International Journal of Thermal Sciences 48 (2009) 1311–1318

Contents lists available at ScienceDirect

International Journal of Thermal Sciences

www.elsevier.com/locate/ijts

Prediction of heat transfer due to presence of copper–water nanofluid using resilient-propagation neural network

Apurba Kumar Santra [a,*], Niladri Chakraborty [a], Swarnendu Sen [b]

[a] *Department of Power Engineering, Jadavpur University, Salt Lake Campus, Block LB, Plot-8, Sector III, Salt Lake, Kolkata 700 098, India*
[b] *Department of Mechanical Engineering, Jadavpur University, Kolkata 700 032, India*

ARTICLE INFO

Article history:
Received 8 April 2008
Received in revised form 10 October 2008
Accepted 12 November 2008
Available online 9 December 2008

Keywords:
ANN
RPROP
Nanofluid
Non-Newtonian
Natural convection
Square cavity

ABSTRACT

Heat transfer due to laminar natural convection of copper–water nanofluid in a differentially heated square cavity has been predicted by Artificial Neural Network (ANN). The nanofluid has been considered as non-Newtonian. The ANN has been trained by a resilient-propagation (RPROP) algorithm. The required input and output data to train the ANN has been taken from the results of numerical simulation that was performed simultaneously where the transport equations has been solved numerically using finite volume approach incorporating SIMPLER algorithm. Results from simulation and resilient-propagation (RPROP) based ANN have been compared. It has been observed that the ANN predicts the heat transfer correctly within the given range of training data. It is further observed that resilient-propagation (RPROP) based ANN is an efficient tool to predict the heat transfer than simulation, which takes much longer time to compute.

1. Introduction

Researchers have observed that thermal conductivity of nanofluid is much higher than that of the base fluids even for low solid volume fraction of nanoparticles in the mixture [1,2]. The nanofluid is stable, introduce very little pressure drop and it can pass through nanochannels [3]. Das et al. [4] has observed that the thermal conductivity for nanofluid increases with increasing temperature. Due to the lack of an accepted theory to predict the effective thermal conductivity of nanofluid, several researchers have proposed different correlations to predict the apparent thermal conductivity of this two-phase mixture. The models proposed by Hamilton and Crosser (HC) [5], Wasp [6], Maxwell–Garnett [7], Bruggeman [8] and Wang et al. [9] to determine the effective thermal conductivity of nanofluid failed to predict it accurately. The experimental results show a much higher thermal conductivity of nanofluid than those predicted by these models. Kumar et al. [10] has considered the effect of temperature on thermal conductivity in his proposed model. Patel et al. [11] has improved the model given in [10] by incorporating the effect of micro-convection due to particle movement. The effect of temperature is justified as Brownian motion increases with temperature, which causes additional convective effect. The above model is applicable for low concentration of solid volume fraction. The model takes into account the effect of particle size through an increase in specific surface area of nanoparticles. It involves an empirical constant '*c*', to link the temperature dependence of effective thermal conductivity to the Brownian motion of the particles. This can be found by comparing the calculated value with experimental data, which comes in the order of 10^4. This empirical constant '*c*' is adjustable and can be thought as a function of particle properties as well as size [11].

Experiments using Al_2O_3–water and CuO–water nanofluid by Putra et al. [12] and TiO_2–water nanofluid by Wen and Ding [13] show that heat transfer due to natural convection decreases with increase in nanoparticle concentration in the nanofluid. At low shear rate the viscosity of nanofluid increases rapidly with the increase of nanoparticle concentration [14]. For natural convection the shear rate is low and hence the viscosity is very high. Recently Santra et al. [15] have numerically shown that with a non-Newtonian assumption of the nanofluid, the heat transfer due to natural convection in the square cavity decreases with the increase in solid volume fraction. The results have a good agreement with the experimental results of Putra et al. [12] and Wen and Ding [13]. However these simulations require much longer computational time and memory allocation.

Over the last two and half decade several artificial intelligence (AI) technique has been utilized in engineering problem solving procedure with a huge reduction in computational time. AI techniques like expert system (ES), knowledge-based system (KBS) and intelligent decision support system (IDSS) try to mimic the mental processes, which take place during the human reasoning. Contrary to that, another AI technique, popularly known as artificial neural network (ANN) tries to simulate the neural activities that take

* Corresponding author. Tel.: +91 33 2335 5813/ 5215; fax: +91 33 2335 7254.
E-mail address: aksantra@pe.jusl.ac.in (A.K. Santra).

1290-0729/$ – see front matter © 2008 Elsevier Masson SAS. All rights reserved.
doi:10.1016/j.ijthermalsci.2008.11.009

Nomenclature

a	actual output
b	predicted output
C_p	specific heat .... J/kg K
E	local error function
Gr_f	Grashof number of fluid, $\rho_f^2 g\beta_f \nabla T H^3/\mu_f^2$
g	acceleration due to gravity .... m/s^2
H, L	dimensional (m) height and width of cavity
k	thermal conductivity .... W/m k
m, n	the respective consistency and fluid behavior index parameter
Nu_y	local Nusselt number of the heater
$\overline{Nu}$	average Nusselt number at the heater
p	pressure .... N/m^2
P	dimensionless pressure $(p - p_0)H^2/\rho_0\alpha^2$
Pr	Prandtl number of fluid, ν_f/α_f
Ra	Rayleigh number, $Gr\,Pr$
T	temperature .... K
T_H, T_C	temperature (K) of the heat source and sink respectively
u, v	velocity components in the x and y directions respectively .... m/s
U, V	dimensionless velocities $(U = uH/\alpha,\ V = vH/\alpha)$
$\Delta W_{ij}^{(t)}$	the incremental changes in the weight values, for tth iteration between neuron i and j
x, y	horizontal and vertical coordinates respectively .... m
X, Y	dimensionless horizontal and vertical coordinates respectively $(X = x/H,\ Y = y/H)$

Greek symbols

α	thermal diffusivity of the fluid .... m^2/s
β	isobaric expansion coefficient .... K^{-1}
ϕ	solid volume fraction
μ	dynamic viscosity .... N s/m^2
ν	kinematic viscosity .... m^2/s
ρ	density of the fluid .... kg/m^3
θ	dimensionless temperature $(T - T_0)/(T_H - T_C)$
ψ	dimensionless stream function
η	factor for change of update values
Δ_{ij}^t	update value for each weight, between neurons of layers i and j at the tth instant of time

Subscripts

t	tth iteration
eff	effective
f	fluid
nf	nanofluid
0	at reference state
s	solid

place in the human brain [16,17]. Different types of ANN have shown a good potential in solving complex engineering problems.

The standard back propagation ANN techniques has been adopted for analysis of different parameters in various mechanical systems like refrigeration system [18], heat transfer [19], internal combustion engines [20], chemical system [21], etc. This back propagation ANN techniques is a generalized approach, which has been derived from the Widrowo Hoff error correction rule [22].

Recently another particular variety of ANN, i.e., resilient propagation (RPROP) has gained focus of the researchers. RPROP was first introduced by Riedmiller and Braun [23,24]. It is a local adaptive learning scheme, performing supervised batch learning in multi-layer perceptions. The basic principle of RPROP is to eliminate the harmful influence of the size of the partial derivative on the weight step as evident in back propagation technique [23]. In RPROP, only the sign of the derivative is considered to indicate the direction of the weight update. The size of the weight change is exclusively determined by a weight-specific, so-called 'update-value'. Hence, this algorithm is different from original back propagation procedure.

In this paper heat transfer due to laminar natural convection of a non-Newtonian copper–water nanofluid in a differentially heated square cavity has been predicted using Artificial Neural Network trained by resilient propagation (RPROP) algorithm for achieving faster solution without compromising the accuracy level. The simulated results of heat transfer obtained by the authors [15] have been used to train the ANN. Training accuracy is measured in terms of Root Mean Square Error between target output and original output. Both the input and output data sets are normalized for faster convergence. The results obtained by the proposed ANN based method is compared with the simulated results. It is found that the convergence characteristic of RPROP algorithm is excellent to yield fairly comparable and faster (approximately 100 times or more) results. To the best of the knowledge of the authors, study of heat transfer augmentation using resilient back propagation algorithm particularly with nanofluid has not been reported so far.

2. Mathematical formulation

2.1. Problem statement

Fig. 1 shows the geometry of the problem. It consists of a two-dimensional square enclosure of height H and width L. For the present case $H = L$ has been considered. The temperatures of the two sidewalls of the cavity are maintained at T_H and T_C. The condition at T_C is taken to be the reference condition. The top and the bottom horizontal walls have been considered to be insulated. The enclosure is filled with a mixture of water and solid spherical copper particles of nanometer size. The nanoparticles are of uniform shape and size. The nanofluid is assumed to be non-Newtonian, incompressible and the flow is laminar. The density variation has

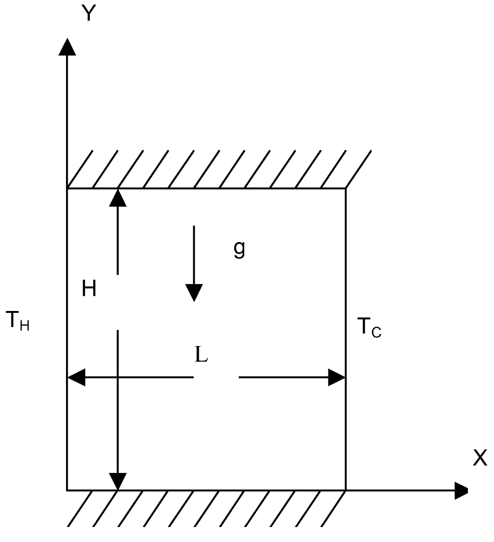

Fig. 1. Geometry of the present problem.

been incorporated only in the body force term by employing the Boussinesq approximation.

2.2. Governing equations and boundary conditions

The non-dimensional transport equations for a steady, two-dimensional flow of a non-Newtonian, fluid are as follows [15];

$$\frac{\partial U}{\partial X}+\frac{\partial V}{\partial Y}=0, \tag{1}$$

$$U\frac{\partial U}{\partial X}+V\frac{\partial U}{\partial Y}=-\frac{\partial P}{\partial X}+\frac{\mu_{\text{app}}}{\rho_{nf,0}\,\alpha_{f,0}}\left[\frac{\partial^2 U}{\partial X^2}+\frac{\partial^2 U}{\partial Y^2}\right], \tag{2}$$

$$U\frac{\partial V}{\partial X}+V\frac{\partial V}{\partial Y}=-\frac{\partial P}{\partial Y}+Gr\,Pr\,Pr\frac{\rho_{f,0}}{\rho_{nf,0}}\left(1-\phi+\phi\frac{\rho_s\beta_s}{\rho_f\beta_f}\right)\theta + \frac{\mu_{\text{app}}}{\rho_{nf,0}\,\alpha_{f,0}}\left[\frac{\partial^2 V}{\partial X^2}+\frac{\partial^2 V}{\partial Y^2}\right], \tag{3}$$

$$U\frac{\partial \theta}{\partial X}+V\frac{\partial \theta}{\partial Y}=\frac{k_{nf}}{k_f}\frac{(\rho C_p)_{f,0}}{(\rho C_p)_{nf,0}}\left[\frac{\partial^2 \theta}{\partial X^2}+\frac{\partial^2 \theta}{\partial Y^2}\right]. \tag{4}$$

The apparent viscosity of the nanofluid is

$$\mu_{\text{app}}=m\left(\frac{\alpha_{f,0}}{h^2}\right)^{(n-1)}\left|\left[2\left\{\left(\frac{\partial U}{\partial X}\right)^2+\left(\frac{\partial V}{\partial Y}\right)^2\right\}+\left(\frac{\partial V}{\partial X}+\frac{\partial U}{\partial Y}\right)^2\right]^{1/2}\right|^{(n-1)}. \tag{5}$$

The boundary conditions, used to solve Eqs. (1) to (4) are as follows:

$$U=V=\frac{\partial\theta}{\partial Y}=0 \quad \text{at } Y=0,\,1.0 \text{ and } 0\leqslant X\leqslant 1.0, \tag{6a}$$

$$\theta=1.0 \quad\text{and}\quad U=V=0 \quad \text{at } X=0 \text{ and } 0\leqslant Y\leqslant 1.0, \tag{6b}$$

$$\theta=0 \quad\text{and}\quad U=V=0 \quad \text{at } X=1.0 \text{ and } 0\leqslant Y\leqslant 1.0. \tag{6c}$$

The effective density of the nanofluid at reference temperature is

$$\rho_{nf,0}=(1-\phi)\rho_{f,0}+\phi\rho_{s,0} \tag{7}$$

and the heat capacitance of nanofluid is

$$(\rho C_p)_{nf}=(1-\phi)(\rho C_p)_f+\phi(\rho C_p)_s. \tag{8}$$

The effective thermal conductivity of nanofluid has been determined by the model proposed by Patel et al. [11]. For the two-component entity of spherical-particle suspension the model gives

$$\frac{k_{\text{eff}}}{k_f}=1+\frac{k_pA_p}{k_fA_f}+ck_pPe\frac{A_p}{k_fA_f} \tag{9}$$

where

$$\frac{A_p}{A_f}=\frac{d_f}{d_p}\frac{\phi}{(1-\phi)} \quad\text{and}\quad Pe=\frac{u_pd_p}{\alpha_f},$$

where u_p is the Brownian motion velocity of the particles which is given by $u_p=(2k_bT)/(\pi\mu_fd_p^2)$.

The effective thermal conductivity can be obtained from Eq. (9).

Eqs. (1) to (4), along with the boundary conditions (6) are solved numerically. From the converged solutions, Nu_y (local Nusselt number) and $\overline{Nu}$ (average Nusselt number) for the hot wall has been calculated in the following ways:

$$Nu_y=-\frac{k_{\text{eff}}}{k_f}\frac{\partial\theta}{\partial X}\bigg|_{0,Y}, \tag{10}$$

$$\overline{Nu}=\frac{1}{H}\int_0^H Nu_y\,dY|_{X=0}. \tag{11}$$

Detail of solution of the governing equations can be obtained from Ref. [15].

3. Artificial neural network

ANN can be divided into two categories based on learning; they are supervised and unsupervised ANN. In unsupervised learning the desired output for input parameters are not specified. In supervised learning, the desired outputs for each input set are provided to the network in the training process and the weights of the network are set on the detail error information obtained from the input–output set supplied to the network. Thus a neural network with supervised learning is required for this purpose [16]. Back-propagation is the first training algorithm to be devised for supervised learning. This algorithm consists of two phases: the forward phase where the activations are propagated from the input to the output layer, and the backward phase, where the error between the observed actual and the requested nominal value in the output layer is propagated backwards in order to modify the weights and bias values by various weight-update rules. In case of simple back-propagation algorithm, the weight update is performed by gradient descent method. Resilient propagation (RPROP) is a variant form of back-propagation.

3.1. Resilient propagation algorithm (RPROP)

The basic principle of RPROP is developed from Manhattan–Learning rule with certain modifications [23,24]. The incremental changes in the weight values, $\Delta W_{ij}^{(t)}$ for a particular iteration is exclusively determined by Eq. (12) and its update in the next iteration is found by Eq. (13).

$$\Delta w_{ij}^{(t)}=\begin{cases}-\Delta_{ij}^{(t)}, & \text{if } \frac{\partial E}{\partial W_{ij}}^{(t)}>0,\\ \Delta_{ij}^{(t)}, & \text{if } \frac{\partial E}{\partial W_{ij}}^{(t)}<0,\\ 0, & \text{else}\end{cases} \tag{12}$$

and

$$W_{ij}^{(t+1)}=W_{ij}^{(t)}+\Delta W_{ij}^{(t)}. \tag{13}$$

Thus RPROP algorithm introduces a personal update value for each weight, $\Delta_{ij}^{(t-1)}$ between neurons of layers i and j at the $(t-1)$th instant of time which evolves during the learning process according to the sight of the local error function E. The gradient information at ith instant over all patterns of the pattern set (batch learning) is denoted by $\frac{\partial E}{\partial W_{ij}}^{(t)}$. The second step of RPROP learning is to determine the new update-values, $\Delta_{ij}(t)$ depending on the topology of the error function. This is based on a sign dependent adaptation process, which works as per Eq. (14).

$$\Delta_{ij}^{(t)}=\begin{cases}\Delta_{ij}^{(t-1)}*\eta^{+}, & \text{if } \frac{\partial E}{\partial W_{ij}}^{(t-1)}*\frac{\partial E}{\partial W_{ij}}^{(t)}>0,\\ \Delta_{ij}^{(t-1)}*\eta^{-}, & \text{if } \frac{\partial E}{\partial W_{ij}}^{(t-1)}*\frac{\partial E}{\partial W_{ij}}^{(t)}<0,\\ \Delta_{ij}^{(t-1)}, & \text{else,}\end{cases} \tag{14}$$

where $0<\eta^{-}<1<\eta^{+}$.

From Eq. (14) it can be concluded that every time the partial derivative of the corresponding weight W_{ij} changes its sign, it indicates that the last update was too big and the algorithm has jumped over a local minimum. The update-value $\Delta_{ij}^{(t)}$ is decreased by the factor η^{-} in such case. If the derivative retains its sign, the update value is slightly increased by the factor η^{+} in order to accelerate convergence in shallow regions. Additionally, in case of a change in sign, there should be no adaptation in the succeeding learning step.

The advantage of this algorithm over back-propagation is that the magnitude of the partial derivative does not influence weight update rather it depends only on the sequence of signs of the derivative. As a further effect, learning is spread equally all over the entire network making it a batch learning process thereby obtaining faster convergence.

3.2. Normalization of data

Both the input and output data sets are normalized since it has been found to have significant effects on learning accuracy and on convergence [25]. Both the inputs and the outputs can be normalized in three ways; by Maximum (Max), by Maximum-Minimum (Max-Min) and Mean & Standard Deviation (M&SD) techniques [25].

Normalization of inputs are done in the following ways

(i) Maximum (Max) method:
The maximum value of the input vector components is determined first as;

$$Inp_{i,\max} = \mathrm{Max}\big(Inp_i(p)\big), \tag{15}$$

where $p =$ number of pattern in the training set and $i = 1, 2, \ldots, N$.
So,

$$Inp_{i,\mathrm{nor}}(p) = Inp_i(p)/Inp_{i,\max}. \tag{16}$$

After this the input variable range is within $(0, 1)$.

(ii) Maximum-Minimum (Max-Min) method:
In this method, the maximum values of the input vector is determined first using Eq. (15) and the minimum is determined by

$$Inp_{i,\min} = \mathrm{Min}\big(Inp_i(p)\big), \tag{17}$$

$$Inp_{i,\mathrm{nor}}(p) = \big(Inp_i(p) - Inp_{i,\min}\big)/(Inp_{i,\max} - Inp_{i,\min}). \tag{18}$$

After normalization the input variable range is within $(0, 1)$.

(iii) Mean & Standard Deviation (M&SD) method:
In this mean & SD method

$$Inp_{i,\mathrm{nor}}(p) = \big(Inp_i(p) - Inp_{i,\mathrm{av}}\big)/\sigma_i, \tag{19}$$

where $Inp_{i,\mathrm{av}} =$ average value of the ith component of the input vector and $\sigma_i =$ standard deviation of the ith component of the input vector.

After normalization the range of the input variable is $(-k_1, k_2)$, where k_1 and k_2 are real positive integer. If the variables are divided by the maximum of k_1 and k_2 then the value will fall within $(-1, 1)$.

4. Result and discussion

4.1. Training of ANN

The ANN architecture proposed here consists of two input parameter, the Rayleigh number (Ra) and the solid volume fraction (ϕ). The architecture has one output, i.e. the Nusselt number ($\overline{Nu}$). The synaptic weights and thresholds of the neural network are fed with suitable initial values before the training begins. As a thumb rule, these matrix elements are initialized in a random manner. Satisfactory network training can only be achieved by proper initialization otherwise it may lead to premature saturation of the network. The training process continues till the desired error limit is reached. A FORTRAN 77 program is developed in house for training the ANN using RPROP algorithm.

For this particular problem we have used the values of $\overline{Nu}$ for $Ra = 10^4$, 5×10^4, 10^5, 5×10^5, 10^6, 5×10^6 and 10^7 and for $\phi = 0.0$ to 5% with an increment of 1% to train the ANN. This is shown in Table 1. Thus total 42 data, which are evenly spread has been used to train the ANN. These data have been obtained from the results of simulation by solving Eqs. (1) to (4) as stated in [15] for a non-Newtonian copper–water nanofluid. As there is no general criterion about deciding the number of nodes in the hidden layers, detailed studies have been carried out on this selection to get the minimum root mean square error (*RMSE*) by trial and error that is unique for the problem at hand. The *RMSE* has been calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}^{N}(a_i - b_i)^2}, \tag{20}$$

here, $a =$ actual output, $b =$ predicted output and $N =$ number of data set.

The guiding criterion of selecting the ANN parameters is to obtain maximum accuracy with minimum computational time. Another 35 data, generated from simulation [15] has been used to validate the ANN model.

4.2. Effect of ANN parameters

Computation has been carried out for different parameters like number of hidden layers, number of node in each hidden layer and the seed value. For each run the number of maximum iteration was kept fixed at 10000. Other parameters were kept unchanged while effect of change of one parameter on the accuracy and time has been studied. The corresponding values of η^+ and η^- were chosen as 1.25 and 0.25 respectively based on literature [25], while value of the initial update of Δ determining the first weight step was considered as 1.0 with the upper bound $\Delta_{\max}$ being set to 50. Maximum allowable *RMSE* value of the output was considered as 0.001.

4.3. Effect of number of hidden layers, nodes and seed

The objective of the work is to reduce the time needed to calculate the heat transfer due to natural convection in a differentially heated square cavity using copper–water nanofluid with maximum accuracy. To achieve this, different number of hidden layers have been considered first. Study has been conducted with single, double and triple hidden layers. In each case the number of connecting nodes has been varied for a wide range to achieve the minimum *RMSE* of the output.

For single hidden layer the number of nodes in the hidden layer has been changed from 3 to 150, keeping the seed value fixed at 600. It was observed that with 39 nodes the error is 3.495051×10^{-3} while with 62 nodes it is minimum with a value of 3.251264×10^{-3}. Keeping the number of nodes fixed, seed value was changed from 550 to 700 for each case. It has been found that with 62 nodes the seed value of 629 gave the minimum *RMSE* of 2.825723×10^{-3} after 6615 iterations.

Study with two hidden layers was conducted thereafter varying the nodes of 1st hidden layer (2nd layer) from 3 to 20. For each 1st hidden layer node, the number of nodes of the 2nd hidden layer (3rd layer) has been varied from 3 to 20. Thus total 18 × 18 cases were studied in this case. It is found that with 18 and 10 numbers of nodes in the 2nd and 3rd layer respectively minimum *RMSE* of 2.352144×10^{-3} was obtained. The seed has been varied from 550 to 700 for these cases. It is found that the minimum *RMSE* has occurred at a seed value of 593 after 7009 iterations. The *RSME* in this case has been found to be 1.908166×10^{-3}.

Studies with three hidden layers (Fig. 2) were performed by varying the number nodes of each hidden layers from 3 to 15 keeping the number of nodes in the other layers fixed. A total of 13 × 13 × 13 cases were considered in this way. This study gave the minimum *RMSE* at 7, 15, 13 nodes for 1st, 2nd and 3rd hidden layers respectively at a seed value 600 after 2741 iteration. The value

Table 1
The input–outputs for training of ANN where average Nusselt number ($\overline{Nu}$) is the output and Rayleigh number (Ra) and solid volume fraction (ϕ) are inputs.

Rayleigh number	Solid volume fraction (%)					
	0.0	1.0	2.0	3.0	4.0	5.0
10^4	2.275919	1.756689	1.601959	1.557881	1.513893	1.529431
5×10^4	3.812990	3.117534	2.653109	2.237802	2.024590	1.947975
10^5	4.719077	3.872105	3.504562	2.926546	2.642346	2.278577
5×10^5	7.568457	6.412295	5.898260	5.153570	4.103301	2.962430
10^6	9.209099	7.875953	7.317188	6.455490	5.548390	4.012743
5×10^6	14.330710	12.478585	11.767651	10.660579	9.663851	7.469041
10^7	17.260213	15.121829	14.336667	13.088992	11.987680	10.649040

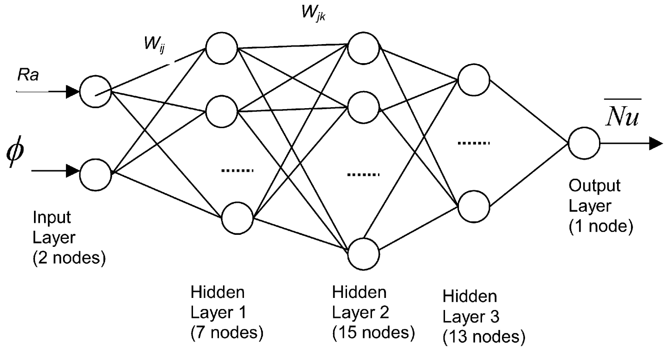

Fig. 2. Schematic diagram of the Artificial Neural Network.

Table 2
Time needed for training.

No. of hidden layers	No. of node in hidden layers	Seed value	No. of iterations	CPU time (in sec)	*RMSE*
Single	39	600	5082	11.5	3.495051×10^{-3}
Single	62	629	6615	17.7	2.825723×10^{-3}
Double	8, 10	593	7009	16.8	1.908166×10^{-3}.
Triple	7, 15, 13	600	2741	7.8	9.95174×10^{-4}

of *RMSE* value was found to be 9.95174×10^{-4}. It is observed that the *RMSE* value for this case is less than the maximum allowable *RMSE* value that was considered to be 0.001.

Since the main objective of the model is to get maximum accuracy with minimum possible time, the *RMSE* and the CPU time for each of the three cases has been observed. The detail of the result has been presented in Table 2. Table 2 shows that though the three hidden layer has maximum number of weights, it gives the most accurate result (minimum *RMSE*) much quicker than the other cases as the number of iterations are less. As the *RMSE* value is minimum, the case of three hidden layers with 7, 15, 13 nodes have been chosen for training. It is to be noted that the literatures available on ANN study using Back Propagation algorithm for mechanical systems either used single hidden layer or sometimes double hidden layer [18,21] for acceptable results. But for the present case, ANN architecture with three hidden layers gave the best result in terms of both accuracy and time.

It has already been stated that normalization of input and output can be done with three ways. Hence altogether nine combinations are possible. To observe the effect of normalization on *RMSE*, all the possibilities of the normalization have been checked and shown in Table 3. This study has been conducted for three hidden layers with 7, 15, 13 nodes in the hidden layers and value of seed have been kept at 600. All the other parameters like initial threshold, η^+ and η^- were kept constant. It has been observed from Table 3 that, normalization by Maximum (Max) for both the inputs and the outputs has been found to provide the best result, as *RMSE* is minimum (9.95174×10^{-4}) for that case.

Table 3
Effect of different kind of normalization for input and output parameters for triple hidden layer configuration.

Input	Output	No. of iterations	Minimum *RMSE*
Maximum	Maximum	2741	9.95174×10^{-4}
Maximum	Mean and S.D	5576	1.03663×10^{-2}
Maximum	Max-Min	9960	1.015341×10^{-2}
Mean and S.D	Maximum	9998	8.000772×10^{-3}
Mean and S.D	Mean and S.D	9997	2.627066×10^{-2}
Mean and S.D	Max-Min	9998	2.611023×10^{-2}
Max-Min	Maximum	9997	5.024211×10^{-3}
Max-Min	Mean and S.D	3314	1.420429×10^{-2}
Max-Min	Max–Min	3430	1.422882×10^{-2}

After training, the average Nusselt number $\overline{Nu}$ was obtained from the same RPROP ANN structure for $Ra = 10^4$, 5×10^4, 10^5, 5×10^5, 10^6, 5×10^6 and 10^7 and for $\phi = 0.5\%$ to 4.5% with an increment of 1%. The results obtained from the ANN have been compared with the result obtained from the simulation [15], which has been shown in Fig. 3. It is evident from Fig. 3 that the values obtained from the ANN model matches well with the values obtained from the simulation. Some deviation has been observed at higher Rayleigh number. But this deviation has been observed near the boundaries of the training data, which is a common feature for any ANN architecture.

Once the RPROP based ANN is trained and its connection weights are assigned then data generation is pretty fast. Test data results reveal that the convergence characteristics of the new method are excellent thus it can be concluded that the training algorithm for the RPROP ANN is quite efficient. The new data generation is quite faster (8 to 10 seconds for ANN and 25–30 min for simulation) compared to the conventional methods as measured by the CPU time requirements. The Mean Relative Error (*MRE*) has been found to be 2.54% while standard deviation in the relative error (STD_R) is 2.46% and BIAS has been found to be 1.1%. Here

$$BIAS = \frac{1}{N}\sum_{i=1}^{N} \Delta A, \tag{21}$$

$$MRE = \frac{1}{N}\sum_{i=1}^{N} ABS(\Delta A), \quad \text{and} \tag{22}$$

$$STD_R = \sqrt{\frac{\sum_{i=1}^{N}(\Delta A - \overline{\Delta A})^2}{N-1}} \tag{23}$$

are calculated based on equations given by Sablani et al. [26].

5. Algorithm of obtaining optimum nodes in the hidden layers

As stated earlier that for each study the number of maximum iteration was kept fixed at 10000. The corresponding values of η^+ and η^- were chosen as 1.25 and 0.25, while value of the initial update of Δ determining the first weight step was considered as 1.0 with the upper bound $\Delta_{\max}$ being set to 50. Maximum allowable

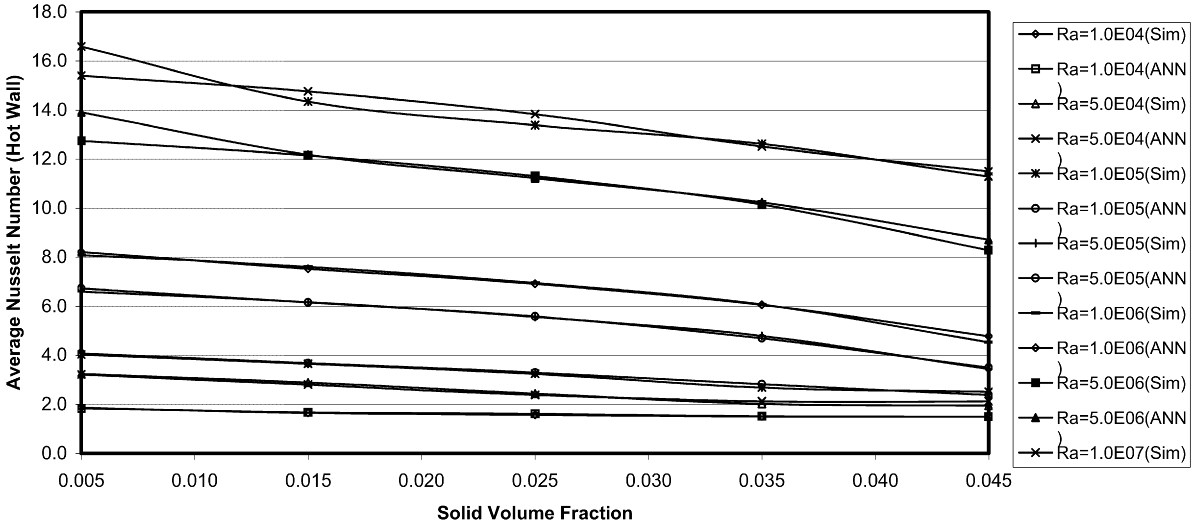

Fig. 3. Comparison of average Nusselt number for different solid volume fraction for different Rayleigh number. Sim – result of simulation, ANN – predicted value from the present ANN model.

Table 4
The input–outputs for training of ANN where average Nusselt number ($\overline{Nu}$) is the output and Rayleigh number (*Ra*) and solid volume fraction (ϕ) are inputs for the second set of study.

Solid volume fraction	Rayleigh number						
	10^4	5.0×10^4	10^5	5.0×10^5	10^6	5.0×10^6	10^7
0.000	2.276	3.813	4.719	7.568	9.209	14.331	17.260
0.005	1.875	3.253	–	–	8.088	12.737	15.401
0.010	1.757	–	3.872	6.412	–	12.479	
0.015	–	2.892	–	6.173	–	12.151	14.762
0.020	–	2.653	–	5.898	–	–	–
0.025	1.575	–	3.249	–	6.951	–	–
0.030	1.558	–	–	–	6.455	10.661	–
0.035	1.509	2.026	–	4.795	–	–	12.511
0.040	–	–	2.642	–	–	–	11.988
0.045	–	1.956	2.514	–	4.535	8.283	11.501
0.050	1.529	1.948	2.279	2.962	4.013	7.469	10.649

RMSE value of the output was considered as 0.001. Initially seed value was fixed at 600.

The steps are follows

1. For single hidden layer the number of nodes in the hidden layer was varied from 3 to 150.
2. For double hidden layers, the number of nodes in the hidden layer was varied from 3 to 20. Thus 18 × 18 combinations was studied.
3. For triple hidden layers, the number of nodes in the hidden layer was varied from 3 to 15. Thus 13 × 13 × 13 combination was studied.
4. Out of the above three studies the minimum *RMSE* (< 0.001) was obtained with triple hidden layer with 7, 15, 13 nodes in the hidden layers.
5. Keeping these nodes fixed, seed value was varied from 551 to 700. Minimum *RMSE* was obtained at 600.
6. Keeping the seed value and number of hidden layers fixed the different normalization technique of input and output data has been studied.

6. Study with random data set

In the previous study the data chosen for the training was in a regular order. In the present case the data has been taken randomly within the simulation range. Forty-six (46) data has been taken for training, while another 31 data has been used to check the accuracy of the ANN. The input data has been listed in Table 4.

Like the previous study the ANN has been trained with singe hidden layer, double hidden layer and triple hidden layer. Number of nodes in the input and output layer were kept at two and one respectively. Only number of nodes in the hidden layers has been varied to obtain minimum *RMSE*. With single hidden layer the minimum *RMSE* has been obtained at 35 nodes, the value of which is 0.003. With double hidden layer, minimum *RMSE* is obtained with 25 and 23 nodes in the 1st and 2nd hidden layer respectively. The value of minimum *RMSE* is 0.00138. With three hidden layers the minimum *RMSE* (0.00211) is obtained at 2, 6, 20 nodes in the first, second and the third layer respectively. For the above study the seed value was kept at 600 and the other parameters are kept same as our previous study.

Hence it is obtained that the minimum *RMSE* occurs with two hidden layer with 25 and 23 nodes. Now keeping the number of nodes and other parameters fixed, the seed value was changed from 501 to 750 and observed that minimum *RMSE* is obtained at 600.

The result obtained from the ANN and the simulation of the 31 data has been compared. The comparison has been shown in Fig. 4. The Mean Relative Error (*MRE*) has been found to be 2.485% while standard deviation in the relative error (STD_R) is 1.952% and BIAS has been found to be −0.2%.

7. Conclusion

Heat transfer due to laminar natural convection of copper–water nanofluid in a differentially heated square cavity, has been

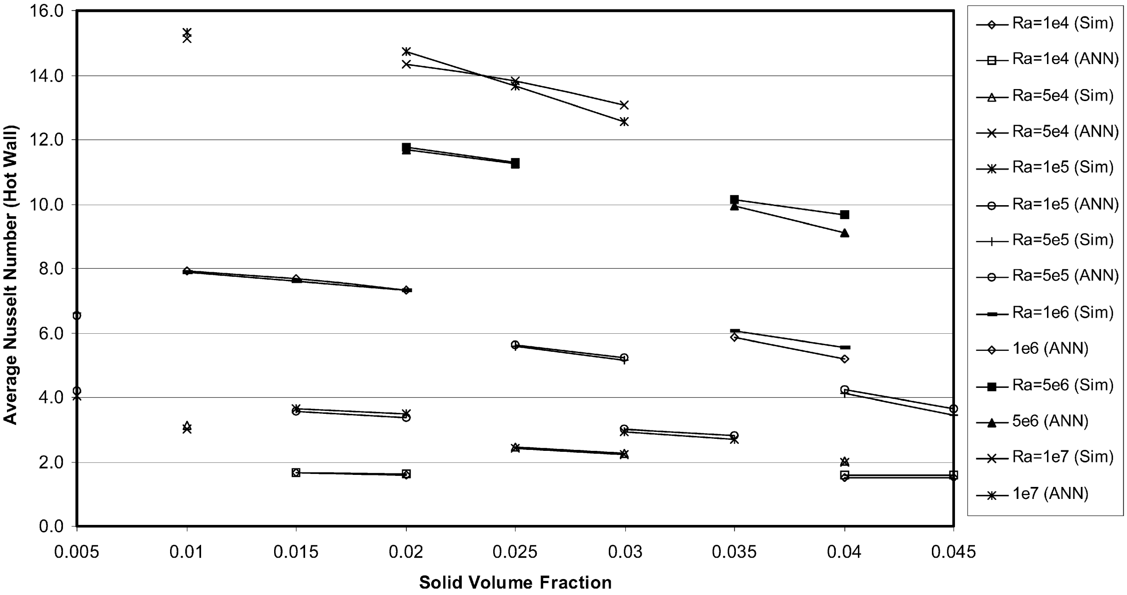

Fig. 4. Comparison of average Nusselt number for different solid volume fraction for different Rayleigh number for random input–output training. Sim – result of simulation, ANN – predicted value from the present ANN model.

predicted by Artificial Neural Network (ANN). The ANN has been trained by a resilient-propagation (RPROP) algorithm with three hidden layers. The ANN has been trained with different Rayleigh number (*Ra*) and solid volume fraction (ϕ) as input while average Nusselt number ($\overline{Nu}$) is considered as output. The required input and output data to train the ANN has been taken from the results of numerical simulation that was performed earlier by the authors [15] where the transport equations for natural convection of non-Newtonian copper–water nanofluid has been solved numerically using finite volume approach using SIMPLER algorithm. The ANN has been trained with data for $Ra = 10^4$, 5×10^4, 10^5, 5×10^5, 10^6, 5×10^6 and 10^7 and for $\phi = 0.0$ to 5% with an increment of 1%. Results has been calculated for $Ra = 10^4$, 5×10^4, 10^5, 5×10^5, 10^6, 5×10^6 and 10^7 and for $\phi = 0.5\%$ to 4.5% with an increment of 1%. Results from simulation as well as ANN for other inputs within the given range of training data have been compared. It has been observed that the ANN predicts the heat transfer correctly within the given range of data. The Mean Relative Error (*MRE*) and the standard deviation in the relative error (*STDR*) have been observed to be 2.54 and 2.46% respectively.

Another set of data containing 46 inputs and outputs within the simulation range has been taken randomly to train the ANN. The remaining 31 input has been used to compare the result of simulation to that of ANN. It is observed that ANN trained with random data also can predict the heat transfer very well. The Mean Relative Error (*MRE*) has been found to be 2.485% while standard deviation in the relative error (*STDR*) is 1.952% and BIAS has been found to be 0.2%.

This RPROP based ANN predicts quite accurately the heat transfer in a differentially heated square cavity using copper–water nanofluid within the training range. The ANN can be extended for different kind of nanofluids. Thus before carrying out the experiment researchers can obtain an idea of heat transfer enhancement very quickly. The RPROP based ANN provides result within 8 seconds while simulation take about 20 to 30 minutes depending on the parameters. This is a very quick and efficient tool to predict the heat transfer than simulation within the trained boundary.

Acknowledgements

We like to acknowledge the support extended by University Grants Commission, Govt. of India. We also thank Jadavpur University, Kolkata for providing all the necessary facilities to carry out this research work.

References

[1] J.A. Eastman, S.U.S. Choi, S. Li, W. Yu, L.J. Thompson, Anomalously increased effective thermal conductivities of ethylene glycol-based nanofluids containing copper nanoparticles, Appl. Phys. Lett. 78 (2001) 718–720.

[2] Y. Xuan, Q. Li, Heat transfer enhancement of nanofluids, Int. J. Heat Fluid Flow 21 (2000) 58–64.

[3] D.W. Zhou, Heat transfer enhancement of copper nanofluid with acoustic cavitation, Int. J. Heat Mass Transfer 47 (2004) 3109–3117.

[4] S.K. Das, N. Putra, P. Thiesen, W. Roetzel, Temperature dependence of thermal conductivity enhancement for nanofluids, J. Heat Transfer 125 (2003) 567–574.

[5] R.L. Hamilton, O.K. Crosser, Thermal conductivity of heterogeneous two-component systems, I & EC Fundamentals 1 (1962) 182–191.

[6] F.J. Wasp, Solid–Liquid Flow Slurry Pipeline Transportation, Trans. Tech. Pub., Berlin, 1977.

[7] J.C. Maxwell-Garnett, Colours in metal glasses and in metallic films, Philos. Trans. Roy. Soc. A 203 (1904) 385–420.

[8] D.A.G. Bruggeman, Berechnung Verschiedener Physikalischer Konstanten von Heterogenen Substanzen, I. Dielektrizitatskonstanten und Leitfahigkeiten der Mischkorper aus Isotropen Substanzen, Ann. Physik Leipzig 24 (1935) 636–679.

[9] B.X. Wang, L.P. Zhou, X.F. Peng, A fractal model for predicting the effective thermal conductivity of liquid with suspension of nanoparticles, Int. J. Heat Mass Transfer 46 (2003) 2665–2672.

[10] D.H. Kumar, H.E. Patel, V.R.R. Kumar, T. Sundararajan, T. Pradeep, S.K. Das, Model for conduction in nanofluids, Phys. Rev. Lett. 93 (2004), 144301-1–144301-3.

[11] H.E. Patel, T. Sundarrajan, T. Pradeep, A. Dasgupta, N. Dasgupta, S.K. Das, A micro-convection model for thermal conductivity of nanofluid, Pramana J. Phys. 65 (2005) 863–869.

[12] N. Putra, W. Roetzel, S.K. Das, Natural convection of nano-fluids, Heat and Mass Transfer 39 (2003) 775–784.

[13] D. Wen, Y. Ding, Natural convective heat transfer of suspensions of titanium dioxide nanoparticles (nanofluids), IEEE Trans. Nanotechnol. 5 (2006) 220–227.

[14] H. Chang, C.S. Jwo, C.H. Lo, T.T. Tsung, M.J. Kao, H.M. Lin, Rheology of CuO nanoparticle suspension prepared by ASNSS, Rev. Adv. Material Sci. 10 (2005) 128–132.

[15] A.K. Santra, S. Sen, N. Chakraborty, Study of heat transfer augmentation in a differentially heated square cavity using Copper–water nanofluid, Int. J. Thermal Sci. 47 (2008) 1113–1122.

[16] L. Fausett, Fundamentals of Neural Networks, Pearson Education, 2004.

[17] S. Haykin, Neural Networks, Addison Wesley Longman Pte. Ltd., Singapore, 2001.

[18] H.M. Ertunc, M. Hosoz, Artificial neural network analysis of a refrigeration system with an evaporative condenser, Appl. Therm. Engrg. 26 (2006) 627–635.

[19] K. Jambunathan, S.L. Hartle, S. Ashforth-Frost, V.N. Fontama, Evaluating convective heat transfer coefficients using neural networks, Int. J. Heat Mass Transfer 39 (1996) 2329–2332.

[20] A. Parlak, Y. Islamoglu, H. Yasar, A. Egrisogut, Application of artificial neural network to predict specific fuel consumption and exhaust temperature for a diesel engine, Appl. Therm. Engrg. 26 (2006) 824–828.

[21] E. Arcaklioglu, Performance comparison of CfCs with their substitutes using artificial neural network, Int. J. Energy Res. 28 (2004) 1113–1125.

[22] B. Widrowo, M.E. Hoff, Adaptive switching networks, in: Parallel Distributive Processing, vol. 1, MIT Press, MA, 1986, pp. 318–362.

[23] M. Riedmiller, H. Braun, A direct adaptive method for faster back propagation algorithm learning: The RPROP algorithm, in: Proc. on IEEE Int. Conf. on Neural Networks (ICNN), San Francisco, 1993.

[24] M. Riedmiller, RPROP – Description and Implementation Details, Technical Report, University of Karlsruhe, January 1994.

[25] N. Chakraborty, S. Chakraborty, P.K. Mukherjee, Adaptive ANN applied to optimise economy in generation system with transmission loss, Journal of the Institution of Engineers (India), EL 77 (1997) 212–219.

[26] S.S. Sablani, A. Kacimov, J. Perret, A.S. Mujumdar, A. Campo, Non-iterative estimation of heat transfer coefficients using artificial neural network models, Int. J. Heat Mass Transfer 48 (2005) 665–679.