WORD LEVEL DISCRIMINATIVE TRAINING FOR HANDWRITTEN WORD RECOGNITION

WEN-TSONG CHEN

Department of Electrical Engineering
University of Missouri – Columbia
239 Engineering Building West
Columbia, Missouri 65211
E-mail: wchen@ece.missouri.edu

PAUL GADER

Department of Computer Engineering and Computer Science
University of Missouri – Columbia
201 Engineering Building West
Columbia, Missouri 65211
E-mail: gader@cecs.missouri.edu

Word level training refers to the process of learning the parameters of a word recognition system based on word level criteria functions. Previously, researchers trained lexicon-driven handwritten word recognition systems at the character level individually. These systems generally use statistical or neural based character recognizers to produce character level confidence scores. In the case of neural networks, the objective functions used in training involve minimizing the difference between some desired outputs and the actual outputs of the network. Desired outputs are generally not directly tied to word recognition performance. In this paper, we describe methods to optimize the parameters of these networks using word level optimization criteria. Experimental results show that word level discriminative training without desired outputs not only outperforms character level training but also eliminates the difficulty of choosing desired outputs. The method can also be applied to all segmentation based handwritten word recognition systems.

1   Introduction

Word level training refers to the process of learning the parameters of the word recognition system based on word level criteria functions. Although HMM-based techniques [1-3] are often trained at the word level, dynamic programming based handwritten word recognition systems have been trained at the character level. These systems generally use statistical or neural based character recognizers to produce character level confidence scores. In the case of neural networks, the objective functions used in training involve minimizing the difference between

393

some desired outputs and the actual outputs of the network. Desired outputs are generally not directly tied to word recognition performance. In fact, previous research results [4, 9, 16-17] indicate that the performance of these modules on isolated character recognition tasks is not a good indicator of performance of these modules in the context of word recognition. Because the training is done in the module level only *i.e.* the training is optimized on the character level it dose not guarantee optimization at the system level (word level). The results imply a need to study system level (word level in our case) training. In word level training, the desired output is set based on a word level criteria function.

In this paper, we describe methods to optimize the parameters of these networks using word level optimization criteria. Experimental results show that word level discriminative training without desired outputs not only outperformed others but also eliminated the difficulty of choosing desired outputs. The method can also be applied to all segmentation based handwritten word recognition system.

The remainder of this section provides a literature review. The literature review includes descriptions of a variety tools used in the research including the baseline handwriting recognition system used for comparison, discriminant functions and discriminant training techniques. In section 2, the word level training framework is described. Finally, the design and results of experiments are provided in section 3.

1.1   Dynamic Programming based Handwritten Word Recognition

The baseline handwritten word recognition system is a segmentation-based, lexicon-driven system; it is similar in structure to others and is fully described in the literature [4-7, 16-17]. The inputs to the baseline system are a word image and a list of possible strings for the word image called a lexicon. The goal of this system is to sort the strings in the lexicon ordered according to scores that indicate the degree to which the word image matches the strings. Therefore, the output of the word recognition system is a sorted lexicon and the first string in the lexicon is considered to be the most likely to represent the word image by the system. Since our system is similar to those of others, the methods presented here should have general applicability [19-22].

The system segments each input word image into a set of primitives; each primitive consisting of a subimage of the word image that should be either a character or a part of a character. Each primitive and union of primitives is assigned a set of confidence values. There is one confidence value for each character class. The confidence values reflect the degree to which the primitive or union of primitives represents that class. For each string in the lexicon, dynamic programming is used to find the best sequence of unions of primitives to match the string using the confidence values. The confidence values in our system are obtained using neural networks. These networks are trained at the character level,

that is, training sets consisting of isolated sets of characters are used to train the neural networks using fuzzy class-coded desired outputs and a standard mean-square error criterion function.

1.2 Discriminant Functions and Discriminative Training

One of the difficulties of handwriting word recognition is the selection of desired outputs of character classifiers during training. Word level discriminative training eliminates the necessity of choosing desired outputs at the word or character level by minimizing a function that related to the standard MSE. It has been applied with some success to speech recognition and landmine detection with hidden Markov models [13-15, 18]. It has not been applied to word level training of dynamic programming based handwritten word recognition. A general overview is given here with specifics in section 2.

Let $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_m$ be a set of vectors. Each vector $\mathbf{x}_i$ is assumed to belong to one of N classes $\mathbf{w}_1$, $\mathbf{w}_2$, ..., $\mathbf{w}_N$. A set of discriminant functions is defined by $D_i(\mathbf{x}; W)$, $i = 1, 2, …, N$, where W is a parameter set and x is an input vector. When $\mathbf{x}_p$ is an input vector, it is classified as class j if

$$j = \arg\max_i D_i(\mathbf{x}_p; W)$$

In [15], Juang and Katagiri designed a method to incorporate the discriminant function in a scalar criterion suitable for a gradient type search algorithm to find a solution. They introduce a misclassification measure that embeds the decision process in the overall minimum classification error formulation. The misclassification measure can be defined as many ways. One of the misclassification measures is as follows

$$d_k(x;W) = -D_k(x;W) + [\frac{1}{N-1}\sum_{i\neq k} D_i(x;W)^\eta]^{1/\eta}$$

where η is a positive number.

If $\eta \to \infty$, the misclassification measure becomes

$$d_k(x;W) = -D_k(x;W) + D_j(x;W)$$

where

$$j = \arg\max_{i\neq k}(D_i(x;w)).$$

That is to say that if x is truly from class k and

$$d_k(x;W) \text{ is } \begin{cases} > 0 & \text{then x is misclassified} \\ \leq 0 & \text{then x is correctly classified.} \end{cases}$$

They then defined a general form of the cost function as

$$C_k(x;W) = C_k(d_k(x;W)).$$

The cost function is a function of the misclassification measure. There are many possible choices for the cost function. One of those many possibilities is translated sigmoid functions which is defined by

$$C_k(d_k) = \frac{1}{1 + e^{-a(d_k + b)}} \qquad a > 0.$$

The final step is to minimize the overall cost function by updating the parameter set W of the classifier using gradient descent. The overall cost function can be expressed as

$$C(W) = \sum_x \sum_{k=1}^{N} C_k(x;W)\delta(x \in \text{class } k)$$

where

$$\delta(\varepsilon) = \begin{cases} 1 & \text{if } \varepsilon \text{ is truth} \\ 0 & \text{other} \end{cases}.$$

The key point of discriminative training is to minimize a measure of misclassification error instead of a criterion function that is only indirectly related to misclassification. This also eliminates the difficulty of choosing desired outputs in the training procedure because there is no longer a need to supply desired outputs with the discriminative learning.

2   Word Level Discriminative Training Technique

In this section, we demonstrate how we applied the discriminative training concept to handwritten word recognition. A training set, T, consists of a set of words together with a set of lexicons; each word has one lexicon associated with it. The word level objective function is formed by

$$\min \sum_{I \in T} C(d(I, \Lambda_I); \Theta)$$

where I is an input image, Λ_I is a lexicon associated with the image I and Θ is a set of parameters of the character classifier. We used the sigmoid function as our cost function, which is

$$C(d; \Theta) = \frac{1}{1 + e^{-ad}} \qquad a > 0.$$

The misclassification measure we used was the extreme case when $\eta \to \infty$, that is

$$d(I, \Lambda; \Theta) = -D(I, L_{truth}; \Theta) + \max_{\substack{L \in \Lambda \\ L \neq L_{truth}}} [D(I, L; \Theta)]$$

where $D(I, L; \Theta)$ is the word level confidence obtained from matching image I to string $L \in \Lambda$ and L_{truth} is a string representing the truth identity of the word image. In our baseline handwritten word recognition system, we can further express D in term of the character level confidences as

$$D(I, L) = \frac{1}{n_i} \sum_{s=1}^{n_i} x_s$$

where n_i is the length of string L and x_s is the character confidence for the segment of the word image that matches to the sth character in the string. If the lexicon is represented as $\Lambda = \{ L_1, L_2, ..., L_m \}$ and if L_i is the string representing the true identity of word image I, then the misclassification measure can be written as

$$d(I, \Lambda; \Theta) = -(\frac{1}{n_i} \sum_{s=1}^{n_i} x_{is}) + \max_{j \neq i} [\frac{1}{n_j} \sum_{r=1}^{n_j} x_{jr}] .$$

Once the word level criterion function is set we can use gradient descent to update the parameter set. We take the derivative of the max function to be

$$\frac{\partial}{\partial z_i} \max(z_1, z_2, K, z_m) = \begin{cases} 1 & \text{if } z_i = \max(z_1, z_2, K, z_m) \\ 0 & \text{else} \end{cases} .$$

3 Experimental Design and Results

3.1 Training Lexicons

The issue is how to generate a lexicon that provides the best recognition rate and more generalization to the other test images. There are two broad choices : use lexicon of real words or artificial lexicons. If we use lexicons of real words, it is difficult to control the distribution of desired output values. This lack of control can lead to unbalanced training. Here, artificial lexicons consist of lists of systematically generated strings. Each string is a sequence of characters that may or may not constitute real words. These lexicons can be generated in such a way that balanced training can be achieved.

The artificial lexicons are generated as follows :

397

Based on the truth strings of the training images, we construct a lexicon (lex_gen) which contains a number of modified strings. Each modified string is constructed by randomly replacing some of the characters in the truth string. If we let n denote the length of a truth string, modified strings are constructed by changing 1, 2, …, n characters.

3.2 Database

The character training data set contains isolated characters extracted from handwritten word images obtained from images of USPS mail addresses. A complete description of this character training data set can be found in [10-11]. This character training data set was used to initialize the parameters of character recognizers. For word level training, we used about 3800 word images which came from the CD-ROM image database produced by the Center of Excellence for Document Analysis and Recognition (CEDAR) at the State University of New York at Buffalo and sponsored by the U.S. Postal Service [12]. These word images were taken from the training database of the CEDAR CD-ROM image database, including word images from "BB", "BC", "BD", "BL", and "BS" databases in both states and cities directories.

The testing image set (bd_317) is bd city word test set, which also came from CD-ROM image database described above. This image set has 317 word images and comes with three different lexicon sets. We used those sets with the average lengths of 1000 (lex3) and 100 (lex2) for each word image.

3.3 Experiments

Word level training was performed on about 3800 word images using word level criterion function using the lex_gen lexicon generation method to generate training lexicons. We generated 20*(length of the truth string) strings for an input word image. All strings used for training for a word image has the same length as its truth string. The training word images were chosen from "BB", "BC", "BD", "BL", and "BS" databases in both states and cities directories in the training database of the CEDAR CD-ROM image database. We chose those images that were legible and have correct segmentations to match to their truth strings. The initial sets of parameters of character classifiers were obtained by training classifiers on separate isolated character sets [10-11]. This sets also include only legible, correctly segmented characters. The testing images were not constrained; none were removed.

398

<u>Word level Discriminative training</u> :

Once the neural network character classifiers were initially trained at the character level, we further trained them using the word level criterion functions for 5000 epochs. In order to compare the effects of word level training with character level training, we also trained these character classifiers on the character level using the same data set as the word level training for 5000 more epochs. More precisely, we constructed a character level training set by collecting all the machine segmented characters from all the words in the training set and further trained the character classifiers using those characters as the training set. These are exactly the same characters that are used by the word level training algorithm. Desired outputs of the character criterion function were obtained by fuzzy k-nearest neighbor algorithm [8].

For every 25 epochs during training, we recorded the parameters of classifiers and tested the performances on the test set (bd_317) with lex2 and lex3 separately. The results are shown in figures 1 and 2. In each case, the system built using word level training outperforms the one built using character level training on testing data. Algorithm 1 shows the details of this word level discriminative training methodology.

<u>Algorithm 1:</u> Word Level Discriminative Training Without using Desired Outputs
For each word image
       For each legal union of primitives
            Form union of primitives;
            Compute feature vectors;
            Compute character scores;
       End for
       Generate lexicon (lex_gen);
       For each string in lex_gen
             Find the highest score using dynamic programming;
            Determine segmentation associated with the highest score;
       End for
       Find the non-truth string in the lexicon that has best score;
       Update parameters of character recognizer using
         truth string of word image
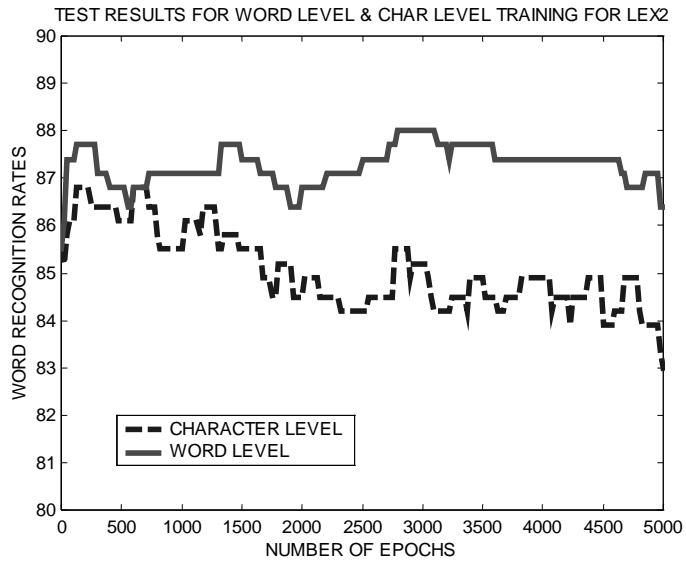   *non-truth string with highest score;*
End for

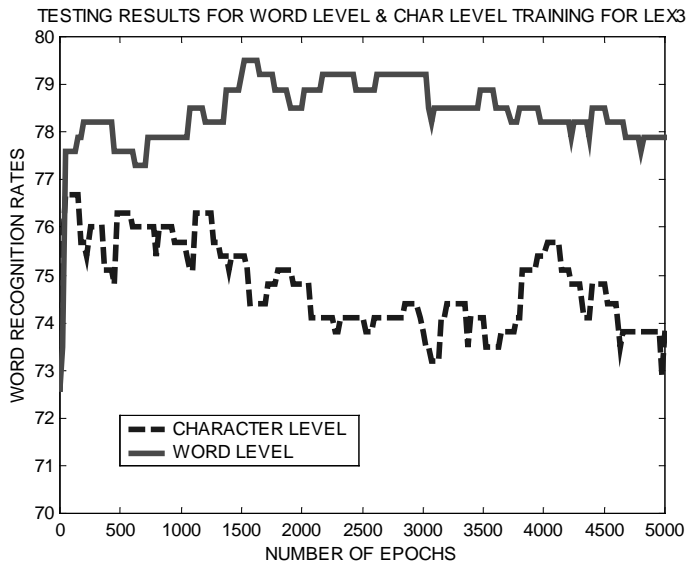**Figure 1**. Testing performance comparison of word level training and character level training on lex2

Figure 2. Testing performance comparison of word level training and character level training on lex3

400

4   Conclusion

Word level training of dynamic programming based word recognition without desired outputs using discriminative training outperformed word level and character level training with desired outputs. One condition for this technique to be successful is that we manually select the correct segmentation of word images according to their truth strings. If the segmentation algorithm is good enough and the dynamic programming can produce correct segmentation during training we can eventually update the parameters dynamically *i.e.* allowing the system to train every time it encounters a word.

References

1. A. M. Gillies, "Cursive Word Recognition using Hidden Markov Models", *Proc. United States Postal Service Advanced Technology Conference,* Washington DC, pp. 557 – 563, 1992.
2. M. Mohamed, "Handwritten Word Recognition Using Generalized Hidden Markov Models", Ph. D. dissertation, University of Missouri-Columbia, 1995.
3. A. Kundu, "Recognition of Handwritten Word: First and Second Order Hidden Markov Model Based Approach", *Pattern Recognition,* vol. 22, no. 3, pp. 457-461, 1988.
4. P. Gader, M. Mohamed, and J. H. Chiang," Comparison of Crisp and Fuzzy Character Networks in Handwritten Word Recognition", *IEEE Trans. Fuzzy Systems*, vol. 3, no. 3, pp. 357-364, 1995.
5. P. Gader, M.A. Mohamed, and J.M. Keller, "Dynamic Programming Based Handwritten Word Recognition Using The Choquet Fuzzy Integral As The Match Function", *Journal of Electronic Imaging*, vol. 5, no. 1, pp. 15-24, 1996.
6. P. D. Gader, M. A. Mohamed, and J. M. Keller, "Fusion of Handwritten Word Classifiers", *Pattern Recognition Letters*, vol. 17, no. 6, pp. 577-584, 1996.
7. P. Gader, M. Whalen, and M. Ganzberger, "Handwritten Word Recognition On A NIST Data Set", *Machine Vision and Its Application*, vol. 8, pp. 31-40, l995.
8. J. Keller, M. Gray, J. Givens, "A Fuzzy K-Nearest Neighbor Algorithms", *IEEE Trans. Syst., Man, Cybern.*, vol.15, no. 4, pp. 580-585, 1985.
9. P. D. Gader, M. Mohamed, and J. H. Chiang, "Handwritten Word Recognition with Character and Inter-Character Neural Networks," *IEEE Trans. Sys. Man Cybernetics*, vol. 27, no. 1, pp. 158-165, 1996.
10. J. H. Chiang, "Hybrid Fuzzy Systems for Handwritten Recognition", Ph. D. dissertation, University of Missouri-Columbia, 1995.
11. J. H. Chiang and P. D. Gader, "Hybrid Fuzzy-Neural Systems in Handwritten Word Recognition", *IEEE Trans. Fuzzy Systems*, vol. 5, no. 4, pp. 497-511, 1997.

12. J. J. Hull, "A Database for Handwritten Text Recognition Research," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 16, no. 5, pp. 550-554, 1994.

13. N. J. Nilsson, *Learning Machines : Foundations of Trainable Pattern Classifying Systems*. New York : McGraw-Hill, 1965.

14. J.K. Hawkins, "Self-Organizing Systems – a Review and Commentary", *Proc. IRE*, Vol. 49, pp.31-48, Jan. 1961.

15. B-H Juang and S. Katagiri, "Discriminative Learning for Minimum Error Classification", *IEEE Trans. Signal processing*, vol. 40, no. 12, pp. 3043-3054, 1992.

16. W-T Chen, P. Gader and H. Shi, "Lexicon-Driven Handwritten Word Recognition Using Optimal Linear Combinations of Order Statistics", *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 21, no. 1, pp. 77-82, 1999.

17. P. Gader and W-T Chen, "Word Level Optimization of Dynamic Programming-Based Handwritten Word Recognition Algoritms", *Proc. SPIE*, Vol. 3651, pp.50-57, Jan. 1999.

18. Y. Zhang, Discriminative Training of Hidden Markov Models for Landmine Detection, MS Thesis, Department of Computer Engineering and Computer Science, University of Missouri – Columbia, Jan. 2000.

19. V. Govindaraju and S. N. Srihari, "System for Reading Handwritten Documents," *Proceedings, IEEE Conf. SMC*, Vancouver BC Canada, pp. 2347-2352, October, 1995.

20. G. Kim and V. Govindaraju, "Handwritten Word Recognition for Real-Time Applications," *Proceedings, Third International Conf. Document Anal. Recognition (ICDAR '95)*, Montreal CA, pp. 24-28, , 1995.

21. F. Kimura, M. Shridhar, and N. Narasimhamurthi, "Lexicon Directed Segmentation - Recognition Procedure for Unconstrained Handwritten Words," *Proceedings, Third International Workshop on Frontiers in Handwriting Recognition*, Buffalo, NY, pp. 122-132, 1993.

22. M. Shridhar, G. Houles, and F. Kimura, "Handwritten Word Recognition Using Lexicon Free and Lexicon Directed Word Recognition Algorithms," *Proceedings, Int'l Conf Document Analysis Recognition (ICDAR '97)*, Ulm Germany, Aug., 1997.