SEQUENCE-DISCRIMINATIVE TRAINING OF RECURRENT NEURAL NETWORKS

Paul Voigtlaender[1], *Patrick Doetsch*[1], *Simon Wiesler*[1], *Ralf Schlüter*[1], *Hermann Ney*[1,2]

[1]Human Language Technology and Pattern Recognition,
Computer Science Department, RWTH Aachen University, Aachen, Germany
[2]LIMSI CNRS, Spoken Language Processing Group, Paris, France

ABSTRACT

We investigate sequence-discriminative training of long short-term memory recurrent neural networks using the maximum mutual information criterion. We show that although recurrent neural networks already make use of the whole observation sequence and are able to incorporate more contextual information than feed forward networks, their performance can be improved with sequence-discriminative training. Experiments are performed on two publicly available handwriting recognition tasks containing English and French handwriting. On the English corpus, we obtain a relative improvement in WER of over 11% with maximum mutual information (MMI) training compared to cross-entropy training. On the French corpus, we observed that it is necessary to interpolate the MMI objective function with cross-entropy.

Index Terms— recurrent neural networks, long short-term memory, sequence-discriminative training, handwriting recognition

1. INTRODUCTION

Neural networks (NNs) have become a key component of modern automatic speech recognition (ASR) and handwriting recognition (HWR) systems. Framewise trained bidirectional long short-term memory recurrent neural networks (BLSTM-RNNs) have been used on several handwriting recognition tasks with great success [1]. In particular on tasks that require long-range contextual information, LSTM-RNNs have been shown to outperform regular feed forward network structures.

Usually, neural networks are trained on frame-level using the cross-entropy (CE) criterion. However, the recognition is done on sequence-level and also incorporates additional knowledge sources like the language model. Furthermore, the evaluation criterion is the word error rate (WER). This mismatch between training and recognition suggests that using training criteria which are based on whole sequences and incorporate additional knowledge sources can be helpful.

Recently, significant improvements using sequence-discriminative criteria for feed forward networks have been reported in literature [2, 3, 4, 5]. Sequence-discriminative training criteria have already been used decades ago for discriminative training of Gaussian mixture hidden Markov models (GHMMs) [6]. Kingsbury et al. [7] proposed using the same lattice-based framework which has been developed for discriminative GHMM training for neural networks.

In this work, we evaluate sequence-discriminative training of deep bidirectional LSTM-RNNs. Since RNNs already make use of the whole sequence and are able to incorporate more contextual information than feed forward networks, it is not clear, whether they can benefit from sequence-discriminative training as much as feed forward networks do. Nevertheless, we find sequence-discriminative training to give substantial improvements over our cross-entropy LSTM-RNN baselines.

Very recently and independently of this work, sequence-discriminative training has been applied to unidirectional LSTM-RNNs for ASR on an in-house database [8]. Our work differs from [8] in that we apply sequence-discriminatively trained LSTM-RNNs to two handwriting recognition tasks, which are publicly available and heavily evaluated. Further, we use bidirectional LSTM-RNNs, a more complex topology, which has been found to be advantageous over unidirectional LSTM-RNNs.

2. LSTM-RNNS IN THE HYBRID APPROACH

In our system, we use the hybrid approach [9] to combine an HMM with a neural network to a neural network hidden Markov model (NN-HMM). The emission probability $p(x|s)$ of observation vector x given HMM state s can be rewritten by Bayes rule as

$$p(x|s) = \frac{p(s|x)p(x)}{p(s)^\alpha} \tag{1}$$

with prior scaling factor α. $p(x)$ is independent of s and can therefore be dropped in search. The state posterior probability $p(s|x)$ is estimated by the neural network.

RNNs extend feed forward networks by recurrent connections. The output of layer l at time t is used as an additional

This work was partially supported by a Google Research Award. H. Ney was partially supported by a senior chair award from DIGITEO, a French research cluster in Ile-de-France.

input of that layer at time $t+1$ which leads to the following equation for the output $y^{(l)}(t)$ of layer l at time t:

$$y^{(l)}(t) = \sigma\left(W^{(l)}y^{(l-1)}(t) + V^{(l)}y^{(l)}(t-1) + b^{(l)}\right), \quad (2)$$

where $W^{(l)}$ is the weight matrix for the output of the underlying layer, $V^{(l)}$ is the recurrent weight matrix, $b^{(l)}$ is the bias of layer l and σ is the activation function. The recurrent connections give the network a memory and enable it to make use of more contextual information. To train RNNs, we use backpropagation through time (BPTT) [10] which is a modification of backpropagation for RNNs. However, training RNNs is a difficult problem as the gradients tend to either blow up or decay exponentially when they are backpropagated through time which is known as the vanishing gradient problem [11]. An effective solution to this problem is the use of the long short-term memory (LSTM) [12] architecture which also further improves the use of contextual information. LSTM introduces the concept of memory cells which are protected by an input, an output and a forget gate which control the flow of information into and out of the cell and has been shown to yield good performance on handwriting recognition tasks [1, 13]. All RNNs in this work are bidirectional [14], which means that the hierarchy of hidden layers is replaced by two independent hierarchies of layers, one which processes its input in forward direction and one which processes its input backwards in time. The outputs of the hierarchy of forward and backward layers are recombined at the output layer which uses a softmax activation function to predict the state posteriors.

3. TRAINING CRITERIA

Let $\{(X_r, W_r)\}_{r=1}^{R}$ be the training set composed of R text line images, each consisting of an observation sequence $X_r := (x_{1,r} \dots x_{T_r,r})$ and a reference transcription $W_r := (w_{1,r} \dots w_{N_r,r})$. The parameters of the neural network are optimized with respect to a training criterion using this training set.

3.1. Cross-entropy training

The cross-entropy criterion (CE) is the most common training criterion for hybrid NN-HMMs. For the CE criterion, an alignment $\{S_r\}_{r=1}^{R}$ with $S_r := (s_{1,r} \dots s_{T_r,r})$ is required for each sequence. The alignment can be computed using the Viterbi algorithm, for example using a GHMM baseline system. The CE objective function for a set of parameters θ is defined as

$$F_{\text{CE}}(\theta) = -\sum_{r=1}^{R}\sum_{t=1}^{T_r} \log p_\theta(s_{t,r}|x_{t,r}). \quad (3)$$

The objective function is optimized with each frame as independent observation. Sequential properties of the input sequence are not covered by the CE objective function.

3.2. Sequence-discriminative training

CE only optimizes the decision on frame-level and does not take the other knowledge sources of the speech recognition system into account. In contrast, sequence-discriminative training criteria are based on whole sequences rather than frames and are formulated in context of the recognition system. In this initial work, we only consider the maximum mutual information (MMI) criterion

$$F_{\text{MMI}}(\theta) = -\sum_{r=1}^{R} \log p(W_r|X_r) \quad (4)$$

$$= -\sum_{r=1}^{R} \log \frac{p(W_r)p_\theta(X_r|W_r)^{\frac{1}{\gamma}}}{\sum_{\tilde{W}} p(\tilde{W})p_\theta(X_r|\tilde{W})^{\frac{1}{\gamma}}}\,. \quad (5)$$

For sequence-discriminative training, usually a weakened LM is used [15]. Further, the acoustic likelihoods are scaled by the inverse of the language model scale $\gamma > 0$ used in recognition.

MMI involves a sum over all possible word sequences, which is approximated by word lattices. As for discriminative GHMM training, the MMI derivative can be computed with the forward-backward algorithm on lattices. Once the MMI derivative with respect to the network output is known, it can be optimized with any gradient-based optimization algorithm.

When training NNs with sequence-discriminative training criteria, in some cases, additional heuristics are needed to obtain improvements compared to the CE baseline [4, 3]. Su et al. [4] conjectured that these problems are related to the sparseness of the lattices. Only a fraction of all labels are contained in the lattice at every time frame. As a solution, they proposed the F-smoothing heuristic which interpolates the MMI criterion with the cross-entropy criterion. The combined objective function is then

$$F_{\text{FSMMI}} = (1 - H) \cdot F_{\text{CE}} + H \cdot F_{\text{MMI}} \quad (6)$$

with $H \in (0, 1)$.

4. IMPLEMENTATION

Our GPU-based implementation of LSTM-RNNs is done in Python using the Theano library [16] which supports automatic differentiation. To calculate the MMI gradient with respect to the network outputs, we combine the Python software with the RWTH Aachen University Open Source Speech Recognition Toolkit (RASR) [17]. RASR does not support RNNs yet, but supports the calculation of MMI derivatives. For every mini-batch of sequences, the python software calculates the softmax activations and transfers them to RASR using a shared memory interface. RASR then uses the activations to calculate the MMI derivatives with respect to the network outputs and transfers them back to the python software which performs the BPTT procedure.

5. EXPERIMENTAL SETUP

Experiments were conducted on two publicly available databases containing offline images of English and French handwriting. A more detailed description of all preprocessing steps and our HMM recognition system is given in [13].

5.1. Databases

For English HWR we used the IAM database [18], which consists of handwritten English sentences. The data is divided into 747 paragraphs for training, 116 paragraphs for development and 336 for testing. A smoothed trigram word-based language model trained on three text corpora lead to a perplexity of 420 and an out of vocabulary (OOV) rate of 4% on the development set. In order to deal with the OOVs, a 10-gram character based language model with a character inventory of 77 characters was trained and combined with the word based language model [19]. The unigram LM used to create lattices on the training data has a perplexity of 310.

The RIMES database [20] is a corpus for French HWR. The training set of the RIMES database contains 1500 sentences and 100 sentences are provided to evaluate the model. Here we use a 4-gram word-based LM with a perplexity of 23 on the validation set and a lexicon with a character inventory of 96 base characters. The unigram LM has a perplexity of 310.

5.2. Preprocessing

The images were cleaned using standard preprocessing techniques like contrast normalization and deslanting. Afterwards image slices are extracted by an overlapping sliding window of width 8. Then the slices are translated to their center of gravity and further normalized by their standard deviation in order to generate fixed-size input frames of 8×32 pixels. The resulting 256-dimensional vector is reduced by PCA to 20 components and augmented by its original moments in horizontal and vertical direction resulting in a 24 dimensional feature vector. On the IAM and RIMES databases $3,742,335$ and $6,991,867$ frames were generated from the training sets, respectively.

5.3. HMM modeling

At first we trained an HMM with a fixed number of six states per character where each of the emission probability distributions was modeled by a Gaussian mixture model (GMM) with 128 densities per state. Afterwards a forced alignment generated by the HMM was used to generate character length statistics to estimate a character dependent variable state HMM topology. With this method we obtained 563 state labels on the IAM database and 702 state labels on the RIMES database which serve as targets for NN training.

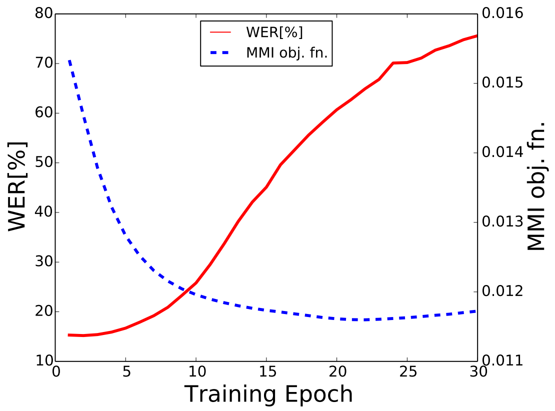

Fig. 1: Evolution of the MMI objective function and WER without heuristics.

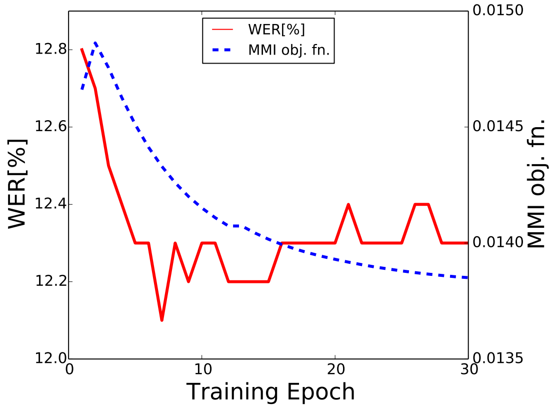

Fig. 2: Evolution of the unsmoothed MMI objective function and WER with active F-smoothing.

6. EXPERIMENTAL RESULTS

Experiments were conducted on the IAM and RIMES handwriting corpora (see Section 5.1). In both cases we first trained an HMM baseline system and obtained an alignment with it. This alignment was then used to train a neural network with CE. In the next step, lattices and lattice alignments were created using the CE network and a unigram LM. The parameters of the CE trained network are then used as initialization for MMI training. The bidirectional LSTM-RNNs were composed of an input layer with one unit for each component of the 24-dimensional input vector, a variable number of hidden layers with 500 memory cells (both forward and

Table 1: Recognition performance on the IAM database.

System	Criterion	WER[%]		CER[%]	
		Dev	Eval	Dev	Eval
GMM	ML	10.7	-	3.8	-
MLP	CE	10.0	14.7	3.3	6.1
	MMI	9.6	13.7	3.1	5.6
LSTM-RNN	CE	9.8	13.8	3.0	5.2
	MMI	8.7	12.7	2.6	4.8

Table 2: Comparison of the proposed system to results reported by other groups on the IAM database.

System	WER[%]		CER[%]	
	Dev	Eval	Dev	Eval
Our system	8.7	12.7	2.6	4.8
Doetsch et al. [21]	8.4	12.2	2.5	4.7
Kozielski et al. [13]	9.5	13.3	2.7	5.1
Pham et al. [23]	11.2	13.6	3.7	5.1

backward) in each layer and a softmax output layer with one unit per state label. Network training was done using BPTT without truncation and stochastic gradient descent using a batch size of 30 sequences, no momentum, and an empirically optimized learning rate for each network. MMI training was performed for at least 30 epochs with a duration of about two hours each.

6.1. IAM

On the IAM corpus, we trained LSTM-RNNs with one to five hidden layers and obtained the best results with three hidden layers. For comparison, we additionally trained multilayer perceptrons (MLPs) with sigmoid units and different numbers of hidden layers and nodes. The best MLP result was obtained with four hidden layers of 2048 nodes each. Table 1 shows our results on IAM. The best LSTM result with CE on the evaluation set was a WER of 9.8% which was improved by MMI by more than 11% relative to 8.7%. We did not use the heuristics of Section 3.2 on IAM, as the training already worked reliably without them and prior experiments with feed forward networks did not show any benefits when using them.

Table 2 compares our results with results reported by other groups on IAM. Doetsch et al. [21] use a modification of the LSTM architecture. Kozielski et al. [13] use discriminatively trained HMMs in the tandem approach [22]. Pham et al. [23] use multidimensional [24] LSTM networks with a Connectionist Temporal Classification (CTC) [25] output layer.

Table 3: Recognition performance on the validation set of the RIMES database.

System	Criterion	WER[%]	CER[%]
GMM	ML	15.7	5.5
LSTM-RNN	CE	12.6	4.4
	MMI	12.1	4.4

Table 4: Comparison of the proposed system to results reported by other groups on the RIMES database.

System	WER[%]	CER[%]
Our system	12.1	4.3
Doetsch et al. [21]	12.9	4.3
Kozielski et al. [13]	13.7	4.6
Pham et al. [23]	12.3	3.3

6.2. RIMES

On the RIMES corpus, we only trained a single LSTM network with three hidden layers with 500 memory cells each. Without the use of heuristics, the WER quickly increased to over 70% when trained with MMI, although the MMI objective function improved (see Figure 1). In order to understand if this problem is related to RNNs, we also trained a multilayer perceptron (MLP) with a single hidden layer. It turned out, that without the use of heuristics, the WER increased even faster for the MLP than for the RNN. In order to improve the result, we used the F-smoothing heuristic (see Section 3.2) and found that a value of $H = 0.75$ yielded good results, although the WER was not very sensitive to the exact value of H. Figures 1 and 2 show the evolution of the MMI part of the objective functions (normalized by the number of frames) and WER during training wit and without F-smoothing, respectively. Note that for Figure 2, the MMI part was rescaled by dividing by H for better comparability. Without F-smoothing the WER quickly degrades. With active F-smoothing, the CE baseline of 12.6% is improved to 12.1% (see Table 3).

Table 4 compares our results with results reported by other groups on RIMES. The systems of Doetsch et al. [21], Kozielski et al. [13], and Pham et al. [23] are the same as on IAM.

7. CONCLUSION

We evaluated sequence-discriminative training of deep LSTM-RNNs using MMI on two real-world handwriting recognition tasks. We obtained a relative improvement in WER of more than 11% on the IAM corpus, which shows that the performance of state-of-the-art LSTM-RNN models can be significantly improved by sequence-discriminative training. We showed that the F-smoothing heuristic is helpful on corpora on which simple MMI does not yield improvements.

8. REFERENCES

[1] Marcus Liwicki, Alex Graves, Horst Bunke, and Jürgen Schmidhuber, “A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks,” in *Proc. of the Int. Conf. on Document Anal. and Recognition (ICDAR)*, 2007, pp. 367 – 371.

[2] Brian Kingsbury, Tara N. Sainath, and Hagen Soltau, “Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization.,” in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc. (Interspeech)*. 2012, ISCA.

[3] Karel Veselý, Arnab Ghoshal, Lukáš Burget, and Daniel Povey, “Sequence-discriminative training of deep neural networks,” in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc. (Interspeech)*, August 2013.

[4] Hang Su, Gang Li, Dong Yu, and Frank Seide, “Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription,” in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, 2013.

[5] Georg Heigold, Erik McDermott, Vincent Vanhoucke, Andrew Senior, and Michiel Bacchiani, “Asynchronous stochastic optimization for sequence training of deep neural networks,” in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 5624–5628.

[6] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, “Maximum mutual information estimation of hidden Markov model parameters for speech recognition,” in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Tokyo, 1986, pp. 49–52.

[7] Brian Kingsbury, “Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling,” in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, 2009, pp. 3761–3764.

[8] Hasim Sak, Oriol Vinyals, Georg Heigold, Andrew Senior, Erik McDermott, Rajat Monga, and Mark Mao, “Sequence discriminative distributed training of long short-term memory recurrent neural networks,” in *Proc. of the Ann. Conf. of the Int. Speech Commun. Assoc. (Interspeech)*, 2014.

[9] Herve A. Bourlard and Nelson Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[10] R. J. Williams and D. Zipser, “Gradient-based learning algorithms for recurrent networks and their computational complexity,” in *Back-propagation: Theory, Architectures and Applications*, Y. Chauvin and D. E. Rumelhart, Eds., chapter 13, pp. 433–486. Hillsdale, NJ: Erlbaum, 1995.

[11] Sepp Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, pp. 107–116, Apr. 1998.

[12] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] Michal Kozielski, Patrick Doetsch, and Hermann Ney, “Improvements in rwth’s system for off-line handwriting recognition,” in *Proc. of the Int. Conf. on Document Anal. and Recognition (ICDAR)*, 2013, pp. 935–939.

[14] M. Schuster and K.K. Paliwal, “Bidirectional recurrent neural networks,” *Trans. on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[15] R. Schlüter, B. Müller, F. Wessel, and H. Ney, “Interdependence of language models and discriminative training,” in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Keystone, CO, USA, Dec. 1999, pp. 119–122.

[16] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio, “Theano: a CPU and GPU math expression compiler,” in *Proc. of the Python for Scientific Computing Conference*, June 2010.

[17] David Rybach, Stefan Hahn, Patrick Lehnen, David Nolden, Martin Sundermeyer, Zoltán Tüske, Simon Wiesler, Ralf Schlüter, and Hermann Ney, “Rasr - the rwth aachen university open source speech recognition toolkit,” in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Hawaii, USA, Dec. 2011.

[18] U.-V. Marti and H. Bunke, “The IAM-database: an english sentence database for offline handwriting recognition,” *Int. J. on Document Anal. and Recognition*, vol. 5, no. 1, pp. 39–46, Nov. 2002.

[19] Michal Kozielski, David Rybach, Stefan Hahn, Ralf Schlüter, and Hermann Ney, “Open vocabulary handwriting recognition using combined word-level and character-level language models,” in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, Vancouver, Canada, May 2013, pp. 8257–8261.

[20] E. Grosicki and H. El Abed, “ICDAR 2009 handwriting recognition competition,” in *Proc. of the Int. Conf. on Document Anal. and Recognition (ICDAR)*, July 2009, pp. 1398 –1402.

[21] Patrick Doetsch, Michal Kozielski, and Hermann Ney, “Fast and robust training of recurrent neural networks for offline handwriting recognition,” in *International Conference on Frontiers in Handwriting Recognition*, Crete, Greece, Sept. 2014.

[22] Hynek Hermansky, Daniel P. W. Ellis, and Sangita Sharma, “Tandem connectionist feature extraction for conventional hmm systems,” in *Proc. of the Int. Conf. on Acoust., Speech, and Signal Process. (ICASSP)*, 2000, pp. 1635–1638.

[23] Vu Pham, Thodore Bluche, Christopher Kermorvant, and Jrme Louradour, “Dropout improves recurrent neural networks for handwriting recognition,” in *Int. Conf. on Frontiers in Handwriting Recognition*, 2014.

[24] Alex Graves and Jürgen Schmidhuber, “Offline handwriting recognition with multidimensional recurrent neural networks,” in *Advances in Neural Inform. Process. Syst. (NIPS)*, Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, Eds. 2008, pp. 545–552, Curran Associates, Inc.

[25] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proc. of the Int. Conf. on Mach. Learning (ICML)*, William W. Cohen and Andrew Moore, Eds. 2006, vol. 148 of *ACM International Conference Proceeding Series*, pp. 369–376, ACM.