Journal of Kirkuk University –Scientific Studies , vol.2, No.3,2007

Solving Linear Equations Using Matrix Splitting for Iterative Discrete-Time Methods in Neural Networks

Essa I. Essa
College of Science-University of Kirkuk

Abstract

The material presented in this paper is the foundation for neural network architectures that can perform (Solving linear equations using matrix splitting for iterative discrete-time methods in neural networks).As announced a neural network consists of many inter connected processing elements (neurons or nodes), I can begins with the presentation of a particular neural network is dependent on the training phase (specifically the training data used). Matrix splitting solved in several preprocessing methods. Many times it's necessary to processes the training data to extract important features from the data can be used to train the network instead of the "raw" data. The preprocessing of the training data can therefore, improve the performance of the neural network. Then, the convergence is achieved using the Richardson and Gauss-Seidel methods, respectively. The same termination criterion was used for both these methods in order to properly compare all the results we see that the SOR iterative method gives the best results, that is, the fastest convergence. Comparing the SOR results with the next-best results (Gauss-Seidel, $\omega = 1$); we see that the SOR method is about 10 times faster.

Introduction

There are four basic iterative discrete-time methods used to solve linear equations of the form

$$A\chi = b \qquad \ldots(1)$$

For $A \in \Re^{n\times m}$ that are based on matrix splitting (Golub et al,1996,N. Higham,1996,R.S. Varga,1992,D.M.Young,1971,Luenberger,2003,and K.E.Atkinson,1989). The solution vector x exists and is unique if and only if A in nonsingular.

All the methods have the basic form

$$Mx(K+1) = Nx(K) + b \qquad \ldots(2)$$

Where K is the discrete-time index. We can express matrix A as the matrix sum

$$A = D - E - F \qquad \ldots(3)$$

Where $D \in \Re^{n\times m}$ is a diagonal matrix, $d = diag[a_{11}, a_{22},..., a_{nn}]$, *and* $E \in \Re^{n\times m}$ *and* $F \in \Re^{n\times m}$ are, respectively, strictly lower and upper triangular matrices.

The entries of *E* and *F* are the negative of the entries of *A* respectively, below and above the main diagonal of *A*. The diagonal elements of *D* are all assumed to be nonzero.

Iterative Methods

Jacobi iterative method:-

Substituting (3) into (1) gives

$$(D-E-F)x = b \qquad \ldots(4)$$

This is then split as

$$Dx = (E+F)x + b \qquad \ldots(5)$$

And an iterative scheme can be written from (5) as

$$Dx(K+1) = (E+F)x(K) + b \qquad \ldots(6)$$

Therefore, according to (2), M=D and N=E+F, since the diagonal elements of D are nonzero, we can write (6) as

$$x(K+1)D^{-1}(E+F)x(K) + D^{-1}b \qquad \ldots(7)$$

Where $K \geq 0$, *and* $x(0)$ is the given initial- condition vector. This is the vector-matrix form of the Jacobi iterative method (Varga,1992), and we call the matrix

$$B = D^{-1}(E+F) \qquad \ldots(8)$$

The Jacobi matrix. The scalar form of (7) is given by

$$x_i(K+1) = -\sum_{\substack{j=1 \\ j\neq i}}^{n}\left(\frac{a_{ij}}{a_{ii}}\right)x_j(K) + \frac{b_i}{a_{ii}} \qquad \ldots(9)$$

Where $1 \leq i \leq n$, $K \geq 0$, *and* $x_i(0)$ is the given initial condition. I see from (9) that in general all the components of vector $x(K)$ must be saved while computing the components of vector $x(K+1)$.However; it seems plausible to use the latest estimates $x_i(K+1)$ of the components x_i of the solution vector X in all subsequent computations. This leads to the second method in this class.

Gauss-Seidel iterative method:-

Starting with(4), $(D-E-F)x = b$, we rearrange (i.e., split) this expression as

$$(D-E)x = Fx + b \qquad \ldots(10)$$

where (*D-E*) is a nonsingular lower triangular matrix. An iterative scheme can be written from (10) as

$$(D-E)x(K+1) = Fx(K) + b \qquad \ldots(11)$$

or

$$x(K+1) = (D-E^{-1})Fx(K) + (D-E)^{-1}b \qquad \ldots(12)$$

For $K \geq 0$, *and* $x(0)$ is the given initial-condition vector. Comparing (11) with (12) reveals $M = D-E$ *and* $N = F$. This is the vector-matrix form of the Gauss-Seidel iterative method (Varga,1992), and the matrix $C = (D-E)^{-1}F$ is called the Gauss-Seidel matrix.

The scalar form of the Gauss-Seidel iterative method can be written from $Dx(K+1) = Ex(K+1) + Fx(K) + b$, and is given by

$$x_i(K+1) = -\frac{1}{a_{ii}}\sum_{j=1}^{i-1} a_{ij}x_j(K+1) - \frac{1}{a_{ii}}\sum_{j=i+1}^{n} a_{ij}x_j(K) + \frac{b_i}{a_{ii}} \qquad \ldots(13)$$

where $1 \leq i \leq n$, $K \geq 0$, *and* $x_i(0)$ is the given initial condition.

Successive overrelaxation iterative method:-

In the case of the successive overrelaxation (SOR) iterative method, we write the split of the A matrix as

$$A = M\omega - N\omega = D - E - F \qquad \ldots(14)$$

where

$$M\omega = \frac{1}{\omega}(D - \omega E) \qquad \ldots(15)$$

and

$$N\omega = \frac{1}{\omega}\left[(1-\omega)D + \omega F\right] \qquad \ldots(16)$$

the parameter ω is called the relaxation factor. Therefore, substituting (15) and (16) into (2) gives:

$$\frac{1}{\omega}(D - \omega E)x(K+1) = \frac{1}{\omega}\left[(1-\omega)D + \omega F\right]x(K) + b \qquad \ldots(17)$$

multiplying both sides of (17) by ω and then premultiplying both sides by D^{-1} give

$$(I - \omega D^{-1}E)x(K+1) = \left[(1-\omega)I + \omega D^{-1}F\right]x(K) + \omega D^{-1}b \qquad \ldots(18)$$

we now define $L \underline{=} D^{-1}E$ (strictly lower triangular), and $U \underline{=} D^{-1}F$ (strictly upper triangular) and substitute these into (18) to give $(I - \omega L)x(K+1) = \left[(1-\omega)I + \omega U\right]x(K) + \omega D^{-1}b$ and premultiplying both sides by $(I - \omega L)^{-1}$ yields

$$x(K+1) = (I - \omega L)^{-1}\left[(1-\omega)I + \omega U\right]x(K) + \omega(I - \omega L)^{-1}D^{-1}b \qquad \ldots(19)$$

for $K \geq 0$ *and* $x(0)$ is the given initial-condition vector. Equation (19) is the vector-matrix form of the SOR iterative method (Varga,1992). The matrix

$$\mathcal{L}_{\omega} = (I - \omega L)^{-1}[(1-\omega)I + \omega U] \qquad \ldots(20)$$

is called successive relaxation matrix. If the relaxation factor lies in the range $0 \leq \omega \leq 1$, then this is underrelaxation. However, if $\omega > 1$, this is overrelaxation. Note that if the relaxation factor is set to $\omega = 1$, then (19) reverts to the vector-matrix form of the Gauss-Seidel method in(12). The scalar form of the SOR iterative method can derived from(17) rewritten as $Dx(K+1) = \omega Ex(K+1) + (1-\omega)Dx(K) + \omega Fx(K) + \omega b$. The scalar form of the SOR iterative method is given by

$$x_i(k+1) = \frac{\omega}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j(k+1) - \sum_{j=i+1}^{n} a_{ij} x_j(k) \right] + (1-\omega)x_i(k) \qquad \ldots(21)$$

where $1 \leq i \leq n$, $k \geq 0$, *and* $x_i(0)$ is the given initial condition. Figure (1) shows a neural network architecture realization of the SOR iterative method. Using the definition of the successive relaxation matrix in (20), Equation (19) can be written as :

$$x(k+1) = \mathcal{L}_{\omega} x(K) + Rb \qquad \ldots(22)$$

where $R \underline{\Delta} \omega(I - \omega L)^{-1} D^{-1}$ .We now define an error vector as:

$$e(k) = x(k) - x \qquad ; k \geq 0 \qquad \ldots(23)$$

where x is the unique vector solution of (1). For this error, from (22) we can write a homogeneous error difference equation as:

$$e(k+1) = \mathcal{L}_{\omega} e(k) \qquad \ldots(24)$$

The relaxation factor ω can be chosen to minimize $\sigma_r[\mathcal{L}_{\omega}]$in order to make $x(k)$ converge to x as rapidly as possible (Atkinson,1989), where $\sigma_r(\bullet)$ is the spectral radius of $\mathcal{L}_{\omega}$, we will call the optimal value of the relaxation factor ω° . The calculation of ω° can be difficult except in simple cases. Typically, it is approximation by trying several values of ω and observing the effect on the speed of convergence. Even with the problem of calculating ω°, the effort is worth it because, of the resulting dramatic increase in the speed of convergence of $x(k)$ *to* x.

Richardson's iterative method:

Another method that can be considered in this class of iterative techniques iterative method (Young,1971). The basic idea is to iterate until the negative of the discrete-time approximation to the first derivative of the solution reaches zero, that is:

$$-\frac{x(k+1) - x(x)}{\beta} = Ax(k) - b \qquad \ldots(25)$$

where $e(k) = Ax(k) - b \quad and \quad k \geq 0$. This expression can be rearranged as:

$$x(k+1) = x(k) - \beta(k)\left[Ax(k) - b\right] \qquad \ldots(26)$$

where $\beta = \beta(k)$, $x(0)$ is the given initial-condition vector, and the optimal iteration parameter, can be determined from (Cichocki et al,1993, Gill P.E. et al,1981,Akaike,H.,1968,Barton, S.A.,1991,and Hertz,J. et al,1991):

$$\beta(k) = \frac{e^T(k)e(k)}{e^T(k)Ae(k)} \qquad \ldots(27)$$

we can derive the scalar form of Richardson's iterative method from the vector-matrix form given in (26) as:

$$x_i(k+1) = x_i(k) - \beta(k)\left[\sum_{j=1}^{n} a_{ij}x_j(k) - b_i\right] \qquad \ldots(28)$$

where $i \leq i \leq n$, $k \geq 0$, and $x_i(0)$ is the given initial condition. If we choose $\beta(k) = 1/a_{ii}$ in (28) we can write the iterative expression as:

$$x_i(k+1) = x_i(k) - \frac{1}{a_{ii}}\left[\sum_{j=1}^{n} a_{ij}x_j(k) - b_i\right] \qquad \ldots(29)$$

which in turn can be written as:

$$x_i(k+1) = -\sum_{\substack{j=1 \\ j\neq i}}^{n}\left(\frac{a_{ij}}{a_{ii}}\right)x_j(k) + \frac{b_i}{b_{ii}} \qquad \ldots(30)$$

where $1 \leq i \leq n, k \geq 0, and$ $x_i(0)$ is the initial condition. This is precisely the Jacobi iterative method given in (9).

Implementation of Neural Networks and Results

The following example show how neural networks are adopted in solving linear equations using matrix splitting for iterative discrete-time methods in neural networks. In this example the Matlab package is used, where the neural networks of figure (1), simulated, and the results are shown in this section.

Example1:-This example compares the performance of three of the four methods presented in this paper for solving linear equations of the form $Ax = b$. In this example the system is given by:

$$\begin{bmatrix} 6 & 5 & 5 & 6 \\ 5 & 9 & 4 & 3 \\ 5 & 4 & 10 & 5 \\ 6 & 3 & 5 & 7 \end{bmatrix} x = \begin{bmatrix} 55 \\ 47 \\ 63 \\ 55 \end{bmatrix} \qquad \ldots(31)$$

Because the condition number for A is relatively small $[cond(A) = 394.8742]$, the *inv* function in (MATLAB version 5.1) can be used to solve the system of equation in (31), the result is given as:

$$x^M = inv(A) * b = \begin{bmatrix} 1.0000 \\ 2.0000 \\ 3.0000 \\ 4.0000 \end{bmatrix} \quad \ldots(32)$$

The successive overrelaxation method requires determining the optimal value for the relaxation parameter ω . Following the procedure given in this paper on the SOR method, the minimum of $\sigma_r[\mathcal{L}(\omega)]$ as a function of ω must be found. When we used the range of values for ω given by $0 \le \omega \le 2.5$ and $\Delta\omega = 1/10.000$. The well-defined minimum establishes the optimal value to use for the relaxation parameter given by $\omega^\circ = 1.7215$ .Each the three methods uses the same set of initial conditions given by $x(0) = [0.0118, 0.0315, 0.1444, -0.0351]^T$. These initial conditions were generated by selecting four random numbers from a Gaussian distribution with zero mean and variance of 0.01. The SOR algorithm was run it took 115 iterations for the algorithm to converge. The termination criterion was defined to be when the absolute error is less than 10^{-7} (that is, $\left\| X^{SOR} - X^M \right\|_2 < 10^{-7}$) then, convergence is achieved using the Richardson and Gauss-Seidel methods, respectively.

Table-I:-Comparison of simulation (MATLAB version 5.1) results using three methods for solving (31).

Method	**Absolute error** $\left\| x - x^M \right\|_2$	**Relative error** $\frac{\left\| x - x^M \right\|_2}{\left\| x^M \right\|_2}$	**Number of iterations required for convergence**
Successive overrelaxation ($\omega^\circ = 1.7215$)	2.1999×10^{-8}	4.0165×10^{-9}	115
Gauss-Seidel	5.3028×10^{-8}	9.6816×10^{-9}	1.00
Richardson	2.7619×10^{-8}	5.0426×10^{-9}	3.400
Jacobi	Diverged	Diverged	Diverged

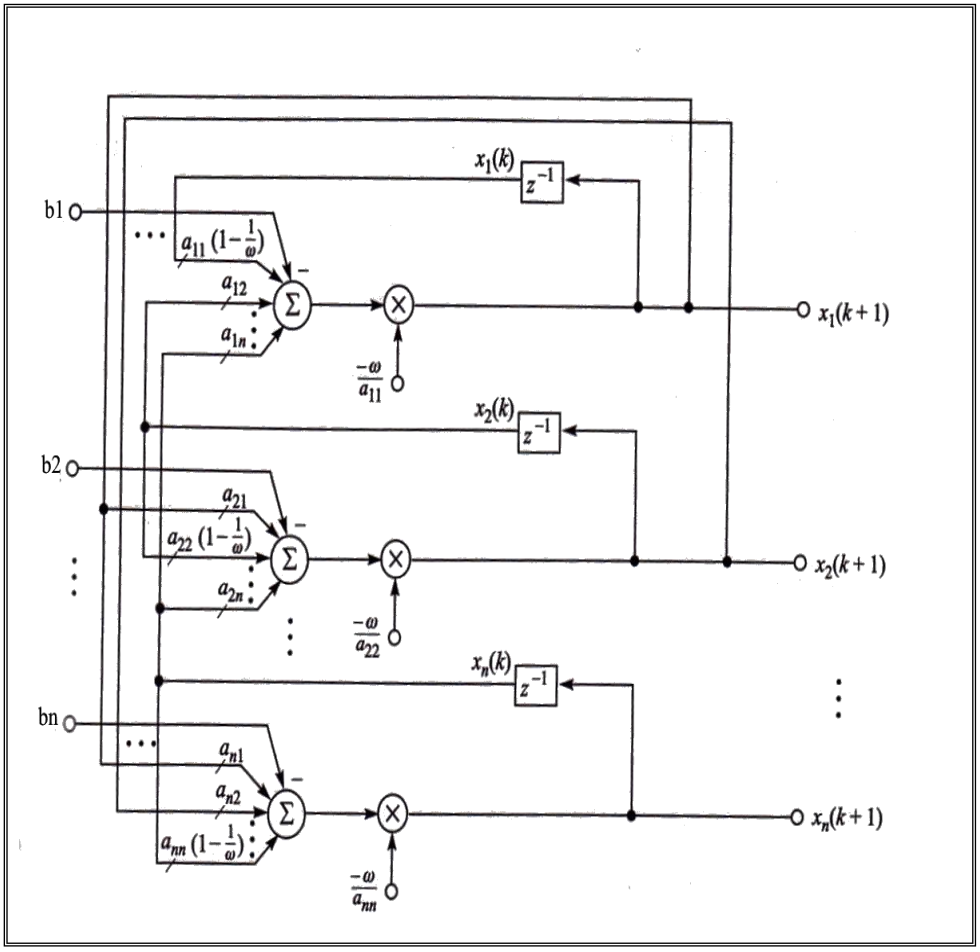

Figure (1): Shows a neural network architecture realization of the SOR iterative method.

Conclusion

- The same termination criterion was used for both these methods in order to properly compare all the results. From Table-I we see that SOR iterative method gives that best result, that is, the fastest convergence. Comparing the SOR results with the next-best results (Gauss-Seidel, $\omega = 1$); we see that the SOR method is about 10 times faster.
- The relaxation factor ω can be chosen to minimize $\sigma_r [\mathcal{L}_\omega]$ in order to make $x(k)$ converge to x as rapidly as possible, where $\sigma_r(\bullet)$ is the spectral radius.

References

- Akaike, H.,(1968): On the use of Linear Model for Identification of Feedback Systems, AISM,.
- A. Cichocki and R. Unbhauen,(1993): Neural Networks for Optimization and Signal Processing, New York: Wiley,.
- Barton, S.A.,(1991): A Matrix Method for Optimizing a Neural Network, Neural Computation,.
- D. M. Young,(1971): Iterative Solution of Large Linear Systems, New York: Academic,.
- Gill, P. E., W. Murray, and M. H. Wright,(1981): Practical Optimization, London: Academic,.
- G. H. Golub and C. F. Van Loan,(1996): Matrix Computations", 3rd ed., Baltimore, MD: Johans Hopkins University Press.
- Hertz, J., Krogh, A., and Palmer, R. G.,(1991): Introduction to The Theory of Neural Computation",Addison-Wesley, New York.
- K. E. Atkinson,(1989): An Introduction to Numerical Analysis,2nd ed.,New York:Wiley,.
- Luenberger, D. G.,(2003): Linear and Nonlinear Programming, 2nd ed., Springer.
- N. Higham,(1996): Accuracy and Stability of Numerical Algorithms, Philadelphia: Society for Industrial and Applied Mathematics.
- R. S. Varga,(1992):Matrix Iterative Analysis, Englewood Cliffs,NJ: Prentice-Hall,.

Journal of Kirkuk University –Scientific Studies , vol.2, No.3,2007

حل المعادلات الخطية باستخدام تقييم المصفوفة لطرق تقييم الوقت المتقطع بواسطة الشبكات العصبية

عيسى إبراهيم عيسى
كلية العلوم ــ جامعة كركوك

الخلاصة

أن المادة التي قدمت في هذا البحث هي من اساسيات (معمارية الشبكة العصبية) التي تكون حلا لـــــ(المعادلات الخطية بأستخدام المصفوفة لطرق تقسيم الوقت المتقطع بواسطة الشبكة العصبية)، وكما هو معروف فان الشبكة العصبية تتكون من العديد من العناصر المترابطة الداخلية(عقد أو عصبونات) ويمكن عرض جزءاً لهذه الشبكة العصبية اعتماداً على طور التدريب (المحدد لتدريب البيانات المستخدمة)،وتقسيم المصفوفة يحل بعدة معالجـــت أولية متعددة.ومن الضروري معالجة هذه الطرق من خلال استخلاص الخصائص المهمة للبيانات المســـتخدمة لتدريب الشبكة عليها بدلاً من أستخدام البيانات على هيئة "سطر". وهذه المعالجة الأولية لها يمكن أن تحســـن أداء الشبكة العصبية والتقارب المنجز بطريقتي(Richardson and Ganss Seidel)على التوالي بنفس معيـــار الانهـــاء المستخدم لكلا الطريقتين، وبمقارنة جميع النتائج ظهر أن طريقة (SOR)المكررة تعطـــي نتـــائج ذات تقـــارب أفضل وأسرع. وبمقارنة نتائج (SOR)مع (نتائج القادم – أفضل)لـــ Gauss-Seidel)$\omega=1$ (ترى ان طريقة الـــ(SOR)اسرع بعشر مرات.