Modeling Long Temporal Contexts for Robust DNN-based Speech Recognition

Bo Li, Khe Chai Sim

School of Computing, National University of Singapore, Singapore
li-bo@outlook.com, simkc@comp.nus.edu.sg

Abstract

Deep Neural Networks (DNNs) have been shown to outperform traditional Gaussian Mixture Models in many Automatic Speech Recognition tasks. In this work, we investigate the potential of modeling long temporal acoustic contexts using DNNs. The complete temporal context is split into several sub-contexts. Multiple sub-context DNNs initialized with the same set of Restricted Boltzmann Machines are fine-tuned independently and their last hidden layer activations are combined to jointly predict the desired state posteriors through a single softmax output layer. From preliminary experiments on the Aurora2 multi-style training task, our proposed system models a 65-frame temporal window of speech signals and yields a 4.4% WER, outperforming the best single DNN by 12.0% relatively. With the local independence assumption, both training and testing of the sub-context DNNs can be done in parallel. Moreover, our system has a relative 48.2% parameter reduction compared to a single DNN with the same amount of hidden units.

Index Terms: deep neural network, split temporal context

1. Introduction

Deep Neural Networks (DNNs) have been adopted in many Automatic Speech Recognition (ASR) systems. Large performance improvements have been reported compared to systems that use Gaussian Mixture Models (GMMs) to represent the state emission probability distributions [1]. Basic DNN-based ASR systems, when trained with large amounts of data, have been found to yield superior performance over advanced GMM-based systems that employ a combination of different optimization techniques [2]. Even for speech under adverse environments, DNNs have obtained comparable performance to the best GMM system with various noise reduction, feature enhancement and model-based compensation methods [3]. However, DNNs are still far from reaching humans’ expectations in real world ASR applications.

One major problem comes from unexpected signal distortions that were not seen in the training data. They can make the posterior probabilities generated from Neural Networks (NNs) unacceptably low. A step towards addressing the unreliable acoustic evidence for NNs might be in expanding the network architectures not only into deeper but also into longer and wider structures, where substantial temporal context attempts to cover whole co-articulation patterns of speech sounds, and multiple processing paths, attending to multiple parts of information-processing paths, attending to multiple parts of information-carrying space, attempt to capitalize on redundancies of coders of information in speech [4, 5]. It is supported by some known properties of human cognition [6] and has been pursued for some time [7–11].

In [12], Morgan also pointed out that while the deep processing structures of DNNs can provide improvements, choice of features and the structure with which they are incorporated, including layer width, can also be significant factors. They studied the effect of varying depth and width of DNN layers given a fixed total number of model parameters on Aurora2 [13]. They found that a large number of layers is not always optimum and for each noise level there is an optimum number of layers. In [14], bottleneck features are evaluated in the hybrid DNN-Hidden Markov Model (HMM) framework. It effectively uses a performance gains are relatively small and it is expected that a good part of the modeling performed in the bottleneck network can be learned in a standalone DNN as well.

In this work, we propose a Deep Split Temporal Context (DSTC) structure for DNNs to directly model long temporal contexts. By assuming independence between the sub-contexts of a long span of speech signals in early acoustic modeling stages, the complete context is split into multiple sub-contexts. They are first modeled independently and then merged at the last hidden layer to jointly give the final predictions. The idea of using sub-components to build networks with higher complexity for difficult speech recognition dates back to almost 25 years ago [15]. It is interesting to justify whether it is still useful for DNNs, which are already quite complex. The most similar work to ours are the original split temporal context (STC) system using shallow networks [16] and a recent system that simply replaces STC’s shallow NNs with DNNs [17]. The major difference lies in how we structure modular DNNs to make our final system equivalent to single DNNs with structure-constrained connections. The remaining part of the paper is organized as follows: Section 2 discusses different DNN structures and presents the proposed DSTC system. Experimental results on Aurora2 [18] are detailed in Section 3. Although there are many limitations to this data set: the small size, the read digit speech and the artificially added noise, it is commonly used and readers can focus more on the comparative results to understand the effect of the proposed method. Section 4 presents some discussions and we conclude the paper in Section 5.

2. Deep Split Temporal Context

Before the success of training DNNs, shallow NNs (Figure 1(a)) with 1 or 2 hidden layers are commonly adopted [19]. Due to the shallow structure, their acoustic modeling capability is limited. One of the early work aiming to improve their performance is the split temporal context (STC) system (Figure 1(b)) [16]. It assumes the independence of the left and right acoustic contexts and models them separately. The two sets of partial predictions generated are then merged with another NN. STC can robustly model long term acoustic dependencies and improve the recognition performance. Instead of treating NNs as black-box models that convert features to posteriors, they can be formulated as a layered feature extraction followed by a lin-
ear classification. We hence revised the original STC system to combine hidden representations rather than output posteriors (Figure 1(c)). Additionally, we believe that, with a good feature representation, linear classifiers are sufficient for phonetic discrimination. The original merging NN is thus simplified to a single softmax output layer, i.e. a linear regression with softmax normalization. In this way, our STC system (Figure 1(c)) with the same number of NN layers and the same set of effective layer sizes as a single NN (Figure 1(a)) will only have slightly more than half of the original NN’s model parameters.

DNN with multiple hidden layers (Figure 1(d)) has shown its superior feature abstraction capabilities, which makes it a perfect fit to STC systems (Figure 1(e)). However, one may argue that DNN is already good enough to capture long term dependencies. It is true that with many layers’ abstraction, DNN has largely outperformed shallow NNs. However, the use of long temporal context will require the increase of not only the depth but also the layer width. It further increases the number of model parameters dramatically. To address this issue, a Deep Split Temporal Context (DSTC) system (Figure 1(f)) is proposed. By assuming the local independence in the input context, we can use smaller DNNs to model each partial context separately. The co-articulation effects in speech caused by human speech production mechanism are modeled by the merging layer in our DSTC system. The local independence assumption is only used in the early stage of acoustic modeling, which will also be validated in later experiments. The training algorithm for the DSTC system is detailed in Algorithm 1. Although there are multiple partial context DNNs in the DSTC system, the training cost is actually reduced compared to training a DNN with the same hidden-capacity on the complete context. The unsupervised pre-training for sub-context DNNs is shared and only the fine-tuning differs, which can be parallelized. In our experiments, we run 200 epochs for the input-to-hidden layer and 100 epochs for all the other layers in the pre-training phase, while maximum 20 epochs are used for the fine-tuning. The main computation burden is the unsupervised pre-training. The extra softmax merging layer is a simple linear regression which is much faster than those DNN trainings.

Algorithm 1: Training a 2-block (i.e. left- and right-context) DSTC system on (2 + w + 1) frames of acoustic contexts with N hidden layers in each DNN using features $\alpha_t$ and labels $\lambda_t$ for each time slice $t$. The final system consists of the left context DNN $M_{\text{left}}$, the right context DNN $M_{\text{right}}$, and the final softmax layer $M_{\text{softmax}}$. Steps 2.1 and 2.2 are carried out in parallel.

1. **Shared Pre-training of the DNN $M_0$.**
   - **Begin**
   - Initialize $n = 0$, $x = \left( \alpha_{t-w}^T, \ldots, \alpha_t^T \right)$ for all $t$, $y = \left( \lambda_t^T \right)$ for all $t$;
   - **While** $n < N$ **do**
     - Train a RBM using $x$;
     - Set $x$ to the hidden activations of the current RBM;
     - Append a randomly initialized classification layer to the stacked RBMs to form our initial DNN, $M_0$;
   - **End while**
   - Stack the $N$ RBMs together;
   - **Begin**
     - Fine-tune the left context DNN $M_{\text{left}}$ **begin**
       - **Begin**
         - Set $x = \left( \alpha_{t-w}^T, \ldots, \alpha_t^T \right)$ for all $t$;
         - Fine-tune $M_0$ until convergence;
         - Remove the final softmax layer to get $M_{\text{left}}$;
         - Forward $x$ through $M_{\text{left}}$ to get $h_{\text{left}}$;
       - **End**
     - Fine-tune right context DNN $M_{\text{right}}$ **begin**
       - Set $x = \left( \alpha_{t}^T, \ldots, \alpha_{t+w}^T \right)$ for all $t$;
       - Fine-tune $M_0$ until convergence;
       - Remove the final softmax layer to get $M_{\text{right}}$;
       - Forward $x$ through $M_{\text{right}}$ to generate $h_{\text{right}}$;
     - **End**
   - **End**
   - **End**

2. **Parallel fine-tuning of partial context DNNs.**
   - **Begin**
     - **Begin**
       - Fine-tune the left context DNN $M_{\text{left}}$ **begin**
         - Set $x = \left( \alpha_{t-w}^T, \ldots, \alpha_t^T \right)$ for all $t$;
         - Fine-tune $M_0$ until convergence;
         - Remove the final softmax layer to get $M_{\text{left}}$;
         - Forward $x$ through $M_{\text{left}}$ to get $h_{\text{left}}$;
       - **End**
     - Fine-tune right context DNN $M_{\text{right}}$ **begin**
       - Set $x = \left( \alpha_{t}^T, \ldots, \alpha_{t+w}^T \right)$ for all $t$;
       - Fine-tune $M_0$ until convergence;
       - Remove the final softmax layer to get $M_{\text{right}}$;
       - Forward $x$ through $M_{\text{right}}$ to generate $h_{\text{right}}$;
     - **End**
   - **End**

3. **Train the final softmax layer $M_{\text{softmax}}$.**
   - Set $x = \left[ h_{\text{left}}^T, h_{\text{right}}^T \right]^T$;
   - Train $M_{\text{softmax}}$ from random initialization.
   - **End**

3. Experiments

In this section, we present the experimental results to justify the effectiveness of our proposed DSTC system. The benchmark...
noisy speech recognition task, Aurora2 [18], is adopted and the multi-style training data is used to train all the acoustic models. This comprises 8,440 utterances from 55 male and 55 female speakers. They are equally split into 20 subsets and each subset contains a few utterances of all training speakers. All the utterances in the same subset share the same noise condition and there are totally 4 different noise scenarios (train, babble, car and exhibition hall) at 5 different SNRs (20dB, 15dB, 10dB, 5dB and the clean condition). All three test sets, A, B and C, are used for evaluation. Set A has the same set of noise as the training set and set B has four new noise types, namely restaurant, street, airport and train station. For set C, there are only two noise scenarios (suburban train and street) but with additional channel distortions. For all three test sets, totally 6 different SNRs are used for evaluation purpose, which have one additional 0dB compared to the training set.

A standard complex back-end GMM-HMM system is built using per-utterance Cepstral Mean and Variance Normalized (CMVN) MFCC features by maximizing the training data likelihood. The word-based HMM for each digit has 16 states and we use 5 states for the silence model, leading to a total number of 181 HMM states. A total number of 3,699 Gaussians are used and the number of Gaussians for each state depends on the amount of samples for that state in the training data. The GMM-HMM system has an average WER of 8.3% over all the three test sets and is used to generate the per-frame DNN training labels. For DNN systems, we use 40-dimensional log Mel filter-bank features and the energy term, together with the first- and second-order derivatives as inputs. Per-utterance CMVN is also adopted for input feature normalization. No language model is used for this task and an equal probability digit-loop is also adopted for input feature normalization. No language and second-order derivatives as inputs. Per-utterance CMVN filter-bank features and the energy term, together with the first-

3.1. Baseline DNN

For the baseline system, we use a context window of 9 frames (w9) and 1,024-dimensional (1024D) hidden layers. The number of hidden layers is increased until degradations are observed. The WERs for each test set and also the average of all three sets are depicted in Figure 2. With 6 hidden (6H) layers, the w9-1024D-6H DNN achieves the WER of 5.3%.

3.2. DSTC System

NNs are inherently capable of modeling wide acoustic contexts. However, the large variations caused by the increase of the context length require a proper NN structure to be well modeled.

![Figure 2: Comparisons of multi-style trained w9-1024D DNNs on Aurora2 with different number of hidden layers.](image)

Table 1: Comparisons of multi-style trained NNs with different structures illustrated in Figure 1 on Aurora2.

<table>
<thead>
<tr>
<th>System</th>
<th>Model Capacity</th>
<th>WER (%)</th>
<th>Set A</th>
<th>Set B</th>
<th>Set C</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1H NN (a)</td>
<td>w9-1024D</td>
<td>6.3</td>
<td>6.6</td>
<td>7.3</td>
<td>6.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w17-1024D</td>
<td>6.8</td>
<td>7.1</td>
<td>8.3</td>
<td>7.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w17-2048D</td>
<td>7.1</td>
<td>7.9</td>
<td>8.8</td>
<td>7.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w17-3072D</td>
<td>7.5</td>
<td>8.9</td>
<td>9.7</td>
<td>8.5</td>
<td></td>
</tr>
<tr>
<td>1H STC (b)</td>
<td>w17-2048D</td>
<td>6.6</td>
<td>6.8</td>
<td>8.0</td>
<td>7.0</td>
<td></td>
</tr>
<tr>
<td>1H STC (c)</td>
<td>w17-2048D</td>
<td>5.8</td>
<td>6.0</td>
<td>6.8</td>
<td>6.1</td>
<td></td>
</tr>
<tr>
<td>6H DNN (d)</td>
<td>w9-1024D</td>
<td>4.6</td>
<td>5.9</td>
<td>5.7</td>
<td>5.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w17-1024D</td>
<td>4.9</td>
<td>5.9</td>
<td>5.9</td>
<td>5.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w17-2048D</td>
<td>4.6</td>
<td>5.6</td>
<td>5.5</td>
<td>5.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w17-3072D</td>
<td>4.7</td>
<td>5.3</td>
<td>5.3</td>
<td>5.1</td>
<td></td>
</tr>
<tr>
<td>6H DSTC (e &amp; f)</td>
<td>w17-2048D M1</td>
<td>4.9</td>
<td>6.4</td>
<td>6.4</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w17-2048D M2</td>
<td>4.8</td>
<td>6.8</td>
<td>6.4</td>
<td>5.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w17-2048D M3</td>
<td>4.5</td>
<td>6.8</td>
<td>6.4</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w17-2048D M4</td>
<td>4.6</td>
<td>6.6</td>
<td>6.3</td>
<td>5.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w17-2048D M5</td>
<td>4.4</td>
<td>6.2</td>
<td>6.1</td>
<td>5.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w17-2048D M6</td>
<td>4.4</td>
<td>5.6</td>
<td>5.6</td>
<td>5.1</td>
<td></td>
</tr>
</tbody>
</table>

It is shown by the dramatic degradations in Table 1 of the 1H shallow NNs (Figure 1(a)) when doubling the contexts from 4 frames on each side to 8 frames, i.e. from w9 to w17 input. All the w17-1H NNs perform worse than the w9-1H NN.

One way to increase the input context and the model capacity while maintaining a reasonable model size is to explore the potential independence structures in the wide context and model them separately with smaller NNs. The original STC system (Figure 1(b)) is firstly built with the existing w9-1024D-1H NN. It gives the WER of 7.0%, which is the lowest among all the existing w17 systems but is still higher than the w9-1024D-1H NN. We believe the reason is the information loss in the early decisions made by each sub-context NNs when combining posteriors. Moreover, the model size is increased rather than being reduced. With our proposed STC system (Figure 1(c)), we can achieve the WER of 6.1% which is a relative 7.6% improvement over the w9-1024D-1H NN.

Another way of modeling the large input variations is to use many layers of nonlinear processing. From WERs of different 6H DNNs (Figure 1(d)) in Table 1, they are much more robust than shallow NNs and can further improve the performance but have the requirement of increasing the model capacity by using higher dimensional hidden layers. The w17-3072D-6H DNN has the lowest average WER of 5.1% with 54.2 million parameters, compared to the w9-1024D-6H DNN’s 5.3% WER and 6.6 million parameters.

Next we use the w9-1024D-6H DNN as partial context DNNs to build our DSTC systems merging at different hidden layers (Figure 1(e)), which are effectively w17-2048D-6H systems. “Mx” indicates that the merging occurs at the xth hidden layer. From Table 1, the DSTC system merging at the last hidden layer performs the best. It indicates the importance of learning a better partial context feature representation over focusing too much on modeling the correlations in early stages. We will thus use only the DSTC that merges at the last hidden layer in following experiments, which gives the WER of 5.1% with only 13.1 million model parameters.
In this experiment, we use a wider context of 33 frames corresponding to 0.33 seconds of speech. Similarly, we build our DSTC systems by reusing the existing DNNs. With the w17-2048D-6H DNN, we split the w33 input window into left and right partial contexts, i.e., 2 blocks. While using the w9-1024D-6H DNN, the input window is split into 4 blocks. For comparison purpose, we also build a single DNN modeling the whole w33 input directly, which can be seen as the 1-block system. From results in Table 2, we achieve the WER of 4.8% with four w9-1024D-6H DNNs to model the 33 acoustic context frames. Although the improvement over the 2-block system is small but the 74.0% relative parameter reduction over the single DNN system is attractive. One probable explanation is that DNN's model capability is more related to the number of hidden units rather than the number of connections. It is similar to the fact that we have 10 billion cortical neurons in the human cerebral cortex but relatively sparse global connections. Otherwise if they were all connected with each other, the brain would be the size of a football stadium [21]. Furthermore, with eight w9-1024D-6H DNNs, we build a w65-8192D-6H DSTC which gives the best 4.4% WER with 46.5 million parameters. It has a relative 12.0% WER reduction and 48.2% parameter reduction over the best single DNN, i.e. the “1-block” w33-4096D-6H system. Additionally, a statistical significant test using \( p = 0.001 \) shows that the difference between these two systems is significant at the level of \( p = 0.001 \).

To validate the local independence assumption, we modify the 4-block DSTC system to overlap each block with its previous one by half of the partial context window. It gives an effective w34-6144D-6H DSTC system and the same 4.8% WER as the 4-block system, which indicates the explicit modeling of the partial context dependencies is unnecessary. This may be because of the inherent sliding window processing of the hybrid system, which has already captured those local dependencies.

### 4. Discussions

Our proposed DSTC system utilizes small partial context DNNs with shared unsupervised pre-training as the first stage high-level feature abstraction learning. It is capable of maintaining a high model capacity while using relatively fewer model parameters. The training cost is relatively lower than a fully connected single DNN with the same model capacity. Most importantly, it improves the generalization capability of DNNs. As a benchmark task, many researchers have reported their results on the Aurora2 dataset. Table 3 compares the best performing DSTC system with previously reported results on its multi-style training task. To our knowledge, we are among the best performing single-model multi-style trained systems. In [22], the authors use a multi-model single-style trained system. A set of speaker-, noise- and SNR-specific GMM-HMM systems are estimated from the training data to establish an environment structure, which is transformed to a target scenario for each testing case. And their 4.6% WER is achieved with a gender dependent version of that system. Our current DSTC system is a generic acoustic modeling approach that does not explicitly handle noisy data (unlike other noise-robustness techniques in Table 3) and yet it gives competitive performance. Effective incorporation of additional information, such as speaker, noise and SNR, into our DSTC system may probably yield further gains in recognizing speech under adverse environments. Our previous work [23] has the same performance but requires the interpolation of posteriors generated from two fully connected DNNs.

### 5. Conclusions

In this work, we proposed a Deep Split Temporal Context (DSTC) system to further improve the robustness of Deep Neural Network (DNN)-based speech recognition systems. The DSTC system uses multiple smaller DNNs to robustly model a long span of speech signals. These partial context DNNs share the same unsupervised pre-training and are fine-tuned in parallel. It reduces the DSTC system training cost compared to a single DNN with fully connected layers. Due to the independent modeling of each partial context, the whole DSTC system has fewer model parameters than a DNN with the same hidden capacity. These are all crucial to real world speech applications. On the Aurora2 multi-style training task, our DSTC system outperforms the best single DNN by 12.0% relative WER (4.4% vs. 5.0%) as well as 48.2% relative model parameter reduction.

### 6. Acknowledgments

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

---

**Table 2: Comparisons of DSTC systems with different number of partial contexts. All the systems have 6 hidden layers. The model sizes are in the order of millions.**

<table>
<thead>
<tr>
<th>System</th>
<th>Model Capacity</th>
<th>Model Size</th>
<th>WER (%)</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-block</td>
<td>w33-4096D</td>
<td>101.3</td>
<td>4.8</td>
<td>5.1</td>
<td>5.3</td>
<td><strong>5.0</strong></td>
<td></td>
</tr>
<tr>
<td>2-block</td>
<td>w33-4096D</td>
<td>51.3</td>
<td>4.4</td>
<td>5.1</td>
<td>5.2</td>
<td><strong>4.8</strong></td>
<td></td>
</tr>
<tr>
<td>4-block</td>
<td>w33-4096D</td>
<td>26.3</td>
<td>4.2</td>
<td>5.2</td>
<td>5.1</td>
<td><strong>4.8</strong></td>
<td></td>
</tr>
<tr>
<td>6-block</td>
<td>w49-6144D</td>
<td>39.4</td>
<td>4.0</td>
<td>4.9</td>
<td>4.9</td>
<td><strong>4.5</strong></td>
<td></td>
</tr>
<tr>
<td>8-block</td>
<td>w65-8192D</td>
<td>52.5</td>
<td>4.0</td>
<td>4.7</td>
<td>4.8</td>
<td><strong>4.4</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Table 3: Reported average WERs (%) of the multi-style trained systems on Aurora2.**

<table>
<thead>
<tr>
<th>System</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projection-based fMLLR [24]</td>
<td>8.6</td>
</tr>
<tr>
<td>Lasso4 [25]</td>
<td>7.9</td>
</tr>
<tr>
<td>MMSE-SPLICE [26]</td>
<td>7.8</td>
</tr>
<tr>
<td>VTS [27], CAUG-LM [28]</td>
<td>7.7</td>
</tr>
<tr>
<td>CMN [27]</td>
<td>7.1</td>
</tr>
<tr>
<td>Extended VTS [29]</td>
<td>7.0</td>
</tr>
<tr>
<td>PLP-Tandem [30]</td>
<td>6.9</td>
</tr>
<tr>
<td>AFE [27, 31]</td>
<td>6.8</td>
</tr>
<tr>
<td>CMVN [27]</td>
<td>6.5</td>
</tr>
<tr>
<td>NAT [27]</td>
<td>6.3</td>
</tr>
<tr>
<td>VTS + CMVN [32]</td>
<td>6.2</td>
</tr>
<tr>
<td>DNN (this work)</td>
<td><strong>5.0</strong></td>
</tr>
<tr>
<td>ESSEM-MCM [22]</td>
<td>4.6</td>
</tr>
<tr>
<td>Spectral Masking [23]</td>
<td><strong>4.4</strong></td>
</tr>
<tr>
<td>DSTC (this work)</td>
<td><strong>4.4</strong></td>
</tr>
</tbody>
</table>

---

1. [http://www1.icsi.berkeley.edu/Speech/docs/sctk-1.2/sctk.htm](http://www1.icsi.berkeley.edu/Speech/docs/sctk-1.2/sctk.htm)
7. References


