Training of hybrid ANN/HMM systems for on-line handwriting word recognition

Emilie CAILLAULT[1], Christian VIARD-GAUDIN[1], Pierre-Michel LALLICAN[2]

[1] Laboratoire IRCCyN UMR CNRS 6597
École polytechnique de l'université de Nantes
Rue Christian Pauc – FR 44306 Nantes cedex 3
{emilie.caillault;christian.viard-gaudin}@univ-nantes.fr

[2] Vision Objects
9 rue du Pavillon
FR 44980 Sainte Luce sur Loire
pmlallican@visionobjects.com

Abstract: On-line handwriting word recognition systems usually rely on hidden Markovs models (HMMs), 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. Training such a hybrid system is not straightforward, this is why not so many attempts are encountered in literature. This paper proposes several different training schemes mixing maximum likelihood (ML) and maximum mutual information (MMI) criteria in the framework of online handwriting recognition with a global optimisation approach defined at the world level.

Paper category: On-going research paper

Keywords: Hybrid TDNN/HMM, global training, MMI/MLE criterion, Discriminant criteria, online cursive handwriting.

Conference Topic(s): Handwriting recognition: On-line and off-line recognition.

1. Introduction

Handwriting word recognition (HWR) can be defined as the classification of the correct word from a given lexicon, according to the word posterior probability. The following elements, namely the writing signal *x*, the word to be recognized *w*, the language model and the handwriting models being linked by the well known Bayes relation (Eq. 1).

$$P(w|x) = \frac{p(x|w)P(w)}{p(x)} \qquad (1) \qquad\qquad \hat{w} = \arg\max_{w} \left\{ p(x|w)P(w) \right\} \qquad (2)$$

Within this relation, the language model accounts for the *a priori* probability *P(w)* for a given word *w*, whereas handwriting models compute the likelihood of an observed signal *x* for a given word *w*. Consequently, the result of the recognition system will consist in maximizing the *a posteriori* probability *P(w|x)*. Therefore, the selected word will be defined by (Eq. 2) using the MAP (Maximum A Posteriori) criterion.

The quantity *p(x|w)*, known as the handwriting model, describes the statistics of sequences of parameterised handwriting observations in the feature space given the corresponding written words. HMMs [1] are the most popular parametric models at the word level. Although HMMs yield good recognition performances under different word recognition experiments [2][3][4], they suffer from some limitations [5]. The assumption of a specific parametric probability density function that describes the emission probabilities associated with the states is arbitrary and constraining. In addition, some statistical independence among input features is usually assumed to simplify parameter estimation. Moreover, the objective function used during learning, based on the Maximum Likelihood criterion, does not guarantee the highest possible classification rate. Based on these remarks, the use of discriminative learning approaches, such as those used with ANN, appears promising: ANN can separate more easily very complex data than generative models [6], they can be trained as non-parametric probabilistic models that exhibit very good generalization capabilities. The idea of combining ANN and HMMS altogether in a hybrid system has been first proposed in the speech community [7][8], and extended soon in the handwriting domain [9].

The training techniques are not straightforward, since back propagation (BP) requires knowledge of the target outputs to compute the gradient of the cost function. In the first attempts, the trainings were done separately and iteratively between ANN and HMMs [10]. The simplicity of the system was counterbalanced by the lack of a global optimisation scheme for the whole system at the word level.

This paper proposes a hybrid architecture for recognizing unconstrained online handwritten words. It is based on a Time Delay Neural Network (TDNN), for the ANN part, and of single-state models for the HMMs at the letter level. The word model being built by a concatenation of the corresponding letters. First, we introduce in section 2 the global system, then we focus on the training stage involved with the such a system, and describe the derivation of a gradient-based algorithm to train the TDNN. Experimental results are reported in section 4, IRONOFF [11] database has been used to evaluate the convergence of the training procedure and the recognition performances.

2. Global presentation of the TDNN/HMM 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, $X_{1,N} = (x_1,\ldots, x_N)$, where x_i describes the ith 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.

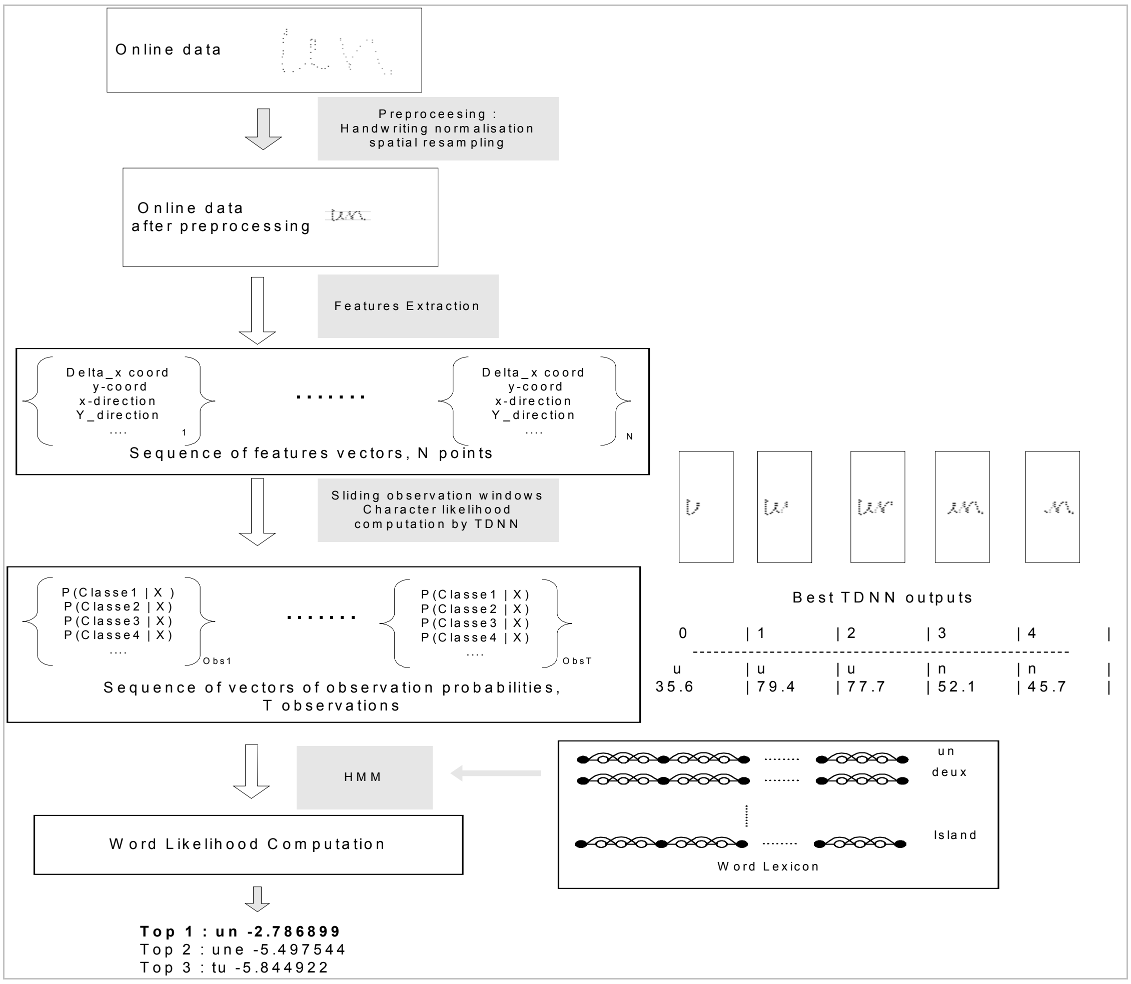

Figure 1. Overview of the on-line cursive words recognition system.

As a NN, we have used in a previous work a standard multi-layer perceptron (MLP) [12] with an explicit multi-segmentation scheme, whereas in this work, we have privileged a TDNN [9] 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 classes: lowercases, uppercases, accents and symbols). Observation probabilities in each emitting state of the basic HMMs are computed by the NN. Transition probabilities model the duration of the letters, actually, as we assume the same duration 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.

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 not a training explicitly at the character level but an optimization of the network to satisfy an objective function defined at the global word level.

3. Word-Level training criteria

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 1 : Objective functions at the word level.

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

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 discriminant 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 [5] 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 1, 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}).

3.1 A generic word level discriminant objective function

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

$$L_G=(1+\varepsilon)\log P(O|\lambda_{trueHMM})-\beta\times[(1-\alpha)\log P(O|\lambda_{bestHMM})+\alpha\log P(O|\lambda_{bestTDNN})] \quad (3)$$

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

With $\varepsilon=\beta=0$, we get the bare ML function, whereas with $\beta=1$ we introduce a discrimination 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.

3.2 Neural network training

Once the objective function is defined, the training of the NN relies on the back-propagation of the gradient error function trough the weight matrices. The gradient of L_G with respect to the NN weights (Eq. 4) can be computed using the chain rule:

$$\frac{\partial L_G}{\partial W_{ji}}=\sum_t \frac{\partial L_G}{\partial v_j(O_t)}\cdot\frac{\partial v_j(O_t)}{\partial W_{ji}} \quad (4)$$

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 neurone j and $x_j(O_t) = b_j(O_t)$ for the TDNN output layer with the HMM notation $\lambda(A,B,\pi)$ [1].

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

$$\frac{\partial L_G}{\partial W_{ji}} = \sum_t \frac{\partial L_G}{\partial v_j(O_t)} \cdot x_i(O_t) \;=\; \sum_t \delta_{j,t} \cdot x_i(O_t) \qquad with \qquad \delta_{j,t} = \frac{\partial L_G}{\partial v_j(O_t)} \tag{5}$$

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: $\delta_{j,t} = Grad_{j,t} - x_{j,t} \sum_k Grad_{k,t}$ (6)

with $Grad_{j,t} = \left((1+\varepsilon) \frac{P(O, q_t = j | \lambda_{trueHMM})}{P(O, | \lambda_{trueHMM})} - \beta * \left[(1-\alpha) \frac{P(O, q_t = j | \lambda_{BestHMM})}{P(O, | \lambda_{BestHMM})} + \alpha \frac{P(O, q_t = j | \lambda_{BestTDNN})}{P(O, | \lambda_{BestTDNN})} \right] \right)$ (7)

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.

Table 2 : Gradient of the Objective function at the NN output level.

Output(j,t) = TrueHMM(j,t)	Output(j,t) = BestHMM(j,t)	Output(j,t) = BestTDNN(j,t)	$Grad_{j,t\,t}$-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	$-\beta\alpha$	*0*	$-\alpha$
F	T	F	$-\beta(1-\alpha)$	*0*	$-(1-\alpha)$
F	T	T	$-\beta$	*0*	-1
T	F	F	$1+\varepsilon$	*1*	*1*
T	F	T	$1+\varepsilon-\beta\alpha$	*1*	$1-\alpha$
T	T	F	$1+\varepsilon-\beta(1-\alpha)$	*1*	$1-(1-\alpha)$
T	T	T	$1+\varepsilon-\beta$	*1*	*0*

4. Experiments and Results

4.1. Training results for one single word

The first experiments consist in evaluating the behavior of the different versions of the objective criterion on the task of learning a single word extracted from the IRONOFF database [11]. We conduct the experiments with the French word “deux” (two), which has been written by 283 different writers. First, we decide to use only one sample to learn the word and to test the generalization capability on the 282 remaining words. We stop the back propagation (BP) iterations as soon as the word used to train the system is well recognized, cf. Table 3-(A). A second experiment uses all the samples of the word for training except one, and test the system on the remaining sample. In that case, 20 epochs of the training set is used, cf. Table 3-(B).

Table 3 : Comparison of training criteria on one example (word "deux").

Criterion		MMIs (1) $\varepsilon=0\ \beta=1\ \alpha=0$	MLE + MMIs(2) $\varepsilon=1\ \beta=1\ \alpha=0$	MLE + TDNN (3) $\varepsilon=1\ \beta=1\alpha=1$	Mixed (4) $\varepsilon=1\ \beta=1\ \alpha=0.5$
(A)	BP iterations	4	87	74	66
	Training Log-likelihood score	**Top 1: deux -29.57** Top 2: dix -29.58 Top 3: de -29.583	**Top 1: deux -9.181** Top 2: du -9.242 Top 3: de -9.253	**Top 1: deux -8.78** Top 2: dix -8.84 Top 3 : six -9.51	**Top 1 : deux -9.62** Top 2 : dix -9.665 Top 3 : six -9.960
	Test recognition rate (1ex trained, 282 others in test)	Top 1 : 24.82 % Top 2 : 34.39 % **Top 45 : 100.00 %**	Top 1 : 0.3546 % Top 2 : 0.3546 % **Top 10 : 100.00 %**	Top 1 : 6.73 % Top 2 : 73.04 % **Top 3 : 100.00 %**	Top 1 : 8.51 % Top 2 : 39.00 % **Top 3 : 100.00 %**
(B)	BP iterations	20×282	20×282	20×282	20×282
	Test recognition rate (282 trained – 1 test)	**Top 1 : 100.00 %**	Top 1 : 99,64 % **Top 2 : 100.00 %**	Top 1 : 98,58 % **Top 2 : 100.00 %**	Top 1 : 98.58 % Top 2 : 99.29 % **Top 3 : 100.00 %**

Of course, using only one sample (A) to train the system leads to poor results. Nevertheless, it is worth noting that MMIs criterion (1) is able very quickly, with only 4 BP iterations, to push the correct word at the top of lexicon (197 words), but it leads to poor generalization results. While with criteria (3) and (4), the training is longer but the generalization capability is better since we reach 100 % of correct recognition within the Top 3 candidates. Conversely, when more samples are used to trained the system (B), the MMIs (1) criterion allows a very good recognition rate, with no error on the test set, the other criteria being also quite satisfying.

An other interesting result is the evolution of the discrimination power of these different criteria. Figure 2 displays the difference between the top 2 candidates. With MMIs (1) criterion, as soon as the training word is at the top1 position, no longer modification of the TDNN is done (since $Grad_{j,t} = 0$), and consequently the difference of likelihoods score remains constant and very close to zero. Whereas with the three other criteria, the difference between the likelihood of the top1 model, which is the true model most of the time, and the second best model still increases even when the word is correctly recognized, meaning that we achieve a better and better modeling of the true model ant at the same time a better discrimination with the remaining set of words of the lexicon.

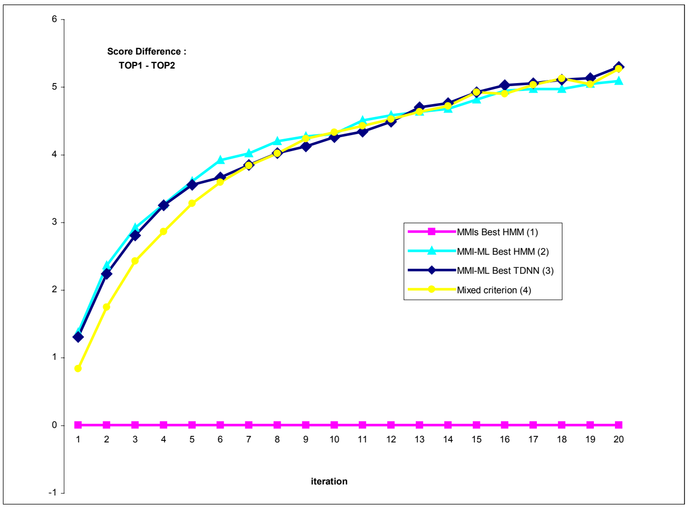

Figure 2 : Difference between scores of Top 1 and Top 2-position words

4.3. Results on the whole IRONOFF database

The whole training set of words (20 898 words representing 197 different labels) is now used for training and a separate set of 10 448 words is used to test the system. The following table presents the results obtained considering the different criteria.

Table 4 : Comparison of recognition rate on IRONOFF database.

Criterion	MMIs (1) $\varepsilon=0\ \beta=1\ \alpha=0$	MLE + MMIs(2) $\varepsilon=1\ \ \beta=1\ \alpha=0$	MLE + TDNN (3) $\varepsilon=1\ \ \beta=1\alpha=1$	Mixed (4) $\varepsilon=1\ \ \beta=1\ \alpha=0.5$
N° epoch	68	99	158	129
TRAINING set rate	83.92	83.82	79.73	87.09
TEST set rate	78.09	81.30	77.36	83.42

One important point is that the system is still being able to converge and achieve quite reasonable recognition rates considering the relative simplicity of the HMM letter 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 23% with respect to the MMIs criterion. In the former case, in addition to the best HMM model, the best TDNN outputs are also involved in the training of the system.

5. Conclusion

We have presented a global scheme defined at the word level to train with different criteria an online handwriting unconstrained word recognition system. All of these criteria show experimentally a convergence of the training process, and the combination of a discriminative learning, based on a MMI criterion, and of a generative modeling based on a MLE criterion gives the best results. An extension of this work using a multi-state modeling at the letter level is currently under development.

6. References

[1] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE*, vol. 77, pp. 257-285, 1989.

[2] M. Gilloux, M. Leroux, J-M Bertille , "Strategies for cursive script recognition using hidden Markov models", *Machine Vision and Applications*, Volume 8 Issue 4, pp 197-205, 1995.

[3] S. Knerr et al, "Hidden Markov Model Based Word Recognition and Its Application to Legal Amount Reading on French Checks", *Computer Vision and Image Understanding*, vol. 70-3, pp. 404-419, 1998.

[4] R. Plamondon, S.N. Srihari, "On-Line and Off-line Handwriting Recognition: A Comprehensive Survey", *IEEE Transactions on PAMI*, Vol.22, No. 1, pp.63-84, 2000.

[5] E. Trentin, M. Gori, "A survey of hybrid ANN/HMM models for automatic speech recognition", *Neurocomputing*, vol. 37, pp. 91-126, March 2001.

[6] C. Bishop, Generative versus Discriminative Methods in Computer Vision**.** Invited Keynote talk at ICPR 2004, Cambridge, presented on 24 August, 2004.

[7] M.A. Franzini, K.F. Lee, A. Waibel, "Connectionist Viterbi training: a new hybrid method for continuous speech recognition', Intern. Conf. on Acoustics, Speech and Signal Processing, Albuquerque, pp. 417-420, 1990.

[8] G. Rigoll, "Maximum mutual information neural networks for hybrid connectionist-HMM speech recognition systems", *IEEE trans. on Speech and Audio Processing*, vol. 2, n° 1, pp. 175-1184, January 1994.

[9] M. Schenkel and I. Guyon and D. Henderson. On-line cursive script recognition using time delay neural networks and hidden Markov models. Proc. ICASSP '94.1994.

[10] H. Bourlard, N. Morgan, "Continuous speech recognition by connectionist statistical methods", *IEEE Trans. On Neural Networks*, vol. 4, n° 6, pp. 893-909, Nov. 1993.

[11] C.Viard-Gaudin, P.-M. Lallican, et al..The Ireste ON/OFF (IRONOFF) Dual Handwriting Database. Fifth International Conference on Document Analysis and Recognition (ICDAR).1999.

[12] Y.H. Tay, P.M. Lallican, et al., "An Analytical Handwritten Word Recognition System with Word-Level Discriminant Training", Proc. Sixth ICDAR, pp. 726-730, Seattle, Sept. 2001.