

MS-TDNN with Global Discriminant Trainings

Abstract

This article analyses the behavior of various hybrid architectures based on a neuro-markovian scheme (MS-TDNN) applied to online handwriting word recognition systems. We have considered different cost functions, including maximal mutual information criteria with discriminant training and maximum likelihood estimation, to train the systems globally at the word level and also we varied the number of states from one up to three to model the basic hidden Markov models at the letter level. We report experimental results for non constrained, writer independent, word recognition obtained on the IRONOFF database.

1. Introduction

In recent years there has been a significant body of work concerning online handwriting recognition systems. Many of these works are restricted to solve the recognition problem for isolated digits of characters. It is much more complex to tackle the problem for unconstrained words. This is why, to alleviate the segmentation problem, one can impose specific constraints to the handwriting, such as using a script style [1, 2]. Systems being able to process unconstrained styles for writer independent recognition still require much research effort to design very performing systems with limited computational and memory resources. Such systems are dedicated to mobile communication devices (personal digital assistant, smart-phone, digital pen).

In this context, state-of-art systems are usually based on hidden Markov models (HMMs) [3][4], which are effective under many circumstances, but do suffer for some major limitations in real world applications. The reason is mainly due to their arbitrary parametric assumption that governs the estimation of a generative model from the data, which is then used in the framework of the Bayesian theory to classify the data. However, it is well known that for classification problems, instead of constructing a model independently for each class, a better solution should be to use a discriminative approach that constructs a unique model to decide where the frontiers between classes are. This why artificial neural networks (ANN) appears to be a promising alternative in this respect, but conversely they failed to model sequence data such as online handwriting due to their variable lengths. As a consequence, by combining HMMs and ANN, we can expect to take advantage of

the robustness and flexibility of the HMMs generative models and of the discriminative power of the ANN [5, 6, 7]. Training such a hybrid system is not straightforward, this is why not so many attempts are encountered in literature.

In this paper, we propose several different global discriminant training schemes for an hybrid neuro-markovian system for recognizing unconstrained online cursive handwritten words. These trainings combine maximum likelihood (ML) and maximum mutual information (MMI) criteria with a global optimization approach defined at the word level.

This paper is organized as follows. In section 2, we present a review of training schemes used in hybrid neuromarkovian systems for handwriting word recognition. In section 3, we describe the proposed overall hybrid system (multi-state HMM/TDNN). Different training strategies are proposed in section 4, and the corresponding results are reported in the next section. Finally conclusions and prospects are given in section 6.

2. Training scheme in online recognition hybrid system.

HMMs provide a reasonable structure for representing sequences of characters or of words. Assuming such a structure, one good use for ANNs might be to provide the distance measure for the local match inside the global match in the search for the best sequence of characters (in the sense of the best match to the data).

Usually, in standard HMM systems, the training of the parameter set θ is usually simplified by maximizing likelihoods only (maximum likelihood estimation), i.e.:

$$\operatorname{argmax}_{\theta} \prod_{j=1}^J p(o_j | w_j, \theta)$$

where w_j is one out of J words of the training set. However, we are interested to actually improve discrimination between the models by maximizing the following criterion during training:

$$P(w_i | O, \theta) = \frac{p(O | w_i, \theta) P(w_i)}{\sum_j p(O | w_j, \theta) P(w_j)}$$

where \sum_j represents the sum over all possible models. It is thus clear that the training of every model should depend on all the other models, yielding proper discrimination between the models.

This is precisely the goal of the hybrid HMM/ANN system used in this work.

The most common procedure used to train this hybrid works at the character level. As ANNs training requires supervision (labeled targets for each pattern), an early problem in applying ANN methods to handwriting recognition was the apparent requirement of hand-labelled signals at this character level. In fact, since the ANN outputs can be used in a DP procedure for global decoding, it is possible to use embedded Viterbi training to iteratively optimize both the segmentation and the ANN parameters. In this procedure, each ANN training is done using labels from the previous Viterbi alignment. In turn, an ANN is used to estimate training set state probabilities, and dynamic programming given the training set models is used to determine the new labels for the next ANN training.

Table 1 gives a brief overview of some hybrid systems for cursive handwriting recognition with a short description of their classifier and mode of training.

Table 1. Training scheme for hybrid system.

Author	classifier	Training scheme
Knerr, Augustin [10]	MLP-HMM Offline Explicit segmentation 44 classes	Separate Training: -MLP Training -HMM Training
Schenkel et al. [5]	TDNN-HMM online 26 classes	Obj. Function : cross entropy (true model) 3 independent steps : -Training TDNN with isolated characters -Training nil-class -Training words
Npen++ [6]	MsTDNN-HMM Online 26 classes * 3	Obj. Function: cross entropy (true model, ML) Three steps: -Training on hand-segmented data with Viterbi constrained to the same duration in each state -Same training with unconstrained Viterbi path. -Training at word level.
Tay et al. [9]	MLP-HMM Offline Explicit segmentation	Global discriminative training Obj. Function: simplified MMI (MMIs) Training at character or word level
Chen et al. [8]	NN-HMM Offline Explicit segmentation	Discriminant training MMIs + discriminative function Training at character then at word level.
Our system	Ms-TDNN-HMM Online 66 classes × 3	Global discriminant training Cost function : ML, MMI, and TDNN discrimination. Training at word level

For online handwriting, most of the classifiers are trained with a ML criterion, which try to maximize the likelihood of the true word model irregardless of all other models. In [8] and [9], a MMI criterion with global training is presented and experimental

comparisons with a ML criterion and a training at character level show the interest of a global discriminant training in the framework of offline cursive words recognition.

The same kind of approach is proposed in this work, applied to an online system, with a new formulation of the objective function mixing generative modeling and discriminant training at the word level.

3. Hybrid MS-TDNN markovian system

Figure 1 gives an overview of the complete on-line recognition system. It is based on an analytic approach with an implicit segmentation and a global word-level training. Thus, it allows to handle dynamic lexicon, and no additional training is required to add new entries in the lexicon. Some pre-processing steps are first introduced in order to normalize the input signal, specifically with respect to size, baseline orientation and writing speed.

From these normalized data, a feature-vector frame is derived, $X_{1,N} = (x_1, \dots, x_N)$, where x_i describes the i^{th} point of the input signal. It will be the input of the NN-HMM learning machine. The role of the NN in this hybrid system is to provide observation probabilities for the sequence of observations, whereas the HMM is used to model the sequence of observations and to compute word likelihoods, based on the lexicon.

We have privileged a Ms-TDNN [6] with no explicit segmentation at the character level but a regular scan of the input signal $X_{1,N}$ to produce the probability observation $O_{1,T}$.

For each entry in the lexicon, a HMM-Word model is constructed dynamically by concatenating letter HMMs (66 characters: lowercases, uppercases, accents and symbols). The characters are modeled with one or several states. So in our TDNN, there are as many outputs as the total number of states. For example, if we describe a character with three states, the number of outputs is 198 (66 letters × 3 states).

Observation probabilities in each emitting state of the basic HMMs are computed by the ANN. Transition probabilities model the duration of the letters, actually, as we do not differentiate durations for every letter, all transition probabilities are set to 1 and are not modified during training. Hence, the likelihood for each word in the lexicon is computed by multiplying the observation probabilities over the best path through the graph using the Viterbi algorithm. The word HMM with the highest probability is the top one recognition candidate.

Training such a system could be imagined either at the character level, or directly at the word level. The character level requires to be able to label the word database at this character level, usually using a post-labeling with the Viterbi algorithm, and to

iterate several cycles of training/recognition/labeling to increase the overall performances. There are some difficulties involved with such a scheme. One is to bootstrap the system with an initial labeling, a second problem is to transform, the posterior probabilities estimated by the ANN into scaled likelihood, a third problem is to deal with inputs that have not been encountered during the training because they do not correspond to any actual character.

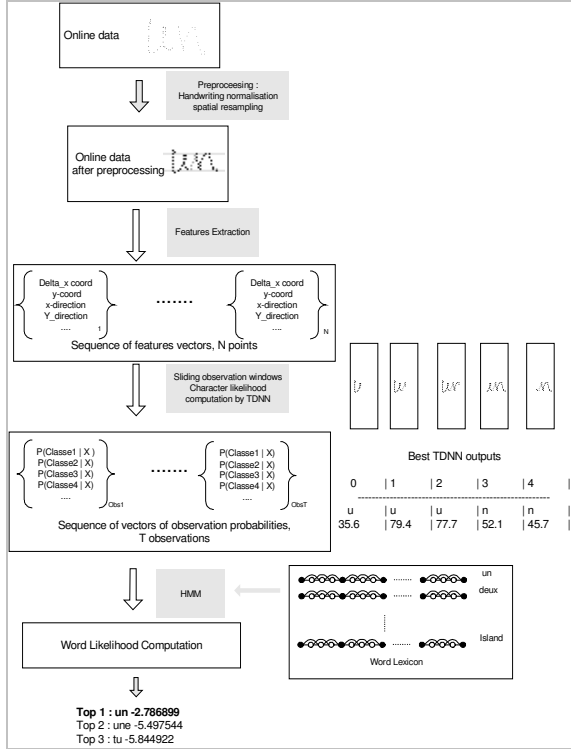


Figure 1. Overview of the on-line cursive word recognition system.

In order to simplify the training process and to improve the word recognition rate, we propose a global training of the hybrid system at the word level. In that case, there is no training explicitly at the character level but an optimization of the network to satisfy an objective function defined at the global word level.

4. Global discriminant training scheme

We propose here a global training approach at the word level based on an objective function that allows to mix discriminant training and generative model training. The proposed solution represents an interesting simplification with respect to approaches requiring several stages, iteratively applied, to carry out the training through the grapheme or letter level to the word level.

The definition of the objective function at the word level is one of the key issues of the training process. Different expressions are proposed in the following table:

Table 2. Objective functions at word level.

Bare ML Criterion	MMI Criterion
$L_{MLE} = \log P(O \lambda_{trueHMM})$	$L_{MMI} = \log \frac{P(O \lambda_{trueHMM})}{\sum \lambda' P(O \lambda')}$
Simplified MMI Criteria	
Lexicon based criterion	TDNN based criterion
$L_{MMIs} = \log \frac{P(O \lambda_{trueHMM})}{P(O \lambda_{bestHMM})}$	$L_{MMI_TDNN} = \log \frac{P(O \lambda_{trueHMM})}{P(O \lambda_{bestTDNN})}$

The training using the bare ML criterion only maximizes the true model regardless of the rest of the models. This does not give the recognizer any discriminative power. With such a criterion, there is a danger that all the weights of the NN are pulled to high values and finally do not converge to the optimal solution. This is referred as the collapse problem [11] and it corresponds to a fatal flaw in the training architecture unless softmax function is used at the output layer. In such a case the sum to 1.0 constraint forces all other character classes to be pushed down if a character class is pulled up. For the MMI criterion, the recognizer is trained to maximize the likelihood of the true model, and at the same time to minimize the likelihood of all other models. The two other expressions, given in Table 2, are each a simplified version of the MMI criterion. They considered, for the remaining models, only the model with the largest likelihood either from a given lexicon (L_{MMIs}) or without lexicon (L_{MMI_TDNN}).

4.1. A generic word level discriminative objective function

We have mixed the different components presented above in a generic objective function defined by the following relation:

$$L_c = (1 + \epsilon) \times \log P(O | \lambda_{trueHMM}) - \beta \times [(1 - \alpha) \log P(O | \lambda_{bestHMM}) + \alpha \log P(O | \lambda_{bestTDNN})]$$

$\alpha, \beta,$ and ϵ being mixture parameters belonging to $[0..1]$.

With $\epsilon = \beta = 0$, we get the bare ML function, whereas with $\beta = 1$ we introduce a discriminant training that takes into account either only the best word-HMM, if $\alpha = 0$, or only the best-TDNN classes if $\alpha = 1$. An intermediate α value interpolates between these two situations.

4.2 Neural network training

Once the objective function is defined, the training of the NN relies on the back-propagation of the gradient error function through the weight

matrices. The gradient of L_G with respect to the NN weights can be computed using the chain rule:

$$\frac{\partial L_G}{\partial W_{ji}} = \sum_t \frac{\partial L_G}{\partial v_j(O_t)} \cdot \frac{\partial v_j(O_t)}{\partial W_{ji}}$$

Where j is the index of the concerned neuron and i a neuron associated from the lower layer, t the temporal indication of observation and $v_j(O_t)$ the synaptic potential of the neuron j for the observation t ; $x_j(O_t) = f(v_j(O_t))$ the output of the neuron j and $x_j(O_t) = b_j(O_t)$ for the TDNN output layer with the standard HMM notation $\lambda(A, B, \pi)$.

Table 3. Gradient of the Objective function at the NN output level.

Output (j,t) = True HMM (j,t)	Output (j,t) = Best HMM (j,t)	Output (j,t) = Best TDNN (j,t)	Grad _{j,t} Gen	Grad _{j,t} ML (ε=0, β=0)	Grad _{j,t} MMI (ε=0, β=1)
F	F	F	0	0	0
F	F	T	-β α	0	- α
F	T	F	-β(1- α)	0	-(1- α)
F	T	T	-β	0	-1
T	F	F	1+ ϵ	1	1
T	F	T	1+ ϵ -β α	1	1- α
T	T	F	1+ ϵ -β(1- α)	1	1-(1- α)
T	T	T	1+ ϵ -β	1	0

By introducing $\delta_{j,t}$ the error term to calculate during the back propagation stage for every neuron, we obtain the following equation:

$$\frac{\partial L_G}{\partial W_{ji}} = \sum_t \frac{\partial L_G}{\partial v_j(O_t)} \cdot x_i(O_t) = \sum_t \delta_{j,t} \cdot x_i(O_t)$$

$$\text{with } \delta_{j,t} = \frac{\partial L_G}{\partial v_j(O_t)}$$

The back propagation in the TDNN hidden layers follows the standard algorithm, just taking in account the TDNN convolutional windows.

Skipping some intermediate calculation, due to lack of space, we obtain at last for the error term to retro-propagate: $\delta_{j,t} = Grad_{j,t} - x_{j,t} \sum_k Grad_{k,t}$ with

$$Grad_{j,t} = \left(\begin{array}{l} (1+\epsilon) \frac{P(O, q_t = j | \lambda_{\text{trueHMM}})}{P(O, \lambda_{\text{trueHMM}})} \\ -\beta * \left[(1-\alpha) \frac{P(O, q_t = j | \lambda_{\text{BestHMM}})}{P(O, \lambda_{\text{BestHMM}})} + \alpha \frac{P(O, q_t = j | \lambda_{\text{BestTDNN}})}{P(O, \lambda_{\text{BestTDNN}})} \right] \end{array} \right)$$

where $P(O, q_t = j | \lambda)$ is computed by dynamic programming (DP). So for each observation O_t , positive gradient is back propagated for the true HMM and negative gradient for best recognized HMM or best recognized TDNN path. The following table illustrates, according that an output of the NN is on the path (True) or not on the path (False) computed by the DP algorithm, the different values taken by the $Grad_{j,t}$ variable.

5. Results on the IRONOFF database

The whole training set of words (20 898 words representing 197 different labels, [12]) is used for training and a separate set of 10 448 words is used to test the system.

5.1. Comparison of criteria

The following table presents the recognition rates and the average gap regarding the classification function between the top 2 candidates. These results are obtained considering the different criteria with only one state by letter-HMM.

Table 4. Recognition rates on IRONOFF.

Criterion	MMIs (1)	MLE+MMIs (2)	MLE+TDNN (3)	Mixed (4)
	ε=0	ε=1	ε=1	ε=1
	β=1	β=1	β=1	β=1
TEST rate	83,82	86,34	82,26	87,09
Score First-Second	4,33	6,78	6,41	6,92

One important point is that the system is being able to converge and achieve quite reasonable recognition rates considering the relative simplicity of the letter-HMM models, which have only one state, and at the same time the important number of different letter classes (66). The simplified MMIs (1) performs better than the MLE+TDNN (3), which does not use the remaining words of the lexicon to train the system. The best recognition rate is achieved with the mixed criteria (4), which allows to reduce the error rate of nearly 20% with respect to the MMIs criterion. In this case, in addition to the best HMM model, the best TDNN outputs are also involved in the training of the system.

5.2 Multi-state models

We train the system with different models for a character : one state, two states, and three states. For each configuration, the number of outputs of the TDNN increases with the number of states S per letter (66×S output neurons).

Table 5 shows the interest to expand the number of states. In all the cases the MMI-MLE criterion achieves better results, and the 3-state model allows a 41% error rate reduction with respect to the basic 1-state model.

It is also worth noting the behavior of the system considering the ranked list of candidates, figure 2. It points out the possible gains that are achievable with some improvements of the system (variable number of states per letter, ligature model between

character) or the introduction of a language model to allow a re-scoring of the proposed list.

Table 5. Recognition performance for different number of states by letter models.

Criterion	Nb states by letter	1	2	3
MMI	Test recognition rate	83.82	87.46	88.22
	Average position	2.15	1.92	1.62
MMIMLE	Test recognition	86.34	90,57	92,01
	Average position	2.07	1,34	1,30

Figure 3 displays the recognitions rates according to the word length for the MMI-MLE criterion. Obviously the short words (up to 3 letters) are poorly recognized. This is mainly due the preprocessing steps, which are applied in the same way whatever the length of the word are, the core line extraction being specifically sensitive to the word length.

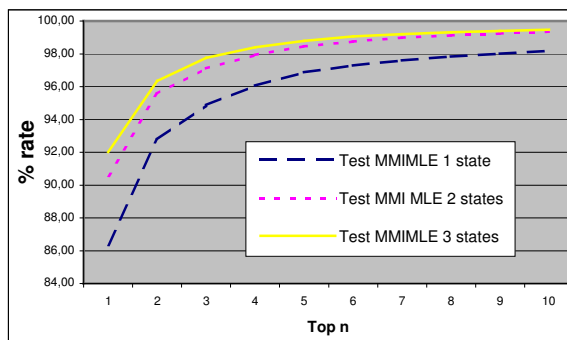


Figure 2. Top n Recognition rate

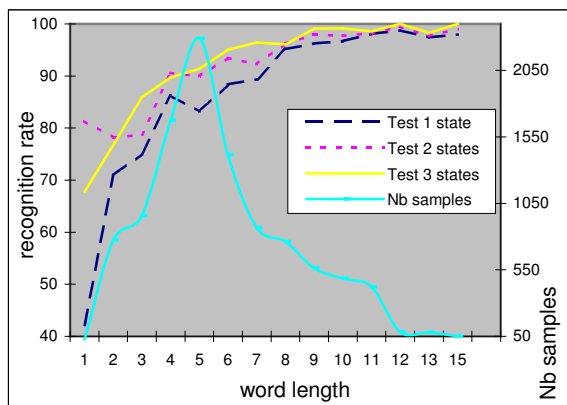


Figure 3. Recognition rate vs. word length

6. Conclusion and future work

We have pointed in this paper the feasibility of training a hybrid system using a generic objective function, mixing generative and discriminant modeling, defined at the word level. We obtain reasonable results without any specific initialization, and without explicit character segmentation, the best results being obtained with the 3-state configuration using the mixed MMI-MLE objective function.

Further investigations will continue to optimize the TDNN structure and the letter modeling. Furthermore, in order to improve the recognition rate, we plan to introduce spatial information with a hybrid TDNN-SDNN (Space Displacement NN) system.

7. References

- [1] E. Anquetil, H. Bouchereau, "Integration of an On-line Handwriting Recognition System in a Smart Phone Device", in Proceedings of the 16th IAPR International Conference on Pattern Recognition (ICPR 2002), pp. 192-195, Quebec, 2002.
- [2] L. Oudot, L. Prevost and M. Milgram, "An activation-verification model for on-line handwriting texts recognition", in Proc. of International Workshop on Frontiers for Handwriting Recognition (IWFHR'9), 2004.
- [3] K.S. Nathan, A.W. Senior and J. Subrahmonia. "Initialization of Hidden Markov Models for Unconstrained On-line Handwriting Recognition", in International Conference on Acoustics, Speech and Signal Processing (ICASSP 1996), pp. 3503-6, 1996.
- [4] J. Hu, S.G. Lim and M.K. Brown, "Writer independent on-line handwriting recognition using an HMM approach" *Pattern Recognition*, January 2000.
- [5] M. Schenkel, I. Guyon, D. Henderson. "On-line cursive script recognition using Time Delay Neural Networks and Hidden Markov Models". *Machine Vision and Applications*, special issue on Cursive Script Recognition, (8):215--223, 1995.
- [6] S. Jaeger, S. Manke, J. Reichert, A. Waibel, "On-Line Handwriting Recognition: The NPen++ Recognizer", *International Journal on Document Analysis and Recognition, IJDAR'00*, volume 3, p. 169-180, 2000.
- [7] Z. Wimmer, S. Garcia-Salicetti, A. Lifchitz, B. Dorizzi, P. Gallinari, T. Artières, « REMUS », <http://www-connex.lip6.fr/~lifchitz/Remus/>.
- [8] W.T. Chen, P. Gader, "Word level discriminative training for handwritten word recognition", Proc. 7th IWFHR, ISBN 90-76942-01-3, pp. 393-402, 2000.
- [9] Y.H. Tay, P.M. Lallican, et al., "An Analytical Handwritten Word Recognition System with Word-Level Discriminative Training", Proc. Sixth ICDAR, pp. 726-730, Seattle, Sept. 2001.
- [10] S. Knerr, E. Augustin, "A Neural Network-Hidden Markov Model Hybrid for cursive Word Recognition", Proc. ICPR, 1998.
- [11] E. Trentin, M. Gori, "A survey of hybrid ANN/HMM models for automatic speech recognition", *Neurocomputing*, vol. 37, pp. 91-126, March 2001.
- [12] C. Viard-Gaudin, P.M. Lallican, S. Knerr, P. Binter, «The IRONOFF Dual Handwriting Database», Proc. 5th ICDAR, pp. 455-458, 1999.