IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 10, No. 2, April, 2023, pp. 565-571
Journal homepage: http://iieta.org/journals/mmep

Design Optimal Neural Network for Solving Unsteady State Confined Aquifer Problem

Muna H. Ali[1,2], Luma N.M. Tawfiq[1*]

[1] Department of Mathematics, College of Education for Pure Science Ibn Al-Haitham, University of Baghdad, Baghdad 10011, Iraq
[2] Department of Mathematics, College of Education, University of Al-Anbar, Al Rumadi 31001, Iraq

Corresponding Author Email: luma.n.m@ihcoedu.uobaghdad.edu.iq

https://doi.org/10.18280/mmep.100225

Received: 22 October 2022
Accepted: 23 December 2022

Keywords:
anisotropic confined aquifers model, BP-training algorithm, FFNNs, neural networks, unsteady state problems

ABSTRACT

This article presents the mathematical model of unsteady state horizontal radial flow in homogenous and anisotropic confined aquifers in polar coordinate which represented an important problem. We design optimal neural network to solve this problem. Also, we estimate the parameters of transmissivity aquifer with high accuracy. The optimality depending on choosing optimum architectural with suitable training algorithm and transfer function. The best design neural network is trained by new training algorithm say LNA. The retrained almost fast by suitable transferring function. Based on the modified architecture, the training and testing phases of the solving process are divided into two. The dataset is separated into three sections during the training phase: 60% of the training data, 20% of the validation data, and 20% of the test data. The total square error (TSE) for the trained phase with using the bach propagation algorithm (BPA) is 4.1499e+01 while it is 9.5898e-03 if we using suggested training algorithm (LNA). The LNA has a much smaller TSE when compared to the BPA. These much smaller values of the TSE indicate that LNA performs better than the BPA for the same number of iterations. So suggested architecture has many advantages such loss function computed on a random sample of the domain, high performance, avoid local minima and can be adapted for the online dynamic modeling, automation, control and robotics applications.

1. INTRODUCTION

Partial differential equations overall unsteady state partial differential equation (USSPDE) arises in some branches of science and engineering. Unsteady-state operation can arise from such things as variations in feedwater quality (and so organic load), permeate flow rate (and hence hydraulic load) and aeration rate. In an experiment the effects of unstable flow and sludge wastage were assessed [1], it was established that the level of carbohydrate in the supernatant before and after each sludge withdrawal increased. Whilst the increase following wastage was thought to be due to the sudden stress experienced by cells due to biomass dilution (which in extreme cases is known to lead to foaming in full-scale plant), increase before sludge withdrawal was attributed to the high MLSS concentration and the resulting low DO level in the bioreactor. It was concluded that unsteady-state operation changed the nature and/or structure (and fouling propensity) of the carbohydrate rather than the overall EPS formation. These findings corroborated results previously reported on effects of transient conditions in feeding patterns: the addition of a pulse of acetate in the feedwater has been shown to decrease significantly the biomass filterability due to the increase in SMP levels produced [2]. More detailed characterization of the impact of a wide range of unsteady-state conditions on the EPS present in activated sludge has recently been presented [3]. Along with changes in DO level, variation in the ratio of monovalent and polyvalent cations present in the feedwater can result in sludge deflocculation, usually leading to increased supernatant SMP levels. In the experiments [4], high monovalent/polyvalent ratios resulted in significant deflocculation and decline in hydraulic performance [5].

Hence being of fundamental importance the existence of suitable methods to find their solution [6]. Since exact solution is available only in limited cases, the construction of efficient numerical [7, 8] or approximate approaches is essential [9-11]. Effective methods, such as the Adomain decomposition method (ADM) [12, 13]; HPM [14, 15] and finite element method (FEM) [16], have been developed for solving unsteady state partial differential equation (USSPDE) in stationary environments [17]. Given unsteady state problem with known parameters and initial or boundary conditions, the numerical methods [18-20] can be used to calculate the approximate solution at some points in the domain (discrete case), producing the results as a table can be interpolated to get the solution elsewhere in the domain. One disadvantage of such methods is that, necessary to discrete the domain as meshes so increase the size of the table significantly therefore memory required [21, 22]. Neural networks (ANNs) provide an optimal representation technique for adaptive USSPDE solutions since they are adjust parameters that can be modified by using suitable training algorithm [23, 24] and because of their ability to approximate well any nonlinear functions on compact space. So, ANNs give solutions are characterized by other advantages over solutions getting by FEM and FDM. One advantage is that the solution is represented by a small

number of parameters, which reduces the amount of required memory [25-27]. Another advantage is that the solution is valid on the domain without need for interpolation. Commonly ANN can be implemented to provide a solution of unsteady state problems many of applications of ANNs for solving PDEs can be found in researches [22].

This article has been arranged as follows: In section 2, define and gives a background of the ANNs. In section 3, mathematical model of unsteady state confined aquifers is presented. Design optimal ANN for solving USSPDE is given and demonstrated through example problems in with implementation and discussions for the results will be given in section 4. Finally, the conclusion is given in section 5.

2. NEURAL NETWORKS

A neural network is a parallel, distributed information processing structure consisting of processing elements interconnected via unidirectional signal channels called connections. Each processing element has a single output connection that branches ("fans out") into as many collateral connections as desired; each carries the same signal, the processing element output signal. The processing element output signal can be of any mathematical type desired. The information processing that goes on within each processing element can be defined arbitrarily with the restriction that it must be completely local; that is, it must depend only on the current values of the input signals arriving at the processing element via impinging connections and on the values stored in the processing element's local memory [13].

Thus for a given input vector *x*, the input to this neuron is $W_j^T x$. We assume that each of the hidden neurons has identical transfer function σ, but that bias b_j. So the output from the j^{th} hidden neuron is $\sigma(W_j^T x + b_j)$.

Now we denote the weight connecting the j^{th} hidden node to the output by υ_j. The output function *g(x)* of the ANN is therefore [1]:

$$g(x) = \sum_{j=1}^{k} \upsilon j \ \sigma(W_j^T x + b_j)$$

Note that σ must be sigmoidal functions, so we choice suitable σ herein defined as [18]:

$$\sigma(n_i) = \frac{e^{n_i} - 1}{e^{n_i} + 1}$$

Then, the ANN input-output equation is

$$\hat{Y} = \Phi(x^T w^T + b^T)\upsilon^T$$

where, $W \in R^{n\times r}$; $\upsilon \in R^{1\times n}$; and $b \in R^{n\times 1}$, are the adjustable input weights, output weights and bias respectively.

The structure of interconnections ANN can be classified to different classes of ANNs architecture such feed forward neural network (FFNN) and Feedback neural network (FBNN). Herein we choose FFNN.

3. MATHEMATICAL MODEL OF THE PROBLEM WITH UNSTEADY STATE CONDITIONS

Unsteady state flow occurs from the moment pumping starts until steady state flow is reached. Consequently, if confined aquifer is an infinite, horizontal, completely and it's of constant thickness is pumped at a constant rate; there will always be unsteady-state flow. In practice, the flow is considered to be unsteady as long as the changes in water level in the well and piezometers are measurable or in other words, as long as the hydraulic gradient is changing in a measurable way [26].

We will solve Eq. (1) that govern the radial flow with homogenous and anisotropic hydraulic conductivities in confined aquifer, where $\frac{\partial h}{\partial t} \neq 0$.

$$\frac{1}{r'}\frac{\partial h}{\partial r'} + \frac{\partial^2 h}{\partial r'^2} = \frac{S}{T}\frac{\partial h}{\partial t} \tag{1}$$

where, *h* is hydraulic head, *S* is storage coefficient, *T* is transmissivity, *t*: time, $r'=r\ T_\theta$, and $r = \sqrt{x^2 + y^2}$ is polar coordinate.

$$r' = \sqrt{\frac{T_y\, x^2 + T_x\, y^2}{T}}, \qquad T = \sqrt{T_x\, T_y}$$

As in Theis equation, suppose

$$u = \frac{r'^2 S}{4Tt} \tag{2}$$

Let $\alpha = \frac{S}{T}$, then:

$$\frac{\partial u}{\partial r'} = \frac{\alpha\, r'}{2t} = \frac{2u}{r'} \quad and \quad \frac{\partial u}{\partial t} = \frac{-\alpha\, r'^2}{4t^2} = \frac{-u}{t} \tag{3}$$

We can write Eq. (1) as following:

$$\frac{1}{r'}\frac{\partial}{\partial r'}\left(r'\frac{\partial h}{\partial r'}\right) = \alpha\frac{\partial h}{\partial t} \tag{4}$$

Then Eq. (4) becomes:

$$\frac{1}{r'}\frac{d}{du}\left(r'\frac{dh}{du}\frac{\partial u}{\partial r'}\right)\frac{\partial u}{\partial r'} = \alpha\frac{dh}{du}\frac{\partial u}{\partial t} \tag{5}$$

Now, from Eq. (2) and Eq. (3), we have:

$$\frac{1}{r'}\frac{d}{du}\left(r'\frac{dh}{du}\frac{2u}{r'}\right)\frac{2u}{r'} = \alpha\frac{dh}{du}\frac{-u}{t} \quad \rightarrow \quad \frac{d}{du}\left(u\frac{dh}{du}\right) = \frac{dh}{du}\frac{-\alpha r'^2}{4t} = -u\frac{dh}{du}$$

So, the following ODE is obtained:

$$\frac{d}{du}\left(u\frac{dh}{du}\right) = -u\frac{dh}{du} \ \rightarrow \ \frac{dh}{du} + u\frac{d^2h}{du^2} = -u\frac{dh}{du} \ \rightarrow \frac{d^2h}{du^2} = -\frac{(1+u)}{u}\frac{dh}{du} \tag{6}$$

Suppose that $h' = \frac{dh}{du}$, then Eq. (6) becomes:

$$\frac{dh'}{du} = -\frac{(1+u)}{u}h' \;\; or \;\; equivalently \;\; \frac{dh'}{h'} = -\frac{(1+u)}{u}\,du \tag{7}$$

Integration Eq. (7) to get:

$$ln(h') = -u - ln(u) + c \;\; \rightarrow \;\; ln(h'u) = c - u \;\; \rightarrow \;\; h'u = e^{c}e^{-u} \tag{8}$$

By Eq. (3) we can get:

$$r' \;\frac{\partial h}{\partial r'} = r' \;\frac{dh}{du}\frac{\partial u}{\partial r'} = r'h'\frac{2u}{r'} = 2uh' \tag{9}$$

In addition, from the fact ($r' = r\,T_\theta$) we get:

$$r\;\frac{\partial h}{\partial r} = \frac{r'}{T_\theta}\;\frac{\partial h}{\partial r'}\;\frac{\partial r'}{\partial r} = \frac{r'}{T_\theta}\;\frac{\partial h}{\partial r'}\;T_\theta = r'\;\frac{\partial h}{\partial r'} \tag{10}$$

From Darcy's law:

$$\lim_{r\to 0} r\;\frac{\partial h}{\partial r} = -\frac{Q}{2\pi T_r} \tag{11}$$

where, $T_r = \frac{T_y\,T_x}{T_y\,cos^2(\theta)+T_x\,sin^2(\theta)}$;

Since $r'=r\,T_\theta$ and from Eqns. (8-11) we have

$$\lim_{r\to 0} r\;\frac{\partial h}{\partial r} = \lim_{r'\to 0} r'\;\frac{\partial h}{\partial r'} = \lim_{u\to 0} 2uh' = -\frac{Q}{2\pi T_r} \tag{12}$$

By Eq. (8) and Eq. (12) we can get:

$$\lim_{u\to 0} 2uh' = \lim_{u\to 0} e^{-u}e^{c} \;\; \rightarrow \;\; -\frac{Q}{4\pi T_r} = e^{c} \tag{13}$$

Substituted Eq. (13) in Eq. (8) to get:

$$h'u = -\frac{Q}{4\pi T_r}\,e^{-u} \;\rightarrow\; h' = -\frac{Q}{4\pi T_r}\,\frac{e^{-u}}{u} \;\rightarrow\; \frac{dh}{du} = -\frac{Q}{4\pi T_r}\,\frac{e^{-u}}{u} \;\rightarrow\; dh = -\frac{Q}{4\pi T_r}\,\frac{e^{-u}}{u}\,du \tag{14}$$

Integrating Eq. (14), to get:

$$h = -\frac{Q}{4\pi T_r}\int_u^\infty \frac{e^{-\tau}}{\tau}\,d\tau + C \tag{15}$$

Using boundary condition $h(\infty,t)=h_0$, we have that $C=h_0$. Finally Eq. (15) becomes:

$$s_r = h_0 - h = \frac{Q}{4\pi T_r}\int_u^\infty \frac{e^{-\tau}}{\tau}\,d\tau \tag{16}$$

The integral term appeared in Eq. (16) is called the Theis well function and denoted by W(u). There are different methods and techniques have been used to approximate the value of well function which vary according to the required of accuracy. Now we can write Eq. (16) as following:

$$s_r = \frac{Q}{4\pi T_r}\,W(u) \tag{17}$$

Now, suppose that the distance r in x-direction is rx, in case θ = 0o and hence we have Tr = Tx and from the fact:

$$r' = \sqrt{\frac{T_y\,x^2 + T_x\,y^2}{T}} = \sqrt{\frac{T_y\,r^2cos^2(\theta) + T_x\,r^2sin^2(\theta)}{T}} \;\; \therefore \; r' = r\sqrt{\frac{T_y\,cos^2(\theta) + T_x\,sin^2(\theta)}{T}} \tag{18}$$

We have $r' = r_x\sqrt[4]{\frac{T_y}{T_x}}$. So, Eq. (2) becomes:

$$u = \frac{r^2 S}{4\,T_x\,t} \tag{19}$$

and Eq. (17) becomes:

$$s_{r_x} = \frac{Q}{4\pi T_x}\,W(u) \tag{20}$$

Similarly, the distance r in y-direction is ry, in case $\theta = \frac{\pi}{2}$ and hence we have Tr = Ty and by Eq. (18) *we have* $r' = r_y\sqrt[4]{\frac{T_x}{T_y}}$, then Eq. (2) becomes:

$$u = \frac{r^2 S}{4\,T_y\,t} \tag{21}$$

and Eq. (17) becomes:

$$s_{r_y} = \frac{Q}{4\pi T_y}\,W(u) \tag{22}$$

The Eqns. (19-22) permit determination of the formation constants *S*, *Tx* and *Ty* by means of pumping tests of wells. These equations can be applied in practice because (l) determined a value for *S* (2) required only one observation well (3) in general a shorter period of pumping, and (4) not required assumption of steady-state flow conditions [20].

Because of the mathematical difficulties encountered in applying Eq. (20) and Eq. (22) then we will use the ANN suggested design to determine the values of parameter.

4. DESIGN OPTIMAL ANN

We suggest three fully inter connection layers, there are 3 neurons in the input layer, 12 neurons in the hidden layer with transfer function say (tanhsig.) and 2 neurons in the output layers say (linsig.) transfer function illustrated in Figure 1. We train the suggested ANN by update back propagation rule. To obtain the mathematical description of Figure 1, *φ(k)* in Eq. (23) is decomposed into the past input, past output and past prediction error parts as in Eq. (25) respectively.

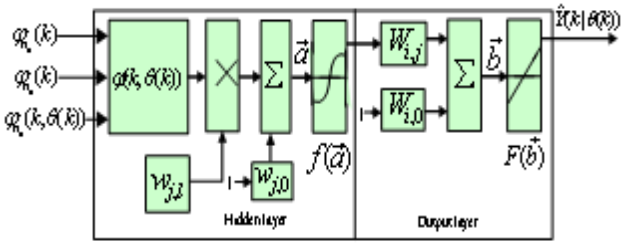

Figure 1. Architecture of the optimal ANN

$$\hat{\theta}(k) = \arg\min_{\theta} \vec{J}(Z^N, \varphi(k), \theta(k)) \tag{23}$$

where, $\hat{\theta}(k)$ is the value of $\theta(k)$ that minimizes Eq. (23) and $\vec{J}(Z^N, \varphi(k), \theta(k))$ is formulated as a total square error (TSE) type error function given as:

$$\vec{J}(Z^N, \varphi(k), \theta(k)) = \frac{1}{2N} \sum_{k=1}^{N} [\varepsilon(k, \theta(k))]^2 \tag{24}$$

$$\varphi_{n_b}(k) = [\, u(k-d), \ldots, u(k-d-m)]^{\tau}$$

$$\varphi_{n_a}(k) = [\, y(k-1), \ldots, y(k-n)]^{\tau} \tag{25}$$

and

$$\varphi_{n_a}(k) = [\, \varepsilon(k-1, \theta(k), \ldots, \varepsilon(k-n, \theta(k))]$$

The outputs of the ANN model illustrated in Figure 1 can be expressed in terms of the network parameters as:

$$\hat{y}(k|\theta(k)) = F_i \left(\sum_{j=1}^{m+n} W_{i,j}\, f_j(\vec{a}) + W_{i,0} \right) \tag{26}$$

where,

$$\vec{a} = \sum_{l_m=1}^{m} W_{j,l_m}\, \varphi_{l_m}(k) + \sum_{i_n}^{n} W_{j,i_n}\, \varphi_{i_n}(k-1) + W_{j,0}$$

j is the number of hidden neurons; (W_{j,l_m} *and* W_{j,i_n}) and $W_{i,j}$ are the hidden and output weights respectively; $W_{j,0}$ and $W_{i,0}$ are the hidden and output biases; $F_i(\vec{b})$ is a linear activation function for the output layer and $f_i(\vec{a})$ is an hyperbolic tangent activation function for the hidden layer given here as:

$$f_i(\vec{a}) = 1 - \frac{2}{e^{2-\vec{a}} + 1} \tag{27}$$

Here, the network weights and biases constitute $\theta(k)$.

The minimization problem of Eq. (23) is re-written as:

$$\hat{\theta}(k) = \arg\min_{\theta} \vec{J}(Z^N, \varphi(k), \theta(k)) \tag{28}$$

It should be noted that there are several ways on how Z^N can be presented to the training algorithm at each time step. The most popular way being the batch mode where all the data set is evaluated at each time step.

The minimization of Eq. (28) is based on an iterative procedure which starts with a randomly initial $\theta(k)$ and updates $\hat{\theta}(k)$ iteratively according to the following typical updating rule:

$$\theta_{I+1}(k) = \theta_I(k) + \Delta\theta_I(k) \tag{29}$$

where, $\theta_I(k)$ denotes $\theta(k)$ at the current iteration I, $\Delta\theta_I(k)$ is the searching direction, $\theta_{I+1}(k)$ is the global minimum and $\hat{\theta}(k) = \theta_{I+1}(k)$ if certain stopping criteria are satisfied.

The most commonly used method for updating $\theta_{I+1}(k)$ is the BPA. This algorithm uses the gradient method [8].

BPA has been reported to be characterized by poor performance [16]. To improve the performance of the BP algorithm, we suggest the following training algorithm say LN- algorithm (LNA). LNA uses the linear approximation error $\tilde{\varepsilon}_{\tau}(k, \theta(k))$ to the error $\varepsilon(k, \theta(k))$ in (27) expressed as:

$$\begin{aligned} \tilde{\varepsilon}_{\tau}(k, \theta(k)) &= \varepsilon(k, \theta_{\tau}(k)) \\ &\quad + [\theta(k) \\ &\quad - \theta_{\tau}(k)]^T \nabla [\varepsilon(k, \theta(k))]\big|_{\theta(k)-\theta_{\tau}(k)} \\ &= \varepsilon(k, \theta_{\tau}(k)) \\ &\quad - [\theta(k) \\ &\quad - \theta_{\tau}(k)]^T \nabla [\hat{y}(k|\,\theta(k))]\big|_{\theta(k)-\theta_{\tau}(k)}^{T} \end{aligned} \tag{30}$$

We obtain the LNA searching direction given as:

$$\Delta\theta_I(k) = -R[\theta_I(k)]^{-1} G[\theta_I(k)] \tag{31}$$

where, $G[\theta_I(k)]$ and $R[\theta_I(k)]$ are the gradient and exact LNT Hessian matrices respectively.

Using standard BPA the LNA can be stated as: $\theta_{I+1}(k) = \theta_I(k) + \Delta\theta_I(k)$.

Thus, we summarize the procedure of training ANN based on suggested update LNA for in a stepwise as follows:

(1) Specify τ, τ_{max}, D, $\lambda_{max} \in [1, 10^3]$ and m, n for $\theta(k)$; where $D = \alpha_a\, I\, [\alpha_h \;\; \alpha_0]$; I is an identity matrix, α_h and α_0 are the weight decay values for the input-to-hidden and hidden-to-output layers respectively.

(2) Initialize $\lambda_{\tau} \in [0.1, 10^{-2}]$, $\delta_{\tau} \in [0.1,\ 10^{-4}], s \in [0.1, 10^{-2}], \theta(k)$ and evaluate $\vec{J}(Z^N, \varphi(k), \theta(k))$ in the following:

$$\begin{aligned} \vec{J}(Z^N, \varphi(k), \theta(k)) = \frac{1}{2N} \Bigg(& \sum_{k=1}^{N} [\varepsilon(k, \theta(k))]^2 \\ & + \theta^T(k) D \theta(k) \Bigg) \end{aligned} \tag{32}$$

(3) While τ=1, compute $G[\theta_{\tau}(k)]$ and $\Delta\,\theta_{\tau}(k)$ using:

$$\theta_{\tau+1}(k) = \theta_{\tau}(k) - \Delta\theta_{\tau}(k)$$

$$\Delta\theta_{\tau}(k) = -[\, R[\widehat{\theta_{\tau}}(k)] + \lambda_{\tau} I]^{-1}\, G[\theta_{\tau}(k)]$$

(4) Evaluate the ratio α_{τ} from the following:

$$\alpha_{\tau} = \frac{\vec{J}(\theta_{\tau}(k)) - \vec{J}(\theta_{\tau}(k) + \Delta\theta_{\tau}(k))}{\vec{J}(\theta_{\tau}(k)) - \vec{J}_{\tau}(\theta_{\tau}(k) + \Delta\theta_{\tau}(k))} \tag{33}$$

(5) Update λ_{τ} according to the following conditions

on α_τ If α_τ>0.75, then $\lambda_\tau \leftarrow 0.5 * \lambda_\tau$ and go to Step 6. If α_τ<0.25, then $\lambda_\tau \leftarrow 2 * \lambda_\tau$ and go to Step 6.

(6) If $\vec{J}(\theta_\tau(k)) - \vec{J}(\theta_\tau(k) + \Delta\theta_\tau(k))$ the $\theta_{\tau+1}(k) \leftarrow \theta_\tau(k) + \Delta\theta_\tau(k)$ set $\tau \leftarrow \tau +$ and $\lambda_{\tau+1} \leftarrow \lambda_\tau$.

(7) If $|S_\tau(\theta_{\tau+1}(k) - \theta_\tau(k))| > S_\tau$ or $\tau > \tau_{max}$ or $\lambda_\tau > \lambda_{max}$(number of iterations); go to Step 8, else go to Step 3.

(8) Accept $\hat{\theta}(k) \leftarrow \theta_{\tau+1}(k)$ and terminate.

5. VALIDATION OF THE TRAINED NETWORK: RESULTS AND DISCUSSION

This section illustrates the implementation of suggested design in previous section and discussed its results. Firstly, we choose the sample data then distributed in three sets: training set, validation set and testing set with the following rate distribution: 60% from data samples for the training set, 20% for validation set and 20% for testing set. We start with training ANN by suggested algorithm then testing ANN and then validating the ANN.

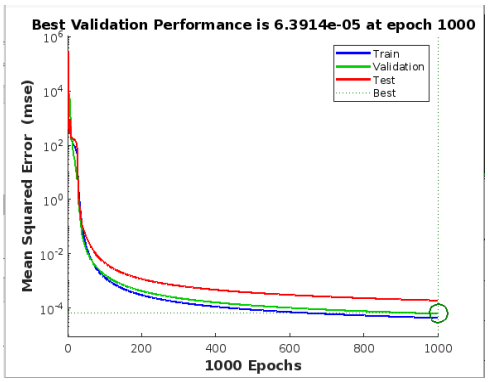

Figure 2. Comparison results between training, testing and validation for ANN

The convergences of the results based on LNA for training suggested design to the mathematical model of unsteady state horizontal radial flow in homogenous and anisotropic confined aquifers in polar coordinate are shown in Figure 2. It can be seen in Figure 2, LNA shows faster but poor convergences with smaller computation times in the test as shown in the Figure 2, when compared to validation. The superior convergences of the LNA compared to the BPA are illustrated in Tables 1, 2 and in Figure 3, for Transmissivity T_x, Table 3, illustrate the value of transmissivity T_x for all different distance r_x computed by suggested design based on LNA (ANN-LNA).

The total square error (TSE) for the ANN trained with the BPA and the LNA are given in the third row of Table 2. The LNA has a much smaller TSE when compared to the BPA. These much smaller values of the TSE and the MPIs indicate that LNA performs better than the BPA for the same number of iterations.

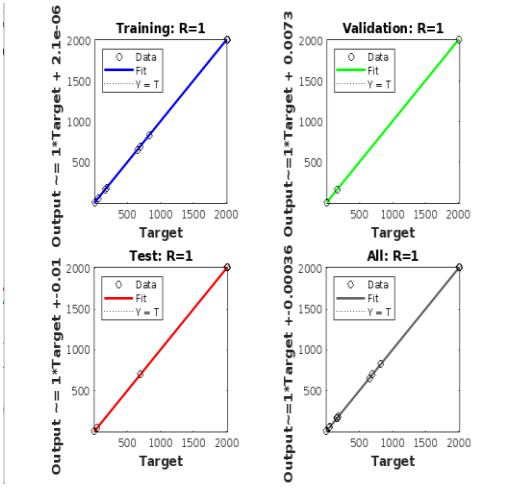

Figure 3. Target for training and validation

Table 1. Training results

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	1000	1000
Elapsed Time	-	00:11:51	-
Performance	2.86e+05	4.44e-05	0
Gradient	1.1e+06	0.158	1e-07
Mu	0.001	0.01	1e+010
Validation Checks	0	0	6

Table 2. Performance comparison of the BPA and LNA training results

S/N		Left rotation angle		Right rotation angle	
		BPA	LNA	BPA	LNA
1	Computation time for model identification (sec)	5.8120e+02	1.5109e+03	5.7036e+02	1.3225e+03
2	Minimum performance index (MPI)	5.2343e-02	4.4952e-05	2.6659e-02	8.0105e-05
3	Total square error (TSE)	4.1499e+01	9.5898e-03	1.1104e+01	2.6953e-03
4	Mean error of one step ahead prediction of training data	1.0442e-01	1.1104e-04	7.0847e-01	2.6953e-04
5	Mean error of one step prediction of test data	3.2832e+01	5.1594e-02	2.8006e+01	1.1955e-03
6	Mean value of 5-step ahead prediction error	4.8886e+01	5.0041e-02	5.1418e+01	5.0736e-02
7	final prediction error (FPE) estimate	3.7989e+01	9.8119e-02	5.1532e+01	1.4494e-02

Table 3. Transmissivity T_x for different distance r_x by ANN-LNA

$r_x = x$	h_x	T_x
6	11.642030326	100
10	13.65248161	100
15	15.83123131	100
20	17.33594252	100
25	17.94977138	100

5. CONCLUSIONS

A new and comprehensive design of ANN based update back propagation training based on update Newton algorithm for the adaptive unsteady state horizontal radial flow in homogenous and anisotropic confined aquifers in polar coordinate has been presented, implemented and demonstrated in this paper with proven results.

Optimal design for ANN training for nonlinear model identification that can be used to design adaptive ANN. For guaranteed stability and positive definitiveness, we used 2nd order derivative of the nonlinear optimization for training nonlinear adaptive design, and fault tolerant capabilities as demonstrated in this article.

Furthermore, the sampling time of suggested design is 0.5 second and/or with/ without noise and disturbances. The average computation time for the serial and parallel implementation are approximately 31.5259 and 4.4117 seconds respectively at each identification. Note that none of the serial or parallel implementation meets the sampling time of 0.5 seconds for the suggested design.

In future we can use other type of ANN for solving the problem or other training algorithm.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the ministry of agriculture and ministry of planning for provided this article data.

REFERENCES

[1] Judd, S., Judd, C. (2011). The MBR Book, Second Edition. Elsevier Ltd.

[2] Cai, J., Meng, Z.B., Khan, A.S., Li, Z.Y., O'Reilly, J., Tong, Y. (2018). Island loss for learning discriminative features in facial expression recognition. In 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG2018), Xi'an, China, pp. 302-309. https://doi.org/10.1109/FG.2018.00051

[3] Muhammad, R.S., Younis, M.I. (2017). The limitation of pre-processing techniques to enhance the face recognition system based on LBP. Iraqi Journal of Science, 58(581B): 355-363.

[4] Goodfellow, I.J., Erhan, D., Carrier, P.L. et al. (2013). Challenges in representation learning: A report on three machine learning contests. In International Conference on Neural Information Processing, pp. 117-124. https://doi.org/10.1007/978-3-642-42051-1_16

[5] Tawfiq, L.N.M., Ali, M.H. (2020). Efficient design of neural networks for solving third order partial differential equations. Journal of Physics: Conference Series, 1530: 012067. https://doi.org/10.1088/1742-6596/1530/1/012067

[6] Lee, M., Lee, Y.K., Lim, M.T., Kang, T.K. (2020). Emotion recognition using convolutional neural network with selected statistical photoplethysmogram features. Applied Sciences, 10(10): 3501. https://doi.org/10.3390/app10103501

[7] Al-Jawary, M.A., Ibraheem, G.H. (2020). Two meshless methods for solving nonlinear ordinary differential equations in engineering and applied sciences. Nonlinear Engineering, 9(1): 244-255. https://doi.org/10.1515/nleng-2020-0012

[8] Hussien, Z.K., Dhannoon, B.N. (2020). Anomaly detection approach based on deep neural network and dropout. Baghdad Science Journal, 17(2): 701-709. http://dx.doi.org/10.21123/bsj.2020.17.2(SI).0701

[9] Anggoro D.A. Supriyanti, W. (2019). Improving accuracy by applying Z-score normalization in linear regression and polynomial regression model for real estate data. International Journal of Emerging Trends in Engineering Research, 7(11): 549-555. http://dx.doi.org/10.30534/ijeter/2019/247112019

[10] Jarah, N.B. (2021). Suggesting multipath routing and dividing the zone to process communication failure in Ad Hoc networks. Iraqi Journal of Science, 62(5): 1702-1709. https://doi.org/10.24996/ijs.2021.62.5.33

[11] Shanthi, T. Sabeenian, R.S. Anand, R. (2020). Automatic diagnosis of skin diseases using convolution neural network. Microprocessors and Microsystems, 76: 103074. https://doi.org/10.1016/j.micpro.2020.103074

[12] Bisht, S., Singh, S.B., Ekata, G. (2020). Reliability and profit function improvement of a cyclic transmission network using artificial neural network. Mathematics in Engineering, Science and Aerospace (MESA), 11(1).

[13] Gupta, R, Batra, CM. (2022). Performance assessment of solar-transformer-consumption system using neural network approach. Baghdad Science Journal, 19(4): 865-874. https://doi.org/10.21123/bsj.2022.19.4.0865

[14] Ghazi, F.F., Tawfiq, L.N.M. (2020). New approach for solving two dimensional spaces PDE. Journal of Physics: Conference Series, 1530: 012066. https://doi.org/10.1088/1742-6596/1530/1/012066

[15] Abdulameer, I.H., Al-Dabbagh, R.D. (2022). Self-adaptive differential evolution based optimized MIMO beamforming 5G Networks. Iraqi Journal of Science, 63(8): 3628-3639. http://dx.doi.org/10.24996/ijs.2022.63.8.37

[16] Salih, T.A., Ali, A.J., Ahmed, M.N. (2020). Deep learning convolution neural network to detect and classify tomato plant leaf diseases. Open Access Library Journal, 7: e6296. https://doi.org/10.4236/oalib.1106296

[17] Hussein, N.A., Tawfiq, L.N.M. (2020). New approach for solving (1+1)-dimensional differential equation. Journal of Physics: Conference Series, 1530: 012098. https://doi.org/10.1088/1742-6596/1530/1/012098

[18] Tawfiq, L.N.M., Altaie, H. (2020). Recent modification of Homotopy perturbation method for solving system of third order PDEs. Journal of Physics: Conference Series, 1530: 012073. https://doi.org/10.1088/1742-6596/1530/1/012073

[19] Jain, R., Nagrath, P. Kataria, G., Kaushik, V.S., Hemanth, D.J. (2020). Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning, Measurement, 165: 108046. https://doi.org/10.1016/j.measurement.2020.108046

[20] Jabbar, M.A., Radhi, A.M. (2022). Diagnosis of Malaria infected blood cell digital images using deep convolutional neural networks. Iraqi Journal of Science, 63(1): 380-396. https://doi.org/10.24996/ijs.2022.63.1.35

[21] Jamil, H.J., Albahri, M.R.A., Al-Noor, N.H., Al-Noor, T.H., Heydari, A.R., Rajan, A.K., Arnetz1, J., Arnetz, B., Tawfiq, L.N.M. (2020). Hookah smoking with health risk perception of different types of tobacco.

Journal of Physics: Conference Series, 1664: 012127. https://doi.org/10.1088/1742-6596/1664/1/012127
[22] Anggoro, D.A., Hajiati, S. (2023). Comparison of performance metrics level of restricted boltzmann machine and backpropagation algorithms in detecting diabetes mellitus disease. Iraqi Journal of Science, 64(2): 907-921. https://doi.org/10.24996/ijs.2023.64.2.35
[23] Sultan, R.N., Flayyih, W.N., Ali, H.M. (2023). Extending Wi-Fi direct single-group network to multi-group network based on Android smartphone. Iraqi Journal of Science, 64(1): 419-438. https://doi.org/10.24996/ijs.2023.64.1.38
[24] Tawfiq, L.N.M., Naoum, R.S. (2007). Density and approximation by using feed forward artificial neural networks. Ibn Al-Haitham Journal for Pure & Applied Sciences, 20(1): 67-81.
[25] Mahmood, S.S., Eidi, J.H., Jasem, J.A. (2022). Positive definiteness of symmetric rank 1 (H-Version) update for unconstrained optimization. Baghdad Science Journal, 19(2): 297-303.
[26] Thenmozhi, K., Reddy, U.S. (2019). Crop pest classification based on deep convolutional neural network and transfer learning. Computers and Electronics in Agriculture, 164: 104906. https://doi.org/10.1016/j.compag.2019.104906
[27] Al-Sarray, B. (2022). Deep learning and machine learning via a genetic algorithm to classify breast cancer DNA data. Iraqi Journal of Science. 63(7): 3153-3168. https://doi.org/10.24996/ijs.2022.63.7.3