.079
	850	0.89724	0.92055	2.597	0.86564	0.89169	3.008



(a) (b)

Fig. 6 Effect of elastic constant μ_3/μ_1 on contact lengths generated by the neural network and from the theoretical solution ($\mu_2/\mu_1 = 0.61$, $R_1/h = R_2/h = 1000$, $\mu_1 h/P = 1000$)

punch radius $R/h = 1000$ is fixed, contact length under that punch increases. Maximum relative error is 12.7%.

Changing in contact lengths a/h and d/h with different values of elastic constant μ_3/μ_1 is given in Fig. 6. Both elastic punches radii are equal ($R_1/h = R_2/h = 1000$). Load factor $\mu_1 h/P = 1000$ is fixed. As upper punch rigidity increases contact lengths a/h and d/h decrease and approach to constant values. This situation is more apparent in Fig. 6(b). μ_3/μ_1 values are selected different from those in the test, train and validation sets. Maximum relative error is 4.70%.

Variation of the contact lengths a/h and d/h with load factor $\mu_1 h/P$ is also given in Fig. 7. Lower and upper elastic punch flexibilities are $\mu_2/\mu_1 = 0.61$ and $\mu_3/\mu_1 = 2$ respectively. While upper punch

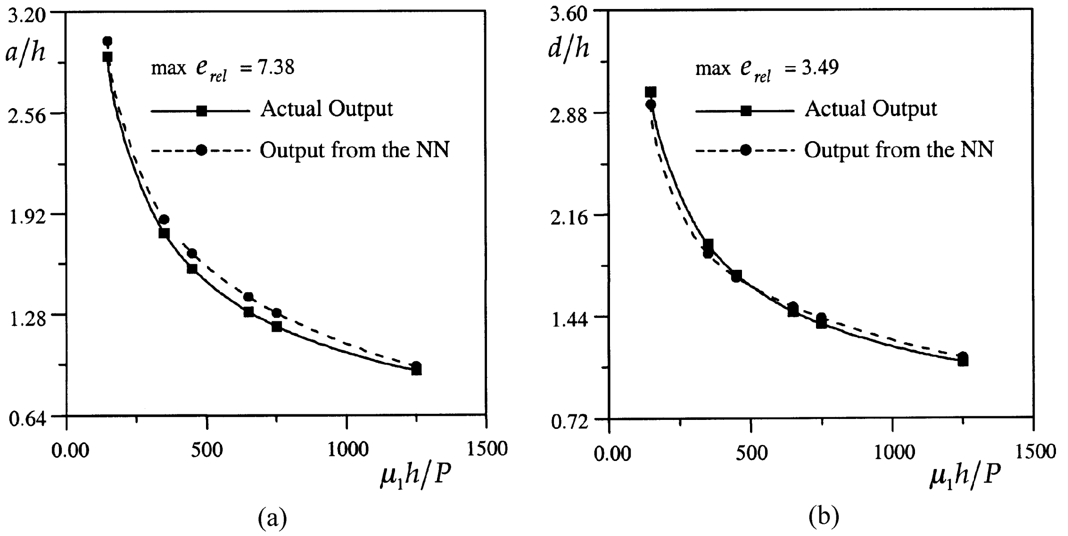


Fig. 7 Effect of load factor $\mu_1 h/P$ on contact lengths generated by the neural network and from theoretical solution $(\mu_2/\mu_1 = 0.61, \mu_3/\mu_1 = 2, R_1/h = 750, R_2/h = 900)$

radius is $R_1/h = 750$, lower punch radius is $R_2/h = 900$. In this case, none of the variables is same as those presented to the network training or testing. As mentioned before, contact lengths decrease with $\mu_1 h/P$ increment. Maximum relative error is 7.38%.

6. Conclusions

In this paper, a backpropagation neural network with one hidden layer has been developed and proved to be very sufficient in predicting the contact lengths between the elastic layer and two elastic circular punches. The training data sets are obtained from elasticity solution. In addition to the other parameters, it is seen that learning rate and momentum term have significant effects on the training process. It was found that neural networks reduce the overall time required for implementations when compared with existing theoretical analysis.

As a result of using the neural network model in the analysis of the relationships between the contact lengths and the load factor, elastic punches' flexibilities and radii, following conclusions can be made:

- If load increases or the elastic layer rigidity decreases, contact lengths *a/h* and *d/h* increase.
- Contact lengths *a/h* and *d/h* increase if elastic layer rigidity is greater than elastic punches rigidities.
- Contact lengths also increase with increase in elastic punches radii.

It is shown that the neural network conclusions agree well with that of theoretical analysis. So, the trained network can be used for on-line prediction of contact lengths. Consequently, application of neural network models to contact mechanics is practical especially for the time consuming problems which require iteration in theoretical solution. NNs may be applied to receding and discontinuous contact problems successfully.

References

Adams, G.G. and Bogy, D.B. (1976), “The plane solution for the elastic contact problem of a semi-infinite strip and half plane”, *J. Appl. Mech.*, **43**, 603-607.

Anderson, D., Hines, E.L., Arthur, S.J. and Eiap, E.L. (1997), “Application of artificial neural networks to the prediction of minor axis steel connections”, *Comput. Struct.*, **63**(4), 685-692.

Artan, R. and Omurtag, M. (2000), “Two plane punches on a nonlocal elastic half plane”, *Int. J. Eng. Sci.*, **38**, 395-403.

Birinci, A. and Erdol, R. (2001), “Continuous and discontinuous contact problem for a layered composite resting on simple supports”, *Struct. Eng. Mech.*, **12**(1), 17-34.

Birinci, A. and Erdol, R. (2003), “A frictionless contact problem for two elastic layers supported by a winkler foundation”, *Struct. Eng. Mech.*, **15**(3), 331-344.

Çakıroğlu, F.L., Çakıroğlu, M. and Erdöl, R. (2001), “Contact problems for two elastic layers resting on elastic half-plane”, *J. Eng. Mech.*, **127**(2), 113-118.

Chandrashekhara, K., Okafor, A.C. and Jiang, Y.P. (1998), “Estimation of contact force on composite plates using impact-induced strain and neural networks”, *Composites Part B*, **29B**, 363-370.

Civelek, M.B. and Erdogan, F. (1974), “The axisymmetric double contact problem for a frictionless elastic layer”, *Int. J. Solids Struct.*, **10**, 639-659.

Civelek, M.B. and Erdogan, F. (1975), “The frictionless contact problem for an elastic layer under gravity”, *J. Appl. Mech.*, **42**(97), 136-140.

Comez, I., Birinci, A. and Erdol, R. (2003), “Double receding contact problem for a rigid stamp and two elastic layers”, *European Journal of Mechanics A/Solids*, **23**, 301-309.

Dempsey, J.P., Zhao, Z.G., Minnetyan, L. and Li, H. (1990), “Plane contact of an elastic layer supported by a winkler foundation”, *J. Appl. Mech.*, **57**, 974-980.

Erdogan, F. and Gupta, G.D. (1972), “On the numerical solutions of singular integral equations”, *Quarterly Journal of Applied Mathematics*, **29**, 525-534.

Erdogan., F. and Ratwani, M. (1974), “The contact problem for an elastic layer supported by two elastic quarter planes”, *J. Appl. Mech.*, **41**(96), 673-678.

Fausett, L. (1994), *Fundamentals of Neural Networks*, Prentice-Hall, New Jersey.

Galin, L.A. (1961), *Contact Problems in the Theory of Elasticity*, North Carolina State College Translation Series, Raleigh.

Geçit, M.R. and Gökpınar, S. (1985), “Frictionless contact between an elastic layer and a rigid rounded support”, *The Arabian Journal for Science and Engineering*, **10**, 243-251.

Geçit, M.R. (1986), “The axisymmetric double contact problem for a frictionless elastic layer indented by an elastic cylinder”, *Int. J. Eng. Sci.*, **24**, 1571-1584.

Goh, A.T.C. (1994), “Nonlinear modelling in geotechnical engineering using neural networks”, *Australian Civil Engineering Transactions*, **CE36**(4), 293-297.

Hong-Guang, N. and Ji-Zong, W. (2000), “Prediction of compressive strength of concrete by neural networks”, *Cement and Concrete Research*, **30**, 1245-1250.

Jaffar, M.J. (1993), “Determination of surface deformation of a bonded elastic layer indented by a rigid cylinder using the chebyshev series method”, *Wear*, **170**, 291-294.

Kang, J. and Song, J. (1998), “Neural network applications in determining the fatigue crack opening load”, *Int. J. Fatigue*, **20**(1), 57-69.

Kuźniar, K., Maciag, E., Obiala, R. and Waszczyszyn, Z. (2000), “Application of neural networks in natural periods identification and simulation of prefabricated buildings response”, *Soil Dynamics and Earthquake Engineering*, **20**, 217-222.

Lai, S. and Serra, M. (1997), “Concrete strength prediction by means of neural network”, *Construction and Building Materials*, **11**(2), 93-98.

Lan, Q., Graham, G.A.C. and Selvadurai, A.P.S. (1996), “Certain two punch problems for an elastic layer”, *Int. J. Solids Struct.*, **33**(19), 2759-2774.

Lee, S.C. and Han, S.W. (2002), “Neural-network-based models for generating artificial earthquakes and response spectra”, *Comput. Struct.*, **80**, 1627-1638.

Seibi, A. and Al-Alawi, S.M. (1997), "Prediction of fracture thoughness using artificial neural networks (ANNs)", *Engineering Fracture Mechanics*, **56**(3), 311-319.

Shahin, M.A., Jaksa, M.B. and Maier, H.R. (2000), "Predicting the settlement of shallow foundations on cohesionless soils using back-propagation neural networks", Research Report No. R 167, The University of Adelaide, Adelaide.

Ufliand, I.S. (1965), *Survey of Articles on the Applications of Integral Transforms in the Theory of Elasticity*, North Carolina State College Translation Series, Raleigh.

Ural, D.N. and Saka, H. (1998), "Liquefaction assessment by neural networks", *Electronic Journal of Geotechnical Engineering*, http://geotech.civen.okstate.edu/ejge/ppr9803/index.html.

Waszczyszyn, Z. and Bartczak, M. (2002), "Neural prediction of buckling loads of cylindrical shells with geometrical imperfections", *International Journal of Non-Linear Mechanics*, **37**, 763-775.

Appendix

Kernels of integral Eqs. (13) and (14) are expressed as follows

$$\begin{aligned} k_1(x,t) = {} & \bigg[\frac{1}{2\Delta}\frac{\mu_2}{\mu_1}\bigg[2e^{(-2\alpha h)}\bigg[(1+2\alpha^2h^2)(1+\kappa_3)-2\alpha h\frac{\mu_3}{\mu_1}(1+\kappa_1)\bigg]+e^{(-4\alpha h)}\bigg(-1+\frac{\mu_3}{\mu_1}(1+\kappa_1)- \\ & \kappa_3)+\bigg(-1-\frac{\mu_3}{\mu_1}(1+\kappa_1)-\kappa_3\bigg)\bigg]-\bigg[-\frac{(1+\kappa_3)}{2\omega}-\frac{(1+\kappa_1)}{2}\bigg]\bigg][\sin\alpha(t+x)-\sin\alpha(t-x)] \end{aligned} \tag{A1}$$

$$k_2(x,t) = \frac{1}{2\Delta}\frac{\mu_2}{\mu_1}(1+\kappa_1)\bigg[(e^{(-\alpha h)}-e^{(-3\alpha h)})\frac{2\mu_3}{\mu_1}+(e^{(-3\alpha h)}+e^{(-\alpha h)})2\alpha h\frac{\mu_3}{\mu_1}\bigg][\sin\alpha(t+x)-\sin\alpha(t-x)] \tag{A2}$$

$$k_3(x,t) = \frac{-1}{2\Delta}\frac{\mu_3}{\mu_1}(1+\kappa_1)\bigg[(e^{(-\alpha h)}-e^{(-3\alpha h)})\frac{2\mu_2}{\mu_1}+(e^{(-3\alpha h)}+e^{(-\alpha h)})2\alpha h\frac{\mu_2}{\mu_1}\bigg][\sin\alpha(t+x)-\sin\alpha(t-x)] \tag{A3}$$

$$\begin{aligned} k_4(x,t) = {} & \bigg[\frac{-1}{2\Delta}\frac{\mu_3}{\mu_1}\bigg[2e^{(-2\alpha h)}\bigg[(1+2\alpha^2h^2)(1+\kappa_2)-2\alpha h\frac{\mu_2}{\mu_1}(1+\kappa_1)\bigg]+e^{(-4\alpha h)}\bigg(-1+\frac{\mu_2}{\mu_1}(1+\kappa_1)-\kappa_2\bigg) \\ & +\bigg(-1-\frac{\mu_2}{\mu_1}(1+\kappa_1)-\kappa_2\bigg)\bigg]-\bigg[\frac{(1+\kappa_1)}{2}+\frac{(1+\kappa_2)}{2\theta}\bigg]\bigg][\sin\alpha(t+x)-\sin\alpha(t-x)] \end{aligned} \tag{A4}$$

where

$$\Delta = -\frac{\mu_3\mu_2}{\mu_1\mu_1}[-1+2e^{(-2\alpha h)}+4\alpha^2h^2e^{(-2\alpha h)}-e^{(-4\alpha h)}] \tag{A5}$$