A CONNECTIONIST RECOGNIZER FOR ON-LINE CURSIVE HANDWRITING RECOGNITION

Stefan Manke and Ulrich Bodenhausen

University of Karlsruhe, Computer Science Department,
D-76128 Karlsruhe, Germany

ABSTRACT

In this paper we show how the Multi-State Time Delay Neural Network (MS-TDNN), which is already used successfully in continuous speech recognition tasks, can be applied both to on-line single character and cursive (continuous) handwriting recognition. The MS-TDNN integrates the high accuracy single character recognition capabilities of a TDNN with a non-linear time alignment procedure (dynamic time warping algorithm) for finding stroke and character boundaries in isolated, handwritten characters and words. In this approach each character is modelled by up to 3 different states and words are represented as a sequence of these characters. We describe the basic MS-TDNN architecture and the input features used in this paper, and present results (up to 97.7% word recognition rate) both on writer dependent/ independent, single character recognition tasks and writer dependent, cursive handwriting tasks with varying vocabulary sizes up to 20000 words.

1. INTRODUCTION

This paper describes a connectionist solution for the problem of single character and cursive (continuous) handwriting recognition. The recognition of continuous handwriting, as it is being written on a touch screen or graphics tablet, has not only scientific but also significant practical value, e.g. for note pad computers or for integration into multi-modal systems. In Figure 1, the example application which we use for on-line demonstrations of our handwriting recognizer is shown. The main advantage of *on-line* handwriting recognition is the temporal information of writing, which can be recorded and used for recognition. Handwritten words can be represented as a time-ordered sequence of coordinates with varying speed and pressure in each coordinate. .As in speech recognition, the main problem of recognizing continuous words is that character or stroke boundaries are not known (in particular if no pen lifts or white space indicate these boundaries) and an optimal time alignment has to be found [1]. The connectionist recognizer, described in this paper, integrates the recognition and segmentation into a single network architecture, the Multi-State Time Delay Neural Network (MS-TDNN), which was originally proposed for continuous speech recognition tasks [2,3,4].

For on-line *single* character recognition, the Time Delay Neural Network (TDNN) [5] with its time-shift invariant architecture has been applied successfully [6]. The Multi-State Time Delay Neural Network (MS-TDNN), an extension of the TDNN, combines the high accuracy character recognition capabilities of a TDNN with a non-linear time alignment procedure (Dynamic Time Warping) [7] for finding an optimal alignment between strokes and characters in handwritten continuous words.

The following section describes the basic network architecture and training method of the MS-TDNN, followed by a description of the input features used in this paper (section 3). Section 4 presents results both on different writer dependent/writer independent, single character recognition tasks and writer dependent, cursive handwriting recognition tasks.

2. THE MULTI-STATE TIME DELAY NEURAL NETWORK (MS-TDNN)

The Time Delay Neural Network provides a time-shift invariant architecture for high performance on-line single character recognition. The Multi-State TDNN is capable of recognizing continuous words by integrating a dynamic time warping algorithm (DTW) into the TDNN architecture. Words are represented as a sequence of characters, where each character is modelled by one or more states. For the results in this paper, each character is

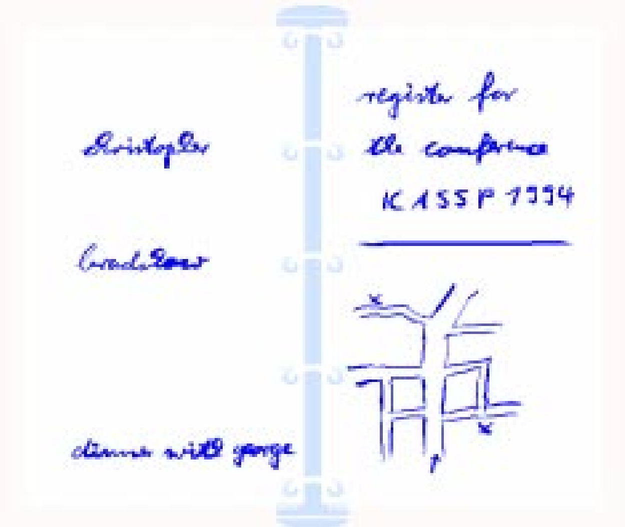

Figure 1: Demonstration application for our on-line continuous handwriting recognizer

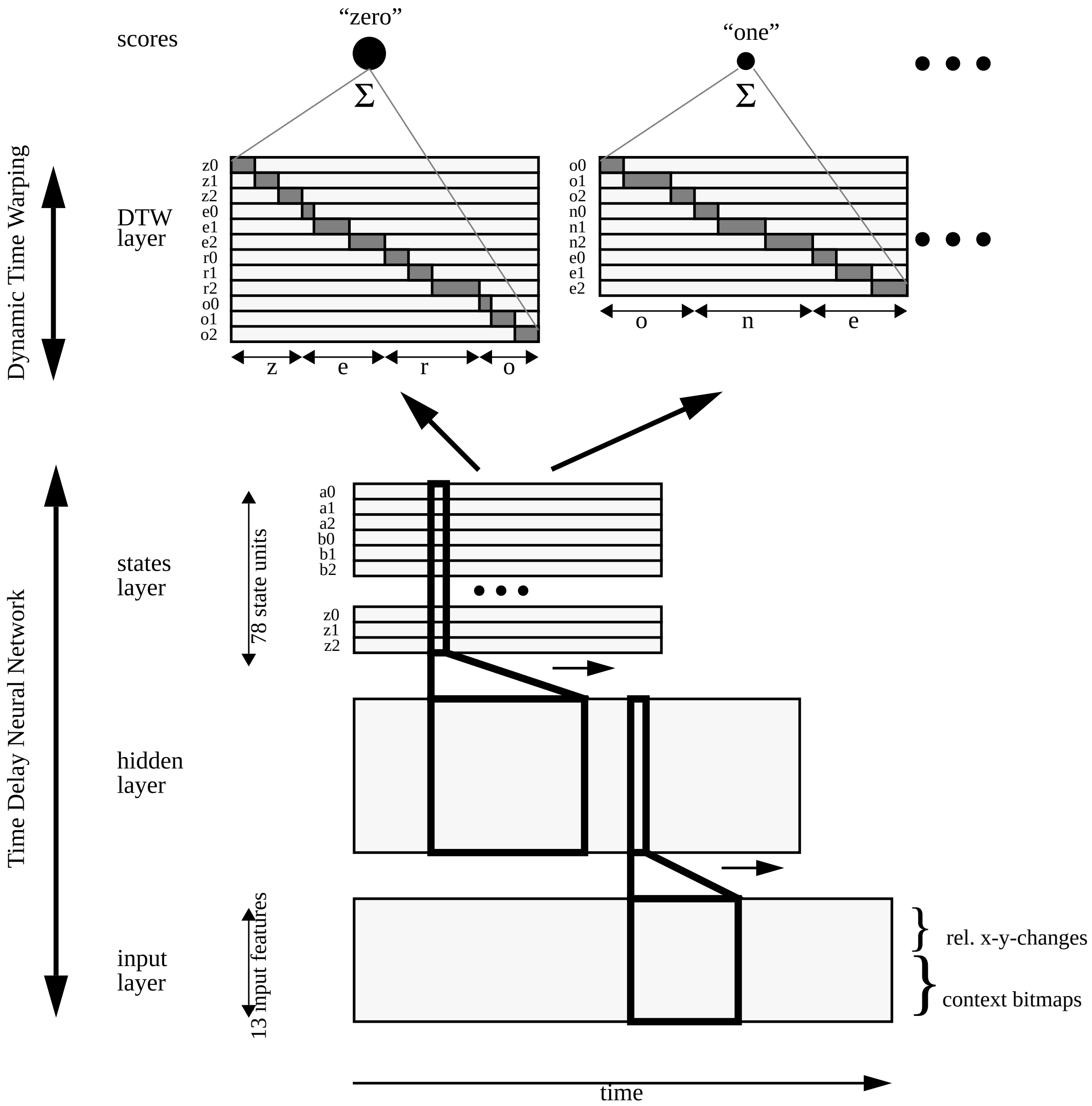

Figure 2: Architecture of the basic MS-TDNN

modelled with 3 states representing the first, middle, and last part of the characters. We use this approach not only for the recognition of cursive words, but also for the recognition of isolated single characters (letters and digits). In Figure 2 the basic architecture of the MS-TDNN is shown. The first three layers constitute a standard TDNN with sliding input windows of certain sizes. This TDNN computes scores for each state and for each time frame in the states layer. In the DTW layer each word to be recognized is modelled by a sequence of states, the corresponding scores for the states are simply copied from the states layer into the word models of the DTW layer. An optimal alignment path is found by the DTW algorithm for each word and the sum of all activations along this optimal path is taken as the score for the word output unit.

All network parameters like the number of states per character, the size of the input windows, or the number of hidden units are optimized manually for the results presented in this paper, but can also be optimized automatically by the Automatic Structure Optimization (ASO) algorithm that we have already proposed in [8,9]. By using the ASO algorithm, no time-consuming manual tuning of these network parameters for particular handwriting tasks and training set sizes is necessary to get optimal recognition performances.

The MS-TDNN is trained with standard back-propagation and training starts in a forced alignment mode, during which the MS-TDNN is trained with hand-segmented training data. For this purpose only a small part of the training data must be labeled manually with character boundaries. Stroke boundaries are deter-

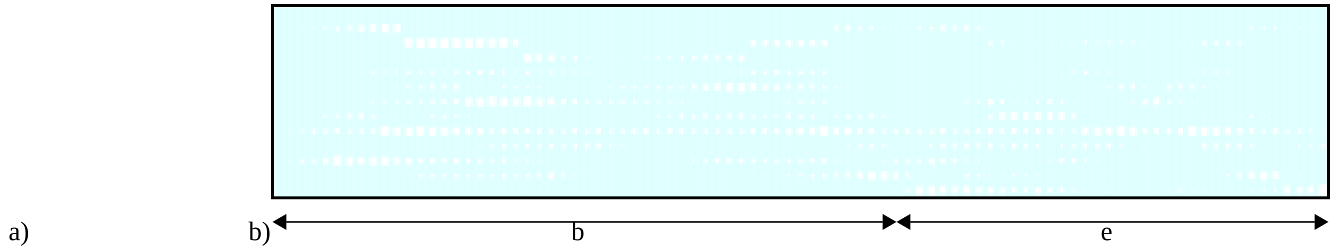

Figure 3: a) the word “be” written on the graphics tablet and b) its input representation

mined automatically. During this forced alignment mode the back-propagation starts at the states layer of the front-end TDNN with fixed state boundaries.

After a certain number of iterations, when the network has successfully learned character boundaries of the first smaller training set, the forced alignment is replaced by a free alignment found by the DTW algorithm. Now training starts at the word level of the MS-TDNN and is performed on unsegmented training data.

During training in forced alignment mode the McClelland objective function [3] is used to avoid the problem with *1-out-of-n* codings for large *n* which appear with the Mean Squared Error. For word level training (where back-propagation starts at the output units) we use an objective function (Classification Figure of Merit [10]), which tries to maximize the distance between the activation of the correct output unit and the activation of the best incorrect output unit.

3. DATA COLLECTION AND PREPROCESSING

The databases used for training and testing of the MS-TDNN were collected at the University of Karlsruhe. All data is recorded on a pressure sensitive graphics tablet with a cordless stylus which produces a sequence of time ordered 3-dimensional vectors (at a maximum rate of 200 dots per second) consisting of the x-y-coordinates and a pressure value for each dot. All subjects had to write a set of single words from a 400 word vocabulary, covering all lower case letters and at least one set of isolated lower case letters, upper case letters, and digits. For the continuous handwriting results presented in this paper only the data of the first author was used.

A relatively straightforward and fast preprocessing is performed on this data. To preserve the dynamic writing information, which is the main advantage of on-line handwriting recognition over pure optical handwriting recognition, these sequences of 3-dimensional vectors are transformed into time-ordered sequences of equally spaced, 13-dimensional feature vectors, which describe relative position changes and the local topological context of each dot. The local context is represented by a 3x3 context pixel-map, which is calculated from a larger 20x20 pixel-map around the current dot. This pixel-map encodes the local shape of the curvature and all previous or future dots in the context of the current dot. For calculating the 20x20 bitmap, all dots of the current word or character are used. So one always knows if a particular dot is visited twice. E.g. in the contextpixel-maps for the dots of the up-stroke of a handwritten “t” the dots of the horizontal stroke of the “t” are already visible). The remaining 4 features describe relative x and y position changes of each dot. Using these 13 features, one can combine the advantages of pure pixel-maps (all dots can be seen at once) with the dynamic writing information (order of writing). We will present these features in more detail in a forthcoming paper.

Figure 2 shows the handwritten word “be” as it has been written on the graphics tablet and its input representation after preprocessing.

4. EXPERIMENTS AND RESULTS

The proposed MS-TDNN architecture was trained and tested both on **writer dependent, cursive (continuous) handwriting** tasks and **writer dependent/writer independent, single character** recognition tasks.

a) writer dependent, cursive (continuous) handwriting

The MS-TDNN was trained in writer dependent mode with 2000 training patterns (isolated words) from a 400 word vocabulary (5 training patterns for each word in the vocabulary). The average word length of this vocabulary is 6.3 characters/word and it covers all single lower case letters.

The trained network was then tested without any retraining on 5 different vocabularies with varying vocabulary sizes from 400 up to 20000 words. Test results for these vocabularies (given as word recognition rates) are shown in Table 1.

Task	Vocabulary Size (words)	Test Patterns	Recognition Rate
msm_400_a	400	800	97.7%
msm_400_b	400	800	96.7%
msm_1000	1000	2000	94.8%
msm_10000	10000	2000	86.6%
msm_20000	20000	2000	83.0%

Table 1: Results for different writer dependent continuous handwriting tasks.

Database *msm_400_a* consists of 800 test patterns from the same 400 word vocabulary the network was trained on.

All othertest sets (*msm_400_b*, *msm_1000*, *msm_10000*, and *msm_20000*) are selected randomly from a 100.000 word vocabulary (Wall Street Journal vocabulary) and are completely different from the *msm_400_a* vocabulary. The average word length of these vocabularies is around 8.2 characters/word.

b) writer dependent/writer independent, single (isolated) characters

We have trained and tested the same network architecture both for writer dependent and writer independent, single character recognition tasks. For the results presented in this paper, we have used separate networks for lower case letters (task a_z), upper case letters (task A_Z), and digits (task 0_9). Writer dependent results for these tasks are shown in Table 2 and writer independent results in Table 3.

Task	Training Patterns	Test Patterns	Recognition Rate
msm_0_9	400	200	99.5%
msm_A_Z	1040	520	99.0%
msm_a_z	1040	520	98.1%

Table 2: Single character recognition results (writer dependent)

Task	Training Patterns	Test Patterns	Recognition Rate
0_9	1600	200 (20 writers)	99.5%
A_Z	2000	520 (20 writers)	93.9%
a_z	2000	520 (20 writers)	91.5%

Table 3: Single character recognition results (writer independent)

5. CONCLUSIONS AND FUTURE WORK

In this paper we have shown how recognition and segmentation of isolated, cursive handwritten words can be integrated into a single architecture, the MS-TDNN, which combines a time-shift invariant Time Delay Neural Network with a non-linear time alignment algorithm for finding stroke and character boundaries. No separate recognition and segmentation steps are necessary in this approach. The MS-TDNN architecture can be used both for single character and cursive handwriting recognition tasks. The results on different writer dependent, cursive handwriting tasks (Table 1) show that the MS-TDNN is capable of learning and finding stroke and character boundaries in handwritten words and classifying these characters and words with high recognition performances. The MS-TDNN performs well not only on the vocabulary it was trained for (see task *msm_400_a*), but also for other vocabularies it has never seen before (task *msm_400_b*), even on much larger vocabularies (*msm_1000*, *msm_10000*, and *msm_20000*). This shows that the MS-TDNN successfully learns to recognize characters and character boundaries independent of the training vocabulary.

Work is in progress to train and test the MS-TDNN architecture on larger writer independent databases, that we have already collected. First results which we have achieved on a smaller witer independent database are about 75% word recognition rate. In addition we are currently testing the MS-TDNN with a first-best/ N-best dynamic programming search driven by a statistical language-model replacing the fixed word models.

Acknowledgments

The authors would like to thank Michael Finke, Hermann Hild, and Alex Waibel for many helpful discussions, and all students and colleagues for contributing to our handwriting databases.

References

[1] M. Schenkel, H. Weissman, I. Guyon, C. Nohl, and D. Henderson. Recognition-based Segmentation of On-line Handprinted Words. *Advances in Neural Network Information Processing Systems (NIPS-5).* Morgan Kaufmann, 1993.

[2] P. Haffner, M. Franzini, and A. Waibel. Integrating Time Alignment and Neural Networks for High Performance Continuous Speech Recognition. *Proceedings of the ICASSP-91.*

[3] H. Hild and A. Waibel. Connected Letter Recognition with a Multi-State Time Delay Neural Network. *Advances in Neural Network Information Processing Systems (NIPS-5).* Morgan Kaufmann, 1993.

[4] C. Bregler, H. Hild, S. Manke, and A. Waibel. Improving Connected Letter Recognition by Lipreading. *Proceedings of the ICASSP-93*, Minneapolis, April 1993.

[5] A. Waibel, T. Hanazawa, G. Hinton, K. Shiano, and K. Lang. Phoneme Recognition using Time-Delay Neural Networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, March 1989.

[6] I. Guyon, P. Albrecht, Y. Le Cun, W. Denker, and W. Hubbard. Design of a Neural Network Character Recognizer for a Touch Terminal. *Pattern Recognition,* 24(2), 1991.

[7] H. Ney. The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, March 1984.

[8] U. Bodenhausen, S. Manke, and A. Waibel. Connectionist Architectural Learning for High Performance Character and Speech Recognition. *Proceedings of the ICASSP-93*, Minneapolis, April 1993.

[9] U. Bodenhausen and S. Manke. Automatically Structured Neural Networks for Handwritten Character and Word Recognition. *Proceedings of the ICANN-93,* Amsterdam, September 1993.

[10] J. Hampshire and A. Waibel. A Novel Objective Function for Improved Phoneme Recognition. *IEEE Transactions on Neural Networks*, June 1990.