

# APPLICATION OF ANN FOR SOLUTION OF PDE IN RF ENGINEERING

Sridhar Pattanaik<sup>1</sup>, Mishra .R.K<sup>2</sup>

<sup>1,2</sup>Department of Electronic Sciences, Berhampur University, Berhampur, India

E-mail : <sup>1</sup>sridhar\_pattanaik@yahoo.co.in , <sup>2</sup>faculty3@rediffmail.com

## Abstract

Artificial Neural Network (ANN) is used to solve the differential equation used in RF Engineering. Solution of Poisson's equation and wave equation are illustrated as example with different boundary conditions like pure Dirichlet, pure Neumann and mixed boundary condition. The formulation considers it as an optimization problem and therefore proposes solutions for both constrained and unconstrained optimization. The results are compared with the analytical results computed using standard established methods and found to be in good agreement.

**Key words:** Boundary Value Problems, Engineering Problems, Neural Networks, Partial Differential Equations (PDEs)

## I. INTRODUCTION

There are many boundary value problems in various fields of engineering like image processing, Electromagnetics, which involves partial differential equation. Only of the few can be solved by analytical methods. Generally, people depend on the numerical methods to solve partial differential equation. Mostly, such numerical methods are computationally slow and give a good result of boundary value problems.

Artificial Neural Network [1-3] is in use for high-level design, providing fast and accurate solutions to the task it has learned. The use of neural networks provides advantages like good generalization properties, less model parameters than other methods, compact solution models and low demand on memory. The function approximation properties of neural network [4-7] make it attractive as trial solutions for differential equation for the boundary value problems. Hence it is supposed to be capable of tackling differential equations in many engineering problems.

In this paper solution of Poisson's equation using ANN is illustrated as example, using both constrained and unconstrained optimization techniques. In the first case, the original constrained optimization problem has to be reduced to unconstrained optimization, which is easier to handle due to the choice of the trial solution. In the latter case the problem is tackled in a constrained manner and care has been taken to meet the constraints like the boundary conditions. In ANN the weights and biases are to be adjusted in order to deal with the minimization problem.

## II. PROPOSED ANN TECHNIQUE FOR PARTIAL DIFFERENTIAL EQUATION SOLUTION

In this section the general method to solve a partial differential equation is discussed and the procedure can be extending for the complex differential equations.

### A. Unconstrained optimization Approach

ANN has been used for solving ordinary and partial differential equation [4-6]. A trial solution is taken as the sum of two terms The term

$$\varphi_i(\bar{x}, \bar{y}) = A(\bar{x}, \bar{y}) + F(\bar{x}, \bar{y}, N(\bar{x}, \bar{y}, wt)) \quad (1)$$

$A(x,y)$  contains no adjustable parameters and satisfies the boundary condition(s).  $N(x,y,wt)$  is a single-output feed-forward neural network with parameters 'wt' and input units fed with the input vectors  $x, y$ . The second term  $F$  is constructed in such a way so as not to contribute to the Bc's  $\varphi(\bar{x}, \bar{y})$ . This term employs a neural network whose weights and biases are to be adjusted in order to deal with the minimization problem. The original constrained optimization problem now has been simplified to an unconstrained one, since the choice of the form of the trial solution satisfies the BC's.

The function construction of  $F$  discussed above may be difficult, as it is dependent on the differential equations. This can be avoided by taking the trial solutions of the differential equation to be  $N(x,y,wt)$  and constructing the cost function by taking the error due to the equation and the error due to the boundary. Minimizing the cost function by training of the network shall determine the weights 'wt' for the problem at hand.

### B. Constrained optimization Approach

Alternatively, one may consider the simpler penalty function method to solve the constrained optimization problem. Other methods like Lagrange multipliers or active set method, can also be used.

## III. ILLUSTRATION OF THE METHODS FOR DIRICHLET BOUNDARY CONDITION

Consider the two-dimensional partial differential equation (Poisson's equation)

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} - f(x, y) = 0 \quad (2)$$

$x \in [0, 1]$ ,  $y \in [0, 1]$  with *Dirichlet* BC:  $\phi(0, y) = f_0(y)$ ,  $\phi(1, y) = f_1(y)$ ,  $\phi(x, 0) = g_0(x)$ ,  $\phi(x, 1) = g_1(x)$

**[a] Solution using unconstrained optimization approach**

$$\phi_t(x, y) = A(x, y) + x(1-x)y(1-y)N(x, y, wt) \quad (3)$$

The boundary conditions are satisfied by constructing  $A(x, y)$  as:

$$A(x, y) = (1-x)f_0(y) + xf_1(y) + y[g_1(x) - \{(1-x)g_1(0) + xg_1(1)\}] + (1-y)[g_0(x) - \{(1-x)g_0(0) + xg_0(1)\}] \quad (4)$$

Then, the cost function is

$$E = \left( \frac{\partial^2 \phi_t(x_i, y_j)}{\partial x^2} + \frac{\partial^2 \phi_t(x_i, y_j)}{\partial y^2} - f(x_i, y_j) \right)^2 \quad (5)$$

Equation 5 is minimized [4, 8] to train the network.

**[b] Solving using constrained optimization Approach**

Here  $\phi_t(x, y) = N(x, y, wt)$  is taken as the trial solution of the neural network instead of equation (3), 'wt' is the weight to be determined by training. In this method, two errors are taken into account for the construction of error function one due to the equation and the other due to the boundaries.

The equation error is given in equation (6) bellow.

$$Eq\_error = \left( \frac{\partial^2 \phi_t(x_i, y_j)}{\partial x^2} + \frac{\partial^2 \phi_t(x_i, y_j)}{\partial y^2} - f(x_i, y_j) \right)^2 \quad (6)$$

As in this problem there are four boundaries, therefore, the boundary errors are given by

$$E_{b1} = (\phi_t(0, y_j) - f_0(y_j))^2 \quad (7)$$

$$E_{b2} = (\phi_t(1, y_j) - f_1(y_j))^2 \quad (8)$$

$$E_{b3} = (\phi_t(x_i, 0) - g_0(x_i))^2 \quad (9)$$

$$E_{b4} = (\phi_t(x_i, 1) - g_1(x_i))^2 \quad (10)$$

The cost function for training the network is given by

$$E = Eq\_error + \lambda(E_{b1} + E_{b2} + E_{b3} + E_{b4}) \quad (11)$$

Equation (11) is used for updating the weights of the neural network.  $\lambda$  is the penalty factor the value is carefully chosen depending on how accurately the boundary condition is to be satisfied.

**Illustrated Example:**

Consider a two-dimensional Poisson's equation

$$\frac{\partial^2 \phi(x, y)}{\partial x^2} + \frac{\partial^2 \phi(x, y)}{\partial y^2} - xe^{-x} + 2e^{-x} + e^{-x}y^3 - 6e^{-x}y = 0 \quad (12)$$

with  $x \in [0, 1]$ ,  $y \in [0, 1]$  with *Dirichlet* BC:  $\phi(0, y) = y^3$ ,  $\phi(1, y) = (1 + y^3)e^{-1}$ ,  $\phi(x, 0) = xe^{-x}$ ,

$$\phi_a(x, y) = xe^{-x} + e^{-x}y^3 \quad (13)$$

A feed forward neural network with back-propagation algorithm [8,9,10] is used to train the network in a sequential way.

#### IV. TRAINING THE ANN

The network is trained to minimize the cost function for the unconstrained and constrained optimizations are equation (5) and equation (11) respectively. The network is trained according to the equations (14)-(17).

$$w_i^{k+1} = w_i^k - \eta \frac{\partial E^k}{\partial w_i} + \alpha(w_i^k - w_i^{k-1}) \quad (4)$$

$$v_{0j}^{k+1} = v_{0j}^k - \eta \frac{\partial E^k}{\partial v_{0j}} + \alpha(v_{0j}^k - v_{0j}^{k-1}) \quad (5)$$

$$v_{1j}^{k+1} = v_{1j}^k - \eta \frac{\partial E^k}{\partial v_{1j}} + \alpha(v_{1j}^k - v_{1j}^{k-1}) \quad (6)$$

$$v_{2j}^{k+1} = v_{2j}^k - \eta \frac{\partial E^k}{\partial v_{2j}} + \alpha(v_{2j}^k - v_{2j}^{k-1}) \quad (7)$$

Where,  $k$  is the iteration number,  $ij$   $v$  are the weights from the input to hidden neuron (except for  $i = 0$ );  $i$   $w$  are the weights from hidden to output and  $j$   $v_0$  is the threshold.  $\eta$ ,  $\alpha$  are the learning rate and momentum respectively. Selection of a value for the learning rate parameter  $\eta$  has a significant effect on the network performance. Usually,  $\eta$  must be a small to ensure that the network will settle to a solution. A small value of  $\eta$  means the network will have to make a large number of iterations. It is often possible to increase the size of  $\eta$  as learning proceeds. The value of

alpha should be positive but less than 1 and represent the momentum term.

## V. RESULTS

The exact solution of the PDE is  $\Phi_a(x, y) = x e^{y^2} + x e^{x^2} + x y^3$  and is compared with the Neural Network output in both the approaches i.e. unconstrained and constrained optimization. The error, in proposed ANN solutions, is calculated by the equation (18).

$$\text{error} = \Phi_a(x, y) - \Phi_t(x, y) \quad (18)$$

The number of input units, hidden neurons and output neurons are two, six and one respectively. The transfer function for the hidden neuron is  $\tanh(\cdot)$  and linear transfer function is used for the output neurons. The network is trained up to tolerance of  $0.5 \times 10^{-3}$  with  $\eta = 0.05$ ,  $\alpha = 0.5$ . In each case a graph is plotted for the analytical result, ANN Result and error. Fig. 1 shows the variation of Analytical, ANN and error for the unconstrained optimization with respect to  $(x, y)$ .

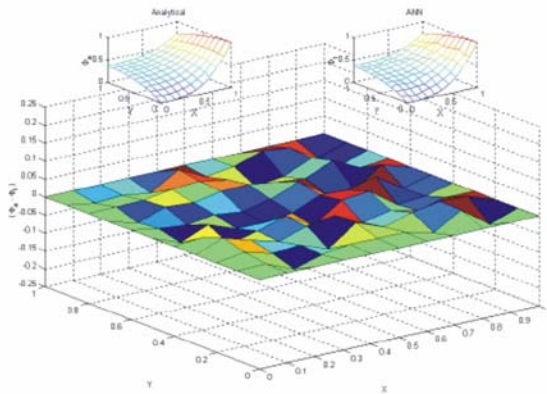


Fig. 1 Comparison of Analytical and ANN method using unconstrained optimization

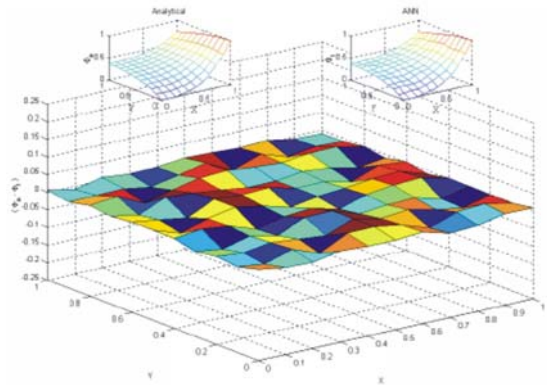


Fig. 2 Comparison of Analytical and ANN method using constrained optimization

Fig. 2 shows the variation of Analytical, ANN and error for the constrained optimization with respect to  $(x, y)$ . The network structure used for the constrained

optimization method is same as that of the unconstrained approach. The network is trained up to tolerance  $0.5 \times 10^{-3}$  with  $\eta = 0.07$ ,  $\alpha = 0.5$  and  $\lambda = 4$ .

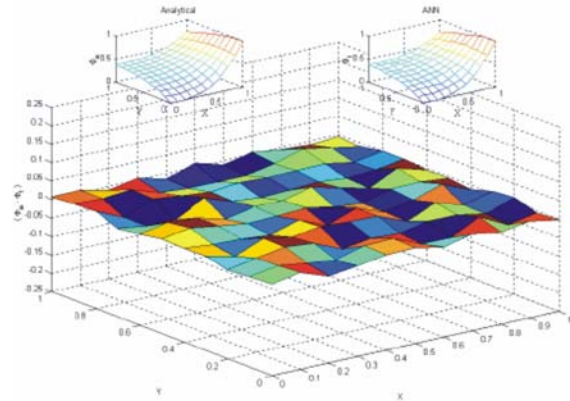


Fig. 3 Comparison of Analytical and ANN method using constrained optimization

Fig. 3 shows the variation of Analytical, ANN and error for the constrained optimization with respect to  $(x, y)$ . The same ANN, as used in the previous case, is used with a penalty of  $\lambda = 1.5$ .

## Illustration of the methods for Neumann and mixed boundary condition

Wave equation plays a major role in the RF engineering. In this section, trials have been made to solve the wave equation for the pure Neumann boundary condition as well as for mixed boundary conditions. Here we are not dealing with the pure Dirichlet boundary conditions as its illustration for the Poisson's equation is already discussed. We will solve the pure Neumann boundary condition with the unconstrained approach and the mixed boundary condition with the constrained approach. The constrained optimization and the unconstrained for the pure Neumann boundary conditions and mixed boundary conditions can be dealt in the similar way discussed for the Poisson's equation.

### [a] Unconstrained Approach for the Neumann Boundary condition.

In order to use ANN we propose the solution of the wave equation to be  $\Phi_t$  such that it satisfies the wave equation as well as the boundary condition given by the equations (19)-(21). 'L' is the length of the element.

$$\left( \frac{\partial^2}{\partial x^2} + k^2 \right) \Phi_t(x) = 0 \quad (19)$$

$$\left. \frac{\partial \Phi_r(x)}{\partial x} \right|_{x=0} = 0 \quad (20)$$

$$\left. \frac{\partial \Phi_r(x)}{\partial x} \right|_{x=L} = 0 \quad (21)$$

With  $x \in [0, 1]$  i.e. the input are normalized.

The trial solution of the wave equation is written as

$$\Phi_r(x) = LN(x, L, wt) - \left( Lx - \frac{x^2}{2} \right) \frac{\partial N(0, L, wt)}{\partial x} - \frac{x^2}{2} \frac{\partial N(L, L, wt)}{\partial x} \quad (22)$$

Where  $N(x, L, wt)$  is the output of a feed forward network with two inputs  $x$  for position coordinate, 'L' length and 'wt' is the weight vector of the ANN. By construction the trial solution satisfies the two boundary conditions and the constrained optimization problem is converted to unconstrained optimization problem.

$$N = \sum_{i=1}^q w_i \zeta(z_i) \quad (23)$$

where  $z_i = v_{1i}x + v_{2i}L + v_{0i}$  (24)

$$\zeta(z) = \tanh(z) \quad (25)$$

The cost function of the network for training is given by (26).

$$Error = L \frac{\partial^2 N(x, L, \bar{p})}{\partial x^2} + k^2 LN(x, L, \bar{p}) + \left[ 1 - k^2 \left( Lx - \frac{x^2}{2} \right) \right] \frac{\partial N(0, L, \bar{p})}{\partial x} - \left[ 1 + k^2 \frac{x^2}{2} \right] \frac{\partial N(L, L, \bar{p})}{\partial x} \quad (26)$$

The square of the error is given by the expression E.

$$E = (Error)^2 \quad (27)$$

The weights of the network are trained according to the normal back propagation algorithms and the weight update equations are given by the equations (14) – (17). Network structure: The network structure is 2-6-1 and the transfer function for the hidden and output layer are 'tanh(.)' and linear respectively. The network is trained upto a tolerance of  $10^{-4}$ . The value of both  $\alpha$  and  $\eta$  in this model is 0.0008.

The accuracy of the method is checked by comparing the result of ANN with that of the closed form solution (Figure 4). The results are in good agreement. The error is due to the training error.

#### [b] Constrained Optimization Approach for the Mixed Boundary condition.

Here we have taken the wave equation in the cylindrical

coordinate system for a circular element. Actually  $\Psi$  is a function of  $r, z$  and  $\phi$  but the variation along  $\phi, z$  is not taken into account for symmetric problems with insignificant contributions from  $z$ .

$$\nabla^2 \Psi + k^2 \Psi = 0 \quad (28)$$

Transforming equation (28) into cylindrical coordinate system.

$$\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial \Psi}{\partial r} \right) + \frac{1}{r^2} \left( \frac{\partial^2 \Psi}{\partial \phi^2} \right) + k^2 \Psi = 0 \quad (29)$$

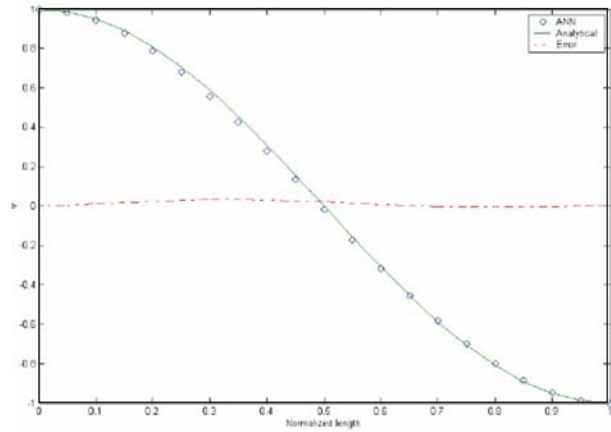


Fig. 4 Comparison of variation of  $Q$  with normalized length of the element using Analytical and ANN model

Neglecting the variation in  $\phi$ .

$$\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial \Psi}{\partial r} \right) + k^2 \Psi = 0 \quad (30)$$

$$\left( r \frac{\partial^2 \Psi}{\partial r^2} + \frac{\partial \Psi}{\partial r} \right) + k^2 r \Psi = 0 \quad (31)$$

The boundary condition for  $\Psi$  are given by equation (32) and (33)

$$\Psi(0) = 0 \quad (32)$$

$$\left. \frac{\partial \Psi}{\partial r} \right|_{r=a} = 0 \quad (33)$$

Here the boundary condition is considered to be mixed i.e. combination of Dirichlet and Neumann. The penalty function approach is used for the solution using ANN. The trial solution is given below.

$$\Phi_r(r) = N(r, a, wt) \quad (34)$$



The cost function to be minimized in this case is given by equation 35.

$$Error = (Err\_equ)^2 + \lambda((Err\_bc1)^2 + (Err\_bc2)^2) \quad (35)$$

Where, (36)

$$Err\_equ = r \frac{\partial^2 N(r, a, wt)}{\partial r^2} + \frac{\partial N(r, a, wt)}{\partial r} + k^2 r N(r, a, wt) \quad (37)$$

$$Err\_bc1 = N(r, a, wt)$$

$$Err\_bc2 = \left( \frac{\partial N(r, a, wt)}{\partial r} \right) \quad (38)$$

The error function to be minimized is given by equation (35). The network is trained according to the normal back propagation algorithms [7] and the weight updated according to equations (14)-(16).

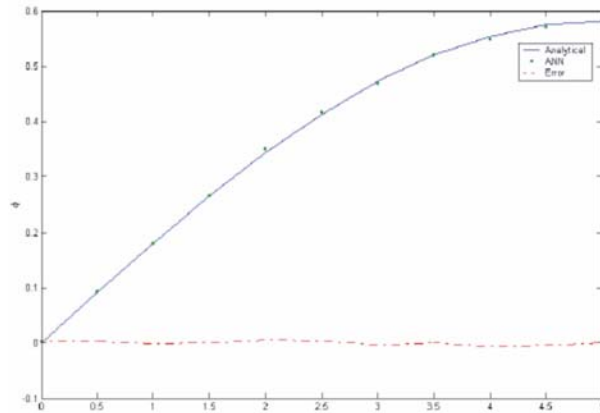


Fig .5 Comparison of variation of  $\phi_t$  with radius 'r' of the circular element using Analytical and ANN model

The network structure used is same as of the pervious case with  $\eta = 0.0018$ ,  $\alpha = 0.008$ , Penalty factor  $\lambda = 4$ . In the neural network method, increasing the number of hidden units can control the accuracy. In this approach time increases linearly with the number of parameters. Even if there are rules and standard algorithms, fixing each parameter of the neural network structure is more of an art than science. Human intervention at every step is necessary during training. That is to say, it mostly depends on experience

## VI. CONCLUSION

A method has been discussed for solving differential equations defined on orthogonal box boundaries that relies upon the function approximation capabilities of feed-forward neural networks and provides accurate and differentiable solutions in a closed analytic form. This

method can be extended to three-dimensional problems. As the dimensionality increases, the number of training points becomes large. This fact becomes a serious problem for methods that consider local functions around each grid point. In the case of the neural method, the number of training parameters remains almost fixed as the problem dimensionality increases. The penalty factor  $\lambda$  should not be increased arbitrarily unless all parameters of the network are normalized consistently. With simpler construction of the trial solution, the constrained optimization approach looks promising than the unconstrained one.

## REFERENCES

- [1] A. Patnaik and R. K. Mishra, March 2000, "ANN Techniques in Microwave Engineering," IEEE Microwave Magazine, vol. 1, no. 1, pp. 55–60,
- [2] K. C. Gupta and P. M. Watson, 1996, "Application of ANN computing to microwave design," 1996 Asia-Pacific Microwave Conference Proceedings, vol. 3, pp. 825–828, 1996.
- [3] A. Patnaik, 2000, "Some Studies on use of Artificial Neural Network for Analysis of Microstrip Antenna," Ph.D. dissertation, Berhampur University, pp 33-34 .
- [4] H. Lee and I. Kang, 1990, "Neural algorithms for solving differential equations," J. Comput. Phys., vol. 91, pp. 110–117,
- [5] A. J. Meade, Jr., and A. A. Fernandez, 1994, "The numerical solution of linear ordinary differential equations by feedforward neural networks," Math. Comput. Modeling, vol. 19, no. 12, pp. 1–25.
- [6] Silvia Ferrari and Robert F. Stengel, Jan. 2005, "Smooth Function Approximation Using Neural Networks", IEEE Trans. Neural Networks, Vol. 16, No. 1, pp. 24–28,
- [7] R. Yentis and M. E. Zaghoul, 1996, "VLSI implementation of locally connected neural network for solving partial differential equations," IEEE Trans. Circuits Syst. I, vol. 43, no. 8, pp. 687–690.
- [8] R. K. Mishra, A. Patnaik, Dec. 1998 "Neural network based CAD model for design of square patch antennas," IEEE Trans., vol. AP- 46, no. 12, pp. 1890–1891.

- [9] K. C. Gupta and P. M. Watson, 1996, "Application of ANN computing to microwave design," 1996 Asia-Pacific Microwave Conference Proceedings, vol. 3, pp. 825-828,
- [10] N. B. Karayiannis and A. N. Venetsanopoulos, 1991, "Fast learning algorithms for neural networks," in Artificial Neural Networks, T. Kohonen et. al., Eds. North Holland: Elsevier Publishers, pp. 1141-1144.



**Mr. Sridhar Pattanaik** - Research Scholar in the Department of Electronic Sciences, Berhampur University is the recipient of the Outstanding Young Scientist Award by Orissa Bigya Congress-Bhubaneswar Chapter of Indian Science Congress in 2002. His research interests include Microstrip Antenna, ANN, Soft Computing.