MODELING LONG TEMPORAL CONTEXTS IN CONVOLUTIONAL NEURAL NETWORK-BASED PHONE RECOGNITION

László Tóth

MTA-SZTE Research Group on Artificial Intelligence
Hungarian Academy of Sciences and University of Szeged
tothl@inf.u-szeged.hu

ABSTRACT

The deep neural network component of current hybrid speech recognizers is trained on a context of consecutive feature vectors. Here, we investigate whether the time span of this input can be extended by splitting it up and modeling it in smaller chunks. One method for this is to train a hierarchy of two networks, while the less well-known split temporal context (STC) method models the left and right contexts of a frame separately. Here, we evaluate these techniques within a convolutional neural network framework, and find that the two approaches can be nicely combined. With the combined model we can expand the time-span of our network to 69 frames, and we achieve a 7.5% relative error rate reduction compared to modeling this large context as one block. We report a phone error rate of 17.1% on the TIMIT core test set, which is one of the best scores published.

Index Terms— Deep neural network, convolutional neural network, maxout, split temporal context, TIMIT

1. INTRODUCTION – RELATION TO PRIOR WORK

The deep neural network (DNN) component of the state-of-the-art HMM/DNN speech recognizers is routinely trained on a large context window of feature vectors. Most commonly, the DNN operates on a block of 9-26 consecutive frames [1, 2, 3], but some studies even go up to 51 frames [4]. Some researchers argue that the impressive gain of DNN-based models over standard HMMs can almost entirely be attributed to this larger context of input – more precisely, to the fact that DNNs can efficiently extract information from this large set of non-linearly correlated features, while Gaussian-based models cannot [5]. But even DNNs may benefit from processing the input in a structured form, instead of using large uniform fully connected layers. The simplest way to do this is to divide the input into smaller chunks, and combine the local pieces of information extracted from these chunks at higher layers. In the case of speech signals, one may chunk the input along either frequency and/or time. A multi-band scheme of decomposing the frequency axis was suggested almost two decades ago [6]. More recently, the best recognition results are achieved by convolutional neural nets (CNNs), which also process smaller, frequency-localized windows separately. Although the main power of CNNs lies in the so-called pooling operation, as a side-result we observed that chunking the input along the frequency axis already improves the results, without any shifting and pooling [7, 8].

As regards the time axis, the classic literature on (shallow) hybrid HMM/ANN systems suggests training the neural network on a context of 9-11 frames [9]. With the advent of deep networks, most research teams apply larger context windows of 17-26 frames [2, 3]. In one of the early studies on DNNs, Mohamed et al. found that increasing the observation context from 11 to 17 or even 27 frames improved the results, but the error rate started to rise at 37 frames [2]. Applying information theoretic methods, Scanlon et al. found that the information about the identity of a given phone spreads out about ± 8 frames around the center frame of the phone [10]. This explains why simply increasing the context size beyond a certain point yields no further gain in performance.

In this paper, we examine whether we can make use of a larger context by splitting it up and processing it in smaller chunks. The better-known way of doing this is to create a hierarchy of two networks. Here, the lower network processes context windows of the usual size, and then a second network is trained on a context of output vectors coming from the first network. This technique is usually referred to as the hierarchical [11, 12] or stacked [13] modelling approach. But the operation of this model can also be interpreted as a convolution in the time domain [14]. We evaluated this approach on the TIMIT phone recognition task, and it significantly improved the recognition scores of both standard DNNs [15] and convolutional networks [7].

A less well-known technique is the split temporal context (STC) method, originally introduced for shallow networks [16]. Here, the observation context of the actual frame is split into a left and a right part, then separate networks are

This publication was supported by the European Union and co-funded by the European Social Fund. Project title "Telemedicine-focused research activities in the fields of mathematics, informatics and medical sciences", project number: TÁMOP-4.2.2.A-11/1/KONV-2012-0073.

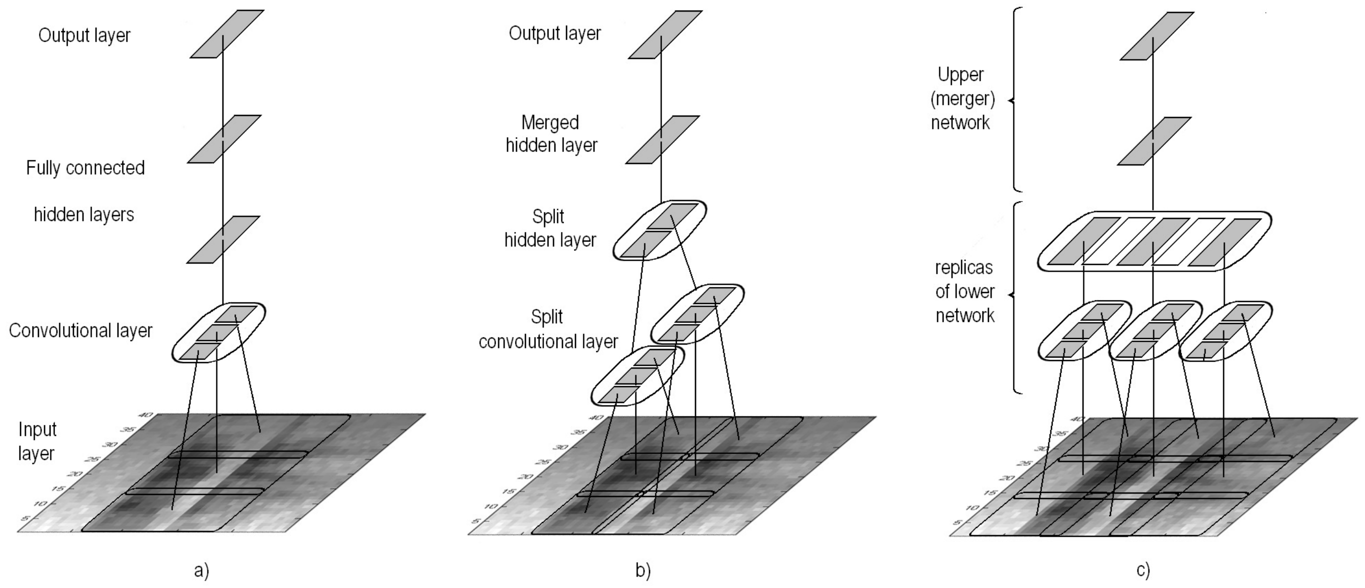

Fig. 1. Illustration of the network structures applied here. For clarity, full connections between layers are denoted here by just a single line. (a) A convolutional network that splits the input along frequency. (b) A convolutional STC network that splits the input along time as well. (c) A hierarchical CNN that combines the local outputs of a sub-network.

trained on the two parts, and their outputs are finally merged. More recently, the STC method was also evaluated in the framework of DNNs. Siniscalchi et al. got an improved LVCSR performance with DNNs using STC [17], while Li and Kim reported both better results and a reduction in the number of parameters when using STC for a noisy speech recognition task [18]. Baccouche et al. experimented with splitting the context into more than two blocks [19].

Here, we evaluate the STC approach within the framework of convolutional neural networks, which, to our knowledge, has not yet been investigated. Moreover, we will combine the STC method and the hierarchical approach to get an improved modeling of a long-span temporal context. We will evaluate the techniques examined on the TIMIT dataset.

2. MODELING LONG TEMPORAL CONTEXTS

2.1. Convolutional neural networks

A convolutional neural network will serve as our baseline system [1, 8, 20, 21, 22]. The structure of this network is demonstrated schematically in Fig.1a. The input to the network consists of the output of 40 mel filter bank channels plus the frame-level energy, along with the corresponding Δ and $\Delta\Delta$ parameters. The lowest, convolutional layer of the network divides the frequency axis into 7 wider frequency bands (the figure shows only 3 of these for clarity), and processes these by using separate sets of dedicated convolutional neurons or 'filters'. The output of the convolutional filters are concatenated and processed by three additional, fully connected layers (only two of these are shown in the figure). All network configurations studied here apply maxout neurons in both the convolutional layer and the fully connected layers [22]. More details on the operation, structure and parameter settings of the baseline model can be found in our earlier papers [7, 22].

While the baseline CNN divides and applies convolution along the frequency axis, it treats the time axis just like a fully connected DNN. A context of 17 consecutive frames served as the input for the baseline model, and the goal of this study was to examine how this time span could be extended using various splitting strategies. It should be added that these splitting strategies do not affect how the model applies convolution along the frequency axis in any way.

2.2. Deep split temporal context

The 'split temporal context' (STC) method was introduced in the framework of shallow HMM/ANN hybrids by Schwarz et al [16]. They assumed that the left and right contexts of the actual frame contain sufficient information for the identification of the frame, and these pieces of information are nearly independent. Based on this, they trained separate subnetworks on the two contexts, and merged their output only at a later stage. More recently, the STC approach was evaluated using DNNs instead of shallow networks, and an improved performance was reported for a large vocabulary task [17]. When using deep networks, Li et al. observed that one can also vary the position of the layer where the merging occurs. They got the best results when all the hidden layers were split, and the merging was performed by the uppermost softmax layer. We will mainly follow their slightly modified STC approach that they call the 'deep split temporal context' (DSTC) technique [18]. We should also add that it is possible to divide the context into more than two blocks [19, 18], but here we will just use a simple left-right context division.

Fig. 1b shows what happens if we apply the STC method to a CNN. As can be seen, the input is split at the actual frame, and the lowest layers are also split to process the left and right parts separately. As an example, in the figure the convolutional and one hidden layer are split. However, we will examine the optimal number of split layers experimentally.

2.3. Hierarchical modeling

The hierarchical modeling method stacks two networks on top of each other (see Fig. 1c). The lower network is trained on a context of acoustic feature vectors, and then a second network is trained on a context of posterior output vectors got from the first network [11, 23]. The time-span of this two-stage model can be easily increased if the upper, merger network operates on vectors that are placed farther apart, instead of using consecutive output vectors from the lower net [12]. Fig. 1c illustrates the case where the upper network combines three output vectors from the lower net, which are two frames apart.

Veselý et al. showed that the hierarchy of the two networks can be trained as one unit, and argued that the compound network can be interpreted as a model that performs convolution along time [14]. We achieved the best known phone recognition accuracy on TIMIT using a model that combines convolution along the frequency axis and a Vesely's-like convolution along the time axis [22].

The hierarchical model of Fig. 1c and the STC model of Fig. 1b are fundamentally different. First, while the subnetworks of the STC model have their own weights, the replicas of the lower network of the hierarchical model share the same weight set. Second, the sub-networks of the STC model are trained to recognize the label of the *same* input frame which is at the center position of the full context. In the hierarchical model, the lower network gives local estimates at several, *different* positions, and the merger network combines these into a single posterior estimate for the center position.

As the goal of STC is to give improved posterior estimates at a given position, while the essence of the hierarchical approach is to merge local estimates obtained at different positions, it is clear that the two approaches can be readily combined. For this purpose, we simply have to apply the STC method to the lower network of the hierarchical model. We will present experiments with this construct later on.

3. EXPERIMENTAL SETUP

The results reported are phone recognition error rates on the well-known TIMIT database. The training set consisted of the standard 3696 'si' and 'sx' sentences. A random 10% of the training set was held out as the 'development set', which was used only for the early stopping of DNN training, and no meta-parameter was tuned on it. For testing we used the *full test set*, as we thought the core test set was too small to draw reliable conclusions. However, since most comparative results are available only for the core test set, the best performing model was evaluated on the core set as well.

To get frame-level labels for training, forced alignment was performed with a conventional context-dependent HMM of 858 tied states. The phone label outputs were mapped to the usual set of 39 labels in the evaluation phase. During decoding a phone bigram language model was used. To be consistent with earlier studies on TIMIT (e.g. [2]), the language model weight and the phone insertion penalty parameters were set to 1.0 and 0.0, respectively.

The DNNs were trained using semi-batch backpropagation, with a batch size of 100. The training target function was the standard frame-level cross-entropy cost. The initial learn rate was set to 0.001 and held fixed while the error on the development set kept decreasing. Afterwards it was halved after each iteration, and the training was halted when the error rate decreased less than 0.1% in two subsequent iterations.

Apart from the softmax output layer, all the neurons of the networks were maxout units. Our baseline CNN consisted of one convolutional layer with 7 times 756 convolutional units, and 3 fully connected layers with 2714 neurons per layer [22].

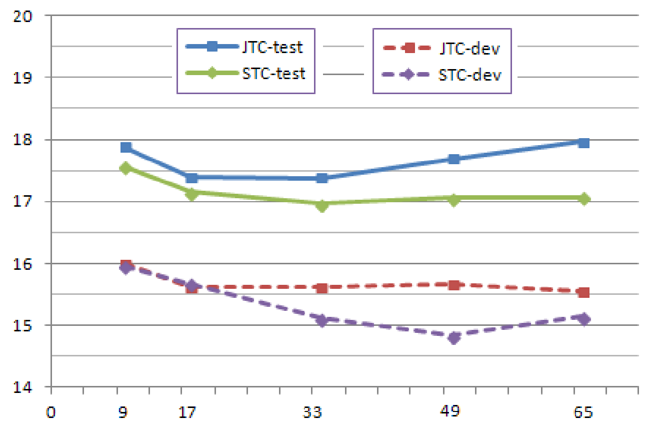

Fig. 2. Phone error rate as a function of the size of the input context (number of frames). Results are shown for the development set and the full test set, for the standard 'joint' (JTC) and the split (STC) temporal context modeling schemes.

4. RESULTS AND DISCUSSION

In the first experiment we examined how the input context size influences the recognition error rate. For this purpose, we set the number of neighbors used to 4, 8, 16, 24 and 32 frames (corresponding to input sizes of 9, 17, 33, 49 and 65 frames, respectively). First, we applied the standard, 'joint' representation method, where the frames are simply concatenated. As Fig.2 shows, on the development set the error rate quickly saturates at a context size of 17 frames, and does not improve

Number of split layers	Dev. set	Test set
1	15.7%	17.2%
2	15.4%	17.0%
3	15.1%	17.0%
4	15.1%	17.0%

Table 1. Phone error rates obtained by varying the number of split layers.

for larger context sizes. On the test set the error even starts to increase beyond 33 frames. This clearly shows that simply increasing the context can be detrimental beyond a point.

Next, we repeated the experiment with the STC model. The input context was split into left and right parts, with 3 frames of overlap. As Li got better results with more layers split [18], we decided to split the convolutional layer and two hidden layers, while the uppermost hidden layer was left intact. As the splitting decreases the number of connections, the number of neurons in the split layers was increased so that the joint and the split models had the same number of parameters.

As can be seen in Fig.2, for larger context sizes the phone error rate of the STC model also saturates and then starts to increase, but the optimum is attained at a later point (49 frames for the development set and 33 frames for the test set). Also, STC gives a moderate but consistent improvement in the error rate. For instance, at 33 frames the gain is 0.4% absolute (about 2.3% relative) on the test set, which proved to be significant with $p < 0.003$ in a paired t-test.

Yet another parameter of STC is the number of layers that are split. To test it, we varied the number of split layers from 1 to 4. That is, in the first case only the convolutional layer was split, while in the latter case the merging of the two network parts was performed by the output layer. Just as in the previous experiment, we ensured that the final parameter count of all models was the same by allocating more neurons for the split layers. The context size in this experiment was set to 33 frames. As shown in Table 1, we found that the merging of the two network parts could be delayed even until the final softmax layer, which accords with the observation of Li et al [18]. In all the subsequent experiments, we used the model that has 3 split and 1 merged hidden layers.

4.1. Combining STC with the hierarchical model

The hierarchical architecture of Fig.1c proved very efficient in our earlier studies [15, 7, 22], and was successfully used by other authors as well [13]. Although in earlier studies we applied two hidden layers in the upper, merger part of our network, here the softmax output layer was adjusted to perform the merging, without any additional hidden layers. The motivation for this was to get a fair comparison, as this way the parameter count of the baseline, the STC and the hierarchical models were roughly the same. The hierarchical model merged five output vectors from the lower sub-networks, the vectors lying 5-5 frames apart, which we found to be the best configuration earlier [15]. Both the baseline 'joint' model and the STC model were turned into a hierarchical model to see how this technique performed by itself, and in combination with STC. In both cases the models with 49 frames of context were extended, hence the hierarchical version of the models covered a time-span of 69 frames.

Network type	Devel. set	Test set
baseline	15.7%	17.7%
STC	14.8%	17.1%
hierarchical	14.8%	17.0%
STC+hierarchical	14.0%	16.6%
STC+hier.+dropout	13.4%	16.2%

Table 2. Phone error rates with STC, with hierarchical modeling, and with their combination.

The results in Table 2 show that, compared to the baseline, the STC method and the hierarchical scheme yield about the same gain. Moreover, as they are based on different paradigms, they can be efficiently combined to get an even better result. Compared to the case of modeling 69 frames the conventional way (cf. Fig 2), the combined model yields about 7.5% relative error rate reduction on the full test set.

In all previous experiments we observed that the improvement on the test set was usually smaller than on the development set, which can be a sign of overfitting. So, we retrained the best model using dropout, which is a simple and recently very popular method to alleviate overfitting [24]. As the last row of Table 2 shows, this way the combined model achieved 13.4% on the development set. By comparison, this score is practically equivalent with our earlier best result [22], but the model applied here contained *two fewer hidden layers*. For comparison, the best model was also evaluated on the core test set, and attained an error rate of 17.1%. Although this is worse than the 16.5% we achieved with using two more hidden layers [22], it still compares favorably with the next best reported scores of 17.4% got using CNNs with multi-resolution scattering features [25], or the 17.7% got with a recurrent network [26].

5. CONCLUSIONS

Here, we proposed a deep convolutional network architecture that combines the split temporal context and the hierarchical modeling approaches in order to expand the time-span of the network. We found that the advantages of the two methods can be combined, and with the compound model we could expand the time-span of our CNN to 69 frames. With this model we got a 7.5% relative error rate reduction compared to modeling this large context as one block, and obtained a phone error rate of 17.1% on the TIMIT core test set, which is among the best reported scores to date.

6. REFERENCES

[1] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. ICASSP*, 2013, pp. 8614–8618.

[2] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. ASLP*, vol. 20, no. 1, pp. 14–22, 2012.

[3] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. E. Hinton, "On rectified linear units for speech processing," in *Proc. ICASSP*, 2013, pp. 3517–3521.

[4] N. Morgan, "Deep and wide: Multiple layers in automatic speech recognition," *IEEE Trans. ASLP*, vol. 20, no. 1, pp. 7–13, 2012.

[5] J. Pan, C. Liu, Z. G. Wang, Y. Hu, and H. Jiang, "Investigation of deep neural networks (DNN) for large vocabulary continuous speech recognition: Why DNN surpasses GMMs in acoustic modeling," in *Proc. ISCSLP*, 2012, pp. 301–305.

[6] A. Janin, D. Ellis, and N. Morgan, "Multi-stream speech recognition: Ready for prime time?," in *Proc. Eurospeech*, 1999, pp. 591–594.

[7] L. Tóth, "Combining time- and frequency-domain convolution in convolutional neural network-based phone recognition," in *Proc. ICASSP*, 2014, pp. 190–194.

[8] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural network concepts to hybrid NN-HMM model for speech recognition," in *Proc. ICASSP*, 2012, pp. 4277 – 4280.

[9] H. Bourlard and N. Morgan, *Connectionist Speech Recognition - A Hybrid Approach*, Kluwer, 1994.

[10] P. Scanlon, D. P. W. Ellis, and R. Reilly, "Using mutual information to design class-specific phone recognizers," in *Proc. Interspeech*, 2003, pp. 857–860.

[11] J. Pinto et al., "Analysis of MLP based hierarchical phoneme posterior probability estimator," *IEEE Trans. ASLP*, vol. 19, no. 2, pp. 225–241, 2010.

[12] Vasquez, D. and Gruhn, R. and Minker, W., *Hierarchical Neural Network Structures for Phoneme Recognition*, Springer, 2013.

[13] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Language ID-based training of multilingual stacked bottleneck features," in *Proc. Interspeech*, 2014, pp. 1–5.

[14] K. Veselý, M. Karafiát, and F. Grézl, "Convolutive bottleneck network features for LVCSR," in *Proc. ASRU*, 2011, pp. 42 – 47.

[15] L. Tóth, "Convolutional deep rectifier neural nets for phone recognition," in *Proc. Interspeech*, 2013, pp. 1722–1726.

[16] P. Schwarz, P. Matějka, and J. Černocký, "Hierarchical structures of neural networks for phoneme recognition," in *Proc. ICASSP*, 2006, pp. 325–328.

[17] S. M. Siniscalchi, D. Yu, L. Deng, and C.-H. Lee, "Speech recognition using long-span temporal patterns in a deep network model," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 201–204, 2013.

[18] B. Li and K. C. Sim, "Modeling long temporal contexts for robust DNN-based speech recognition," in *Proc. Interspeech*, 2014, pp. 353–357.

[19] M. Baccouche, B. Besset, P. Collen, and O. Le Blouch, "Deep learning of split temporal context for automatic speech recognition," in *Proc. ICASSP*, 2014, pp. 5459–5463.

[20] O. Abdel-Hamid, L. Deng, and D. Yu, "Exploring convolutional neural network structures and optimization techniques for speech recognition," in *Proc. Interspeech*, 2013, pp. 3366 – 3370.

[21] T. N. Sainath, B. Kingsbury, A. Mohamed, and B. et al. Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR," in *Proc. ASRU*, 2013, pp. 315–320.

[22] L. Tóth, "Convolutional deep maxout networks for phone recognition," in *Proc. Interspeech*, 2014, pp. 1078–1082.

[23] H. Ketabdar and H. Bourlard, "Enhanced phone posteriors for improving speech recognition systems," *IEEE Trans. ASLP*, vol. 18, no. 6, pp. 1094–1106, 2010.

[24] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[25] V. Peddinti, T N. Sainath, S. Maymon, B. Ramabhadran, D. Nahamoo, and V. Goel, "Deep scattering spectrum with deep neural networks," in *Proc. ICASSP*, 2014, pp. 210–214.

[26] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.