Sequence Labelling in Structured Domains with Hierarchical Recurrent Neural Networks

Santiago Fernández[1] and **Alex Graves**[1] and **Jürgen Schmidhuber**[1,2]
[1] IDSIA, Galleria 2, 6928 Manno-Lugano, Switzerland
[2] TU Munich, Boltzmannstr. 3, 85748 Garching, Munich, Germany
{santiago,alex,juergen}@idsia.ch

Abstract

Modelling data in structured domains requires establishing the relations among patterns at multiple scales. When these patterns arise from sequential data, the multiscale structure also contains a dynamic component that must be modelled, particularly, as is often the case, if the data is unsegmented. Probabilistic graphical models are the predominant framework for labelling unsegmented sequential data in structured domains. Their use requires a certain degree of *a priori* knowledge about the relations among patterns and about the patterns themselves. This paper presents a hierarchical system, based on the connectionist temporal classification algorithm, for labelling unsegmented sequential data at multiple scales with recurrent neural networks only. Experiments on the recognition of sequences of spoken digits show that the system outperforms hidden Markov models, while making fewer assumptions about the domain.

1 Introduction

Assigning a sequence of labels to an unsegmented stream of data is the goal of a number of practical tasks such as speech recognition and handwriting recognition. In these domains, the structure at multiple scales is often captured with hierarchical models to assist with the process of sequence labelling.

Probabilistic graphical models, such as hidden Markov models [Rabiner, 1989, HMMs] and conditional random fields [Lafferty *et al.*, 2001, CRFs], are the predominant framework for sequence labelling. Recently, a novel algorithm called connectionist temporal classification [Graves *et al.*, 2006, CTC] has been developed to label unsegmented sequential data with recurrent neural networks (RNNs) only. Like CRFs, and in contrast with HMMs, CTC is a discriminant algorithm. In contrast with both CRFs and HMMs, CTC is a general algorithm for sequence labelling, in the sense that CTC does not require explicit assumptions about the statistical properties of the data or explicit models of the patterns and the relations among them.

Hierarchical architectures are often used with HMMs. There exist efficient algorithms for estimating the parameters in such hierarchies in a global way, i.e. a way that maximises the performance at the top of the hierarchy. Nonetheless, HMMs do not efficiently represent information at different scales, indeed they do not attempt to abstract information in hierarchical form [Nevill-Manning and Witten, 1997].

The only paper known to us that describes a hierarchy of CRFs has been presented recently by Kumar and Hebert [2005] for classifying objects in images. The hierarchy is not trained globally, but sequentially: i.e. estimates of the parameters in the first layer are found and, then, with these parameters fixed, those in the second layer (and transition matrices) are estimated.

This paper uses a hierarchical approach to extend the applicability of CTC to sequence labelling in structured domains. Hierarchical CTC (HCTC) consists of successive levels of CTC networks. Every level predicts a sequence of labels and feeds it forward to the next level. Labels at the lower levels represent the structure of the data at a lower scale. The error signal at every level is back-propagated through all the lower levels, and the network is trained with gradient descent. The relative weight of the back-propagated error and the prediction error at every level can be adjusted if necessary, e.g. depending on the degree of uncertainty about the target label sequence at that level, which can depend on the variability in the data. In the extreme case in which only the error at the top level is used for training, the network can, potentially, discover structure in the data at intermediate levels that results in accurate final predictions.

The next section briefly introduces the CTC algorithm. Section 3 describes the architecture of HCTC. Section 4 compares the performance of hierarchical HMMs and HCTC on a speech recognition task. Section 5 offers a discussion on several aspects of the algorithm and guidelines for future work. Final conclusions are given in section 6.

2 Connectionist Temporal Classification

CTC is an algorithm for labelling unsegmented sequential data with RNNs only [Graves *et al.*, 2006]. The basic idea behind CTC is to interpret the network outputs as a probability distribution over all possible label sequences, conditioned on the input data sequence. Given this distribution, an objective function can be derived that directly maximises the probabilities of the correct labellings. Since the objective function is differentiable, the network can then be trained with standard backpropagation through time [Werbos, 1990].

The algorithm requires that the network outputs at different times are conditionally independent given the internal state of the network. This requirement is met as long as there are no feedback connections from the output layer to itself or the network.

A CTC network has a softmax output layer [Bridle, 1990] with one more unit than the number of labels required for the task. The activation of the extra unit is interpreted as the probability of observing a "blank", or no label at a given time step. The activations of the other units are interpreted as the probabilities of observing the corresponding labels. The blank unit allows the same label to appear more than once consecutively in the output label sequence. A trained CTC network produces, typically, a series of spikes separated by periods of blanks. The location of the spikes usually corresponds to the position of the patterns in the input data; however, the algorithm is not guaranteed to find a precise alignment.

2.1 Classification

For an input sequence $\mathbf{x}$ of length T we require a label sequence $\mathbf{l} \in L^{\leq T}$, where $L^{\leq T}$ is the set of sequences of length $\leq T$ on the alphabet L of labels. We begin by choosing a label (or blank) at every timestep according to the probability given by the network outputs. This defines a probability distribution over the set L'^T of length T sequences on the extended alphabet $L' = L \cup \{blank\}$ of labels with *blank* included. To disambiguate the elements of L'^T from label sequences, we refer to them as *paths*.

The conditional probability of a particular path $\pi \in L'^T$ is given by:

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

where y^t_k is the activation of output unit k at time t.

Paths can be mapped into label sequences by first removing the repeated labels, and then removing the blanks. For example, the path (a,-,a,b,-), and the path (-,a,a,-,-,a,b,b) would both map onto the labelling (a,a,b). The conditional probability of a given labelling $\mathbf{l} \in L^{\leq T}$ is the sum of the probabilities of all the paths corresponding to it:

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

where $\mathcal{B} : L'^T \mapsto L^{\leq T}$ is the many-to-one map implied by the above process.

Finally, the output of the classifier $h(\mathbf{x})$ is the most probable labelling for the input sequence:

$$h(\mathbf{x}) = \arg\max_{\mathbf{l}} p(\mathbf{l}|\mathbf{x}). \tag{3}$$

In general, finding this maximum is intractable, but there are several effective approximate methods [Graves *et al.*, 2006]. The one used in this paper assumes that the most probable path will correspond to the most probable labelling:

$$h(\mathbf{x}) \approx \mathcal{B}(\pi^*) \tag{4}$$
$$\text{where } \pi^* = \arg\max_{\pi} p(\pi|\mathbf{x})$$

and π^* is just the concatenation of the most active outputs at every time-step.

2.2 Training

The objective function for CTC is derived from the principle of maximum likelihood. That is, it attempts to maximise the log probability of correctly labelling the entire training set. Let S be such a training set, consisting of pairs of input and target sequences $(\mathbf{x}, \mathbf{z})$, where the length of sequence $\mathbf{z}$ is less than or equal to the length of the input sequence $\mathbf{x}$. We can express the objective function to be minimised as:

$$O^{ML}(S) = -\sum_{(\mathbf{x},\mathbf{z}) \in S} ln\big(p(\mathbf{z}|\mathbf{x})\big). \tag{5}$$

The network can be trained with gradient descent by differentiating equation (5) with respect to the network outputs. This can be achieved by calculating the conditional probabilities $p(\mathbf{z}|\mathbf{x})$ with a dynamic programming algorithm similar to the forward-backward recursions used for HMMs [Graves *et al.*, 2006].

We define $\alpha_t(s)$ as the probability of having passed through sequence $\mathbf{l}_{1\ldots s}$ and being at symbol s at time t:

$$\begin{aligned}\alpha_t(s) &= P(\pi_{1\ldots t} : \mathcal{B}(\pi_{1\ldots t}) = \mathbf{l}_{1\ldots s}, \pi_t = s|\mathbf{x}) \\ &= \sum_{\substack{\pi: \\ \mathcal{B}(\pi_{1\ldots t})=\mathbf{l}_{1\ldots s}}} \prod_{t'=1}^{t} y^{t'}_{\pi_{t'}},\end{aligned} \tag{6}$$

and $\beta_t(s)$ as the probability of passing through the rest of the label sequence $(\mathbf{l}_{s\ldots|\mathbf{l}|})$ given that symbol s has been reached at time t:

$$\begin{aligned}\beta_t(s) &= P(\pi_{t+1\ldots T} : \mathcal{B}(\pi_{t\ldots T}) = \mathbf{l}_{s\ldots|\mathbf{l}|}|\pi_t = s, \mathbf{x}) \\ &= \sum_{\substack{\pi: \\ \mathcal{B}(\pi_{t:T})=\mathbf{l}_{s:|\mathbf{l}|}}} \prod_{t'=t+1}^{T} y^{t'}_{\pi_{t'}}.\end{aligned} \tag{7}$$

To allow for blanks in the output paths, for every label sequence $\mathbf{l} \in L^{\leq T}$ we consider a modified label sequence $\mathbf{l}'$, with blanks added to the beginning and the end and inserted between every pair of labels. The length of $\mathbf{l}'$ is therefore $|\mathbf{l}'| = 2|\mathbf{l}| + 1$. In calculating $\alpha_t(s)$ and $\beta_t(s)$, we allow all transitions between blank and non-blank labels, and also those between any pair of *distinct* non-blank labels. The sequences can start with either *blank* or the first symbol in $\mathbf{l}$, and can end with either *blank* or the last symbol in $\mathbf{l}$.

From equations (1), (2), (6) and (7), we find that for any t:

$$p(\mathbf{l}|\mathbf{x}) = \sum_{s=1}^{|\mathbf{l}'|} \alpha_t(s)\beta_t(s). \tag{8}$$

Noting that the same label (or blank) may be repeated several times in a single labelling l, we define the set of positions where label k occurs as $lab(\mathbf{l}, k) = \{s : \mathbf{l}'_s = k\}$, which may be empty, and differentiate equation (8) with respect to the network outputs:

$$\frac{\partial p(\mathbf{l}|\mathbf{x})}{\partial y^t_k} = \frac{1}{y^t_k} \sum_{s \in lab(\mathbf{l},k)} \alpha_t(s)\beta_t(s). \tag{9}$$

Setting $\mathbf{l} = \mathbf{z}$ and differentiating the objective function, we obtain the *error signal* received by the network during

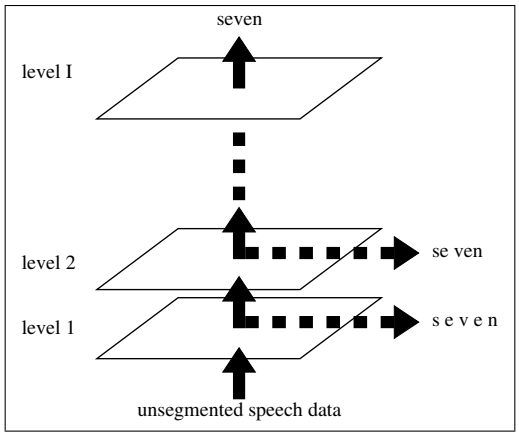

Figure 1: Schematic representation of an HCTC network labelling an unsegmented input data stream. Outputs at each level correspond to the probability of having detected a particular label (or emitting the *blank* label) at a particular time.

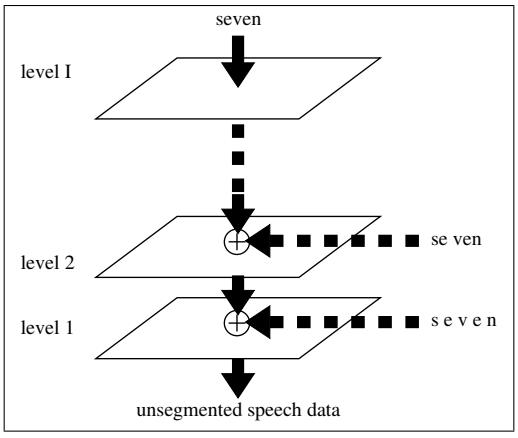

Figure 2: Schematic representation of the error signal flow on an HCTC network. The network is trained globally with gradient descent. External error signals injected at intermediate levels are considered optional. If applied, their contribution to the total error signal can be adjusted.

training:

$$\frac{\partial O^{ML}(\{(\mathbf{x},\mathbf{z})\})}{\partial u_k^t} = y_k^t - \frac{1}{Z_t} \sum_{s \in lab(\mathbf{z},k)} \alpha_t(s)\beta_t(s) \quad (10)$$

where u_k^t and y_k^t are the *unnormalised* and *normalised* outputs of the softmax layer, respectively, and

$$Z_t = \sum_{s=1}^{|\mathbf{l}'|} \alpha_t(s)\beta_t(s) \quad (11)$$

is a normalization factor.

3 Hierarchical Connectionist Temporal Classification

A hierarchy of CTC networks is a series $\mathbf{g} = (g_1, g_2, \ldots, g_I)$ of CTC networks, with network g_1 receiving as input the external signal and all other networks g_i receiving as input the output of network g_{i-1}. This architecture is illustrated in Figure 1.

Note that every network g_i in the hierarchy has a softmax output layer. This forces the hierarchical system to make decisions at every level and use them in the upper levels to achieve a higher degree of abstraction. The analysis of the structure of the data is facilitated by having output probabilities at every level.

Because of the modular design, the flexibility of the system allows using other architectures (such as feeding the external inputs to several levels in the hierarchy) as long as the mathematical requirements of the CTC algorithm are met (see section 2).

3.1 Training

In general, HCTC requires target sequences for every level in the hierarchy, given a particular input sequence in the training set. Each example in the training set S consists of a pair $(\mathbf{x}, Z)$, where $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ is the external input sequence, and $Z = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_I)$ is the set of target sequences, where $|\mathbf{z}_i| \leq T \; \forall i$.

The system is trained globally with gradient descent. Figure 2 illustrates the error signal flow for an HCTC network. The error signal due to the target sequence $\mathbf{z}_i$, received by network g_i, is given by equation (10). To simplify the notation, we define:

$$\Delta_i^{target} := \frac{\partial O_i^{ML}(\{(\mathbf{v}_i,\mathbf{z}_i)\})}{\partial u_{i,k}^t} \quad (12)$$

where the input sequence v_i is the external input sequence for network g_1 and the output of network g_{i-1} for all levels $i \neq 1$.

The error signal Δ_i^{target} is back-propagated into the network g_i, and then into all lower levels $j : j < i$. The contribution of this term to the error signal at the *unnormalised* activation $u_{i,k}^t$ of the kth output unit of network $g_i : i \neq I$ is:

$$\Delta_i^{backprop} = y_k^t \sum_{m \in M} \delta_m \left(w_{mk} - \sum_{k' \in K} w_{mk'} y_{k'}^t \right) \quad (13)$$

where M is the set of units in level $i+1$ which are connected to the set of softmax output units, K, in level i; δ_m is the error signal back-propagated from unit $m \in M$; w_{mk} is the weight associated with the connection between units $m \in M$ and $k \in K$; and y_k^t are the output activations of the softmax layer at level i.

Finally, the total error signal Δ_i received by network g_i is the sum of the contributions in equations (12) and (13). In general, the two contributions can be weighted depending on the problem at hand. This is important if, for example, the target sequences at some intermediate levels are uncertain or not known at all.

$$\Delta_i = \begin{cases} \Delta_i^{target} & \text{if } i = I, \\ \lambda_i \Delta_i^{target} + \Delta_i^{backprop} & \text{otherwise,} \end{cases} \quad (14)$$

Digit	Phonemes
ZERO	Z - II - R - OW
ONE	W - AX - N
TWO	T - OO
THREE	TH - R - II
FOUR	F - OW - R
FIVE	F - AY - V
SIX	S - I - K - S
SEVEN	S - EH - V - E - N
EIGHT	EY - T
NINE	N - AY - N
OH	OW

Table 1: Phonetic labels used to model the digits in the experiments.

with $0 \leq \lambda_i \leq 1$. In the extreme case where $\lambda_i = 0$, no target sequence is provided for the network g_i, which is free to make any decision that minimises the error $\Delta_i^{backprop}$ received from levels higher up in the hierarchy. Because training is done globally, the network can, potentially, discover structure in the data at intermediate levels that results in accurate predictions at higher levels of the hierarchy.

4 Experiments

In order to validate the HCTC algorithm we chose a speech recognition task because this is a problem known to contain structure at multiple scales. HMMs remain state-of-the-art in speech recognition, and therefore HCTC performance is compared with that of HMMs.

The task was to find the sequence of digits spoken in an standard set of utterances using, at an intermediate level, the sequence of phonemes associated to every word.

4.1 Materials

The speaker-independent connected-digit database, TIDIGITS [Leonard and Doddington, 1993], consists of more than 25 thousand digit sequences spoken by over 300 men, women and children. The database was recorded in the U.S. and it is dialectically balanced. The utterances were distributed into a test and a training set. We randomly selected five percent of the training set to use as a validation set. This left 11922 utterances in the training set, 627 in the validation, and 12547 in the test set.

The following eleven digits are present in the database: "zero", "one", "two", "three", ..., "nine" and "oh". Utterances consist of a sequence of one to seven digits. The representation of the digits at the phonetic level used in the experiments can be seen in Table 1. Nineteen phonetic categories were used, nine of which are common to two or more digits.

Samples were digitized at 20 kHz with a quantization range of 16 bits. The acoustic signal was transformed into mel frequency cepstral coefficients (MFCC) with the HTK Toolkit [Young *et al.*, 2005]. Spectral analysis was carried out with a 40 channel Mel filter bank from 130 Hz to 6.8 kHz. A pre-emphasis coefficient of 0.97 was used to correct spectral tilt. Twelve MFCC plus the 0th order coefficient were computed on Hamming windows 25.6 ms long, every 10 ms. Delta and Acceleration coefficients were added giving a vector of 39 coefficients in total. For the network, the coefficients were normalised to have mean zero and standard deviation one over the training set.

4.2 Setup

The HMM system was implemented with the HTK Toolkit [Young *et al.*, 2005]. Three-states left-to-right models were used for each one of the nineteen phonetic categories. A silence model and a "short pause" model (allowed at the end of a digit) were also estimated. Observation probabilities were modelled by a mixture of Gaussians. The grammar model allowed any sequence, preceded and followed by silence, of one or more digits. Neither linguistic information nor probabilities of partial phone sequences were included in the system.

The number of Gaussians and the insertion penalty that optimised performance on the validation set was selected. Using the training set, the number of Gaussians was increased in steps of two until performance on the validation set stabilised or, as in our case, decreased slightly (96 Gaussians). Every time the number of Gaussians was increased, the parameter estimation algorithm (HERest) was applied twice and results were collected on the validation set with insertion penalties varying from 0 to -100 in steps of -5. The best performance on the validation set was obtained with 80 Gaussians and an insertion penalty of -85. The total number of parameters for the HMM is 384,126.

A 2-level HCTC was used. The top, second level, had as target the sequence of digits corresponding to the input stream. The bottom, first level, used as target the sequence of phonemes corresponding to the target sequence of digits for the top level. Each CTC network uses the bi-directional LSTM recurrent neural network [Graves and Schmidhuber, 2005; Graves *et al.*, 2006], primarily because on a phoneme recognition task better results than for other RNNs have been reported [Graves *et al.*, 2005]. Also, the CTC formalism is best realised with a bi-directional recurrent neural network [Schuster and Paliwal, 1997] because the network output at a particular time depends on both past and future events in the input sequence.

Within each level, the input layer was fully connected to the hidden layer and the hidden layer was fully connected to itself and the output layer. In the first level, the input layer was size 39, the forward and backward layers had 128 blocks each, and the output layer was size 20 (19 phonetic categories plus blank). In the second level, the input layer was size 20 also, the forward and backward layers had 50 blocks each and the output layer was size 12 (eleven digits plus the blank label). The input and output cell activation functions were a hyperbolic tangent. The gates used a logistic sigmoid function in the range $[0, 1]$. The total number of weights in the HCTC network is 207,852.

Training of the HCTC network was done by gradient descent with weight updates after every training example. In all cases, the learning rate was 10^{-4}, momentum was 0.9, weights were initialized randomly in the range $[-0.1, 0.1]$ and, during training, Gaussian noise with a standard deviation of 1.0 was added to the inputs to improve generalisation.

System	LER
HMM	0.89 %
HCTC ($\lambda_1 = 1$)	0.61 ± 0.04 %
HCTC ($\lambda_1 = 0$)	0.51 %

Table 2: Label Error Rate (LER) on TIDIGITS. Results for HCTC ($\lambda_1 = 1$) are means over 5 runs with different random weights, $\pm$ standard error. This gives a 95 % confidence interval of $(0.50; 0.72)$, with a t-value of 2.8 for 4 degrees of freedom and a two-tailed test. For HCTC ($\lambda_1 = 0$) the best result obtained is shown; this coincides with the best result obtained with $\lambda_1 = 1$.

Performance was measured as the normalised edit distance (label error rate; LER) between the target label sequence and the output label sequence given by the system.

4.3 Results

Performance rates for the systems tested can be seen in Table 2. The best HMM-based system achieved an error rate of 0.89% in continuous digit recognition. The label error rate for HCTC was on average 0.61 % (95 % confidence interval of $(0.50; 0.72)$). This is an improvement of more than 30 % with respect to the HMM and with approximately half the number of parameters.

The best result without injecting the error signal associated to the phoneme labels ($\lambda_1 = 0$) was 0.51 %, which is as good as the best performance achieved with $\lambda_1 = 1$. In this case, however, the pattern of activations of seven output units (instead of twenty) was enough to encode the dynamic structure of the data at the lower level in order to make accurate predictions at the top level of the sequence of digits in the utterance (see Figure 3). Some of these seven output units become active/inactive at specific points in time for each digit pattern. And some of them are active for different digits. Nonetheless, they cannot be associated directly to phonetic categories and might encode another type of information in the signal.

5 Discussion and Future Work

Inserting levels in the hierarchy without a corresponding target label sequence is interesting if the target labels are not known, or, for instance, if the phonetic labels used are suspected to be invalid due to, e.g., the presence of dialectal variations in the dataset. If the sources of variability in the data cannot be specified accurately, a system with less constraints might be capable of making more effective decisions.

Experimentally, this means that the system will be more difficult to train for a particular goal. HCTC with and without target label sequences at the phonetic level achieved similar performance, albeit by different means. Nevertheless, HCTC with $\lambda_1 = 0$ suffered to a larger extent from the problem of local minima.

Assuming that reasonable target label sequences can be specified *a priori* for intermediate levels, a possible solution is to train HCTC until the error has decreased significantly, and then remove the contribution to the error signal of the target label sequences for intermediate levels. This will free a partially trained network to explore alternatives that maximise performance at the top level of the hierarchy.

In the future, we would also like to explore ways of improving system scalability. For example, large vocabulary speech recognition requires systems that work with many thousands of words. Using output layers of that size for HCTC is not currently practical. Instead of assigning a unique output unit to every possible label, other methods can be explored such as assigning labels to specific activation patterns in a group of output units. This will require modifying CTC's training algorithm.

Another aspect we would like to investigate is the potential of HCTC for word spotting tasks. As a discriminant method, HCTC may improve detection rates due to its capability of discriminating between keywords and non-speech or other speech events. Besides this, HCTC provides estimates of *a posteriori* probabilities that can help to directly assess the level of confidence in the predictions. This is in contrast to generative approaches, such as HMMs, which use unnormalized likelihoods.

6 Conclusion

This paper has presented the HCTC algorithm, which uses a hierarchy of recurrent neural networks to label sequences of unsegmented data in structured domains. HCTC is capable of finding structure at multiple levels and using it to achieve accurate predictions further up in the hierarchy. The use of neural networks offers a flexible way of modelling the domain while at the same time allowing the system to be trained globally. Experimental results demonstrated that this approach outperformed hidden Markov models on a speech recognition task.

Acknowledgments

This research was funded by SNF grant 200021-111968/1.

References

[Bridle, 1990] John S. Bridle. *Neurocomputing: Algorithms, architectures and applications*, chapter Probabilistic interpretation of feedforward classification network outputs, pages 227–236. Springer-Verlag, 1990.

[Graves and Schmidhuber, 2005] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5–6):602–610, June/July 2005.

[Graves *et al.*, 2005] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional LSTM networks for improved phoneme classification and recognition. In *Proceedings of the 2005 International Conference on Artificial Neural Networks*, Warsaw, Poland, 2005.

[Graves *et al.*, 2006] 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*, Pittsburgh, PA, USA, 2006.

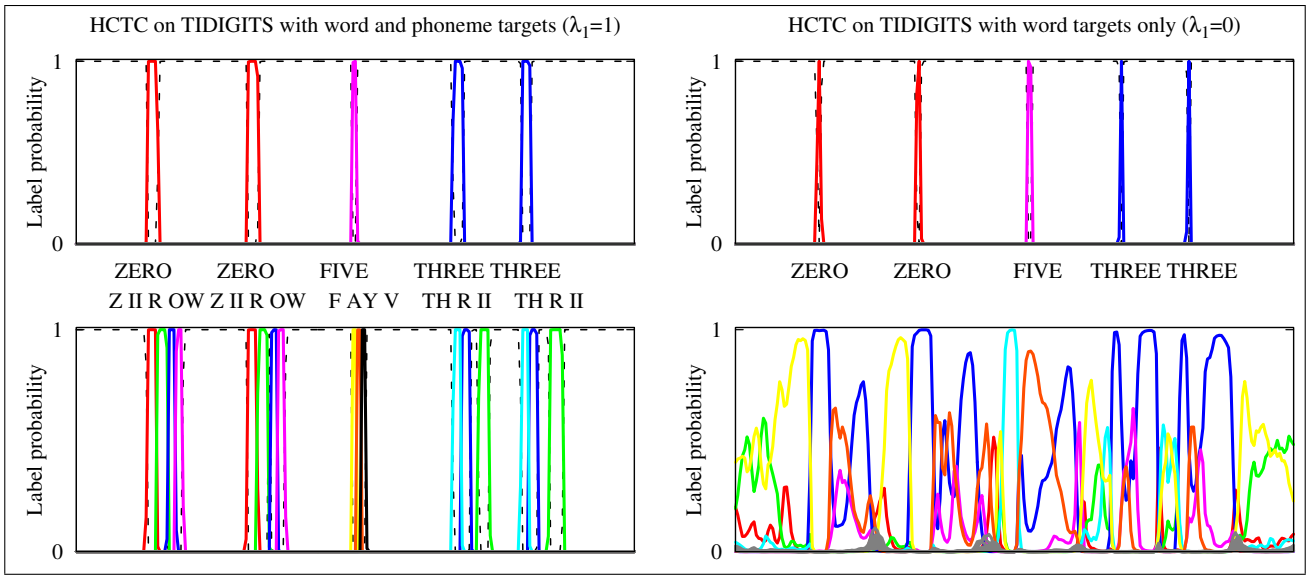

Figure 3: HCTC output on TIDIGITS. The network on the left was trained with target label sequences at the phoneme and word levels (i.e. $\lambda_1 = 1$), whereas the network on the right received an error signal only at the word level ($\lambda_1 = 0$). Activations are shown for the output layers at the word level (top) and phoneme level (bottom). For $\lambda_1 = 0$, the activations at the lower level do not represent phonetic labels but some other kind of information which requires only seven (instead of twenty) output units. Both networks achieved the same word error rate (0.51 %).

[Kumar and Hebert, 2005] Sanjiv Kumar and Martial Hebert. A hierarchical field framework for unified context-based classification. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, Beijing, China, 2005.

[Lafferty *et al.*, 2001] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, 2001.

[Leonard and Doddington, 1993] R. Gary Leonard and George Doddington. *TIDIGITS*. Linguistic Data Consortium, Philadelphia, 1993.

[Nevill-Manning and Witten, 1997] Craig G. Nevill-Manning and Ian H. Witten. Identifying hierarchical structure in sequences: A linear-time algorithm. *Journal of Artificial Intelligence Research*, 7:67–82, 1997.

[Rabiner, 1989] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[Schuster and Paliwal, 1997] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45:2673–2681, 1997.

[Werbos, 1990] Paul J. Werbos. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[Young *et al.*, 2005] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book version 3.3*. Cambridge University Engineering Department, 2005.