CONNECTIONIST TEMPORAL CLASSIFICATION: LABELLING UNSEGMENTED SEQUENCE DATA WITH RECURRENT NEURAL NETWORKS

Sebastian Hirt

ABSTRACT

The Paper "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks" [1] introduces a novel recurrent neural network (RNN) training method called Connectionist Temporal Classification (CTC). It aims to improve sequence learning by removing the need of pre-segmenting data and post processing. The authors show its advantages compared to a baseline Hidden Markov Model (HMM) and a hybrid HMM-RNN approach by experimentation with the TIMIT speech corpus.

1. INTRODUCTION

In recent years machine learning has become a fast emerging research topic, which is driven among others by the need of systems for handwriting recognition, speech recognition or gesture recognition. In this context, methodologies transcribe sequence data into a sequence of labels. For instance, in speech recognition an acoustic signal is mapped to a sequence of words.

However, labelling sequence data still remains an ubiquitous problem within current state of the art algorithms such as recurrent neural networks as they require pre-segmented training data and post-processing to transform their outputs into label sequences. To apply RNNs directly to sequence labelling, RNNs are combined with HMMs, which are traditionally used for sequence labelling tasks [2][3], in so-called hybrid systems [4][5]. Yet, these hybrid systems inherit drawbacks such as the requirement for task specific knowledge for the HMM and do not exploit the full potential of RNNs for sequential modeling.

To tackle this problem Alex Graves et al. introduce a novel RNN training algorithm called "Connectionnist Temporal Classification (CTC)". It aims to improve sequence learning by removing the need of pre-segmenting data and post-processing outputs into labels. This is achieved by introducing a 'blank' label and directly predicting a probability distribution over all possible label sequences from unsegmented input data. The residual of the abstract is structured as follows.

Section 2 describes CTC and the training of the RNN. In Section 3 the experiments on the TIMIT data set are described, analyzed and compared to other methods. Finally, key results are summarized and discussed.

2. METHODS

2.1. Connectionist Temporal Classification

This section describes the approach which enables a RNN being applied directly to sequential data. Labelling sequential data is defined as temporal classification, which is defined as the mapping $h : \mathcal{X} \mapsto \mathcal{Z}$, where $\mathcal{X} = (R^m)^*$ and $\mathcal{Z} = L^*$ denote the input space (the set of all sequences of m dimensional real valued vectors) and the target space (the set of target labelling sequences).

For this reason, we use a RNN that outputs a conditional probability distribution over label sequences and a classifier that maps those to a unique sequence of labels. The probability distribution is calculated by using a softmax output layer with L + 1 units. The first L units represent the probability of the corresponding label and the extra unit represents the probability of a 'blank' or no label. Hence, this RNN defines a probability distribution,

$$p(\pi|\mathbf{x}) = \prod_{t=1}^{T} y_{\pi_t}^t, \forall \pi \in L'^T, \tag{1}$$

where π denotes one possible sequence of labels (paths), $y_{\pi_t}^t$ the probability of label π_t at time step t and L'^T the set of all possible paths of length T. Applying subsequently a many-to-one mapping $\mathcal{B}$ to sum up all probabilities of paths with the same meaning, yields

$$p(\mathbf{l}|\mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} p(\pi|\mathbf{x}). \tag{2}$$

Formally, the mapping $\mathcal{B}$ removes all blanks and repeated labels from the paths π to merge them into a unique labeling sequence $\mathbf{l}$. Given the probability distribution by equation 2 we can define a classifier which outputs the most probable labelling for a given input. This task is also referred to as decoding. Decoding can be implemented by "best path decoding" or "prefix search decoding".

2.2. Training the Network

In the last section we have described how CTC enables RNNs to be applied to sequential classification without presegmenting the input. However, the efficient training of the RNN remains an open problem. To train the RNN we use

the principle of maximum likelihood [6] and first define the objective function as

$$O^{ML}(S, \mathcal{N}_w) = -\sum_{(\mathbf{x},\mathbf{z})\in S} \ln\left(p(\mathbf{z}|\mathbf{x})\right). \tag{3}$$

Minimizing this function translates to maximizing the conditional probability of a target sequence $\mathbf{z}$ given a input sequence $\mathbf{x}$ for all sequences in the training set. As seen in equation 2 one needs to sum over all paths π that correspond to that sequence, which are too many to have an efficient calculation. To tackle this problem path probabilities are deduced via the

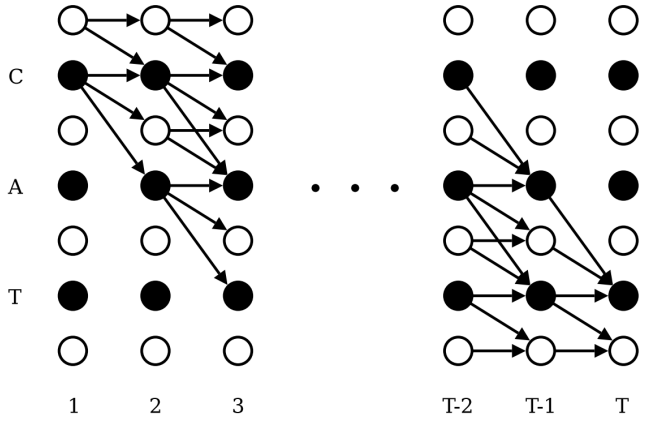

Fig. 1. Forward-backward algorithm applied on the labelling sequence "CAT". Nodes represent labels/blanks at certain points in time. α gets updated forward (in arrow direction), β backwards (against arrow direction) [1].

efficient forward-backward algorithm (Figure 1). Within this algorithm, a forward variable,

$$\alpha_t(s) \stackrel{\text{def}}{=} \sum_{\substack{\pi\in N^T:\\ \mathcal{B}(\pi_{1:t})=\mathbf{l}_{1:s}}} \prod_{t'=1}^{t} y^{t'}_{\pi_{t'}}, \tag{4}$$

and a backward variable,

$$\beta_t(s) \stackrel{\text{def}}{=} \sum_{\substack{\pi\in N^T:\\ \mathcal{B}(\pi_{t:T})=\mathbf{l}_{s:|\mathbf{l}|}}} \prod_{t'=t}^{T} y^{t'}_{\pi_{t'}}, \tag{5}$$

are defined, where $\alpha_t(s)$ and $\beta_t(s)$ represent the probability of the partial target sequence $\mathbf{l}_{1:s}$ and $\mathbf{l}_{s:|\mathbf{l}|}$ going through node s at time t. Using a recursive algorithm to determine the backward and forward variables, we can represent the probability of a target sequence by equation,

$$p(\mathbf{l}|\mathbf{x}) = \sum_{s=l}^{|\mathbf{l}'|} \frac{\alpha_t(s)\beta_t(s)}{y^t_{\mathbf{l}'_s}}, \tag{6}$$

where $y^t_{\mathbf{l}'_s}$ is the probability of label s at time t.

3. EVALUATION

3.1. Label error rate

To evaluate the method, the label error rate (LER) is used. It is defined as,

$$LER(h, S') = \frac{1}{Z}\sum_{(\mathbf{x},\mathbf{z})\in S} ED(h(\mathbf{x})), \tag{7}$$

where S' is the test set, Z the total number of labels in S' and $ED(\mathbf{p}, \mathbf{q})$ describes the edit distance between two sequences $\mathbf{p}$ and $\mathbf{q}$ [7].

3.2. Setup and Data

The methods where tested on the task of speech recognition. The TIMIT [8] data set, which contains segments of English speech audio data and the corresponding transcription, is used. It is split into training, test and validation set. For experimentation a bidirectional Long Short-Term Memory architecture [9] was used as the RNN for CTC. It was then compared to baseline HMMs and hybrid HMMs [10].

3.3. Results

System	LER
Context-independent HMM	38.85 %
Context-dependent HMM	35.21 %
BLSTM/HMM	33.84 ± 0.06 %
Weighted error BLSTM/HMM	31.57 ± 0.06 %
CTC (best path)	31.47 ± 0.21 %
CTC (prefix search)	30.51 ± 0.19 %

Table 1. Label Error Rate (LER) (smaller is better) of CTC compared to HMMs and hybrid methods, evaluated on the TIMIT speech corpus [1].

Table 1 shows the results of the experiments. The LER (Equation 7) decreases from top to bottom. The inclusion of a RNN (BLSTM) shows a clear improvement of 2-5% compared to the HMMs. Furthermore the proposed CTC method shows an additional 1-2% improvement over those systems. The best result is achieved by using the "prefix search decoding" for selecting the correct labeling sequence, as it selects the optimal labeling sequence every time.

4. CONCLUSIONS

The method described in this paper is removing the need of pre-segmented input data in sequence labelling tasks. At the same time it outperforms traditional methods like HMMs and HMM-RNN hybrids on the speech recognition task. The authors intend to use a hierarchical structure of classifiers and improvements of the RNN training to further explore CTC.

5. REFERENCES

[1] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, New York, NY, USA, 2006, ICML '06, p. 369–376, Association for Computing Machinery.

[2] Lalit Bahl, Peter Brown, Peter De Souza, and Robert Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1986, vol. 11, pp. 49–52.

[3] A_P Varga and Roger K Moore, "Hidden markov model decomposition of speech and noise," in *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1990, pp. 845–848.

[4] Herve Bourlard and Nelson Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, 01 1994.

[5] Y. Bengio, "Markovian models for sequential data," *Neural Computing Surveys*, vol. 2, 07 1999.

[6] Christopher M Bishop et al., *Neural networks for pattern recognition*, Oxford university press, 1995.

[7] Eric Sven Ristad and Peter N Yianilos, "Learning string-edit distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 522–532, 1998.

[8] J. Garofolo, Lori Lamel, W. Fisher, Jonathan Fiscus, D. Pallett, N. Dahlgren, and V. Zue, "Timit acoustic-phonetic continuous speech corpus," *Linguistic Data Consortium*, 11 1992.

[9] Felix A. Gers, Nicol N. Schraudolph, and Jürgen Schmidhuber, "Learning precise timing with lstm recurrent networks," *J. Mach. Learn. Res.*, vol. 3, pp. 115–143, 2002.

[10] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition.," 01 2005, pp. 799–804.