Pattern Recognition Letters 90 (2017) 58–64

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Combination of context-dependent bidirectional long short-term memory classifiers for robust offline handwriting recognition

Youssouf Chherawala*, Partha Pratim Roy, Mohamed Cheriet

Synchromedia Laboratory, École de Technologie Supérieure, 1100 Notre-Dame Ouest, Montreal (QC), Canada

ARTICLE INFO

Article history:
Received 22 May 2016
Available online 21 March 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
Handwriting recognition
BLSTM
Context-dependent model
RIMES database

ABSTRACT

The BLSTM classifier has been recently introduced for sequence labeling tasks and provides state-of-the-art performance for handwriting recognition. Its recurrent connections integrate the context at the feature level over a long range. Nevertheless, this context is not explicitly modeled at the label level. Explicit context-modeling strategies have been applied to HMMs with improvement of the recognition rate. In this paper, we study the effect of context modeling on the performance of the BLSTM classifier. The baseline approach, consisting of context-independent character label, is compared with several context-dependent approaches, modeling the left and right contexts. The results show that context-dependent models improve the recognition rate, and demonstrate the ability of the BLSTM classifier to deal with a large number of character models, without clustering. Furthermore, the context-dependent and context-independent models are complementary, and their combination leads to a robust recognition. We tested our approach with promising results on the RIMES database of Latin script documents.

1. Introduction

The bidirectional long short-term memory (BLSTM) [11] neural network has been recently introduced for sequence labeling tasks and provides state-of-the-art performance for handwriting recognition. It shows better results than the well-known HMM model in recent studies, thanks to its discriminative training and recurrent connections that integrate the context at the feature level over a long range. Nevertheless, this context is not explicitly modeled at the label level. Explicit context modeling strategies have been applied to HMMs with improvement of the recognition rate. Indeed, in cursive scripts, the shape of a character depends on its context, usually described by the characters preceding and following it. This is illustrated in Fig. 1.

Context-dependent models were first introduced for speech recognition, before being applied to online [17,28] and offline handwriting recognition. Schüßler and Niemann [27] used a hierarchy of context, from monographs, bigraphs, trigraphs and word models during the HMM recognition. The most specialized contexts are favored over simpler ones to build the word model, provided that sufficient data are available for their training. Fink and Plotz [9] showed that the improvement using context-dependent models for handwriting recognition is marginal compared to its application to speech recognition. Two hypotheses were provided to explain this result. First, the physical phenomena leading to the context dependency in speech are well understood, while those leading to context dependency in handwriting are not clear. Second, proper training of context-dependent models requires much more data for handwriting recognition than for speech recognition. The lack of data to accurately learn each model is one of the main issues of context-dependent models. Bianne-Bernard et al. [4] proposed a knowledge-driven strategy to efficiently tackle this limitation. A decision tree is used for state clustering between context-dependent models for Latin and Arabic scripts. Hamdani et al. [15] proposed a set of questions specific to the Arabic script for the construction of the decision tree. Natarajan et al. [20] applied left and right context models in their multi-lingual offline handwriting recognition system. Finally, the IFN/ENIT database [23] of Arabic words defines context-dependent character labels, however, the context is limited to the position of the character in the subword (beginning, middle or end).

In this paper, we study the effect of context modeling on the performance of the BLSTM classifier. It is the main novelty of the paper and to the best of our knowledge, we are the first to investigate on it. The baseline BLSTM, with context-independent character label is compared with context-dependent BLSTM models. The results show that context-dependent models improve the recognition rate, and demonstrate the ability of the BLSTM classifier to deal with a large number of character labels, without using clustering. Also, we show that context-dependent and context-

* Corresponding author.
E-mail addresses: ychherawala@synchromedia.ca (Y. Chherawala), proy@synchromedia.ca (P.P. Roy), mohamed.cheriet@etsmtl.ca (M. Cheriet).

http://dx.doi.org/10.1016/j.patrec.2017.03.012
0167-8655/

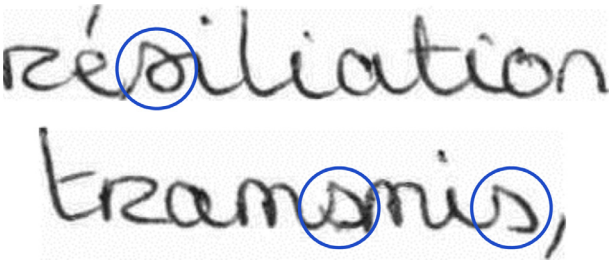

Fig. 1. French words 'résiliation' and 'transmis' from the same writer. The shape of the character 's' is context dependent.

independent character labels are complementary, and that their combination provides an improved recognition performance. In this paper, we are specifically interested in Latin-script documents.

This paper is an extension of the work published in [6]. In particular, this extension studies advanced context-modeling approaches and suitable schemes for context-dependent models combination. The contributions of this paper are the following: 1) explicit use of the context at the label level to improve the performance of the BLSTM classifier, 2) exploration of suitable features for context-dependent models, 3) robust recognition system by combining classifiers.

The paper is organized as follows. The background on the BLSTM classifier and the type of context models investigated are presented respectively in Sections 2 and 3. The feature extraction process is detailed in Section 4. The combination rules for the different models are described in Section 5. The experiments are detailed and discussed in Section 6. Finally, the conclusions are drawn in Section 7.

2. BLSTM

The BLSTM is a recurrent neural network (RNN), that is, connections between artificial neurons form a directed cycle. It provides a 'memory' of the previous internal state of the network. In this section, we describe the LSTM hidden layer and the CTC output layer, both forming the BLSTM network.

2.1. Long short-term memory (LSTM) layer

The LSTM network neurons have a specific architecture, referred as memory block. Each memory block contains a memory cell, and its interaction with the rest of the network is controlled by three gates, namely: an input gate, an output gate and a forget gate. This allows the memory cell to preserve its state over a long range of time and to model the context at the feature level. The 1D sequence recognition is improved by processing the input signal in both directions, i.e., one layer processes the signal in forward direction while another layer processes it in backward direction. The output of both layers is combined at the next layer as a feature map. Similarly to the convolutional neural network architecture [18], it is possible to have multiple forwards and backward layers in each LSTM layer as well as multiple feature maps at the output layer, and to stack multiple LSTM layers using max-pooling subsampling.

2.2. Connectionist temporal classification (CTC) layer

Usually, most of the RNNs require pre-segmented training data or postprocessing to transform its outputs into transcriptions. To avoid such a process, the CTC output layer has been designed for sequence labeling. This layer is trained to predict the probability of an output label sequence given an observation sequence. There is one output unit for each distinct character label, plus a 'blank' label for character transition. Each unit activation provides the probability to observe the corresponding label for each sequence time. The objective function of the CTC layer is defined as the negative log probability of the network correctly labeling the entire training set. Once the network is trained, the labeling of an input sequence $\mathbf{O}$ involves a decoding process of the network output, in which the word $\hat{w}$ from a lexicon L that generates the most probable path π is chosen:

$$\hat{w} = \mathcal{B}\left(\underset{\pi \in \bigcup_{w \in L} \mathcal{B}^{-1}(w)}{\arg\max} \; P(\pi|\mathbf{O}) \right) \tag{1}$$

where $\mathcal{B}$ is a function that maps a path on the network output to a word. The confidence of the network in the recognition of a word w is related to $P(\pi_w|\mathbf{O})$, where π_w is the most probable path corresponding to w. Nevertheless, this probability is not normalized over the lexicon. Therefore, we formulate the probability of the recognized word $\hat{w}$ knowing the input sequence $\mathbf{O}$ as follows:

$$P(\hat{w}|\mathbf{O}) = \frac{P(\pi_{\hat{w}}|\mathbf{O})}{\sum_{w \in L} P(\pi_w|\mathbf{O})} \tag{2}$$

This value is useful to compare the confidence of different systems, especially in the context of multiple system combination. In practice, it is normalized over the N-best word hypotheses, instead of the whole lexicon. We used $N = 5$ in the experiments section.

3. Context modeling strategies

We propose several context modeling strategies for the BLSTM classifier. The difference lies in the way the characters are labeled. For context-dependent models, the labels explicitly include information about the context. Therefore, context modeling only involves the use of more specific character labels. As a result, the CTC output layer will have more units, each sensitive to a specific character and its context. The label sequence of the words of the lexicon remains context-dependent for the recognition. There is no other modification to the BLSTM classifier previously presented. We first present basic modeling approaches followed by advanced ones.

3.1. Basic modeling

The first basic approach is the context-independent (CI) character model. The character labels remain unchanged and it will serve as our baseline. The second approach considers the left context (LC) of each character, i.e., the character preceding the current one. For this, the label of each character is modified to include the label of the preceding one. Finally, the third model considers the right context (RC), i.e., the character following the current one. The label of each character is modified to include the label of the following one. To summarize, the CI model uses unigrams for each character label while the LC and RC models use bigrams. In this example, the French word 'transmis' is modeled by the following label sequences for each strategy:

- **Context independent**: w = 't' 'r' 'a' 'n' 's' 'm' 'i' 's'.
- **Left context**: w = '#t' 'tr' 'ra' 'an' 'ns' 'sm' 'mi' 'is'.
- **Right context**: w = 'tr' 'ra' 'an' 'ns' 'sm' 'mi' 'is' 's#'.

where # represents the word boundary.

3.2. Advanced modeling

We first propose a hybrid context modeling approach that combines context-dependent and context-independent labels. We explicitly model character transitions by context-dependent labels