Neural Network Learning Through Optimally Conditioned Quadratically Convergent Methods Requiring NO LINE SEARCH

Homayoon S.M. Beigi
T.J. Watson Research Center
International Business Machines
P.O. Box 704 /Room J2-N41
Yorktown Heights, New York 10598
E-Mail: *beigi@watson.ibm.com*

Keywords: Learning, Neural Networks, Quasi Newton Methods, OCR Unconstrained Optimization, Handwriting Recognition

Abstract

Neural Network Learning algorithms based on Conjugate Gradient Techniques and Quasi Newton Techniques such as Broyden, DFP, BFGS, and SSVM algorithms require exact or inexact line searches in order to satisfy their convergence criteria. Line searches are very costly and slow down the learning process. This paper will present new Neural Network learning algorithms based on Hoshino's weak line search technique and Davidon's Optimally Conditioned **line search free** technique. Also, a practical method of using these optimization algorithms is presented such that they will avoid getting trapped in local minima for the most part. The global minimization problem is a serious one when quadratically convergent techniques such as Quasi Newton methods are used. Furthermore, to display the performance of the proposed learning algorithms, the more practical algorithm based on Davidon's minimization technique is used in conjunction with a cursive handwriting recognition problem. For comparison with other algorithms, also a few small benchmark tests are conducted and reported.

1 Introduction

Quadratically Convergent Optimization Techniques have been applied to the problem of learning in neural networks to create learning techniques with superior speeds and precision compared to previously used first order methods such as the backpropagation technique. These methods have shown many orders of magnitude faster convergences and higher final accuracies. [1, 2, 3, 4] At their core, most of these methods basically use some kind of Conjugate Gradient or Quasi Newton technique. To ensure convergence, most quadratic learning techniques discussed in the literature require some sort of exact or inexact line search to be done. For each step in a line search, usually a set of patterns should be presented to the network. One measure of the speed of learning is the number of pattern presentations to the network to attain proper learning state.

Learning algorithms based on Conjugate Gradient Techniques and Quasi Newton Techniques such as Broyden, Davidon-Fletcher-Powell (DFP) and Projected Newton-Raphson require exact line searches to be done in order to satisfy theoretical convergence criteria and also to practically converge in many cases. [1] Exact line searches are very costly and slow down the learning process. Some Quasi Newton techniques get away with inexact line searches. Among these methods we could name Quasi Newton techniques based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update, Pearson's two updates, Greenstadt's two updates and Self-Scaling Variable Metric (SSVM) updates. [1, 2, 3] Some of these techniques such as SSVM and BFGS meet their promise of inexact line searches better than others. However, none of the above methods produce learning algorithms which do not require line searches.

Hoshino presented a Variable Metric (Quasi Newton) method which theoretically requires only inexact line searches to be done. [5] Davidon on the other hand introduced a Quasi Newton method which does not require any line search to be done most of the time. [6] Davidon's algorithm only needs a very crude line search to be done in some special cases which will be discussed later. The theory behind these two methods is discussed and some practical information is provided regarding the use of these methods.

Neural Network related objective functions normally possess many local minima by nature and getting trapped in these local minima usually makes the job of learning much more difficult. These new learning algorithms also feature answers to the problem of global optimization and avoiding the local minima through restructuring the network.

The Line Search Free Learning (LSFL) algorithm is then used in developing a new unconstrained handwriting recognizer which could be used for both on-line and off-line handwriting recognition. Results of accuracy of this recognizer are also presented. In addition, to evaluate the rate of convergence of the new learning technique, its performance is compared to the best reported results on small benchmark problems. [3]

The next few sections present the above topics under respective headings. Then the handwriting recognizer is described and finally results are shown and some conclusions are drawn regarding the practicality of the new proposed Neural Net Learning techniques.

For a complete description of the derivation of the minimization function generated from a general feedforward neural network architecture refer to [3]. In [3], a very handy notation is used to simplify the mapping between a neural network learning problem and unconstrained minimization.

2 Inexact Line Search Learning (ILSL)

As explained in [1, 2, 3], in any feedforward neural network, the problem of learning could be posed as the problem of minimizing the sum of the squares of errors generated from the differences between the outputs of the network and the desired values for those outputs. These squares of errors are summed at the outer layer over all output neurons and all patterns presented to the network. Let us denote this objective function of minimization by E. Quasi Newton techniques are quadratically convergent methods which use the gradient information of the objective function to evaluate some approximate of the inverse of the Hessian (matrix of second partial derivatives) of the function. This matrix is used to generate a Newton-like direction for minimization. The minimization variables are the weights connecting the outputs of neurons in one layer to the inputs of consequent layers and the activation function parameters of the neurons. Activation function parameters could be regarded as thresholds of these functions in neurons. In a Quasi Newton minimization technique usually two levels of iterations are done. One iteration level is to approximate the inverse of the Hessian at any iteration k, H_k. Once this approximation is made at each iteration, several iterations are done to minimize the objective function along the direction suggested by the inverse Hessian weighted gradient of the objective function. This minimization is the objective of a line search algorithm. Namely, that value of λ_k^i, λ_k^* is found which minimizes the objective function E_{k+1} through a change in the state variable

x from x_k to x_{k+1} along the Quasi Newton direction s_k given by (1). See (2).

$$s_k = -H_k g_k \tag{1}$$

$$x_{k+1} = x_k + \lambda_k^* s_k \tag{2}$$

To ensure convergence, most Quasi Newton techniques require some line search to be done. This is to ensure that the function will always decay at each iteration. Some methods such as the BFGS and SSVM techniques do not require an exact line search in practice. [1, 3] These methods require a line search that will reduce the objective function value by an amount dictated in their converge requirements. Hoshino in [5] presented a Variable Metric method with the following update for the inverse Hessian approximate,

$$H_{k+1} = H_k + \frac{1}{\Delta g_k^T \Delta x_k + \Delta g_k^T H_k}$$

$$(\eta_k \Delta x_k \Delta x_k^T - \Delta x_k \Delta g_k^T H_k - H_k \Delta g_k \Delta x_k^T - H_k \Delta g_k \Delta g_k^T H_k) \tag{3}$$

$$where,\ \eta_k = \left[1 + \frac{2\Delta g_k^T H_k \Delta g_k}{\Delta g_k^T \Delta x_k}\right]$$

An attractive feature of Hoshino's Quasi Newton minimization is his theoretical approach to the use of his update with inexact line searches. Inexact line searches are desired to reduce the number of function evaluations (number of presentations of the training set to the Neural Net). To evaluate the Quasi Newton direction of the update at any iteration $k+1$, Hoshino uses a modified gradient which is forced to be perpendicular to the search direction. Consider the line search terminating at x_{k+1}. Also, consider x'_{k+1} to be the true minimum in the direction of search. Furthermore, denote the step from x_k to the true minimum x'_{k+1} by $\Delta x'_k$ and define the scalar ϵ_k such that,

$$\Delta x'_k = \epsilon_k \Delta x_k$$

Therefore, the gradient at the true minimum, g'_{k+1} will be given by,

$$\begin{aligned} g'_{k+1} &= g_{k+1} + (g_{k+1} - g_k)\,\epsilon_k \\ &= g_{k+1} + \epsilon_k \Delta g_k \end{aligned} \tag{4}$$

The true minimum would be at the point where the gradient is perpendicular to the direction of search or,

$$\Delta x_k^T g'_{k+1} = 0 \tag{5}$$

Solving for the ϵ_k which satisfies condition (5) gives,

$$\epsilon_k = -\frac{\Delta x_k^T g_{k+1}}{\Delta x_k^T \Delta g_k}$$

Using this scalar factor, the expression for the modified gradient is given by,

$$g'_{k+1} = g_{k+1} - \frac{\Delta x_k^T g_{k+1}}{\Delta x_k^T \Delta g_k} \Delta g_k$$

This new gradient gives the following Quasi Newton direction at step $k+1$,

$$s_{k+1} = -H_{k+1} g'_{k+1}$$

Hoshino does not use this modified gradient for his update to the inverse Hessian approximate. This gradient is only used to obtain the Quasi Newton step. [5] also gives a stability analysis for this technique. Despite the special handling of inexact line searches, Hoshino's update has practically generated matrices with larger condition numbers than those created by the BFGS and SSVM methods. [1, 2, 3] This makes Hoshino's technique converge more slowly.

3 Line Search Free Learning (LSFL)

In 1975, Davidon [6] made an important contribution to the improvement of Quasi-Newton methods by introducing his optimally conditioned method which is to a certain sense free of line searches. Schnabel [7] has devoted most of his Ph.D. dissertation to evaluating Davidon's method. Updates generated by this method are optimally conditioned in the same sense as approached by Oren and Spedicato in their SSVM updates. [3] Historically, optimally conditioned updates were chosen such that the ratio of the condition number of H_{k+1} to that of H_k would be minimized. This would produce updates with invariance under orthogonal transformations. However, Davidon and SSVM updates are chosen such that they would minimize the condition number of $(H_k^{-1}H_{k+1})$ through minimizing the ratio of (λ_1/λ_N) in the eigen-value problem,

$$H_{k+1}u = \lambda H_k u$$

This conditioning produces updates which are invariant under all invertible linear transformations.

Equation (6) is a general Quasi Newton update which includes DFP and BFGS as its special cases. In (6), the value of θ_k is chosen such that the condition number of $(H_k^{-1}H_{k+1})$ is minimized. Davidon uses Δx_0 as the initial value of w_k and in the following iterations, w_k is updated using equation (7). However, due to the non-quadratic nature of Neural Net learning functions, $w_k = \Delta x_k$ is used in LSFL.

$$H_{k+1} = H_k + \frac{e_k w_k^T + w_k e_k^T}{w_k^T \Delta x_k} - \frac{e_k^T \Delta g_k w_k w_k^T}{\left(w_k^T \Delta g_k\right)^2}\theta_k \xi_k \xi_k^T \tag{6}$$

where,

$$e_k = \Delta x_k - H_k \Delta g_k$$

and,

$$\xi_k = \frac{e_k}{e_k^T \Delta g_k} - \frac{w_k}{w_k^T \Delta g_k}$$

$$w_{k+1} = w_k e_k^T \Delta g_k - e_k w_k^T \Delta g_k \tag{7}$$

Davidon's minimization algorithm has three very important features. The first feature is that in order to improve numerical stability and accuracy of the algorithm, a Jacobian matrix J_k is updated in the place of H_k. This matrix is the square root of H_k in the sense that,

$$H_k = J_k j_k^T$$

This is a common practice in numerical analysis. The condition number of the Jacobian J_k is of order of the square root of the condition number of matrix H_k. This smaller condition number improves numerical stability of the update, in practical applications. Also, an update in H_k given by,

$$H_{k+1} = \left(1 + uv^T\right) H_k \left(1 + vu^T\right)$$

in terms of the Jacobian is given by the update,

$$J_{k+1} = \left(1 + uv^T\right) J_k \tag{8}$$

In addition, update (8 requires fewer operations and thus produces less roundoff error. Some rank-two updates in the Broyden single parameter family, such as DFP, BFGS, and SSVM updates turn into rank-one updates in the Jacobian J_k.

The second feature of Davidon's algorithm stems from the fact that one could approximate the gradient at the minimum of a quadratic function by a linear interpolation similar to Hoshino's approach discussed in the previous section. [8] In this approach, the actual change of gradient and step size are not used. Instead, their projections are used to enable the algorithm to avoid doing any line searches. Therefore, line searches are avoided if a quadratic approximation of the objective function is acceptable. Otherwise, a simple line search is done to ensure sufficient reduction in the value of the objective function. Shanno and Phua have devoted a paper to the discussion of these projections. [9] Despite the use of these projections, Davidon's update still maintains a positive definite approximation to the inverse Hessian for both quadratic and non-quadratic functions. This fact ensures quadratic convergence of this algorithm. In real implementation, an equivalent of (6 is used which is given by (9).

$$H_{k+1} = H_k + \frac{e'_k \Delta x'^T_k + \Delta x'_k e'^T_k}{\Delta x'^T_k \Delta x'_k} - \frac{e'^T_k \Delta g'_k \Delta x'_k \Delta x'^T_k}{\left(\Delta x'^T_k \Delta g'_k\right)^2} \theta'_k \xi'_k \xi'^T_k \tag{9}$$

where,

$$\Delta x'_k = Q^T_k \Delta x_k,$$

$$\Delta g'_k = Q^T_k \Delta g_k,$$

$$e'_k = \Delta x'_k - H_k \Delta g'_k,$$

$$\xi'_k = \frac{e'_k}{e'^T_k \Delta g'_k} - \frac{w'_k}{w'^T_k \Delta g'_k},$$

and Q_k is the matrix projecting onto $[w^T_k(\Delta x_k - H_k \Delta g_k)]$ in H^{-1}_k.

The third feature of this algorithm is that the directions w of (6) need not be orthogonal to the error of the Quasi-Newton step. These directions are chosen such that the updates and new directions satisfy the following conditions:

$$\begin{aligned} &(H_{k+1} - H_k) z = 0 \forall z \in Z_H \\ &w \perp Z_H \perp \Delta x_k - H_k \Delta g_k \end{aligned} \tag{10}$$

where,

$$Z_H = [\Delta g_0, \ldots, \Delta g_{k-1}]$$

This ensures approximate inverse Hessian matrices which satisfy the following condition,

$$H_{k+1} \Delta g_j = \Delta x_j \qquad j \le k \tag{11}$$

This condition ensures convergence to the minimum of a quadratic function after N updates to the inverse Hessian approximate when inexact line searches (or even no line searches) are done.

Davidon's update, when no projections are used and the inverse Hessian approximate is updated directly, can be described by equations (12), (13) and (14).

$$H_{k+1} = \mu_k \left(H_k - \frac{H_k \Delta g_k \Delta g^T_k H_k}{\Delta g^T_k H_k \Delta g_k} + \theta_k v_k v^T_k \right) + \frac{\Delta x_k \Delta x^T_k}{\Delta x^T_k \Delta g_k} \tag{12}$$

$$a = \Delta g_k^T H_k \Delta g_k \qquad\qquad b = \Delta x_k^T \Delta g_k \tag{13}$$

$$\begin{aligned} \theta &= b\frac{(c-b)}{(ac-b^2)} \qquad & when\ b \leq \frac{2ac}{(a+c)} \\ \theta &= \frac{b}{(b-a)} \qquad & when\ b > \frac{2ac}{(a+c)} \end{aligned} \tag{14}$$

(12) is the most general forms of Quasi Newton update which encompasses Broyden's single parameter family, as well as SSVM, Hoshino and Davidon's update. [1, 2, 3]

4 Global Minimization

Local convergence has always been a major problem in Neural Network learning algorithms. In most cases, to avoid this, impractical applications, different initial conditions are used until one case would converge to the desirable global minimum. This is, however, not very practical and some supervision is necessary. Also, due to the high degrees of nonlinearity of Neural Net objective functions, picking optimal initial conditions are next to impossible. Classically, when training a Neural Net for a problem, the architecture of the network is fixed to some intuitively sound architecture with certain number of hidden units. Then, a learning algorithm is used with different initial conditions until convergence to a near global minimum is attained. Some researchers have also used global optimization techniques such as the statistical method of Simulated Annealing. [10] Simulated annealing, however, is known, due to its statistical nature, to be very costly in terms of number of presentations needed for convergence.

In the LSFL learning algorithm presented in this paper, the author has used a way of restructuring the network adaptively. In this method the network starts out with one hidden neuron. Then, the LSFL algorithm is applied to the network until a local minimum is obtained. Once a local minimum is reached, another neuron is added to the hidden layer and the weights of corresponding this new neuron are initialized to some random values. However, weights and activation function parameters corresponding to the original hidden neuron are not touched. Therefore, for the original hidden unit, the final state of the previous run is the new initial condition. The idea behind this theory is that new sets of dimensions are introduced by the introduction of this new neuron and the algorithm can once again start moving. This is with the assumption that the weights of the original neuron do not need to be changed much. Philosophically, in this way, the problem is being broken up and each hidden neuron handles a part of the problem. This methodology is then further applied each time a new local minimum is reached until there is no longer any improvement with the addition of new hidden neurons. At this point, it is assumed that either the algorithm has reached a global minimum or with the set of initial conditions we started, there is no more improvement possible. [1, 2, 3] have noted lots of cases when the global minimum was not reached given a fixed set of hidden units. Most of the cases discussed in those references have reached a global minimum by applying this new neuron addition technique. In addition, this method of finding the minimum reduces the amount of computation since in the beginning, a reduced network is used to do partial learning.

5 Handwriting Recognizer

Two different handwriting recognizers were developed using a feedforward neural network with the LSFL algorithm. The first was an unconstrained handwriting recognizer. By unconstrained handwriting we mean writing which consists of any combination of discretely written characters and connected writing. In other words, no constraint is imposed in the method of writing. This is the most difficult problem in handwriting recognition due to problems with segmentation. [11, 12] For this method a set of spatial features were extracted from the handwriting (a total of nine features per pattern presented to the network.) These features were associated with a fixed size vertical slice of the writing after it was normalized to a standard size. Each slice contains some overlap with its neighboring slices. Recognition tests were conducted on handwritten digits. Each digit, on the average, generated about three of these slices. The nine dimensional feature vector associated with each of these slices was passed to the network for training. At the output layer, there was one neuron per digit, namely ten output neurons were present. For each pattern presented at the time of training, the correct neuron in the output was asked to generate a 1 when the rest of the nine neurons were asked to generate a 0.

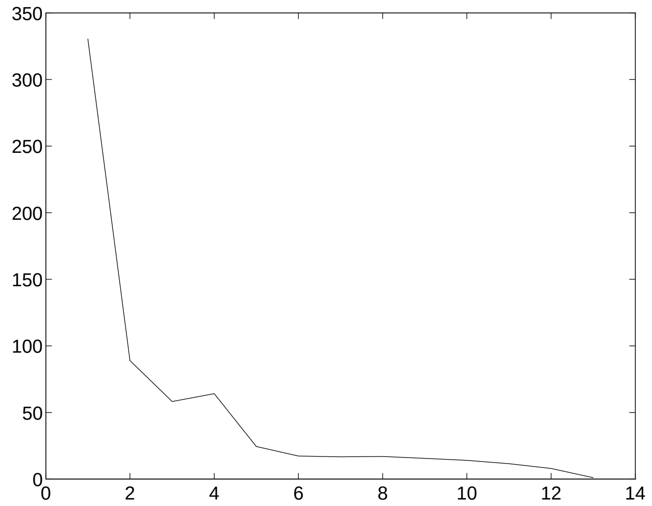

Figure 1: Sum of Squares of Output Error for Digit Recognition Training vs. Number of Neurons in the Hidden Layer + 1

The network was trained using the LSFL algorithm by being presented with five samples of each digit. It started with one neuron, as discussed in the previous section, and converged with a final number of 12 neurons in the hidden layer. Figure [1] shows a graph of the output error versus the number of hidden neurons plus one. The first error in the graph was obtained from the initial state of the network with one hidden neuron before learning stated. Later, a set of six samples of each digit were processed and run through the network. These samples were different from those samples used for training. A total accuracy of over 87% was obtained in this test. At the recognition stage, a reliability value was outputted by each output neuron associated with the ten different digits. These reliabilities were added over neighboring slices to evaluated the most likely path and eventually the intended digit string. For this purpose, in a truly unconstrained recognition system, any efficient search technique could be used. [12]

The second test was done on a set of discrete images of handwritten characters. The test was done on a limited basis just on the first six capital letters of the alphabet, $A - F$. A total of 4 neurons were necessary for final convergence to a zero output error. Due to insufficient data and time, only two samples of each character were used for this training. Figure [2] shows the output of the network versus the number of neurons used. This network was used to test two samples for each character and 100% accuracy was obtained. These numbers due to the lack of data and larger tests are not very

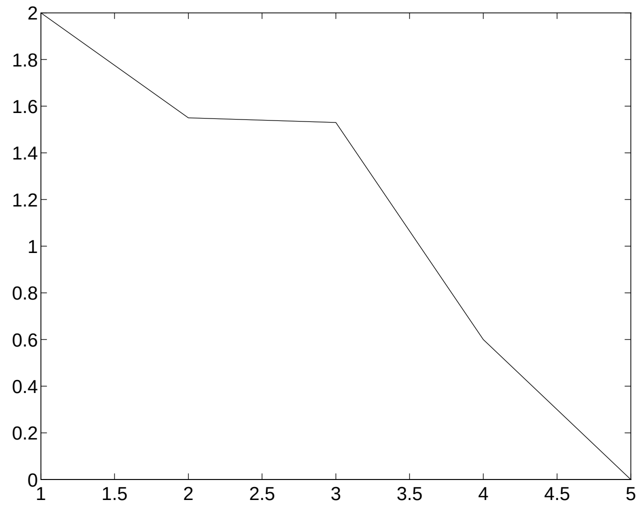

Figure 2: Sum of Squares of Output Error for Optical Character Recognition vs. Number of Neurons in the Hidden Layer + 1

reliable. However, they show the practicality of constructing such recognizers if more time were to be spent on picking better features and collecting data.

6 Other Results and Conclusion

Finally, to compare the performance of the LSFL algorithm with the best reported results in learning, two small benchmark test were conducted. The networks for these two tests had two and four hidden neurons respectively. Table of figure [3] shows the results of these tests conducted on the LSFL network and the results of the benchmark tests done on Steepest Descent, BFGS, and SSVM algorithms. [3] In the table E denotes the final error after convergence; P denotes the number of presentations of the training data to the network; G denotes the number of gradient evaluations; and F denotes the number of floating point operations done in the process of learning.

One could clearly see that for the Exclusive-OR (XOR) problem, there is not much gain by using the LSFL algorithm. However, by increasing the size of the problem to the Encoder problem, we could clearly see that avoiding excessive presentation of the training patterns due to the reduced number of line searches enhances the performance of the learning algorithm. In this case there were almost the same number of presentations as there were gradient evaluations (Inverse Hessian Updates). This shows that for much larger problems, the LSFL algorithm could save lots of computation. In addition, due to the special global minimization, lots of practical problems could take advantage of quadratically convergent algorithms without having to worry about getting trapped in local minima. In addition, since there is no real line search necessary for the LSFL algorithm, its application is much simpler. Finally, two very useful handwriting recognizers were developed which could be enhanced further and applied to more general recognition problems.

In conclusion we should note that since the ILSL method produces Inverse Hessian updates which have larger condition number than the already reported BFGS and SSVM methods, it is not considered in these tests. Also, after thorough examination of different techniques, the author recommends using the LSFL algorithm with Global Minimization handling as the best overall encountered learning algorithm for practical problems. The only disadvantage that this algorithm has, and this is a common problem in using all Quasi Newton based techniques, is the need for storing a huge matrix of the inverse Hessian approximate or its Jacobian.

		SD	BFGS	SSVM	LSFL
XOR	E	2.8e−5	4e−16	0	1.8e−3
	P	8553	15	11	17
	G	8465	6	4	9
	F	8.0e6	2.54e4	1.1e4	2.3e4
Encoder	E	8.0e−5	1.0e−17	3.7e−6	3.4e−3
	P	1352	36	33	11
	G	1194	11	10	9
	F	3.8e6	5.1e5	1.4e5	7.5e4

Figure 3: Comparison of LSFL vs. Best Reported Learning Technique on Benchmark Problems

References

[1] Homayoon S.M. Beigi and C. James Li, "A New Set of Learning Algorithms for Neural Networks," ISMM International Symposium, Computer Applications in Design, Simulation and Analysis, New Orleans, LA, March 1990, pp. 277-280.

[2] Homayoon S.M. Beigi and C. James Li, "Neural Network Learning Based on Quasi-Newton Methods with Initial Scaling of Inverse Hessian Approximate," The 1990 Long Island Student Conference on Neural Networks, Old Westbury, NY, April 21, 1990, pp. 49-52.

[3] Homayoon S.M. Beigi and C. James Li, "Learning Algorithms for Neural Networks Based on Quasi-Newton Methods with Self-Scaling," the ASME Journal of Dynamic Systems, Measurement, and Control, March 1993, pp.

[4] Homayoon S.M. Beigi and C. James Li, "New Neural Network Learning Based on Gradient-Free Optimization Methods," The 1990 Long Island Student Conference on Neural Networks, Old Westbury, NY, April 21, 1990, pp. 9-12.

[5] S. Hoshino, "A Formulation of Variable Metric Methods," J. Inst. Maths Applics, Vol. 10, 1972, pp. 394-403.

[6] W. C. Davidon, "Optimally Conditioned Optimization Algorithms Without Line Searches," Mathematical Programming, Vol. 9, 1975, pp. 1-30.

[7] Robert B. Schnabel, "Analyzing and Improving Quasi-Newton Methods for Unconstrained Optimization," Ph.D. Thesis, Cornell University, August 1977.

[8] M. J. Box, "A Comparison of Several Current Optimization Methods," The Computer Journal, Vol. 9, 1966, pp. 67-77.

[9] D. F. Shanno and K. H. Phua, "Numerical Comparison of Several Variable Metric Algorithms, MIS Technical Report 21, University of Arizona, 1977.

[10] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," Science, Vol. 220, 1983, pp. 671-680.

[11] T. Fujisaki, H.S.M. Beigi, C.C. Tappert, M. Ukelson, and C.G. Wolf, "On-line Recognition of Unconstrained Handprinting: a stroke-based system and its evaluation," From Pixels to Features III: Frontiers in Handwriting Recognition, S. Impedovo and J. C. Simon (eds.), Elsevier Publishers, New York, 1992, pp. 297-312.

[12] Tetsu Fujisaki, Krishna Nathan, Wongyu Cho, Homayoon Beigi, "On-line Unconstrained Handwriting Recognition by a Probabilistic Method," Pre-Proc. of IWFHRIII, Buffalo, New York, May 25-27, 1993, pp.235-241.