21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Long-Short Term Memory Neural Networks Language Modeling for Handwriting Recognition

Volkmar Frinken
Centre de Visió per Computador
Universitat Autònoma de Bacelona
Bellaterra, Spain
vfrinken@cvc.uab.es

Francisco Zamora-Martínez
Departamento Ciencias Físicas, Matemáticas
y de la Computación
Universidad CEU Cardenal Herrera
Alfara del Patriarca, Valencia (Spain)
francisco.zamora@uch.ceu.es

Salvador España-Boquera and María José Castro-Bleda
Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València, Valencia, Spain
{*sespana,mcastro*}*@dsic.upv.es*

Andreas Fischer
Department of Computer Science
University of Fribourg, Fribourg, Switzerland
andreas.fischer@unifr.ch

Horst Bunke
IAM
University of Bern, Bern, Switzerland
bunke@iam.unibe.ch

Abstract

Unconstrained handwritten text recognition systems maximize the combination of two separate probability scores. The first one is the observation probability that indicates how well the returned word sequence matches the input image. The second score is the probability that reflects how likely a word sequence is according to a language model. Current state-of-the-art recognition systems use statistical language models in form of bi-gram word probabilities. This paper proposes to model the target language by means of a recurrent neural network with long-short term memory cells. Because the network is recurrent, the considered context is not limited to a fixed size especially as the memory cells are designed to deal with long-term dependencies. In a set of experiments conducted on the IAM off-line database we show the superiority of the proposed language model over statistical n-gram models.

1 Introduction

The task of transcribing images of continuous handwritten text into a computer-readable form is challenging and has been a focus of research for several decades [7]. It has been shown that it is beneficial to not only take the input image into account when estimating the most likely word sequence, but also the target language [9]. Thus, modern modern recognizers use a statistical approach in which the returned word sequence is the one that maximizes the probability w.r.t. the input image according to a model trained on handwritten text as well as the language according to a model trained on a language corpus.

State-of-the-art recognition systems model the language using statistical n-grams (typically bi-grams). However, modeling the target language using n-gram probabilities has severe disadvantages. First, the size of the contextual information taken into account when estimating a word's probability is limited to the last $n-1$ words for a given n-gram order, hence ignoring longer span dependencies. Secondly, the number of distinct n-grams increases exponentially with n. Hence, even in a large training corpus, many word combinations do not occur at all or they occur with a frequency not high enough for a robust occurrence probability estimation.

In a neural network (NN) language model, a word is represented as an element of a continuous vector space which has been shown to be useful for approximating the probabilities of low-frequency word sequences [2, 12, 13]. The limitations of restricting the input to a

978-4-9906441-1-6

fixed-size context in a feed-forward NN language model have been addressed in the domain of speech recognition using recurrent neural networks [11]. For the task of handwriting recognition, however, no similar work seems to exist. Hence, we propose in this paper recurrent NN language models for handwriting recognition. Furthermore, we propose to use an architecture specifically designed to deal with long-term dependencies, called long-short term memory (LSTM) network [3]. We demonstrate on the IAM off-line database that an increase in context length decreases dramatically the perplexity of the testing set. A similar trend, however, can not be observed on the handwriting recognition task by re-ranking N-best lists. Nevertheless, the proposed language model outperforms the state-of-the-art reference system.

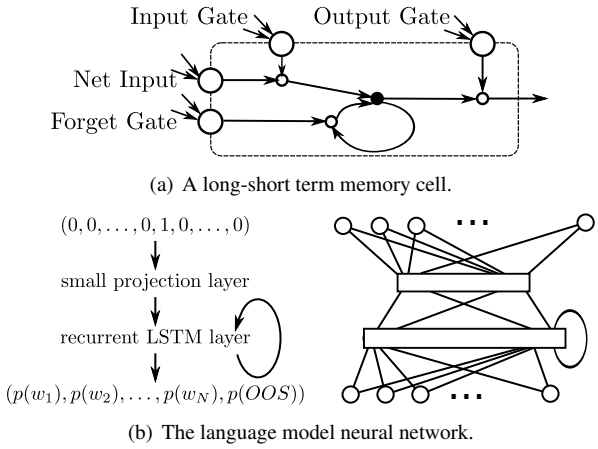
The rest of this paper is structured as follows. Section 2 introduces the LSTM language models. The complete handwriting recognition process is outlined in Section 3. An experimental evaluation is presented in Section 4 and conclusions are drawn in Section 5.

2 LSTM Neural Networks

For both language modeling and continuous text recognition, interdependencies between different elements in the input sequence are crucial. Recurrent neural networks can be used like a memory to store information over several time steps. However, these networks suffer from the so-called *vanishing gradient problem* which describes the inability to learn long-term sequence dependencies.

A recently proposed solution to this problem is *long-short term memory* (LSTM) cells [5]. Consequently, both the proposed language model and the recognition system underlying the experiments are based on this architecture to store information over arbitrarily long time steps. As shown in Fig. 1(a), the core of the cell stores the net input with a recurrent connection in the middle of the cell to store the memory. Three nodes are used to control the information flow from the network into the cell and from the cell back into the network. The net input node receives values from the network and forwards its activation into the cell. However, the value is only passed into the core if a second node, the input gate, is activated, or open. Similarly, an output gate regulates if the core's value is fed into the network.

Each of the activation functions of the nodes in an LSTM cell is differentiable, hence the entire LSTM cell realizes a differentiable function, which renders the entire network suitable for standard back-propagation training.



(a) A long-short term memory cell.

(b) The language model neural network.

Figure 1. The LSTM neural network.

2.1 LSTM Neural Network Language Models

The goal of handwriting recognition is to transcribe a text image X by returning a word sequence $\hat{w} = w_1 w_2 \ldots w_n$ that maximizes a combination of the occurrence probability $p(\hat{w}|X)$ and the language model probability of the target language $p(\hat{w}|LM)$. For a word sequence, we factorize the latter probability as

$$p(\hat{w}|LM) = p(w_1|LM) \prod_{i=2}^{n} p(w_i|w_1 \ldots w_{i-1}, LM)$$

and evaluate each factor separately. In this paper we propose to estimate the probabilities $p(w_i|w_1 \ldots w_{i-1}, LM)$ using a neural network containing two hidden layers, a projection layer and a recurrent LSTM layer (see Fig. 1(b)).

Given a *shortlist* S of the most frequent words, the neural network has one input node for each word in the shortlist and three additional ones: one to indicate an *out-of-shortlist* (OOS) word, one for the *start-of-sentence* (SOS) tag, and one for the *end-of-sentence* (EOS) tag. Thus, each word is represented as a binary vector of length $|S| + 3$.

To reduce the huge dimensionality of this encoding, the words are linearly mapped onto a projection layer. The next layer is the LSTM layer containing the recurrent LSTM memory cells. The output layer has again the same dimensionality as the input layer. Thus, the probabilities $p_{\text{LSTM}}(w_i|w_1 \ldots w_{i-1}, LM)$ are obtained by normalizing the output activations via the *softmax* function after feeding the input sequence $w_1 \ldots w_{i-1}$ into the network one word at a time.

For OOS words, the probability mass given by the activation of the OOS-node is split across the remaining vocabulary, according to factors estimated on the

training corpus. Finally, the probabilities are smoothed with a bi-gram language model of the entire vocabulary. This is done by a weighted sum of both probabilities. The weights are estimated on the validation set.

For training, a sequence $w_0, \ldots, w_{i-1}$ is fed to the network and the cross entropy error between the word w_i and the network's output acts as the objective function that is to be minimized.

3 Recognition System

The advantage of using LSTM NN recurrent models is that they can consider the entire context. Keeping track of all the different word histories, however, leads to an exponential increase in decoding time. Therefore we have opted to use a state-of-the-art recognition system and prune the search space by means of N-best lists which are then re-ranked using the LSTM NNs.

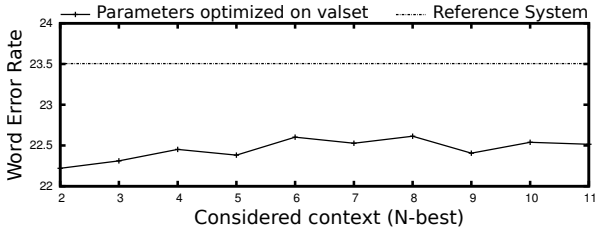
The considered recognition system itself is also based on LSTM neural networks and has been proven to achieve a high performance. Given a bi-gram language model, the system is able to generate N-best lists of recognition hypotheses for each text line. For more details on the handwriting recognition system we refer to [4].

4 Experimental Evaluation

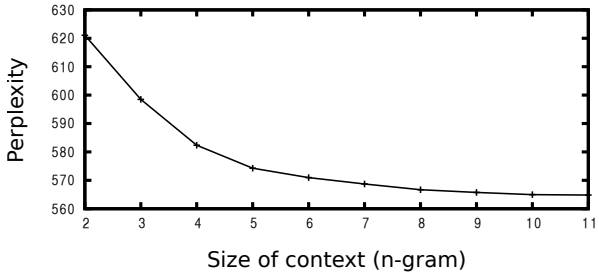
To evaluate the performance of the proposed language model we re-ranked N-best lists of a BLSTM NN recognizer according to the newly obtained word sequence probabilities. The experiments are done on the IAM off-line database of modern English handwritten data [10]. The database consists of texts from the LOB corpus [6] and is separated into writer disjunct training, validation, and testing sets with a size of 6,161, 920, and 2,781 text lines, respectively. Ten BLSTM NN recognizers were trained on the training set and the one performing best on the validation set was selected.

To create the novel language model, we trained eight LSTM neural networks on a union of the Brown and Wellington corpus [1, 8] as well as the part of the LOB corpus not used for validation or testing[1]. Finally, a linear combination of the four networks having the lowest cross entropy error on the validation set is used as the LSTM NN language model. We used a shortlist size of 10K words, a projection layer of size 193 and 100 LSTM cells in the recurrent layer. The size of the projection layer was taken from previous experiments, while the size of the LSTM layer was selected as a good trade-off between the computational cost of training and

[1] All in all 3.34M words in 162.6K sentences.



(a) recognition accuracy



(b) perplexity

Figure 2. Recognition word error rate and perplexity on the testing set.

decrease of training error. The bi-gram model used to obtain the N-best lists is estimated on the same set on which the LSTM NNs are trained. The complete dictionary of 103K words is used. One network consists of 1.2M weights and requires therefore about 10MB, hence storing four networks in memory uses about as much memory as a tri-gram LM.

The experiments are done as follows: the BLSTM NN recognizer decodes each text line of the validation and testing set using the bi-gram model. Each text line is stored as an N-best list with $N = 5,000$. For each hypothesis, the recognition probability and language model probability are stored separately. Then, the language model probability is replaced by the one estimated using the LSTM NN language model.

A *grammar scale factor* and a *word insertion penalty* parameter is used to balance the word probabilities given by the language model and the observation probabilities given by the recognizer [9]. For all systems, both parameters are optimized on the validation set.

In Fig. 2(a) the decoding using the LSTM NN language model using various context lengths is compared to the reference system which achieves a word error rate on the testing set of 23.5%. The proposed language model, with a word error rate between 22.2% and 22.4%, constantly outperforms the reference system. This increase is statistically significant ($\alpha = 0.05$).

Additionally, we show in Fig. 2(b) the perplexity of the testing set, using the LSTM NN language model as a

function of the context size. One can clearly see the correlation between the context that is taken into account and the perplexity. The perplexity of a language model indicates how unpredictable a given text is. Hence the lower the perplexity, the better can the word sequences be explained by the model. It can clearly be seen that the neural networks make use of long-term dependencies in a text line and move more probability mass to the correct words, the longer the considered context is. The perplexity of the bi-gram and tri-gram reference models are far above the shown range.

5 Conclusions

In this paper we propose the use of recurrent neural network language models containing LSTM memory cells. This architecture is specifically designed for long term dependencies within sequences. This renders them very useful for the task of language modelling.

Following the approach of feed-forward NN language models, the proposed LSTM NN language model focuses on a shortlist of 10K words. During application, the words, represented as 10,003-dimensional vectors, are mapped linearly into a smaller subspace. The low-dimensional word representations are then fed into the recurrent LSTM layer of the neural network which finally predicts the occurrence probabilities for the text word that follows the input sequence.

We demonstrate how contextual information decreases the perplexity of a testing set using the LSTM NN language model. Furthermore, we show for the writer independent handwriting recognition task on the IAM off-line database that the proposed language model can be used to re-rank generated N-best lists to decrease the word error rate. However, word error rate does not follow the trend of the perplexity reduction w.r.t. the context length, possibly because the N-best lists were created using bi-grams. Hence, a further step would be to integrate the LSTM NN LM directly into the decoding process.

Language models that consider an arbitrary long context work best the longer the input sequence becomes. Hence, moving from single line to whole document recognition seems to be an interesting application of recurrent NN LM in the future.

Acknowledgments

We thank Alex Graves for kindly providing us with the BLSTM Neural Network source code. This work has been supported by the European project FP7-PEOPLE-2008-IAPP: 230653, the Spanish Government under project TIN2010-18958, as well as the Swiss National Science Foundation (Project CRSI22_125220).

References

[1] L. Bauer. Manual of Information to Accompany The Wellington Corpus of Written New Zealand English. Technical report, Department of Linguistics, Victoria University, Wellington, New Zealand, 1993.

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3(2):1137–1155, 2003.

[3] F. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning Precise Timing with LSTM Recurrent Networks. *Journal of Machine Learning Research*, 3:115–143, 2002.

[4] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 2009.

[5] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[6] S. Johansson, E. Atwell, R. Garside, and G. Leech. The tagged lob corpus: Users' manual. Technical report, The Norwegian Computing Centre for the Humanities, 1986.

[7] A. L. Koerich, R. Sabourin, and C. Y. Suen. Large Vocabulary Off-Line Handwriting Recognition: A Survey. *Pattern Analysis and Applications*, 6(2):97–121, 2003.

[8] H. Kucera and W. N. Francis. *Manual of Information to accompany A Standard Corpus of Present-Day Edited American English, for use with Digital Computers*. Brown University, Department of Linguistics, Providence, Rhode Island, 1964. Revised 1971. Revised and amplified 1979.

[9] U.-V. Marti and H. Bunke. Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System. *Int'l Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[10] U.-V. Marti and H. Bunke. The IAM-Database: An English Sentence Database for Offline Handwriting Recognition. *Int'l Journal on Document Analysis and Recognition*, 5:39–46, 2002.

[11] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur. Recurrent Neural Network based Language Models. In *11th annual Conf. of the International Speech Communication Association*, number 9, pages 1045–1048, 2010.

[12] H. Schwenk. Continuous Space Language Models. *Computer Speech and Language*, 21(3):492–518, 2007.

[13] F. Zamora-Martínez, M. J. Castro-Bleda, and H. Schwenk. N-Gram-Based Machine Translation Enhanced with Neural Networks for the French-English BTEC-IWSLT'10 Task. In *Int'l Workshop on Spoken Language Translation*, pages 45–52, 2010.