Faster Training Algorithms in Neural Network Based Approach For Handwritten Text Recognition

Haradhan Chel *h.chel@cit.ac.in*
Dept. of Electronics and Communication
CIT Kokrajhar, Assam, India

Aurpan Majumder *reach2am@gmail.com*
Dept. of Electronics and Communication
NIT Durgapur, West Bengal, India

Debashis Nandi *debashisn2@gmail.com*
Dept. of Information Technology,
NIT Durgapur, West Bengal, India

Abstract

Handwritten text and character recognition is a challenging task compared to recognition of handwritten numeral and computer printed text due to its large variety in nature. As practical pattern recognition problems uses bulk data and there is a one step self sufficient deterministic theory to resolve recognition problems by calculating inverse of Hessian Matrix and multiplication the inverse matrix it with first order local gradient vector. But in practical cases when neural network is large the inversing operation of the Hessian Matrix is not manageable and another condition must be satisfied the Hessian Matrix must be positive definite which may not be satishfied. In these cases some repetitive recursive models are taken. In several research work in past decade it was experienced that Neural Network based approach provides most reliable performance in handwritten character and text recognition but recognition performance depends upon some important factors like no of training samples, reliable features and no of features per character, training time, variety of handwriting etc. Important features from different types of handwriting are collected and are fed to the neural network for training. It is true that more no of features increases test efficiency but it takes longer time to converge the error curve. To reduce this training time effectively proper train algorithm should be chosen so that the system provides best train and test efficiency in least possible time that is to provide the system fastest intelligence. We have used several second order conjugate gradient algorithms for training of neural network. We have found that *Scaled Conjugate Gradient Algorithm*, a second order training algorithm as the fastest for training of neural network for our application. Training using SCG takes minimum time with excellent test efficiency. A scanned handwritten text is taken as input and character level segmentation is done. Some important and reliable features from each character are extracted and used as input to a neural network for training. When the error level reaches into a satisfactory level (10^{-12}) weights are accepted for testing a test script. Finally a lexicon matching algorithm solves the minor misclassification problems.

Keywords: Transition Feature, Sliding Window Amplitude Feature, Contour Feature, Scaled Conjugate Gradient.

1. INTRODUCTION

As the computing technologies and high speed computing processors has been developed, pattern recognition field got a wider dimension. Recognition of handwritten text is a real challenging task and research on which started from early sixties. Along with the up gradation of computational techniques researchers always tried to realize human perception and to implement into mathematical logic and designed different perception to train the computer system.

Researchers took wide varieties approaches [7],[9], [20] to recognize handwritten text. In most of the research works some common logical steps were used for recognition of handwritten text such as Segmentation, Feature Extraction and pattern classification technique. Among different classification techniques popular approaches are Neural Classifier [6], [14], [23], Hidden Markov Model [9], Fuzzy Classifier, or hybridized technique like Neuro-fuzzy, Neuro-GA, or some other statistical methods [25]. Neural Classifier has high discriminative power [1],[11],[12] for different patterns. Handwritten text contains wide varieties of styles in nature. It is really difficult to get real character level segmentation. A lot of research works has been done on segmentation of hand written text in last two decades. Improper segmentation results inaccurate feature extraction and poor recognition performance. Similarly reliable and only discriminative features give better recognition performance. If number of features is increased recognition performance increases but it increases computational complexities and leads to much longer time of training of the neural network. The most important task is to choose a proper training algorithm which train faster the network with large no of features and provide best recognition performance. Many of these algorithms are based on Gradient Descent Algorithm such as Back Propagation Algorithm [15], [18]. But all the algorithms are practically inapplicable in large scale systems because of its slow nature and its performance also depends on some user dependent parameters like learning rate and momentum constant. Some second order training algorithm such as Fleture Reeves [5], Polak Riebier[16],[19] or Powell-Beale Restarts algorithm[24] may be applied in appropriate applications. However *Scaled Conjugate Gradient* algorithm [22] results in much better recognition performance. By virtue of Scaled Conjugate Algorithm fastest training is obtained with almost 98 percent recognition efficiency. This paper not only shows the comparison of different second order training algorithm but also the reliable feature extraction and efficient lexicon matching technique. A small relevant description of Scaled Conjugate Algorithm is also given in later section. Different feature extraction schemes are also described in brief and finally the result of all experiments are shown.

2. IMAGE PREPARATION AND SEGMENTATION

A handwritten text written over A4 size paper is scanned through optical scanner and stored in bitmap image format. The image is then converted in to a binary image. One important point may be mentioned regarding image preparation is choosing the proper threshold of intensity level so that image does not get any discontinuity in any continuous part. Image is segmented [4],[8],[3] text level to word level and word level to character level segmentation all relevant information such as no of word, no of characters, no of lines are stored. Pixel coordinates of each words and characters in the bitmap image are stored very cautiously .The algorithm for segmentation used in image separation may be described below [20].

Algorithm 1:
Step1: The base lines such as upper, lower and middle base lines are calculated.

Step 2: Word is over segmented.

Step 3: The word segmentation points are modified using various rules.

Step 4: Undesired and incorrect segmentation points are removed.

Step 5: The correct segmentation points are used for final segmentation of words.

Step 6: Position of characters and words in the script are stored.

3. FEATURE EXTRACTION

It was mentioned in the introduction that reliable and meaningful features increase the recognition performance. Reliable feature are those features which produce almost same numeric value irrespective of slight positional variations in different sample images of same character. In this experiment four kinds of features extraction schemes are used and a total 100 nos. of features were collected per character. Before feature extraction all the segmented character is resized [17] in a fixed size (60 x40). One runtime separated unbounded character image, bounded image and resized image is shown in figure 1. Short description of all four kinds of features is described in the following subsections.

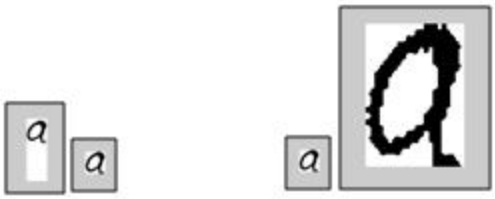

FIGURE 1: Unbounded Image, Bounded Image and Resized Image.

3.1 Transition Feature

Assume that the character image has a fixed dimension of X rows and Y columns. An M x N grid is superimposed over the image as shown in figure 2. The grid is prepared in such a way that all start and end values of each row and column got an integer value. The start values of row m and column n may be written as under.

$$x_1 = y_1 = 1 \tag{1}$$

$$x_m = Int\left[\frac{(m-1)X}{M-1}\right] \qquad m = 2,3 \ldots M \tag{2}$$

$$y_n = Int\left[\frac{(n-1)Y}{N-1}\right] \qquad n = 2,3 \ldots N \tag{3}$$

$Int\,[x]$ refers to nearest integer value to x. Assume the intensity of coordinate(x,y) of the X x Y image is A(x,y). The original image is scanned along every row and column and the gradient information along each row and columns are collected.

$$\Delta(x_m, y) = \mathrm{A}(x_m, y+1) - A(x_m, y) \tag{4}$$
$$where \quad y = 1,2 \ldots \ldots . \quad Y-1$$

And

$$\Delta(\mathrm{x}, \mathrm{y_n}) = A(x+1, y_n) - A(x, y_n) \tag{5}$$

$$where \quad x = 1,2 \ldots \ldots . \quad X-1$$

As the image is a binary image both $\Delta(x_m, y)$ $\Delta(\mathrm{x}, \mathrm{y_n})$ return one of the three values 0, 1 or -1. 0 indicate no transition 1 means white to black transition and -1 indicate transition from black to white. A 5x5 grid over character image is shown on figure 2.a and transitions are also shown in figure 2.b. In this case we would consider only -1 value and in each row and columns of the M x N grid total number of transitions in every row and columns are important features of the image. As it was mentioned earlier that the approach of the experiment is neural network based and the immediate operation after feature extraction is training, it is found experimentally that neural network works reliably and faster if the input feature values are within 0 to 1 limit. As in this case all values are more than 1, they are normalized between 0 and 1 by a nonlinear transfer function as mentioned below.

$$f(x) = \frac{1}{1 + e^{-kx}} \quad \ldots\ldots\ldots\ldots\ldots 0 \leq x \leq 5 \tag{6}$$

k is a constant and values may be chosen between .3 to 1 for better result. The upper limit of x is taken as 5 because maximum possible white to black transition in handwritten characters found 5. In this way we found (M+N) nos. feature for each handwritten character. At the time of scanning the original image through all M rows and N columns using M x N grid and the coordinates of first and last transitions in each rows and columns are stored and a new kind of transition features were collected. Suppose in m^{th} row first and last transition occurs at (x_m, y_f) and (x_m, y_l) then two features can be collected from that row as mentioned below.

$$F_1 = \frac{y_f}{Y} \tag{7}$$

$$F_2 = \frac{Y - y_l}{Y} \tag{8}$$

Similarly if in n^{th} column of the M x N grid first and last transition occurs at (x_f, y_n) and (x_l, y_n)then two features can be collected from that column as mentioned below.

$$F_3 = \frac{x_f}{X} \tag{9}$$

$$F_4 = \frac{X - x_l}{X} \tag{10}$$

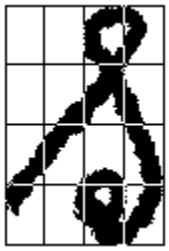

FIGURE 2.a: 5 x 5 Grid Over Original Image.

FIGURE 2.b: 5 x 5 Grid Over Grid Image.

3.2 Sliding Window Amplitude Feature

In this feature extraction scheme the image is subdivided into some overlapping windows such that half of the window breadth and width are overlapped with previous and next window along row and column wise except the windows at boundaries and corners. The windows at left boundary have no windows left at left side that it will overlap. Similarly top, bottom and right boundary side windows do not overlap with any windows beyond their boundary. The corner block has only two blocks to overlap for example the top left corner has only right and down windows to overlap. Black pixel density in each block or window is calculated. The original image of handwritten character has a fixed size of X rows and Y columns. As shown in figure (2.a) the image is superimposed with M x N grid. The four corners of the rectangular window R(m,n) are $[(x_{m,}y_{n)}\ (x_m,\ y_{n+2)}\ (x_{m+2,}\ y_{n)}\ (x_{m+2,}\ y_{n+2})]$ .Values of x_m and y_n is defined in eqn. 1-3. Each block produces one feature and the feature is normalized between 0 and 1 and a total (M-2) (N-2) features are collected with normalized feature values [13]. In our experiment we have taken both M and N as 8 and a total 36 features were collected per character.

$$f = \frac{no\ of\ black\ pixel\ in\ the\ block}{total\ no\ of\ pixel\ in\ the\ block}$$

3.3 Contour Feature

The most important feature extraction scheme is contour feature [2][3]. Image is scanned and filtered. The filtered image is superimposed with an 11 X 1 grid as shown in figure (3.a) to find the external boundary intersection points of the character and total 22 such points are collected. All the 22 points are numbered in a circular fashion that from the top to bottom as 1 to 11 and from bottom to top as 12 to 22. All points are connected with the next point to draw an exterior boundary contour as shown in figure (3.b). For example the n_{th} line is bound between two coordinate (x_n,y_n) and (x_{n+1}, x_{n+1}).. The alignment of all the boundary lines is unique property of that image and it differs from character to character. The two unique parameters of a straight line that is the gradient and a constant value which solely depend on the coordinates of the two points are taken as feature normalizing by a nonlinear hyperbolic function. Equation of a straight line and its representation by two points may be mentioned by equation 12-14. Suppose a straight line in equation is bounded between two points (x_1, y_1) and (x_2, y_2). The m is the gradient and c is a constant parameter of the straight line. Both m and c are two unique parameters which contain the boundary patterns of the characters.

$$y = mx + c \tag{12}$$

$$m = \frac{y_1 - y_2}{x_1 - x_2} \tag{13}$$

$$c = \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1} \tag{14}$$

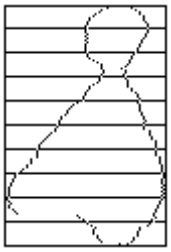

FIGURE 3.a: 11 X 1 grid superimposed over boundary image.

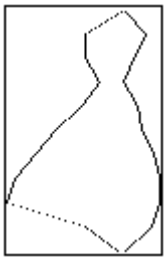

FIGURE 3.b: Contour of image represented with straight line.

Both the m and c may have any positive and negative values between negative infinity and positive infinity. As the immediate process after feature extraction is training of neural network with feature matrix all the feature values should be normalized between 0 and 1 as it was shown by experiment that neural network shows better response if the inputs are in range within 0 to 1 limit. For normalizing all the values of gradient (m) and constant (c) a hyperbolic nonlinear function is used as transfer function which is shown in equation 15 and a graphical representation is also shown in figure (4).In equation 15 any value of x maps the output between 0 and 1.k is a constant and value of it depends upon the no of features extracted. For 22 nos. of contour feature a value range from k=.3 to .5 is selected.

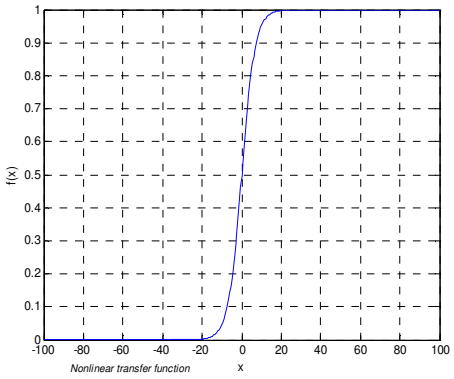

FIGURE 4: Nonlinear Hyperbolic Transfer Function.

$$f(x) = \frac{1}{1 + e^{-kx}} \tag{15}$$

If number of feature is increased its value should be decreased and similarly if number of features is decreased its value should be increased. In figure 4 plots for x versus f(x) is shown for an x range from -100 to 100. In this way 22 nos. features are collected and finally put in to the feature matrix.

3.4 Shadow Feature

In this feature extraction scheme the resized image is segmented in to four segments as shown in figure (5). This feature is similar feature like transition feature as described in section 3.1. The term 'Shadow' is taken symbolically from the concept of formation of shadow when light falls over an object. When light is fallen from the upper side of the subsection it creates three shadows one is over the ground and other two are over the two inclined side. The phenomenon is shown in figure (6). When light is projected at a perpendicular dirction to the horizontal line over the object shown in segment -1 shadow is formed over two sides AC and BC of the triangle and shadow also falls in the ground. The length of the shadows in side AC and BC are DC and EC respectively. The length of the shadow fallen over the ground is D'E' which is basically projection of point D and E over the ground. The features may be defined from these parameters in the following way.

$$Shadow\ feature = \frac{Length\ of\ the\ shadow\ over\ the\ side}{length\ of\ the\ side}$$

Using the above definition following three nos. of features can be extracted from each segment.

$$SF(1) = DC/AC \qquad SF(2) = EC/BC \qquad and \quad SF(3) = D'E'/AB$$

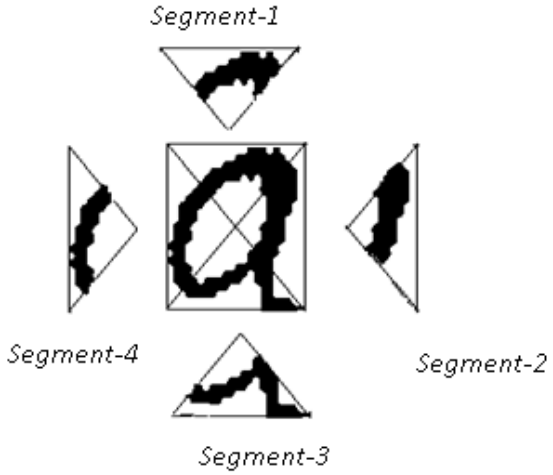

FIGURE 5: Four Subsections of the Resized Image

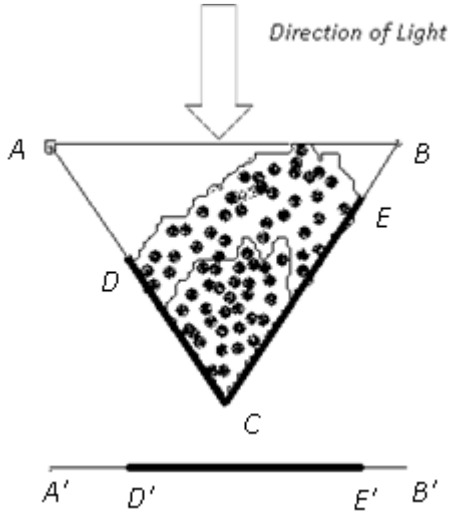

FIGURE 6: Shadow at Ground and Over Two Inclined Sides.

In the similar way features are extracted from all four segments and a total 12 nos. of features are added in the feature matrix for each character.

In our experiment four kinds of features extraction schemes are applied and we have taken 30 transition features, 36 sliding window amplitude features, 22 contour features and 12 shadow feature and a total 100 numbers of features are collected. Content of the training script are 10 sets of capital no(i.e from A-Z) and 10 sets of small letters (i.e from a-z) which are processed, filtered and resized as described in the previous sections. Finally the feature matrix having dimension 520x 100 is fed to the neural network for training. The description of the neural network and the algorithms used for training are described in the next section.

4. TRAINING OF NEURAL NETWORK

Training and design of neural network is the most important part in this experiment. Here we have used single and multilayer feed forward network with different first and second order back propagation training algorithm. It is not surprising that increase in no of feature improves

recognition efficiency. But in opposite side it takes a longer time to train the neural network. In our experiment we used four kinds of feature extraction scheme. Name of the feature extraction scheme and no of features of each type are mentioned in the previous section. But theses large nos. of features takes very longer time to train the network if a proper training algorithm is not chosen. We have used different conjugate gradient methods for training and the speed of training improved significantly. All the algorithms showed better performance in respect to the speed of the training and recognition performance of the Neural Network are second order conjugate direction method such as Fleture-Reeves [15], [21], Powell Beale [24], Polak Ribiere [16],[19] and Scaled Conjugate Gradient algorithm by Moller [22] .A comparative study also been given among the above algorithms and a comparison of converging performance of all the second order conjugate gradient methods is presented in result section. It has also been found experimentally that among the entire training algorithms Powell Beale method and Scaled Conjugate Gradient methods have shown better performance in respect of quick convergence of the error curve. Using *Scaled Conjugate Gradient* algorithm [22] the Hessian matrix of error equation is always positive over all iterations. But in all other algorithms mentioned above it is uncertain. This property of SCG algorithm increases learning speed reliably in successive iteration. Let us we define an error function Taylor Series expansion.

$$E(\widetilde{w} + \Delta\widetilde{w}) = \mathrm{E}(\widetilde{w}) + \mathrm{E}'(\widetilde{w})^{\mathrm{T}}\Delta\widetilde{w} + \Delta\widetilde{w}^{\mathrm{T}}\mathrm{E}''(\widetilde{w})\Delta\widetilde{w} + \cdots \quad (16)$$

Suppose the notations are used as $A = E'(\widetilde{w})$ and $H = E''(\widetilde{w})$. Weight vector in n^{th} iteration may be mentioned as $\widetilde{w}_n$. H is the Hessian matrix and A is the local gradient vector.
We will use a second order approximated error equation for further calculations and the same equation may be as follows

$$E(\widetilde{w} + \Delta\widetilde{w}) = \mathrm{E}(\widetilde{w}) + \mathrm{E}'(\widetilde{w})^{\mathrm{T}}\Delta\widetilde{w} + \Delta\mathrm{w}^{\mathrm{T}}\mathrm{E}''(\widetilde{w})\Delta\widetilde{w} \quad (17)$$

The solution of the quadratic difference equation may be found as follows [15].

$$\Delta\widetilde{w} = \mathrm{H}^{-1}\mathrm{A} \quad (18)$$

The above solution can be achieved subject to condition that H is positive definite matrix. The above equation states that how much amount of shifts is required for all the weights so that the error curve converges significantly. The above equation (18) is the essence of Newtown's Method [15] and from the equation it is found that the equation can be converged in one step if the inverse of the Hessian matrix is calculated. But in practical situation this phenomenon does not occur because of the some constraints as mentioned below.

a) When number of Weights is more calculation of inverse of the Hessian Matrix is computationally expansive and highly time consuming matter.

b) There is no guarantee that the inverse of the Hessian Matrix will be positive definite.

c) The system converges in one step if and only if the error equation is perfectly quadratic in nature. But generally all error equation has some higher order terms.

To avoid the above limitations the only way was found to achieve the solution through an iterative process. Suppose a set of nonzero vectors $p_1, p_2, p_3 \ldots\ldots p_N$ are basis vectors in R^N and are H conjugate. If H is a positive definite matrix then the following conditioned must be satisfied [10].

$$\tilde{p}_k^T H\, \tilde{p}_i = 0 \qquad for\ all\ k\ and\ i\ except\ k = i$$

$$\tilde{p}_k^T H\, \tilde{p}_i > 0 \qquad for\ k = i \quad (19)$$

Let $x_n = \tilde{p}_n^T H\, \tilde{p}_i$. When k= i= n then x_n is nonzero quantity.

$$x_n = \tilde{p}_n^T H\, \tilde{p}_n \quad (20)$$

$$= \tilde{p}_n^T \tilde{s}_n$$

Here $\tilde{s}_n = H\,\tilde{p}_n = E''(\tilde{w}_n)\,\tilde{p}_n$. The idea to estimate the term s_n with a non symmetric approximation [22] it may be written as:

$$\tilde{s}_n = \frac{E'(\tilde{w_n}+\sigma_n\tilde{p_n})-E'(\tilde{w_n})}{\sigma_n} \qquad \text{Where } 0 < \sigma_n \ll 1 \tag{21}$$

The above equation is the essence of second order error estimation used by Hestene [10] in conjugate gradient method. Scaled Conjugate Gradient method is a slightly different concept applied over it. In eqn.(20) the sign of x_n may be negative or positive but by definition if H is a positive definite matrix then x_n must be a greater than zero. Hestene [10] combines the concept of introducing a dumping scalar value and modifies the equation as in eqn. 22. The other steps are same to the conjugate direction method. The function of the scalar parameter λ is to compensate

$$s_n = \frac{E'(\tilde{w_n}+\sigma_n\tilde{p_n})-E'(\tilde{w_n})}{\sigma_n} + \lambda_n\tilde{p_n} \text{ Where } 0 < \sigma_n \ll 1 \tag{22}$$

the indefiniteness of value of $E''(w_k)$ when it is negative. In every iteration sign of x_n is checked i.e. whether $x_n > 0$ or $x_n < 0$. When x_n is less than zero the value of λ_n is increased and similarly when x_n is greater than zero the value of λ_n is decreased. All other steps are similar to the conjugate direction method.

Two up gradation equation governs the whole process. Firstly the weight up gradation equation (eqn. 23) and secondly the basis vector up gradation equation (eqn. 24). The target is to find such a solution set of weight vector that H become positive definite. As the iterations go forward the first order error gradient $E'(w_n)$ decreases and at last it reaches very near to zero. The equations are as follows

$$\tilde{w}_{n+1} = \tilde{w}_n + \alpha_n\tilde{p}_n \tag{23}$$

$$\tilde{p}_{n+1} = \tilde{r}_n + \beta_n\tilde{p}_n \tag{24}$$

Where $r_n = -E'(w_n)$, α_n and β_n may be calculated in each iteration by the following equations.

$$\alpha_n = \frac{\tilde{p}_n^T\tilde{r}_n}{\tilde{p}_n^T H\tilde{p}_n} = \frac{\tilde{p}_n^T\tilde{r}_n}{x_n} \tag{25}$$

And $$\beta_n = \frac{\langle\tilde{r}_{n+1},\tilde{r}_{n+1},\rangle - \langle\tilde{r}_{n+1},\tilde{r}_n\rangle}{\tilde{p}_n^T H\tilde{p}_n} \tag{26}$$

The most important matter is to discuss that how the value of λ in eqn. 22 is chosen. As described earlier that, the function of λ_n is to check the sign of x_n in each iteration and to set a proper value of λ_n so that $\tilde{x}_n$ get a positive value.

Let it is found that $x_n \leq 0$ and λ_n is raised by $\overline{\lambda_n}$ so that s_n get a new value

$$\tilde{\tilde{s}}_n = \tilde{s}_n + (\overline{\lambda_n} - \lambda_n)\tilde{p}_n \tag{27}$$

$$\widetilde{x_n} = \tilde{p}_n^T\tilde{\tilde{s}}_n \tag{28}$$

Putting the value of $\tilde{\tilde{s}}_n$ from eqn. no 27 in eqn. 28

$$\widetilde{x_n} = x_n + (\overline{\lambda_n} - \lambda_n)|\tilde{p}_n|^2 > 0 \tag{29}$$

$$\Rightarrow \overline{\lambda_n} > \lambda_n - \frac{x_n}{|\tilde{p}_n|^2} \quad (30)$$

From eqn. 29 some guide line is found that the new λ_n should be greater than by -$\frac{x_n}{|\tilde{p}_n|^2}$ but no such particular value can be obtained that so that the optimal solution can be obtained. However we have used

$$\overline{\lambda_n} = 3\left(\lambda_n - \frac{x_n}{|\tilde{p}_n|^2}\right) \quad (31)$$

The above assumption is put into eqn. 29 we get

$$\widetilde{x_n} = -2x_n + 3\lambda_n|\tilde{p}_n|^2 \quad (32)$$

Combining eqn. 32 and 25 we found the modified value of α_n

$$\alpha_n = \frac{\tilde{p}_n^T \tilde{r}_n}{-2x_n + 3\lambda_n|\tilde{p}_n|^2} \quad (33)$$

the above equation clears that with the increase of the value of λ_n decrease the height of step size and decrease in λ_n increases the height of step size which agrees to all the assumptions made earlier.

In the above steps it was realized that how the Hessian Matrix remain positive in all iterations using a scalar parameter λ. But the second order approximation of error equation which was used during entire calculation cannot assure for the best performance though we get positive definite Hessian matrix over all iterations. The above calculation steps assure that the error curve will be converging towards minima but in some situation choosing a proper value of λ is required otherwise the rate of convergence become slow. So a proper mechanism for scaling λ is adopted. A parameter ρ is defined which measure in what extent the second approximation of the error curve matches the original error curve. The following equation defines ρ as

$$\rho_n = \frac{E(\tilde{w}_n) - E(\tilde{w}_n + \alpha_n\tilde{p}_n)}{E(\tilde{w}_n) - E_{2q}(\alpha_n\tilde{p}_n)} \quad (34)$$

Here ρ_n measures how finely the second order approximated error equation $E_{2q}(\alpha_n\tilde{p}_n)$ matches to the $E(\tilde{w}_n + \alpha_n\tilde{p}_n)$. The above equation tells than the value of λ nearer to 1 means better the approximation. For faster convergence λ is scaled by the following equations.

$$\lambda_n = \frac{1}{3}\lambda_n \qquad if\ \rho_n > .75$$

$$\lambda_n = \lambda_n + \frac{x_n(1 - \rho_n)}{|p_n{}^2|} \qquad if\ \rho_n < .25 \quad (35)$$

In our experiments the initial values was chosen as λ_1=10^{-3} and $\sigma = 10^{-5}$ was taken and in later steps $\sigma_n = \frac{\sigma}{|\tilde{p}_n|^2}$ was assumed. Initially weights are chosen as random nos. between 0 and 1. At starting of training in first iteration $p_1 = r_1 = -E'(\tilde{w}_1)$ is assumed. The algorithm may be summarized as below.

a) Initialization of parameters like σ,p,r in first iteration.

b) Calculation of second order parameters like s,x and σ

c) Check whether Hessian matrix H is positive definite or not

d) If false adjust the value of s by increasing λ and recalculate s again.

e) Calculate ρ and readjust the value of λ

f) Calculate step size

g) If error > minimum error limit go to next iteration.

h) Accept the weight vector for test.

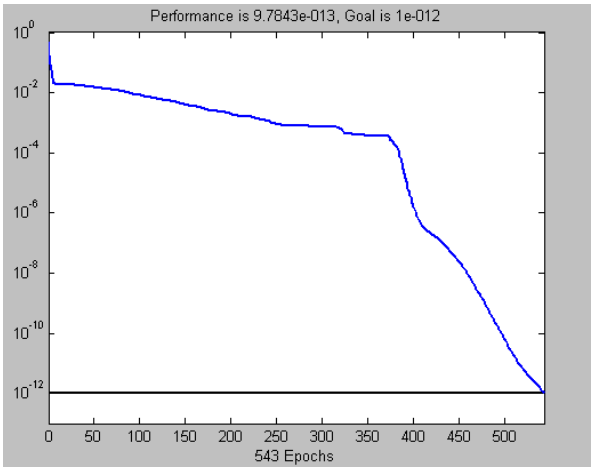

FIGURE 7: epoch versus error curve.

During the training operation the epoch versus error data is collected and the nature of the convergence is also noted. A run time error curve is plotted in figure 7 which shows that the nature of convergence is very fast and it takes just one or two min for completion of training and error limit reaches in 10^{-12} range within only 543 epoch.

5. RESULT AND DISCUSSION

This experiment is an extension of [26] where three types of feature extraction schemes were applied. Here a new kind of feature extraction scheme 'Shadow Feature' was implemented in section 3.4. It accelerated both the training performance and recognition performance. Though the maximum recognition performance did not changed but most of the time it showed better performance. The results are shown in Table 1 and Table 2. The experiment for recognition of handwritten text was conducted in MATLAB environment where two types of images are used as input i.e. train image and test image. The purpose of the experiment was to recognize individual's handwriting by a neural network which is trained to identify the patterns of handwriting of the same person. Here both the train and test script was written by same person and the texts are casually written over the script. For this reason this experiment was not conducted and compared by any standard data base like CEDER or IAM data base of handwritten text. Two major features are highlighted in this experiment which differs it from all other researches [6],[13],[23] done earlier may be mentioned that is i) the superfast training speed ii) high recognition performance. Ten sets of handwritten capital and small letters of English alphabets are taken as training script and various handwritten texts written by same person are used as test script. A sample train and test scripts are shown in figure 8 and figure 9. All the characters are written in natural way over a sheet of A4 size. In both training and testing stage common character level segmentation and feature extraction is done over the train and test script. Training characters are English alphabets so there is no concept of forming words but test script is a text or may be called as group of words without any special symbol like ;, : and ?. In case of test script the segmentation is done from text level to word level and later from word level to character level. The start and end locations of each word in the text are stored. After recognition of all the characters in test script the computer printed words are regrouped by the NN output characters using the start and end location information of each words. Character level and Word level efficiency may be defined as follows.

Character Level Efficiency = (No of characters recognized correctly)/(Total no of characters in the script)

Word Level Efficiency = (No of words recognized correctly)/(Total no of words in the script)

One character misclassification in a word decreases the word level recognition performance. But it gets easily be improved if the same word is checked through a dictionary for validity of the word. If the word exists in the dictionary it returns the same word otherwise nearest word of same length is returned. For implementing this one dictionary is formed in MATLAB as an m-file which contains more than two thousand common words

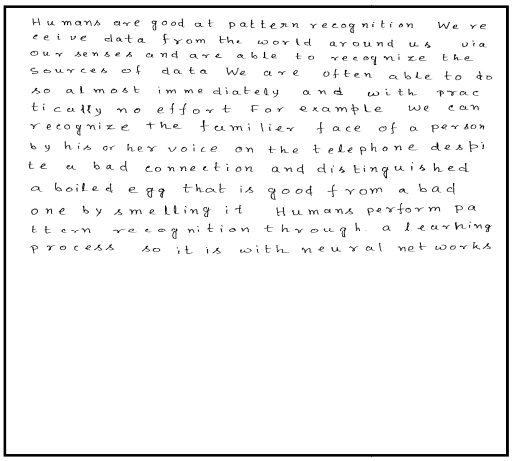

FIGURE 8: Train Script.

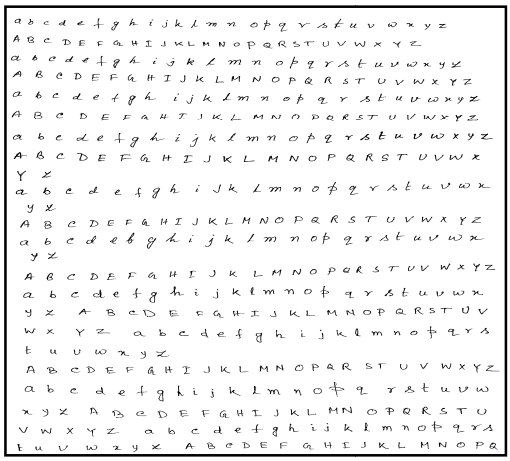

FIGURE 9: Test Script.

No of Features / character	NN structure	Training algorithm	No of epochs	Train efficiency	Error limit
100	100,120,52	SCG	543	100%	10^{-12}
100	100,120,52	Powell-Beale	1412	100%	10^{-12}
100	100,120,52	Fleture Reeves	4322	100%	10^{-12}

TABLE 1: Comparison Between Different Training Algorithms.

taken from a standard oxford dictionary. In this paper various algorithms were used for training such as a) Scaled Conjugate direction Method b) Powell Beale method and c) Fleture-Reeves method among the three algorithm Scaled conjugate algorithm shown the best training speed. A comparative statement is described in table.1.the above table shows that total 100 nos. of features were taken and NN structure indicates 52 nos. of output neurons for 52 characters (a-z and A-Z) and the result was obtained for 120 nos. of hidden neurons. Both the three algorithms provide good convergence of the error curve but Scaled Conjugate Gradient algorithm provides fastest training speed and the training completes within only 543 epoch which takes less than one minute for reaching 100 percent train efficiency. The neural network trained with ten sets of handwritten alphabets that is total 520 nos. of characters when asked to make response to test feature matrix its response is really satisfactory only a few misclassification arises for very similar type of characters and for very bad handwritings. A character level recognition performance is shown in table 2 which shows that using double hidden layer with 150 and 100 hidden neurons we get maximum 83.60 percent case sensitive character level recognition and a maximum 92.32 percent case insensitive character level recognition performance is achieved. Though the degree of recognition is good but it may further be improved by lexicon matching technique which was discussed earlier. Some other researchers have reported like in [27] for 27 classes ouput recognition performance was 82 percent and 94 percent for lowercase and uppercase characters. In [12] a different type of feature extraction

scheme was applied and which gave 86 percent and 84 percent for lowercase and uppercase characters. In [13] a neural network based segmentation method has been presented and a global feature extraction scheme has been applied which shows character segmentation with a performance in the rage of 56.11 percent for case sensitive and 58.50 percent which was further improved by a lexicon matching method and result was found as 85.71 percent. Compared to all the mentioned works this approach has given better result from three different aspects i) Very less number of features per characters was taken and it showed better result ii) It gives higher speed as scaled conjugate algorithm has been used for neural network training and iii) better recognition performance compared to the referred [12] [13] [27] works. Word label performance after lexicon matching is shown in table 3. This shows that the case insensitive performance shown in table 3 improves by maximum 99.02 percent and because some of the word which contain lesser misclassified characters get corrected by the lexicon matching technique which increases word level.

NN structure	MSE obtained	Train efficiency %	Test efficiency (Character Level) %	
			Case sensitive	Case insensitive
100,160,52	$9.63x10^{-13}$	100	81.43	89.42
100,200,52	$9.75x10^{-13}$	100	82.64	89.42
100,250,52	$9.32x10^{-13}$	100	79.25	88.70
100,120,80,52	$9.44x10^{-13}$	100	74.17	89.42
100,150,100,52	$9.69x10^{-13}$	100	83.60	92.32
100,200,100,52	$9.98x10^{-13}$	100	83.60	92.32

TABLE 2: Character Level Classification Performance.

NN Structure	Without Lexicon Matching		After Lexicon Matching	
	Character Level (%)	Word Level (%)	Character Level (%)	Word Level (%)
100,160,52	89.17	62.87	98.43	96.74
100,200,52	89.42	48.65	99.02	97.83
100,250,52	88.70	47.56	99.02	97.83
100,120,80,52	87.34	38.96	98.43	95.65
100,150,100,52	92.32	66.13	98.43	98.91
100,200,100,52	89.72	47.56	99.02	98.91

TABLE 3: Performance Before and After Lexicon Matching.

recognition and when word get corrected the misclassified characters are also changes to its correct version. As a whole both the word and character level performance increases. In this experiment maximum word level and character level recognition performance achieved as 98.91 and 99.02 percent respectively. Though the result of the experiment was very well and most of the words and characters were recognized correctly except a few misclassifications that were found during experiment which may be shown in table 4.

6. CONCLUSION

Reliable feature extraction methods are shown which is most important in Neural based approach for pattern recognition. When first order standard back propagation algorithms fails to produce result in a bulky neural network in a limited time frame, second order training algorithm work surprisingly. In this paper we focused both the training speed and recognition performance of handwritten alphabet based text. When no. of features are higher and no. of output classes are 52 all first order training algorithms basically fails or generates very poor result in training. Basic reason is the slower convergence of error curve and proper second order training algorithm become suitable replacement of those algorithms. Using this algorithm, Hessian matrix of error equation always remain positive definite and in every iteration the error curve converges in a faster way. In this paper comparison

Original	Recognized	Original	Recognized
H	M ,n	n	m
r	v, Y	Y	r,y
e	c,l,C	y	r,Y
a	Q, u, l ,G	c	C
f	t	C	c
s	a,b,	O	o
l	x	o	O,b,D
l	j	j	i
b	o	t	f,F,q

TABLE 4: General Misclassifications.

between several second order training algorithms has been shown and it was found experimentally that Scaled Conjugate Gradient algorithms works with fastest speed and recognition performance is also excellent. Training part was very fast but regarding the complexity of the test script, scripts characters are simple, easy to understand by human eye. Test script contains some inter-line horizontal space and some inter-word and inter-character space also. This pattern is not always available in natural handwriting. So it needs more experimental effort for faithful conversion from difficult handwriting to text conversion. However, this approach may be a true guideline for future research for giving computer an intelligence which a human being applies everyday and at every moment.

7. REFERENCES

[1] S-B. Cho,"Neural-Network Classifiers for Recognizing Totally Unconstrained Handwritten Numerals", IEEE Trans. on Neural Networks, vol.8, 1997, pp. 43-53.

[2] Verma, B. "A Contour Code Feature Based Segmentation For Handwriting Recognition", 7th IAPR International conference on Document Analysis and Recognition, ICDAR'03, 2003, pp. 1203-07.

[3] N.W. Strathy, C.Y. Suen and A. Krzyzak, "Segmentation of Handwritten Digits using Contour Features", ICDAR '93,1993, pp. 577-580.2003.

[4] R G Casey and E Lecolinet "A Survey of Methods and Strategies in Character Segmentation," IEEE Trans. Pattern analysis and Machine Intelligence, vol. 18, 1996, pp. 690-706.

[5] Fletcher, R., and C.M. Reeves "Function minimization by conjugate gradients" the computer journal, vol-7 149-153, 1964.

[6] D. Gorgevik and D. Cakmakov,An Efficient Three-Stage Classifier for Handwritten Digit Recognition, ICPR, vol. 4, 2004, pp. 507-510.

[7] Gernot A. Fink, Thomas Plotz, "On Appearance-Based feature Extraction Methods for Writer-Independent Handwritten Text Recognition" Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05) 1520-5263/05.

[8] C. E. Dunn and P. S. P. Wang, "Character Segmentation Techniques for Handwritten Text A Survey Proceedings of the II n International Conference on Pattern Recognition, The Hague,The Netherlands,1992 pp 577-580.

[9] Matthias Zimmermann and Horst Bunke "Optimizing the Integration of a Statistical Language Model in HMM based Offline Handwritten Text Recognition" .Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04) 1051- 4651/04.

[10]. Hestenes, M." conjugate Direction Methods In Optimization", Springer-verlag, New York, 1980.

[11] S-W. Lee,"Off-Line Recognition of Totally Unconstrained Handwritten Numerals Using Multilayer Cluster Neural Network". IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 18,1996, pp. 648-652.

[12] P. D. Gader, M. Mohamed and J-H. Chiang, 'Handwritten Word Recognition with Character and Inter-Character Neural Networks", IEEE Transactions on Systems, Man, and Cybernetics-Part B:Cybernetics, vol .27, 1997, pp. 158-164.

[13] Blumenstein and B. Verma. “Neural–based solutions for the segmentation and recognition of difficult handwritten words from a benchmark database”. In Proc. 5th International Conference on Document Analysis and Recognition, pages 281–284 Bangalore, India, 1999.

[14] J-H. Chiang, “A Hybrid Neural Model in Handwritten Word Recognition”, Neural Networks, vol. 1I, 1998, pp. 337-346.

[15] Simon Haykin ‘Neural Networks A comprehensive Foundation’,second edition.

[16] R.O.Duda ,P.E. Hart , D.G.stock ‘Pattern classification’ second edition.

[17] William K.Pratt ‘DIGITAL IMAGE PROCESSING’ Third edition.

[18]. S.Rajeshkaran and G.A. Vijayalakshmi Pai ‘Neural Networks, Fuzzy Logic, and Genetic Algorithms Synthesis and Applications.’ Eastern Economy Edition.

[19]. K.M.Khoda, Y.Liu and C. Storey “Generalized Polak –Ribiere Algorithm” journal of optimization theory and application: vol 75,No 2,November 1992.

[20] Verma, B.; Hong Lee;’ A Segmentation based Adaptive Approach for Cursive Hand written Text Recognition Neural Networks, 2007. IJCNN 2007. International Joint Conference on 12-17 Aug. 2007 Page(s):2212 – 2216 Digital Object Identifier 10.1109/IJCNN.2007.4371301.

[21] Fletcher, R.(1975). “practical methods of optimization “. New York: John Wiley & Sons.

[22] Martin Fodslette Moller. “A Scaled Conjugate Gradient Algorithm For Fast Supervised Learning.” Neural Networks, vol 6:525-533, 1993.

[23] M. Blumenstein and B. Verma “Neural-based Solutions for the Segmentation and Recognition of Difficult Handwritten Words from a Benchmark Database” In Proc. 5th International Conference on Document Analysis and recognition, pages 281–284, Bangalore, India, 1999.

[24] Y. H. Dai and Y. Yuan, Convergence properties of the Beale-Powell restart algorithm, Sci.China Ser. A, 41 (1998), pp. 1142-1150.

[25] U. Pal, N. Sharma, T. Wakabayashi, F. Kimura, "Off-line handwritten character recognition of Devanagari script", Proceedings of 9th international conference on document analysis and recognition , vol. 1, pp. 496-500, 2007.

[26] Haradhan Chel, Aurpan Majumder, Debashis nandi, “Scaled Conjugate Gradient Algorithm in Neural NetworkBased Approach for Handwritten Text Recognition” D. Nagamalai, E. Renault, M. Dhanushkodi (Eds.): CCSEIT 2011, CCIS 204, pp. 196–210, 2011.

[27] P. Gader, M. Whalen, M. Ganzberger, and D. Hepp. “Handprinted word recognition on a NIST dataset. Machine Vision and Applications, 8:31–41, 1995