Solving Linear Programming Problem by Using Neural Network Model

[1]Prateeksha Chouksey, [2]K. N. Hande
[1,2]Dept. of CSE, Smt. Bhagwati Chaturvedi College of Engg., Nagpur, India

Abstract
Linear Programming (LP) is the process of taking various linear inequalities together and find out the "best" optimal solution. Linear programming problems arise in real-life economic situations where profits are to be maximized or costs to be minimized. This paper shows the flexibility of neural networks for modeling and solving diverse linear programming problems. Advantage of using neural networks to solve problems includes powerful computation and less time required. The proposed system is to build the neural network model by using back propagation algorithm for solving linear programming problem. Here the neurons are trained by using back propagation algorithm. This system will provide powerful computation and the computational time required is low.

Keywords
Linear Programming, Neural Network, Back Propagation, Training and Testing

I. Introduction

Linear programming problems arise in a wide variety of scientific and engineering fields including regression analysis, function approximation, signal processing, image restoration, parameter estimation, filter design, robot control, etc. However, traditional numerical methods may not be economical for digital computers since the computing time needed for an answer is greatly enthusiastic about the dimension and also the structure of the matter and also the complexness of the rule used. One promising approach to handle these optimization issues with high dimension and dense structure is to use artificial-neural-network-based implementation.

An Artificial Neural Network (ANN),

Usually called neural network (NN), is a mathematical model or computational model that is inspired by biological neural networks. A neural network consists of a group of interconnected artificial neurons, and it processes data employing a connectionist approach to computation. Generally an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are generally used to model complex relationships between inputs and outputs or to find patterns in data. The main benefits of using ANN on linear programming problems are the following.

- The ability of coaching the neurons.
- The facility of implementation of circuit in hardware.
- The capability of mapping complicated systems while not necessity of knowing the ultimate mathematical models related to them.

Linear programming problems can be represented in canonical form:

Maximize $c^T x$

Subject to $Ax \leq b$

And $x \geq 0$.

Where x is the vector of variables that is to be determined, c and b are vectors of known coefficients, A is a known matrix of coefficients, and (.)T is the transpose matrix. The expression to be maximized or minimized is called the objective function (cTx in this case). The inequalities $Ax \leq b$ is the constraints which specify a convex polytype over which the objective function is to be optimized. In this situation, two vectors are comparable after they have same dimensions. If each entry within the first is less-than or equal-to the corresponding entry within the second then we are able to say the first vector is less-than or equal-to the second vector. A detailed literature is provided in session II. In session III, a method is proposed to solve linear programming problem by using neural network. Back propagation algorithm is used to train the network.

II. Related Work

The wide used methodology for linear programming problem is revised simplex method, a univariate search technique. It suffers the drawback of slow convergence with the tendency of variable pop in and out of the basis matrix. The numbers of iterations are more and increase in computational effort and time. This gives wide scope to a search of new algorithm. In the bounded complex algorithm a univariate search technique is used, but the entering variables are arranged not only based on the maximum contribution if that variable to the objective function but also on the type of the constraints. Once the priority arrangement of the promising variables is obtained the iterations are performed as in the Simplex Method by selecting a leaving variable based on bounds of the variables. Redundancies, if any, in the LP model will waste computational effort. There is a wide literature on the detection and elimination of redundant constraints in linear programming models. Heuristic approach is using an intercept matrix to identify redundant constraints prior to the start of the solution process. During this heuristic approach, tendency of variables to pop in and pop out of the basis is eradicated when eliminating the redundancies. The eradication of pop-in and pop-out considerably reduces the number of iterations. A major reduction within the computation effort is achieved for linear programming problems of different sizes in heuristic approach.

The Hopfield neural network has proven to be a powerful tool for solving some of the optimization problems. Tank and Hopfield first proposed a neural network for solving mathematical programming problems, where a linear programming problem (LPP) was mapped into a closed-loop network [1]. Tank and Hopfield network may not be a solution of original problem. This work has inspired many researchers to investigate other neural networks for solving linear and nonlinear programming problems.

Kennedy & Chua extended the Tank and Hopfield network by developing a neural network for solving nonlinear programming problems, by satisfaction of the Karush–Kuhn–Tucker optimality conditions [2]. The proposed network uses the penalty parameter and it produces the approximate solutions only. But it generates the implementation problem when penalty parameter is large.

Hasan Ghasabi-Oskoei and Nezam Mahdavi-Amir [3] present a high-performance and efficiently simplified new neural network which improves the existing neural networks for solving general linear and quadratic programming problems. In this proposed network there is no need of parameter setting, results in a simple

hardware requiring no analog multiplier. It is shown to be stable and converges globally to the exact solution. By using this network, the linear and quadratic programming problem and their duals can be solved simultaneously. The features of this network are high accuracy of optimal solution and low cost implementation.

L.R. Arvind Babu and B. Palaniappan [4] propose an algorithm, called, Hybrid Algorithm, which trains the constraint and parameter before applying the formal methodology. Linear Programming Problems are mathematical models used to represent real life situations in the form of linear objective function and constraints various methods are available to solve linear programming problems. This linear programming problem can be solved by often including all possible constraints but some of them may not generate optimal solution. The presence of redundant constraints does not generate the optimal solution but it may consume extra computation time. To reduce computation time, redundant constraints identification methods are used. But this will reduce the accuracy due to reduction of loops and constraints. So to achieve accuracy in optimal solution and computational time the new algorithm is proposed called hybrid algorithm.

For implementation of the problem in neural network G.M. Nasira, S. Ashok Kumar and T.S.S. Balaji [5] proposed a new and simple hybrid (primal-dual) algorithm which finds the optimal solution for a class of integer linear programming problems. This integer linear programming problem is a special case of linear programming problem. Generally, this type of problem is solved by using simplex method such as Gomory method and Bound technique but this new algorithm does not use this simplex method. Since the results of this proposed new hybrid (primal- dual) algorithm is very fast and efficient in comparison to the results of the problems solved by simplex method.

Neeraj Sahu and Avanish Kumar [8] propose a solution of the Linear Programming Problems based on Neural Network Approach. There is no location restriction and this network is implemented only in simple hardware. Here they proved to be completely stable to exact solution without any multipliers. Further, this network is used to solve linear programming problem and it's dual simultaneously.

Ue - Pyng Wen, Kuen - Ming Lan and Hsu - Shih Shi [12] propose a Hopfield neural network for solving mathematical programming problem. The major advantage of Hopfield neural network is its structure is accomplished on an electronic circuit, possibly on a VLSI (very large-scale integration) circuit, for an on-line solver with a parallel-distributed method. The structure of Hopfield neural network utilizes three common ways, penalty functions, Lagrange multipliers, and primal and dual method to construct an energy function. When the function reaches at stable state, it produces the approximate solution to the problem. Under the categories of those methods, they tend to organize Hopfield neural networks by three types of mathematical programming problems: linear, non-linear, and mixed-integer.

Hong - Xing Li and Xu Li Da [13] propose the flexibility of neural networks for modeling and solving the diverse mathematical problems. The advantages of using neural networks to solve problems are clear visualization, powerful computation and easy to implement in hardware. They first introduce the well known exclusive OR (XOR) problem, followed by Taylor series expansion and Weierstrass's first approximation theorem.

The conventional linear programming and quadratic programming is extended as extended linear- quadratic programming (ELQP) problem. Xiaolin Hu [14] proposes solving extended linear-quadratic programming with general polyhedral sets by using recurrent neural networks. The general projection neural network is used to reduce the network complexity and the dimensions based on specific types of constraints. All of these designed general projection neural networks are globally convergent to the solutions of the corresponding extended linear-quadratic programming problems under mild conditions.

III. Proposed Work

Linear programming (LP) is the process of taking various linear inequalities together and find out the "best" optimal solution. This linear programming problem is used in various real-life applications like economic problem to find out the profit to be maximized or cost to be minimized. Linear programming problem is define as, consider the objective function

Min P(x) = cTx;

Subject to: $Ax \geq b$

And $x>=0$ $(= g\ j\ (x) \geq b\ j, j=1, 2... m)$

Where x is the vector of variables that is to be determined, c and b are vectors of known coefficients, A is a known matrix of coefficients, and (.)T is the transpose matrix. The expression to be maximized or minimized is called the objective function (cTx in this case). The inequalities $Ax \leq b$ is the constraints which specify a convex polytype over which the objective function is to be optimized. These linear programming problems can be solved by various methods i.e. polynomial time, interior point method, simplex method etc. These linear programming problems can also be solved by using neural network. Neural network is software or hardware simulation of a biological neuron. This neural network is used to learn to recognize the patterns in the data. Once this neural network has been trained on the samples of the data, it can make predictions by detecting similar patterns in future data. The typical structure of neural network is shown in below diagram.

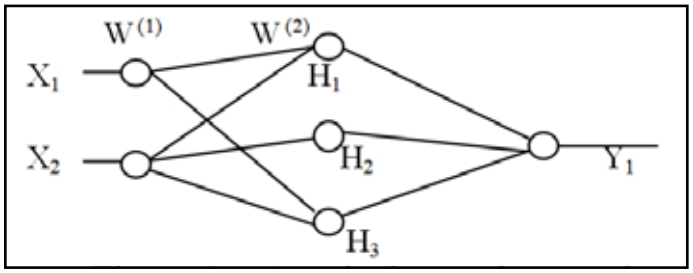

Fig. 1: A Typical Neural Network

The typical neural network comprises of inputs X1, X2, corresponding to independent variables. A hidden layer as the first layer, and second layer is the output layer whose output unit is Y1 corresponding to dependent variables. In between there is hidden layer H1, H2……corresponding to intermediate variables. These interact by means of weight variables i.e. W(1), W(2).The activation function in neural network model is defined as

$F(x) = P(x) + \sum_{j=1}^{m} (gj+(x))^2$

This paper presents the linear programming problem which will be solved by using neural network. The linear programming problem can be solved by various algorithms of neural network i.e. feed forward algorithm, hybrid algorithm, Hopfield neural network etc. But this paper presents the linear programming problem which will be solved by using back propagation algorithm. Use of back propagation is rear in linear programming problem solution. It has its own advantages and disadvantages. Based on the detail literature survey and literature Gap following work is proposed. It contains defining linear programming problem and solution with neural network approach. Back propagation algorithm is

used to train the network. Main Idea is to distribute the error function across the hidden layers, corresponding to their effect on the output.

Process is divided into three modules

1. Implementation of neural network model.
2. Implementation of linear programming problem.
3. Solving linear programming problem by using neural network.

Step 1: Implementation of neural network model.

The neural network model will be implemented by using back propagation algorithm. Back propagation algorithm is used to train the network. In first phase neural network model will be established which will be checked for working of back propagation algorithm on any simple problem. The simulation technique is used to implement the network model.

Step2:Implementation of linear programming problem.

In this step the linear programming problem will be solved in MATLAB. Simple procedure will be used like Simplex Algorithm to solve the linear programming problem. It will be run for some problems to prepare test cases which will be used in training stage in final module.

Step 3: Solving linear programming problem by using neural network.

The last module is designed to solve LPP by neural network. Working of the model is explained as follows.

Consider a LP problem in the following standard form:

Find x that

Maximizes: bTx

Subject to the constraints: A x<= c, x >=0 (1)

Where x and b ε Rn, A ε Rm x n and c ε Rm.

The dual problem of (1) is:

Find y that

Minimizes: c T y

Subject to the constraints: A T y >=b, y >= 0 (2)

Mathematically, the outputs of the above primal and dual neurons can be described by the following nonlinear dynamical system:

$dx/dt = b - AT\ (y + k\ (dy/dt))$, $x >= 0$ (3)

$dy/dt = -c + A\ (x + k\ (dx/dt))$, $y >= 0$ (4)

It can be seen that (3) and (4) are equivalent to a system of second order differential equations. The Euler method is used to solve differential equations (3) and (4) which will be solved in neural network processing.

Algorithm:

Input:

1. Pair of Linear Programming Problems with solution
2. Minimum error value

Output:

Training Stage:

1. Initialize the weights in the network (often randomly)
2. Do

For each example data (e) in the training set

i. O= neural-net-output (network, e);

ii. T = output for e

iii. Calculate error (T - O) at the output units

iv. Compute delta_wi for all weights from Hidden layer to output layer;

v. Backward pass Compute delta_ wi for all weights from input layer to hidden layer;

vi. Backward pass continued

vii. Update the weights in the network until dataset classified correctly or stopping Criterion satisfied.

III. Return the Network

A. Testing Stage

Simulate the network with new input (testing) problem. In the processing T is calculated by using Euler's method. By using various test cases neural network will be train and that network will be used to solve linear programming problem. Expected output is a solution for the problem.

IV. Conclusion

This paper presents a detailed survey of literature which was carried out in neural network model for solving linear programming problems and Literature gap for solving linear programming problem by neural network. Very less work has been proposed in this area as compared to other. Based on the literature, a new approach is proposed to solve linear programming problem by using back propagation algorithm. Further work includes implementation of above mentioned work.

References

[1] Tank, D.W., Hopfield, J.,"Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit", IEEE Transactions on Circuits and Systems, 33(5), pp. 533–541, 1986.

[2] Kennedy, M.P., Chua, L.O.,"Neural networks for nonlinear programming", IEEE Transactions on Circuits and Systems, 35(5), pp. 554–562, 1988.

[3] Hasan Ghasabi-Oskoei a, Nezam Mahdavi-Amiri,"An efficient simplified neural network for solving linear and quadratic programming problems", Applied Mathematics & computation Journal, Elsevier, pp. 452–464, 2006.

[4] L.R. Arvind Babu, B. Palaniappan,"Artificial Neural Network Based Hybrid Algorithmic Structure for Solving Linear Programming Problems", International Journal of Computer and Electrical Engineering, Vol. 2, No. 4, August, 2010, pp. 1793-8163.

[5] G.M. Nasira, S. Ashok Kumar, T.S.S. Balaji,"Neural Network Implementation for Integer Linear Programming Problem", 2010 International Journal of Computer Applications, Vol. 1, No. 18.

[6] Mohsen Alipour,"A Novel Recurrent Neural Network Model For Solving Nonlinear Programming Problems With General Constraints", Australian Journal of Basic and Applied Sciences, 5(10), pp. 814-823, 2011.

[7] Khanh V. Nguyen,"A Nonlinear Neural Network for Solving Linear Programming Problems".

[8] Neeraj Sahu, Avanish Kumar,"Solution of the Linear Programming Problems based on Neural Network Approach", IJCA, Vol. 9, No. 10, November 2010.

[9] Maa, C.-Y., Shabbat, M.A.,"Linear and quadratic programming neural network analysis", IEEE Transactions on Neural Networks, 3(4), pp. 580–594, 1992.

[10] Alaeddin Malek, Maryam Yashtini,"A Neural Network Model for Solving Nonlinear Optimization Problems with Real-Time Applications", W. Yu, H. He and N. Zhang (Eds.): ISNN 2009, Part III, LNCS 5553, pp. 98–108, 2009. © Springer-Verlag Berlin Heidelberg 2009.

[11] Xingbao Gao, Li-Zhi Liao,"A New One-Layer Neural Network for Linear and Quadratic Programming", IEEE Transactions on Neural Networks, Vol. 21, No. 6, June 2010.

[12] Ue - Pyng Wen, Kuen - Ming Lan, Hsu - Shih Shi,"A review of Hopfield neural networks for solving mathematical programming problems", European Journal of Operational Research, Elsevier, pp. 675- 687, 2009.
[13] Hong - Xing Li, Xu Li Da,"A neural network representation of linear programming", European Journal of Operational Research, Elsevier, pp. 224-234, 2000.
[14] Xiaolin Hu,"Applications of the general projection neural network in solving extended linear-quadratic programming problems with linear constraints", Neurocomputing, Elsevier, pp. 1131 -1137, 2009.