Solving Ordinary and Partial Differential Equation Using Legendre Neural Network DDY2 Method

Ahmat Saiful Latif, Sumardi*, Imam Solekhudin, Ari Suparwanto

***Abstract*—In this paper, we present a new method based on modification on single layer Legendre Neural Network(LeNN) method to be Legendre Neural Network DDY2 (LeNNDDY2) method to solve ordinary and partial differential equation. The activation function on the hidden layer is changed by Legendre polynomial expansion. The optimization method used in weight and bias updates is the DDY2 conjugate gradient method. For example problems, the numerical results have been compared with the other methods and gotten better results.**

***Index Terms*—differential equation, artificial neural network, Legendre polynomial, DDY2 conjugate gradient**

I. INTRODUCTION

DIFFERENTIAL equations are used as a powerful tool in solving many problems in various fields of human knowledge, such as physics, computer science, biology, chemistry, mechanics, economics, etc. The real problem is modeled into differential equations, then, by solving the differential equation the answer is described. Usually, many of these problems of the differential equation do not have analytical solutions. thus, we need a method to approximate the solution.

In this modern era, many researchers especially in the numerical field, use a deep learning method to solve certain problems like predict data or classify data in the form of an approximation of some data. In this case, a function that outputs will be formed expected by the existing problem-solving. The famous deep learning is Artificial Neural Network (ANN). Because the output of the ANN method is also a function to approximate data then the ANN method can be used to solve a first order partial differential equation. In previous research, the ANN method was able to solve a ordinary differential equation problems quite well. The output of the ANN method in the form of an approximation, the function is an excess of the ANN method compared another method that only has the output of the point-by-point approximation point.

A method to solve the first order ordinary differential equation using Hopfield neural network models was introduced by Lee and Kang[1]. Meade and Fernandez[2] solved linear and nonlinear ordinary differential equations using feed forward neural network architecture and B-splines of degree one. Lagaris et al.[3] used multi layer perceptron in their network to solve both ordinary and partial differential equation. In comparison with Jamme and Liu[4], Malek and Shekari[5] presented the potential of the hybrid and optimization technique to deal with differential equation of lower order as well as higher order. Recently, the application of Legendre neural network for solving differential equation is presented by Mall and Cakraverty[6].

In the next section, we introduce the Legendre neural network DDY2 method. In section 3, the proposed method for solving first and second order differential equation is introduced. The examples of ordinary differential equation are initial value problem, boundary value problem and system of the first ordinary differential equations. The examples of partial differential equation are advection, Laplace and Poisson equation. In section 4, we presented numerical examples. Finally, the conclusion is outlined in Section 5.

Manuscript received October 20, 2020; revised January 31, 2021.This work was supported by Research Grant PDUPT contract number 2833/UN1.DITLIT/D IT-LIT/PT/2020 provided by the Directorate General of the Higher Education of the Republic of Indonesia.

Ahmat Saiful Latif is a Master student at Department of Mathematics, Universitas Gadjah Mada, Yogyakarta, INDONESIA. E-mail:ahmat.saiful.latif@mail.ugm.ac.id

*Corresponding author, Sumardi is a lecturer at Department of Mathematics, Universitas Gadjah Mada, Yogyakarta, INDONESIA. E-mail:mardimath@ugm.ac.id

I. Solekhudin is a lecturer at Department of Mathematics, Universitas Gadjah Mada, Yogyakarta, INDONESIA. E-mail:imams@ugm.ac.id

Ari Suparwanto is a lecturer at Department of Mathematics, Universitas Gadjah Mada, Yogyakarta, INDONESIA. E-mail:ari_suparwanto@ugm.ac.id

II. LEGENDRE NEURAL NETWORK DDY2 METHOD

In this section, we have introduced a structure and learning algorithm that has been used in LeNNDDY2 method.

A. Structure of Legendre Neural Network DDY2 method

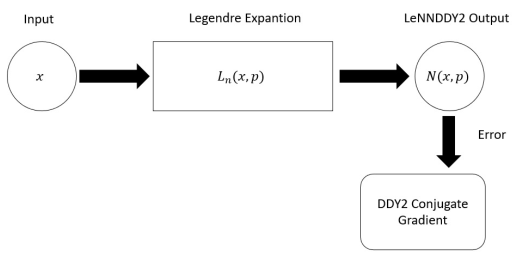

Fig. 1. Structure of Legendre Neural Network DDY2

The structure of LeNNDDY2 model can be seen in Fig.1. These structures consist of an input node, output node and the hidden layer is transform to Legendre polynomials expansion. Defined $L_n(q)$ is Legendre polynomial, with n is the order and $q \in [-1, 1]$.

The first few Legendre polynomials are [7]

$$\begin{aligned} L_0(q) &= 1, \\ L_1(q) &= u \\ L_2(q) &= \frac{1}{2}(3q^2 - 1). \end{aligned}$$

The higher order Legendre polynomials may be generated by the following recursive formula

$$L_{n+1}(q) = \frac{1}{n+1}\left[(2n+1)uL_n(q) - nL_{n-1}(q)\right]. \quad (1)$$

As input data we consider a vector $x = (x_1, x_2, ..., x_r)$ of dimension r. The enhanced pattern is obtained by using the Legendre polynomials

$$\begin{bmatrix} 1, L_1(w_{11}x_1 + u_{11}), ..., L_n(w_{n1}x_1 + u_{n1}) \\ 1, L_1(w_{12}x_2 + u_{12}), ..., L_n(w_{n2}x_2 + u_{n2}) \\ . \\ . \\ . \\ 1, L_1(w_{1h}x_h + u_{1h}), ..., L_n(w_{nh}x_h + u_{nr}) \end{bmatrix}$$

.

B. Learning algorithm of Legendre Neural Network DDY2

Error backpropagation learning algorithm is used for updating the network parameters (weights and bias) of Legendre Neural Network DDY2 (LeNNDDY2). As such, the gradient of an error function concerning the network parameter is determined. The linear function is considered an activation function. The gradient conjugate DDY2 [8] is used for learning to minimize the error function. The weights and bias are initialized randomly and then the weights and bias are updated as follows

$$W_{k+1} = W_k + \alpha_k d_k, \quad (2)$$

where W is the parameters(weights and bias), α_k is the step length, and d_k is the search direction defined by

$$d_k = \begin{cases} -g_k, k = 1, \\ -g_k + \beta_k d_{k-1}, k \geq 2, \end{cases} \quad (3)$$

where $g_k = \nabla E_p(W)$ is the gradient of E_p at W_k, and β_k is the parameter defined by

$$\beta_k = \begin{cases} \frac{g_k^T\left(g_k - \frac{g_k^T d_{k-1}}{\|d_{k-1}\|^2} d_{k-1}\right)}{d_{k-1}^T(g_k - g_{k-1}) + \mu g_k^T d_{k-1}}, & g_k^T d_{k-1} \geq 0 \\ 0, & other \end{cases}. \quad (4)$$

The value of α_k can be determined using Secant method.

III. Proposed the Method for Ordinary and Partial Differential Equation

In this section, we describe the formulation of first and second-order partial differential equations, especially on advection, two dimensional Laplace equation and Poisson Equation problem. The formulation in general form of the differential equation (which represents ordinary as well as partial differential equations) may be written as [2]

$$G(x, y(x), \nabla y(x), \nabla^2 y(x), ..., \nabla^k y(x)) = 0, x \in \bar{D} \subseteq R^n \quad (5)$$

subject to some initial or boundary conditions, where $y(x)$ is the solution, G is the function that defines the structure of the differential equation and ∇ is a differential operator. $\bar{D}$ donates the discretized domain over a finite set of points in R^n.

Let $y_t(x, p)$ denotes the trial solution with adjustable parameters(weights)p, thus the problem is transformed into the following minimization problem

$$\underset{p}{Min} \sum_{x_n \in \bar{D}} \left(G(x_n, y_t(x_n, p), \nabla y_t(x_n, p), ..., \nabla^k y_t(x_n, p)\right)^2. \quad (6)$$

The trial solution $y_t(x, p)$ satisfies the initial or boundary conditions and be written as the sum of two terms:

$$y_t(x, p) = B(x) + F(x, N(x, p)) \quad (7)$$

where $A(x)$ satisfies initial boundary conditions and contains no adjustable parameters. $N(x, p)$ is the output of feed forward neural network with parameters p and input x. The second term $F(x, N(x, p))$ does not contribute to initial or boundary conditions but this is used to a neural network whose weights are adjustable to minimize the error function.

The formulations of the first and second order differential equations are the following.

A. First order ODE

The firts order ODE may be represent as

$$\frac{dy}{dx} = f(x, y), x \in [a, b] \quad (8)$$

subject to $y(a) = A$.

The LeNNDDY2 trial solution is

$$y_t(x, p) = A + (x - a)N(x, p) \quad (9)$$

where $N(x, p)$ is output of LeNNDDY2 model defined by

$$N(x, p) = \sum_{j=0}^{n} o_j v_j, \quad (10)$$

and

$$o_j = L_j(w_{j+1}x + u_{j+1}), j = 0, 1, ..., n. \quad (11)$$

The error function is written as

$$E_p = \sum_{i=1}^{m} \left(\frac{dy_t(x_i, p)}{dx_i} - f\left(x_i, y_t(x_i, p)\right)\right)^2. \quad (12)$$

Derivative of $y_t(x, p)$ with respect to x is given as

$$\frac{dy_t(x, p)}{dx} = N(x, p) + (x - a)\frac{dN(x, p)}{dx}. \quad (13)$$

B. Second order ODE

Let us consider second order initial value problem as

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right), x \in [a, b] \quad (14)$$

with initial condition $y(a) = A, y'(a) = A'$.

The LeNNDDY2 trial solution is

$$y_t(x, p) = A_0 + A_1(x - a) + (x - a)^2 N(x, p). \quad (15)$$

where $N(x, p)$ is output of LeNNDDY2 model defined by Eq. and Eq.

The error function is written as

$$E_p = \sum_{i=1}^{m} \left(\frac{d^2 y_t(x_i, p)}{dx_i{}^2} - f\left(x_i, y_t(x_i, p), \frac{dy_t(x_i, p)}{dx_i}\right)\right)^2. \quad (16)$$

From Eq. we get (by differentiating)

$$\frac{dy_t(x,p)}{dx} = A_1 + 2(x-a)N(x,p) + (x-a)^2\frac{dN}{dx} \quad (17)$$

$$\frac{d^2y_t(x,p)}{dx^2} = 2N(x,p)+4(x-a)\frac{dN}{dx}+(x-a)^2\frac{d^2N}{dx^2}. \quad (18)$$

Next, a second order boundary value problem may be written as

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right), x \in [a,b] \quad (19)$$

with boundary condition $y(a) = A, y(b) = B$.

Corresponding LeNNDDY2 trial solution for the above boundary value problem is

$$y_t(x,p) = \frac{bA-aB}{b-a} + \frac{B-A}{b-a}x + (x-a)(x-b)N(x,p) \quad (20)$$

As such, the error function may be obtained as

$$E_p = \sum_{i=1}^{m}\left(\frac{d^2y_t(x_i,p)}{dx_i{}^2} - f\left(x_i, y_t(x_i,p), \frac{dy_t(x_i,p)}{dx_i}\right)\right)^2. \quad (21)$$

here

$$\frac{dy_t}{dx} = \frac{B-A}{b-a} + (2x-a-b)N + (x-a)(x-b)\frac{dN}{dx}. \quad (22)$$

C. System of first ODEs

Here we consider the following system of initial value first order differensial equations

$$\frac{dy_r}{dx} = f_r(x, y_1, ..., y_l), x \in [a,b], \quad (23)$$

subject to $y_r(a) = A_r, r = 1, 2, ..., l$.

Corresponding trial solution has the following form

$$y_{tr}(x, p_r) = A_r + (x-a)N_r(x, p_r), \quad (24)$$

for every $r = 1, 2, ..., l$.

For each r, $N_r(x, p_r)$ is the output of the Legendre Neural Network DDY2 with parameter x and parameter p_r difined by

$$N_r(x, p_r) = \sum_{j=0}^{n} o_{jr}v_{(j+1)r}. \quad (25)$$

and

$$o_{jr} = L_j(w_{(j+1)r}x_i + u_{(j+1)r}) \quad (26)$$

where $j = 0, 1, ..., n$ and $r = 1, 2, .., l$.

Then the corresponding error function with adjustable network parameters may be written as

$$E_p = \sum_{i=1}^{m}\sum_{r=1}^{l}\left(\frac{dy_{tr}(x_i,p_r)}{dx_i} - f\left(x_i, y_{t1}(x_i,p_1), ..., y_{tl}(x_i,p_l)\right)\right)^2 \quad (27)$$

From the Eq. we have

$$\frac{dy_{tr}(x,p_r)}{dx} = N_r(x,p_r) + (x-a)\frac{dN_r(x,p_r)}{dx} \quad (28)$$

for each $r = 1, 2, ..., l$.

D. First order partial differential equation

Let us consider first order partial differential equation

$$\frac{\partial}{\partial x}\psi(x,y) + \frac{\partial}{\partial y}\psi(x,y) = 0, \quad (29)$$

subject to $\psi(0,y) = f_0(y), \psi(1,y) = f_1(y)$, $\psi(x,0) = g_0(x)$ and $\psi(x,1) = g_1(x)$, where $x \in [0,1], y \in [0,1]$. Corresponding trial solution has the following form

$$\psi_t(x,y) = B(x,y) + x(x-1)y(y-1)N(x,y,p), \quad (30)$$

where $B(x,y)$ is chosen so as to satisfy the boundary condition and $N(x,y,p)$ is output of LeNNDDY2 model defined by

$$N(x,p) = \sum_{j=0}^{n} o_j v_{j+1}. \quad (31)$$

and

$$o_j = L_j(w_{(j+1)k}x_i + w_{(j+1)k}y_i + u_{j+1}) \quad (32)$$

for each $j = 0, 1, ..., n$. Then the error function to be minimized is given by

$$E_p = \sum_{i=1}^{m}\left(\frac{\partial}{\partial x}\psi_t(x_i,y_i) + \frac{\partial}{\partial y}\psi_t(x_i,y_i)\right)^2. \quad (33)$$

E. Laplace equation

Let us consider two dimensional Laplace equation as bellow

$$\frac{\partial^2}{\partial x^2}\psi(x,y) + \frac{\partial^2}{\partial y^2}\psi(x,y) = 0, \quad (34)$$

subject to $\psi(0,y) = f_0(y), \psi(1,y) = f_1(y)$, $\psi(x,0) = g_0(x)$ and $\psi(x,1) = g_1(x)$, where $x \in [0,1], y \in [0,1]$. Corresponding trial solution has the following form

$$\psi_t(x,y) = B(x,y) + x(x-1)y(y-1)N(x,y,p), \quad (35)$$

where $B(x,y)$ is chosen so as to satisfy the boundary condition and $N(x,y,p)$ is output of LeNNDDY2 model defined by

$$N(x,p) = \sum_{j=0}^{n} o_j v_{j+1}. \quad (36)$$

and

$$o_j = L_j(w_{(j+1)k}x_i + w_{(j+1)k}y_i + u_{j+1}) \quad (37)$$

for each $j = 0, 1, ..., n$. Then the error function to be minimized is given by

$$E_p = \sum_{i=1}^{m}\left(\frac{\partial^2}{\partial x^2}\psi_t(x_i,y_i) + \frac{\partial^2}{\partial y^2}\psi_t(x_i,y_i)\right)^2. \quad (38)$$

F. Poisson equation

Here we consider the following two dimensional Poisson equation

$$\frac{\partial^2}{\partial x^2}\psi(x,y) + \frac{\partial^2}{\partial y^2}\psi(x,y) = f(x,y), \quad (39)$$

subject to $\psi(0,y) = f_0(y), \psi(1,y) = f_1(y)$, $\psi(x,0) = g_0(x)$ and $\psi(x,1) = g_1(x)$, where $x \in [0,1], y \in [0,1]$.

Corresponding trial solution has the following form

$$\psi_t(x,y) = B(x,y) + x(x-1)y(y-1)N(x,y,p), \quad (40)$$

TABLE I
COMPARISON AMONG ANALYTICAL, LENNDDY2 AND LENN RESULTS(EXAMPLE 1)

input	analitik	LeNN	LeNN DDY2
0	1	1	1
0.1	0.87397	0.89900	0.88484
0.2	0.78904	0.81792	0.79985
0.3	0.73861	0.76030	0.74506
0.4	0.71924	0.72933	0.72049
0.5	0.72926	0.72779	0.72620
0.6	0.76794	0.75792	0.76221
0.7	0.83493	0.82140	0.82854
0.8	0.92993	0.91926	0.92524
0.9	1.05254	1.05180	1.05234
1.0	1.20218	1.21852	1.20986

where $B(x, y)$ is chosen so as to satisfy the boundary condition and $N(x, y, p)$ is output of LeNNDDY2 model defined by

$$N(x, p) = \sum_{j=0}^{n} o_j v_{j+1}. \quad (41)$$

and

$$o_j = L_j(w_{(j+1)k}x_i + w_{(j+1)k}y_i + u_{j+1}) \quad (42)$$

for each $j = 0, 1, ..., n$.

Then the error function tp be minimized is given by

$$E_p = \sum_{i=1}^{m} \left(\frac{\partial^2}{\partial x^2}\psi_t(x_i, y_i) + \frac{\partial^2}{\partial y^2}\psi_t(x_i, y_i) - f(x_i, y_i) \right)^2. \quad (43)$$

IV. NUMERICAL EXAMPLE

In this section, we consider various example, such us a initial value problem, a boundary value problem, a system of coupled first order ordinary differential equation, a first order partial differential equation and two dimensional Laplace and Poisson equation problems. The example problems are solved and computed with MATLAB.

Example 1. Let us consider the first order ordinary differential equation as follows:

$$\frac{dy}{dx} + \left(x + \frac{1 + 3x^2}{1 + x + x^3} \right) y = x^3 + 2x + \frac{x^2 + 3x^4}{1 + x + x^3}$$

with initial conditions $y(0) = 0$ and $x \in [0, 1]$. The exact solution is

$$y(x) = \frac{e^{-x^2/2}}{1 + x + x^3} + x^2$$

Following the procedure of the present method, we write the LeNNDDY2 trial solution

$$y_t(x) = 1 + xN(x, p)$$

The network was trained using grid of ten equidistance points in $[0, 1]$. Five wight and five bias in the first five Legendre polynomials expansion also five wight between Legendre expantion and output layer are considered. Comparison among analytical, Legendre Neural Network DDY2(LeNNDDY2) and Legendre Neural Network(LeNN) results has been shown in Table.I These comparison are also depicted in Fig.2. Plot comparison between LeNNDDY2 error (analytical and LeNNDDY2) and LeNN error (analytical and LeNN) is cited in Fig.3.

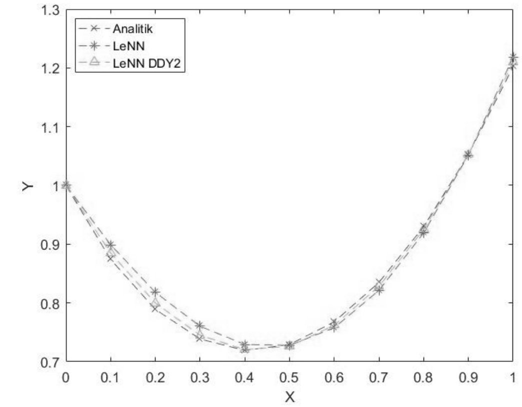

Fig. 2. Plot of Analytical, LeNN and LeNNDDY2 results (Example 1)

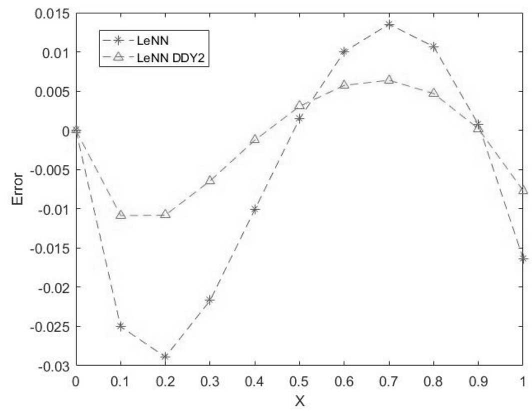

Fig. 3. Error plot between LeNN and Error LeNNDDY2 (Example 1)

Example 2. Let us consider the second order ordinary differential equation problem as follow:

$$\frac{d^2y}{dx^2} + \frac{2}{x}\frac{dy}{dx} + 4(2e^y + e^{y/2}) = 0$$

with initial condition $y(0) = 0, y'(0) = 0$ and $x \in [0, 1]$. The exact solution is

$$y = -2\ln(1 + x^2)$$

and the related LeNNDDY2 trial solution is written as

$$y_t(x) = x^2N(x, p).$$

We train the network for ten equidistant point in the domain $[0, 1]$ with first five Legendre polynomials expansion. Table II shows comparison among analytical, LeNNDDY2 and LeNN result. The comparison among analytical, LeNNDDY2 and LeNN results are also depicted in Fig.4. Fig.5 shows comparison between LeNNDDY2 error and LeNN error.

Example 3. Here, we consider a second order boundary value problem as follow:

$$\frac{d^2y}{dx^2} + \frac{1}{5}\frac{dy}{dx} + y = \frac{1}{5}e^{-x/5}\cos(x)$$

with boundary condition $y(0) = 0$ and $y(1) = \sin(1)e^{-0.2}$.

The exact solution of the problem is

$$y(x) = e^{-x/5}\sin(x)$$

TABLE II
COMPARISON AMONG ANALYTICAL, LENNDDY2 AND LENN RESULTS(EXAMPLE 2)

input	analitik	LeNN	LeNN DDY2
0	0	0	0
0.1	-0.01990	-0.01990	-0.02001
0.2	-0.07844	-0.07822	-0.07861
0.3	-0.17236	-0.17172	-0.17239
0.4	-0.29684	-0.29591	-0.29670
0.5	-0.44629	-0.44547	-0.44614
0.6	-0.61497	-0.61473	-0.61499
0.7	-0.79755	-0.79810	-0.2977
0.8	-0.98939	-0.99045	-0.98947
0.9	-1.18665	-1.18759	-1.18642
1.0	-1.38629	-1.38667	-1.38637

TABLE III
COMPARISON AMONG ANALYTICAL, LENNDDY2 AND LENN RESULTS(EXAMPLE 3)

input	analitik	LeNN	LeNN DDY2
0	0	0	0
0.1	0.09786	0.09828	0.09784
0.2	0.19088	0.19081	0.19085
0.3	0.27831	0.27753	0.27828
0.4	0.35948	0.35829	0.35946
0.5	0.43380	0.43280	0.43380
0.6	0.50079	0.50060	0.50081
0.7	0.56006	0.56107	0.56008
0.8	0.61129	0.61337	0.61131
0.9	0.65429	0.65644	0.65430
1.0	0.68894	0.68894	0.68894

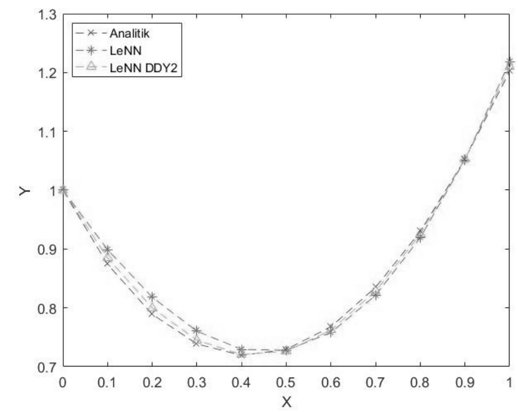

Fig. 4. Plot of Analytical, LeNN and LeNNDDY2 results (Example 2)

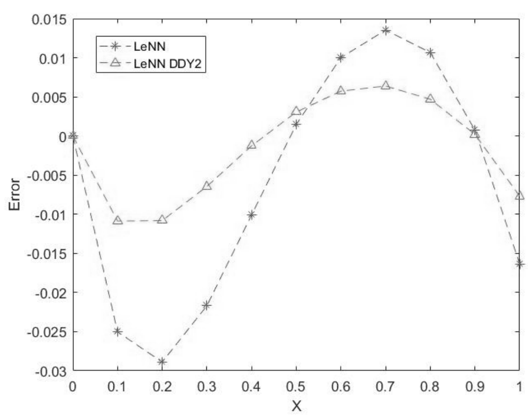

Fig. 5. Error plot between LeNN and LeNNDDY2 (Example 2)

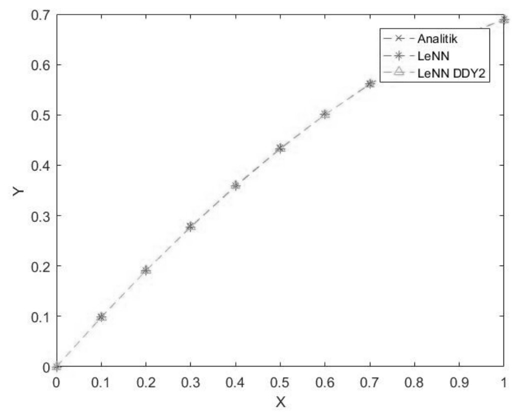

Fig. 6. Plot of Analytical, LeNN and LeNNDDY2 results (Example 3)

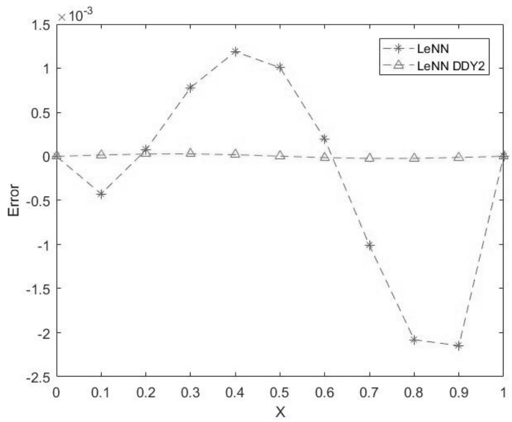

Fig. 7. Error plot between LeNN and LeNNDDY2 (Example 3)

and the trial solution is written as

$$y_t(x) = x^2 N(x,p)$$

Now, the network is trained for ten equidistant point in the domain $[0,1]$ with dive Legendre polynomial expansion. Comparison among analytical, LeNNDDY2 and LeNN result are given in Table III. Fig.6 shows comparison between analytical, LeNNDDY2 and LeNN. Fig.7 shows comparison between LeNNDDY2 error (analytical and LeNNDDY2) and LeNN error(analytical and LeNN).

Example 4. Next, we take a system of coupled first order ordinary differential equation

$$\frac{dy_1}{dx} = \cos(x) + y_1^2 + y_2 - (1 + x^2 + \sin^2(x))$$

$$\frac{dy_2}{dx} = 2x - (1 + x^2)\sin(x) + y_1 y_2$$

with initial condition $y_1(0) = 0, y_2(0) = 1$ and $x \in [0,1]$.

Corresponding exact solution are

$$\begin{aligned} y_1(x) &= \sin(x) \\ y_2(x) &= 1 + x^2 \end{aligned}$$

In the case, the LeNNDDY2 trial solution are

$$\begin{aligned} y_{t_1}(x) &= xN_1(x,p_1) \\ y_{t_2}(x) &= 1 + xN_2(x,p_2) \end{aligned}$$

We consider ten equidistant points in $[0,1]$ and the result are compared between analytical, LeNNDDY2 and LeNN results. Comparison among analytical, LeNNDDY2

TABLE IV
COMPARISON AMONG ANALYTICAL, LENNDDY2 AND LENN y_1 RESULTS(EXAMPLE 4)

input	analitik	LeNN	LeNN DDY2
0	0	0	0
0.1	0.09983	0.09938	0.09912
0.2	0.19867	0.19858	0.19812
0.3	0.29552	0.29604	0.29551
0.4	0.38942	0.39034	0.38992
0.5	0.47943	0.48024	0.48014
0.6	0.56464	0.56481	0.56513
0.7	0.64422	0.64341	0.64413
0.8	0.71736	0.71581	0.71670
0.9	0.78333	0.78222	0.78275
1	0.84147	0.84337	0.84260

TABLE V
COMPARISON AMONG ANALYTICAL, LENNDDY2 AND LENN y_2 RESULTS(EXAMPLE 4)

input	analitik	LeNN	LeNN DDY2
1	1	1	1
0.1	1.10100	1.01421	1.01343
0.2	1.04000	1.04374	1.04279
0.3	1.09000	1.09137	1.09062
0.4	1.16000	1.15901	1.15867
0.5	1.25000	1.24784	1.24796
0.6	1.36000	1.35823	1.35876
0.7	1.49000	1.48987	1.49062
0.8	1.64000	1.64176	1.64240
0.9	1.81000	1.81227	1.81223
1	2	1.9992	1.99762

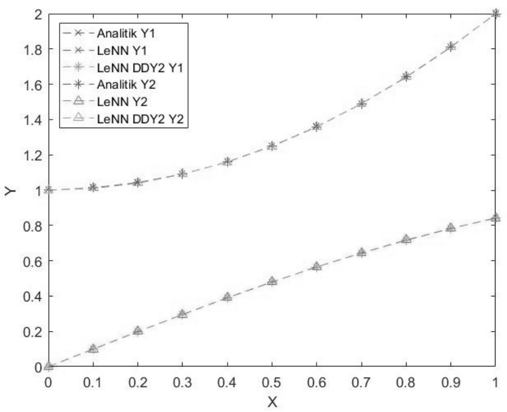

Fig. 8. Plot of Analitical, LeNN and LeNNDDY2 results (Example 4)

and LeNN results are given in Table IV and Table V also depicted in Fig.8. Fig.9 and Fig.10 shows comparison between LeNNDDY2 error and LeNN error.

Example 5. Consider the first order partial differential equation problem

$$\nabla\psi(x,y) = 0, x, y \in [0,1]$$

with initial and boundary condition

$$\begin{aligned} \psi(x,0) &= \sin x, \\ \psi(0,y) &= -\sin y. \end{aligned}$$

The analytic solution of the problem is

$$\psi(x,y) = \sin(x-y).$$

Consider the initial and boundary condition, the trial solution was constructed as

$$\psi_t(x,y) = x^y \sin(x) - y^x \sin(y) + xyN(x,y,p).$$

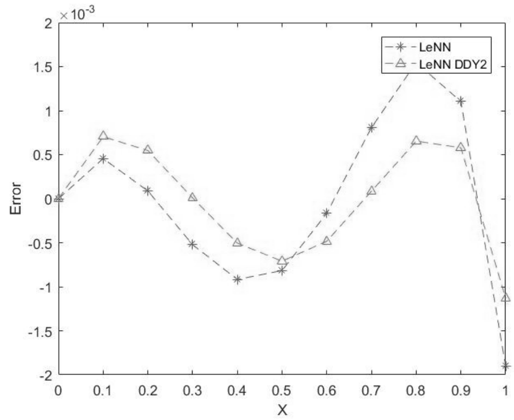

Fig. 9. Error plot between LeNN and LeNNDDY2 (Example 4)

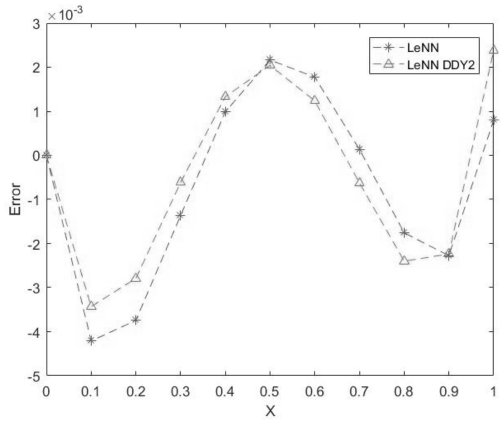

Fig. 10. Error plot between LeNN and LeNNDDY2 (Example 4)

TABLE VI
ANALYTICAL RESULTS(EXAMPLE 5)

	x					
y	0	0.2	0.4	0.6	0.8	1
0	0	0.1987	0.3894	0.5646	0.7174	0.8415
0.2	-0.1987	0	0.1987	0.3894	0.5646	0.7174
0.4	-0.3894	-0.1987	0	0.1987	0.3894	0.5646
0.6	-0.5646	-0.3894	-0.1987	0	0.1987	0.3894
0.8	-0.7174	-0.5646	-0.3894	-0.1987	0	0.1987
1	-0.8415	-0.7174	-0.5646	-0.3894	-0.1987	0

TABLE VII
LENN RESULTS(EXAMPLE 5)

	x					
y	0	0.2	0.4	0.6	0.8	1
0	0	0.1987	0.3894	0.5646	0.7174	0.8415
0.2	-0.19871	0.0042	0.2266	0.4401	0.6327	0.7970
0.4	-0.3894	-0.2133	0.0098	0.2419	0.4645	0.6675
0.6	-0.5646	-0.4268	-0.2259	0.0020	0.2356	0.4635
0.8	-0.7174	-0.6205	-0.4544	-0.2470	-0.0185	0.2204
1	-0.8415	-0.7764	-0.6429	-0.4587	-0.2408	-0.0014

The network is trained here for eight equidistant points in given domain. Result of analytical, LeNN and LeNNDDY2 shown at Table VI, Table VII and Table VIII also depicted at Fig.11 , Fig.12 and Fig.13. Error of LeNN method and LeNNDDY2 method is cited in Fig.14 and Fig.15.

Example 6. Consider a two dimensional Laplace equation problem

$$\nabla^2\psi(x,y) = 0, x, y \in [0,1]$$

TABLE VIII
LeNNDDY2 results(Example 5)

	x					
y	0	0.2	0.4	0.6	0.8	1
0	0	0.1987	0.3894	0.5646	0.7174	0.8415
0.2	-0.1987	0.0009	0.2029	0.3920	0.5627	0.7155
0.4	-0.3894	-0.2047	-0.0013	0.1977	0.3886	0.5649
0.6	-0.5646	-0.3893	-0.1937	0.0052	0.2308	0.3992
0.8	-0.7174	-0.5656	-0.3889	-0.1965	-0.0024	0.1844
1	-0.8415	-0.7137	-0.5665	-0.3920	-0.1958	-0.0001

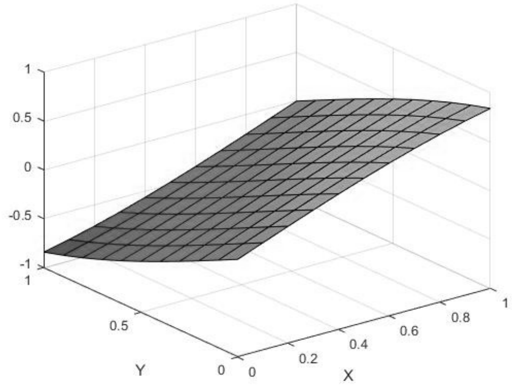

Fig. 11. Plot of Analytical result (Example 5)

with boundary condition

$$\begin{aligned}\psi(x,y) &= 0,\ \forall x \in \{(x,y) \in D | x = 0, x = 1, y = 0\}\\ \psi(x,y) &= \sin \pi x,\ \forall x \in \{(x,y) \in D | y = 1\}\end{aligned}$$

where $D = [0,1] \times [0,1]$. The analytical solution is

$$\psi(x,y) = \frac{1}{e^{\pi} - e^{-\pi}} \sin \pi x \left(e^{\pi y} - e^{-\pi y}\right).$$

Using the boundary condition, the trial solution was constructed as

$$\psi_t(x,y) = y \sin \pi x + x(x-1)y(y-1)N(x,y,p).$$

The network is trained here for ten equidistant points in given domain. Table IX, Table X and Table XI shown the analytical, LeNN and LeNNDDY2 result. Fig.16, Fig.17 and Fig.18 are depicted are analytical, LeNN and LeNNDDY2 results. Error between analytical results and LeNN result

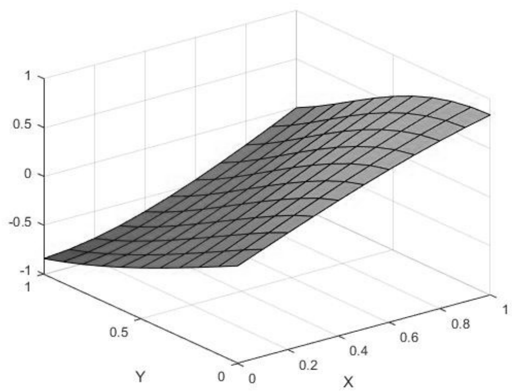

Fig. 12. Plot of LeNN result (Example 5)

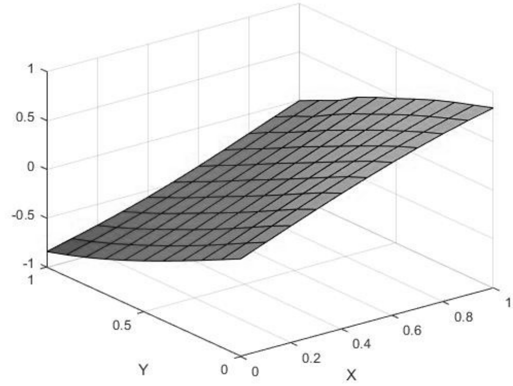

Fig. 13. Plot of LeNNDDY2 result (Example 5)

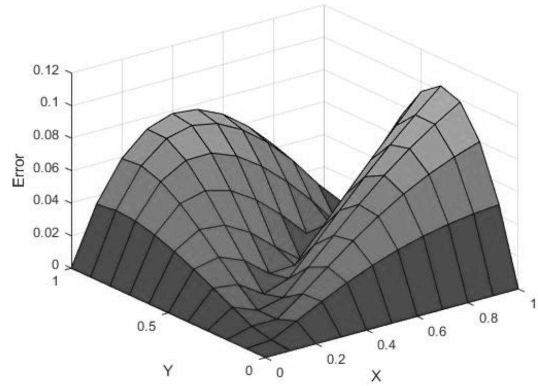

Fig. 14. Error plot of LeNN result (Example 5)

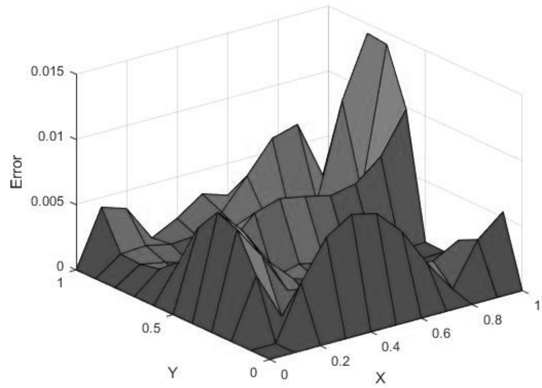

Fig. 15. Error plot of LeNNDDY2 result (Example 5)

are cited in Fig.19. Fig.20 cited plot of the error between analytical result and LeNNDDY2 result.

TABLE IX
Analytical results(Example 6)

	x					
y	0	0.2	0.4	0.6	0.8	1
0	0	0	0	0	0	0
0.2	0	0.0341	0.0552	0.0552	0.0341	0
0.4	0	0.0822	0.1330	0.1330	0.0822	0
0.6	0	0.1637	0.2649	0.2649	0.1637	0
0.8	0	0.3121	0.5050	0.5050	0.3121	0
1	0	0	0.5878	0.9511	0.9511	0

TABLE X
LeNN results(Example 6)

	x					
y	0	0.2	0.4	0.6	0.8	1
0	0	0	0	0	0	0
0.2	0	0.0338	0.0548	0.0548	0.0338	0
0.4	0	0.0819	0.1327	0.1327	0.0819	0
0.6	0	0.1638	0.2652	0.2652	0.1638	0
0.8	0	0.3124	0.5057	0.5057	0.3124	0
1	0	0	0.5878	0.9511	0.9511	0

TABLE XI
LeNNDDY2 results(Example 6)

	x					
y	0	0.2	0.4	0.6	0.8	1
0	0	0	0	0	0	0
0.2	0	0.0341	0.0553	0.0553	0.0341	0
0.4	0	0.0822	0.1332	0.1332	0.0822	0
0.6	0	0.1636	0.2649	0.2649	0.1636	0
0.8	0	0.3121	0.5051	0.5051	0.3121	0
1	0	0	0.5878	0.9511	0.9511	0

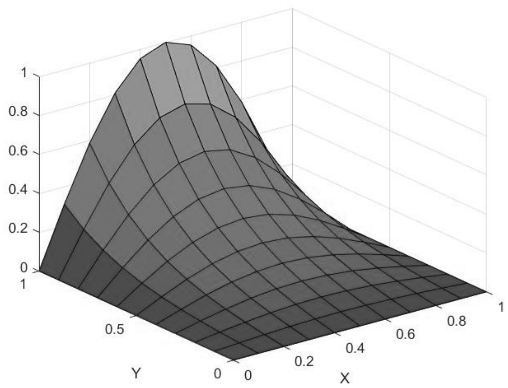

Fig. 16. Plot of Analytical result (Example 6)

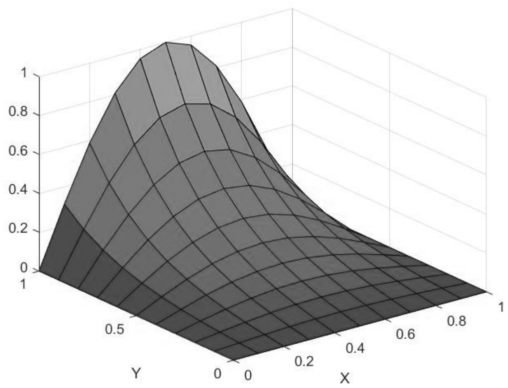

Fig. 17. Plot of LeNN result (Example 6)

Example 7. Consider a two dimensional Poisson equation problem

$$\nabla^2\psi(x,y) = e^{-x}(x-2+y^3+6y), x,y \in [0,1]$$

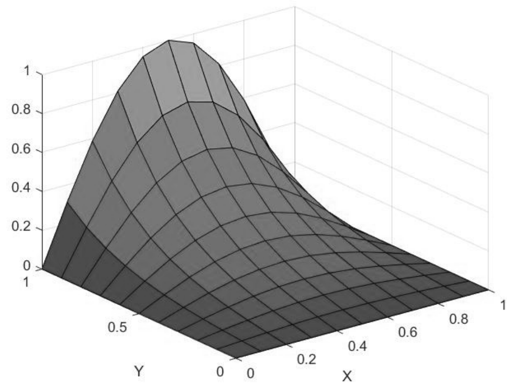

Fig. 18. Plot of LeNNDDY2 result (Example 6)

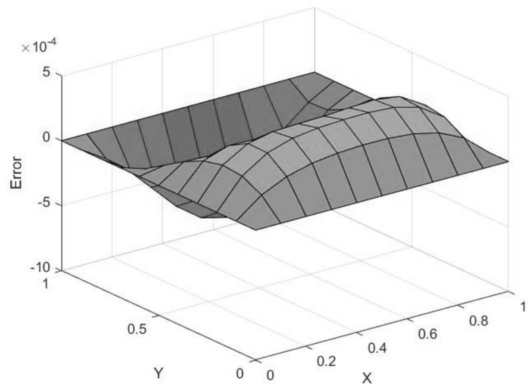

Fig. 19. Error plot of LeNN result (Example 6)

with boundary condition

$$\begin{aligned} \psi(0,y) &= y^3, \\ \psi(1,y) &= (1+y^3)e^{-1}, \\ \psi(x,0) &= xe^{-x}, \\ \psi(x,1) &= e^{-x}(x+1). \end{aligned}$$

Analytical solutions for the problem may be obtained as

$$\psi(x,y) = e^{-x}(x+y^3).$$

Based on the procedure in the previous section, we write

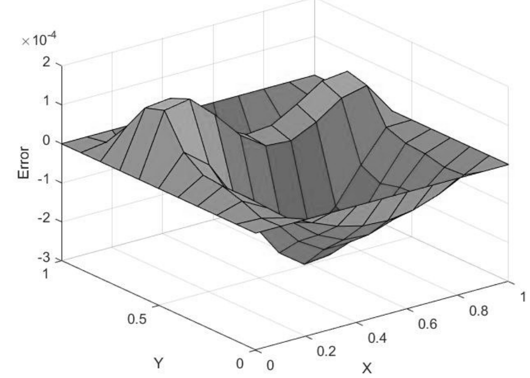

Fig. 20. Error plot of LeNNDDY2 result (Example 5)

the LeNNDDY2 trial solution for the problem is

$$\psi_t(x, y) = A(x, y) + x(1 - x)y(1 - y)N(x, y, p)$$

with the value of $A(x, y)$ as follow

$$\begin{aligned} A(x, y) &= (1 - x)y^3 + x(1 + y^3)e^{-1} \\ &\quad +(1 - y)x(e^{-x} - e^{-1}) \\ &\quad y[(1 + x)e^{-x} - (1 - x - 2xe^{-1})]. \end{aligned}$$

The network is trained here for ten equidistant points in given domain. The analytical, LeNNDDY2 and LeNN are shown in Table XII, Table XIII and Table XIV. Comparison between analytical, LeNNDDY2 and LeNN results are depicted in Fig.21,Fig.22 and Fig.23. Plot of the error function is cited in Fig.24 and Fig.25.

TABLE XII
ANALYTICAL RESULTS(EXAMPLE 5)

	y					
x	0	0.2	0.4	0.6	0.8	1
0	0	0.1637	0.2681	0.3293	0.3595	0.3679
0.2	0.0080	0.1703	0.2735	0.3337	0.3631	0.3708
0.4	0.0640	0.2161	0.3110	0.3644	0.3882	0.3914
0.6	0.2160	0.3406	0.4129	0.4478	0.4565	0.4473
0.8	0.5120	0.5829	0.6113	0.6103	0.5895	0.5562
1	1	0.9825	0.9384	0.8781	0.8088	0.7358

TABLE XIII
LENN RESULTS(EXAMPLE 7)

	y					
x	0	0.2	0.4	0.6	0.8	1
0	0	0.1637	0.2681	0.3293	0.3595	0.3679
0.2	0.0080	0.1730	0.2779	0.3382	0.3666	0.3708
0.4	0.0640	0.2169	0.3129	0.3667	0.3907	0.3914
0.6	0.2160	0.3379	0.4099	0.4456	0.4561	0.4473
0.8	0.5120	0.5794	0.6068	0.6064	0.5878	0.5562
1	1	0.9825	0.9384	0.8781	0.8088	0.7358

TABLE XIV
LENNDDY2 RESULTS(EXAMPLE 7)

	y					
x	0	0.2	0.4	0.6	0.8	1
0	0	0.1637	0.2681	0.3293	0.3595	0.3679
0.2	0.0080	0.1704	0.2756	0.3374	0.3659	0.3708
0.4	0.0640	0.2142	0.3112	0.3673	0.3908	0.3914
0.6	0.2160	0.3369	0.4107	0.4485	0.4578	0.4473
0.8	0.5120	0.5797	0.6089	0.6100	0.5899	0.5562
1	1	0.9825	0.9384	0.8781	0.8088	0.7358

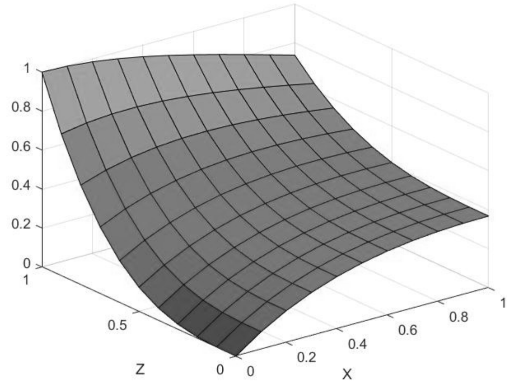

Fig. 21. Plot of Analytical result (Example 7)

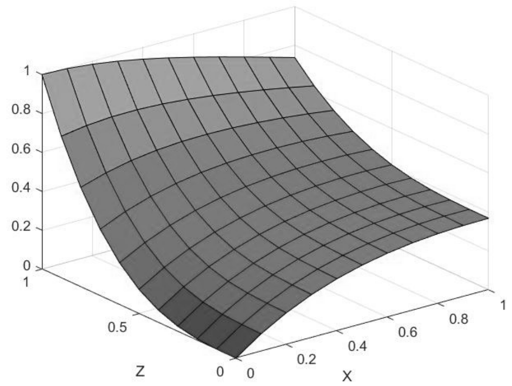

Fig. 22. Plot of LeNN result (Example 7)

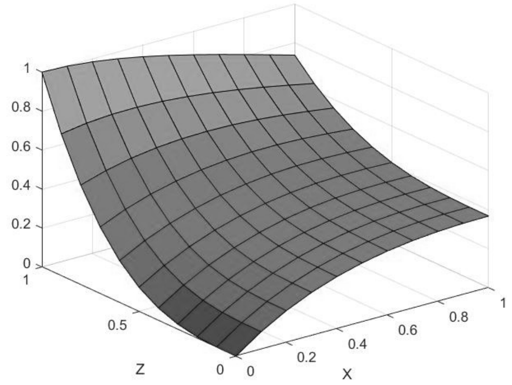

Fig. 23. Plot of LeNNDDY2 result (Example 7)

V. CONCLUSION

This paper presents a new approach to solve ordinary and partial differential equation. The examples of the PDE are advection, Laplace and Poisson equation. Here we have considered single layer artificial neural network architecture. In architecture, the hidden layer is replaced by Legendre polynomial expansion. A gradient conjugate DDY2 is used to minimize the error function of the backpropagation algorithm to update the network parameters(weights and bias). Based on some comparison examples in the previous section, the proposed method has a better result with a smaller error value.

REFERENCES

[1] H. Lee and L. Kang, "Neural Network Algoritms to Solving Differential Equations," in *Lecture Notes in J.Comput.Phys.91 1990*, pp. 110-117.

[2] A. J.Maede Jr. and A. A. Fernandez, "The Numerical Solution of Linear Ordinary Differential Equations by Feed Forward Neural Network," in *Lecture Notes in Math. Comput. Model 19 1994*, pp. 1-25.

[3] I. E. Lagaris., A. Likas, and D. I. Fotiadis "Artificial Neural Network for Solving Ordinary and Partial Differential Equations," in *Lecture Notes in IEEE Trans. Neural Netw. 9 1998*, pp. 987-1000.

[4] B. Liu. and B. James, "Solving Ordinary Differential Equations by Neural Network," in Proceeding of 13th European Simulation Multy-Conference Modelling and Simulation: A Tool for the Next Millenium, Warsaw, Poland, June 14 1999.

[5] A. Melek., and R. Shekari, "Numerical Solution for High Order

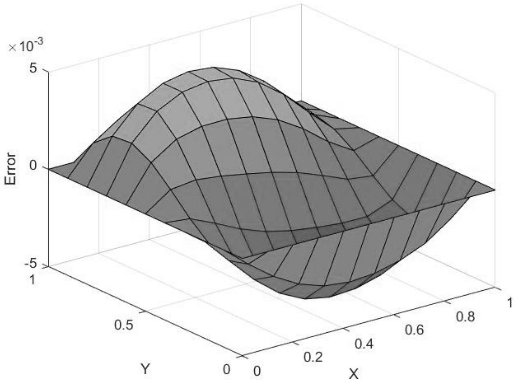

Fig. 24. Error plot of LeNN result (Example 7)

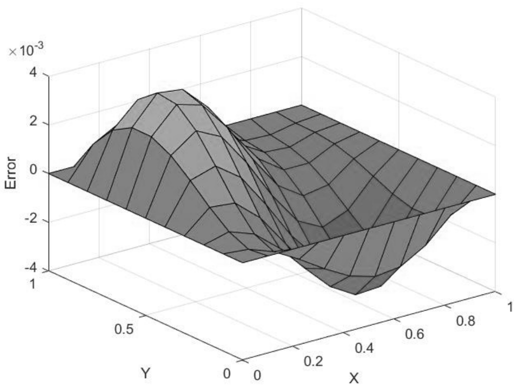

Fig. 25. Error plot of LeNNDDY2 result (Example 7)

Differential Equations using A Hybrid Neural Network Optimization Method," in *Lecture Notes in Applied Mathematics and Computation 183 2006*, pp. 260-271.

[6] S. Mall and S. Chakraverty, "Application of Legendre Neural Network for Solving Ordinary Differential Equation," in *Lecture Notes in Applied Soft Computing 43 2016*, pp. 347-356.

[7] J. C. Patra and C. Bornand, "Nonlinier Dynamic System Identification using Legendre Neural Network,"in IEEE Int. Joint Conf 2010.

[8] Z. Zhibin, Z. Dongdong and W. Shuo, "Two Modified DY Conjugate Gradient Methods for Unconstrained Optimization Problems,"in *Lecture Notes in Applied Mathematics and Computation 2020.*