# DEEP CONVOLUTIONAL NEURAL NETWORKS FOR LVCSR

*Tara N. Sainath*[1]*, Abdel-rahman Mohamed*[2]*, Brian Kingsbury*[1]*, Bhuvana Ramabhadran*[1]

[1]IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, U.S.A.
[2]Department of Computer Science, University of Toronto, Canada
[1]{tsainath, bedk, bhuvana}@us.ibm.com, [2]asamir@cs.toronto.edu

## ABSTRACT

Convolutional Neural Networks (CNNs) are an alternative type of neural network that can be used to reduce spectral variations and model spectral correlations which exist in signals. Since speech signals exhibit both of these properties, CNNs are a more effective model for speech compared to Deep Neural Networks (DNNs). In this paper, we explore applying CNNs to large vocabulary speech tasks. First, we determine the appropriate architecture to make CNNs effective compared to DNNs for LVCSR tasks. Specifically, we focus on how many convolutional layers are needed, what is the optimal number of hidden units, what is the best pooling strategy, and the best input feature type for CNNs. We then explore the behavior of neural network features extracted from CNNs on a variety of LVCSR tasks, comparing CNNs to DNNs and GMMs. We find that CNNs offer between a 13-30% relative improvement over GMMs, and a 4-12% relative improvement over DNNs, on a 400-hr Broadcast News and 300-hr Switchboard task.

***Index Terms***— Neural Networks, Speech Recognition

## 1. INTRODUCTION

Recently, Deep Neural Networks (DNNs) have achieved tremendous success for large vocabulary continuous speech recognition (LVCSR) tasks, showing significant gains over state-of-the-art Gaussian Mixture Model/Hidden Markov Model (GMM/HMM) systems on a wide variety of small and large vocabulary tasks [1, 2, 3, 4, 5]. Convolutional Neural Networks (CNNs) [6, 7] are an alternative type of neural network that can be used to model spatial and temporal correlation, while reducing translational variance in signals.

CNNs are attractive compared to fully-connected DNNs that have been used extensively as acoustic models for a variety of reasons. First, DNNs are not explicitly designed to model translational variance within speech signals, which can exist due to different speaking styles [6]. This requires us to apply various speaker adaptation techniques to reduce feature variation. While DNNs of sufficient size could indeed capture translational invariance, this requires large networks with lots of training examples. CNNs on the other hand capture translational invariance with far fewer parameters by replicating weights across time and frequency. Second, DNNs ignore input topology, as the input can be presented in any (fixed) order without affecting the performance of the network [6]. However, spectral representations of speech have strong correlations, and modeling local correlations with CNNs has been shown to be beneficial in other fields [8].

In fact, CNNs have been heavily explored in the image recognition and computer vision fields, offering improvements over DNNs on many tasks [8], [9]. Recently, CNNs have been explored for speech recognition [10], also showing improvements over DNNs, however on a small vocabulary tasks with shallow networks. While [10] introduced a novel framework to model spectral correlations, one of the limitations of this spectral modeling approach was that the network was limited to one convolutional layer, unlike most CNN work which uses multiple convolutional layers [8]. In this paper, we explore spatial modeling similar to that done in the image recognition community, which allows for multiple convolutional layers and encourages deeper networks.

The first part of this paper explores the appropriate architecture for CNNs on LVCSR tasks. Specifically, we investigate how many convolutional vs. fully connected layers are needed, the optimal number of hidden units per layer, the optimal pooling strategy and the best type of input feature to be used with CNN.

Given this analysis, we then explore using CNNs on a 50-hr English Broadcast News (BN) task, in both hybrid [1, 5, 11] and neural network feature-based [12] setups. Naturally, our best pretrained DNN system offers a 14% relative improvement over the GMM/HMM, consistent with gains observed in the literature with DNNs vs. GMM/HMMs [2]. Comparing DNNs to CNNs, we find that a CNN hybrid system offers a 4% relative improvement over the hybrid DNN, and the CNN-based features offer a 7% relative improvement over the hybrid DNN. Given that we obtain the best performance with CNN-based features, we then explore using CNN-based features on two larger scale tasks - namely a 300-hr Switchboard task where the CNN offers between a 4-7% relative improvement over the DNN, and a 400-hr BN task where the CNN offers between a 10-12% relative improvement over the DNN.

The rest of this paper is organized as follows. Exploration of the appropriate CNN architecture for LVCSR tasks is described in Section 2. Initial results using the proposed CNN architecture on 50-hr BN are presented in Section 3, while results on larger tasks are described in Section 4. Finally, Section 5 concludes the paper.

## 2. CNN ARCHITECTURE

In this section, we describe CNNs in more detail and highlight experiments performed to learn the optimal CNN architecture for LVCSR.

### 2.1. CNN Description

A typical convolutional network architecture is shown in Figure 1. In a fully-connected network like DNNs, each hidden activation $\mathbf{h}_i$ is computed by multiplying the entire input $\mathbf{V}$ by weights $\mathbf{W}$ in that layer. However, in a CNN, each hidden activation is computed by multiplying a small local input (i.e. $[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$) against the weights $\mathbf{W}$. The weights $\mathbf{W}$ are then shared across the entire input space, as indicated in the figure. After computing the hidden units, a max-pooling layer helps to remove variability in the hidden units (i.e. convolutional band activations), that exist due to speaking styles,

channel distortions, etc. Specifically, each max-pooling unit receives activations from $r$ convolutional bands, and outputs the maximum of the activations from these bands. Most CNN work in image recognition has the lower network layers be convolutional, while the higher network layers are fully connected. In this section, we will explore how many convolutional vs. fully connected layers are needed, what is the optimal number of hidden units per layer, what is the best pooling strategy, and the best input feature type for CNNs.

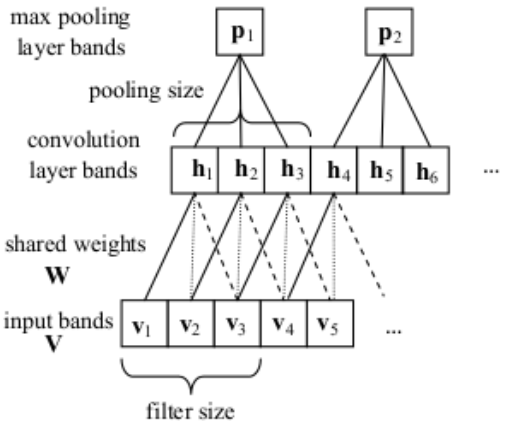


**Fig. 1**. Diagram showing a typical convolutional network architecture consisting of a convolutional and max-pooling layer. In this diagram, weights with the same line style are shared across all convolutional layer bands. Note this figure shows non-overlapping pooling, which is different than [10].

### 2.2. Experimental Setup

We perform preliminary experiments to learn the behavior of CNNs for speech on a 50-hr English Broadcast News task [2]. The acoustic models are trained on 50 hours of data from the 1996 and 1997 English Broadcast News Speech Corpora. Results are reported on the EARS `dev04f` set. Unless otherwise indicated, we use 40 dimensional log mel-filterbank coefficients, which exhibit local structure, to train the CNNs. The CNNs and fully-connected DNNs use 1,024 hidden units per each fully connected layer, and 512 output targets.

Following a recipe similar to [11], during fine-tuning, after one pass through the data, loss is measured on a held-out set and the learning rate is reduced by a factor of 2 if the held-out loss has not improved sufficiently over the previous iteration. Training stops after we have reduced the step size 5 times. All DNNs and CNNs are trained with cross-entropy, and results are reported in a hybrid setup.

### 2.3. Number of Convolutional vs. Fully Connected Layers

Most CNN work in image recognition makes use of a few convolutional layers before having fully connected layers. The convolutional layers are meant to reduce spectral variation and model spectral correlation, while the fully connected layers aggregate the local information learned in the convolutional layers to do class discrimination. However, the CNN work done thus far in speech [10] introduced a novel framework for modeling spectral correlations, but this framework only allowed for a single convolutional layer. We adopt a spatial modeling approach similar to the image recognition work, and explore the benefit of including multiple convolutional layers.

Table 1 shows the WER as a function of the number of convolutional and fully connected layers in the network. Note that for each experiment, the number of parameters in the network is kept the same. The table shows that increasing the number of convolutional layers up to 2 helps, and then performance starts to deteriorate. Furthermore, we can see from the table that CNNs offer improvements over DNNs for the same input feature set.

| # of Convolutional vs. Fully Connected Layers | WER |
|---|---|
| No conv, 6 full (DNN) | 24.8 |
| 1 conv, 5 full | 23.5 |
| 2 conv, 4 full | 22.1 |
| 3 conv, 3 full | 22.4 |

**Table 1**. WER as a Function of # of Convolutional Layers

### 2.4. Number of Hidden Units

CNNs explored for image recognition tasks perform weight sharing across all pixels. Unlike images, the local behavior of speech features in low frequency is very different than features in high frequency regions. [10] addresses this issue by limiting weight sharing to frequency components that are close to each other. In other words, low and high frequency components have different weights (i.e. filters). However, this type of approach limits adding additional convolutional layers [10], as filter outputs in different pooling bands are not related. We argue that we can apply weight sharing across all time and frequency components, by using a large number of hidden units compared to vision tasks in the convolutional layers to capture the differences between low and high frequency components. This type of approach allows for multiple convolutional layers, something that has thus far not been explored before in speech.

Table 2 shows the WER as a function of number of hidden units for the convolutional layers. Again the total number of parameters in the network is kept constant for all experiments. We can observe that as we increase the number of hidden units up to 220, the WER steadily decreases. We do not increase the number of hidden units past 220 as this would require us to reduce the number of hidden units in the fully connected layers to be less than 1,024 in order to keep the total number of network parameters constant. We have observed that reducing the number of hidden units from 1,024 results in an increase in WER. We were able to obtain a slight improvement by using 128 hidden units for the first convolutional layer, and 256 for the second layer. This is more hidden units in the convolutional layers than are typically used for vision tasks [6], [8], as many hidden units are needed to capture the locality differences between different frequency regions in speech.

| Number of Hidden Units | WER |
|---|---|
| 64 | 24.1 |
| 128 | 23.0 |
| 220 | 22.1 |
| 128/256 | 21.9 |

**Table 2**. WER as a function of # of hidden units

### 2.5. Optimal Feature Set

Convolutional neural networks require features which are locally correlated in time and frequency. This implies that Linear Discriminant Analysis (LDA) features, which are very commonly used in speech, cannot be used with CNNs as they remove locality in frequency [10]. Mel filter-bank (FB) features are one type of speech feature which exhibit this locality property [13]. We explore if any

additional transformations can be applied to these features to further improve WER. Table 3 shows the WER as a function of input feature for CNNs. The following can be observed:

- Using VTLN-warping to help map features into a canonical space offers improvements.
- Using fMLLR to further speaker-adapt the input does not help. One reason could be that fMLLR assumes the data is well modeled by a diagonal model, which would work best with decorrelated features. However, the mel FB features are highly correlated.
- Using delta and double-delta (d + dd) to capture further time-dynamic information in the feature helps.
- Using energy does not provide improvements.

| Feature | WER |
|---|---|
| Mel FB | 21.9 |
| VTLN-warped mel FB | 21.3 |
| VTLN-warped mel FB + fMLLR | 21.2 |
| VTLN-warped mel FB + d + dd | 20.7 |
| VTLN-warped mel FB + d + dd + energy | 21.0 |

**Table 3**. WER as a function of input feature

In conclusion, it appears VTLN-warped mel FB + d+dd is the optimal input feature set to use. This feature set is used for the remainder of the experiments, unless otherwise noted.

### 2.6. Pooling Experiments

Pooling is an important concept in CNNs which helps to reduce spectral variance in the input features. Similar to [10], we explore pooling in frequency only and not time, as this was shown to be optimal for speech. Because pooling can be dependent on the input sampling rate and speaking style, we compare the best pooling size for two different 50 hr tasks with different characteristics, namely 8kHZ speech - Switchboard Telephone Conversations (SWB) and 16kHz speech, English Broadcast News (BN). Table 4 indicates that not only is pooling essential for CNNs, for all tasks pooling=3 is the optimal pooling size. Note that we did not run the experiment with no pooling for BN, as it was already shown to not help for SWB.

| | WER-SWB | WER- BN |
|---|---|---|
| No pooling | 23.7 | - |
| pool=2 | 23.4 | 20.7 |
| pool=3 | 22.9 | 20.7 |
| pool=4 | 22.9 | 21.4 |

**Table 4**. WER vs. pooling

## 3. RESULTS WITH PROPOSED ARCHITECTURE

In this section, we explore using the proposed CNN architecture from Section 2 in both a hybrid [11] and neural network-based feature [12] setup, and compare them to two state-of-the art techniques used for LVCSR tasks, namely generatively pre-trained DNNs and GMM/HMMs. Our experiments are conducted on the same 50-hr English Broadcast News (BN) task used in Section 2, and results reported on both the EARS `dev04f` and `rt04` sets, used for development and testing respectively.

### 3.1. Experimental Setup

The GMM system is trained using our standard recipe [14], which is briefly described below. The raw acoustic features are 13-dimensional MFCC features with speaker-based mean, variance, and vocal tract length normalization (VTLN). Temporal context is included by splicing 9 successive frames of MFCC features into supervectors, then projecting to 40 dimensions using LDA. Next, a set of feature-space speaker-adapted (FSA) features are created using feature-space maximum likelihood linear regression (fMLLR). Finally, feature-space discriminative training and model-space discriminative training are done using the boosted maximum mutual information (BMMI) criterion. At test time, unsupervised adaptation using regression tree MLLR is performed. The GMMs use 2,220 quinphone states and 30K diagonal covariance Gaussians.

The hybrid DNN is trained using FSA features as input, with a context of 9 frames around the current frame. In [11], it was observed that a 5-layer DNN with 1,024 hidden units per layer and a sixth softmax layer with 2,220 output targets was an appropriate architecture for BN tasks. All DNNs are pre-trained generatively using the procedure outlined in [11]. During fine-tuning, the DNN is first trained using the cross-entropy objective function, followed by Hessian-free sequence-training [2]. The DNN-based feature system is also trained with the same architecture, but uses 512 output targets. A PCA is applied on top of the DNN before softmax to reduce the dimensionality from 512 to 40 [1]. Using these DNN-based features, we apply maximum-likelihood GMM training, followed by feature and model-space discriminative training using the BMMI criterion, and then do an MLLR at test time. The GMM acoustic model has the same number of states and Gaussians as the baseline GMM system.

The hybrid and CNN-based feature systems are trained using the optimal architecture and feature set from Section 2, namely VTLN-warped mel-FB with delta + double-delta. The number of parameters of the CNN matches that of the DNN, with the hybrid system having 2,220 output targets and the feature-based system 512 targets. No pre-training is performed, only cross-entropy and sequence-training.

### 3.2. Results

Table 5 shows the performance of CNN-based feature and hybrid systems, and compares this to DNN and GMM/HMM systems. The table indicates that the DNN hybrid offers a 13% relative improvement over the GMM/HMM, consistent with gains observed in the literature with DNNs vs. GMM/HMMs [2]. However, the CNN systems are far better than the DNNs. The CNN hybrid offers between a 3-5% relative improvement over this DNN hybrid, and the CNN-based feature system offers between a 5-6% relative improvement over the hybrid DNN. Given that we obtain the best performance with CNN-based features, we explore the performance of CNN-based features on two larger tasks in the next section.

| model | `dev04f` | `rt04` |
|---|---|---|
| Baseline GMM/HMM | 18.8 | 18.1 |
| Hybrid DNN | 16.3 | 15.8 |
| DNN-based Features | 16.7 | 16.0 |
| Hybrid CNN | 15.8 | 15.0 |
| CNN-based Features | **15.2** | **15.0** |

**Table 5**. WER for NN Hybrid and Feature-Based Systems

[1] Note that a PCA is used instead of an autoencoder [12] for dimensionality reduction because the performance of the two methods is very similar, and PCA training is much faster

## 4. RESULTS ON LARGER TASKS

In this section, we explore the performance of CNN-based features on two larger scale tasks.

### 4.1. Broadcast News

#### *4.1.1. Experimental Setup*

First, we explore scalability of CNNs on 400 hours of English Broadcast News [15]. Development is done on the DARPA EARS `dev04f` set. Testing is done on the DARPA EARS `rt04` evaluation set. The raw acoustic features are 19-dimensional perceptual linear predictive (PLP) features with speaker-based mean, variance, and VTLN, followed by an LDA and then fMLLR. The GMMs are then feature and model-space discriminatively trained using the BMMI criterion. At test time, unsupervised adaptation using regression tree MLLR is performed. The GMMs use 5,999 quinphone states and 150K diagonal-covariance Gaussians.

The generatively pre-trained DNN hybrid system use the same fMLLR features and 5,999 quinphone states as the GMM system described above, with a 9-frame context around the current frame, and use five hidden layers each containing 1,024 sigmoidal units. The DNN-based feature system is trained with 512 output targets. The DNN training begins with greedy, layerwise, generative pre-training, followed by cross-entropy training and then sequence training.

The CNN-based feature system is trained with VTLN-warped mel-FB with delta + double-delta features. The first convolutional layer has 128 hidden units, second has 256 hidden units, the three fully connected layers have 1,024 hidden units, and the softmax layer has 512 output targets. Again, the number of parameters of the CNN matches that of the DNN. No pre-training is performed, only cross-entropy and sequence-training. After 40-dimensional features are extracted with PCA, we apply maximum-likelihood GMM training, followed by discriminative training using the BMMI criterion, and then do an MLLR at test time.

#### *4.1.2. Results*

Table 6 shows the performance of the CNN-based features compared to both DNNs and GMM/HMMs. The CNN-based features offer between a 13-18% relative improvement over the GMM/HMM system, and between a 10-12% relative improvement over the DNN-based features. This helps to strengthen the hypothesis that CNNs are better than DNNs for speech tasks.

| model | `dev04f` | `rt04` |
|---|---|---|
| Baseline GMM/HMM | 16.0 | 13.8 |
| Hybrid DNN | 15.1 | 13.4 |
| DNN-based Features | 14.9 | 13.3 |
| CNN-based Features | **13.1** | **12.0** |

**Table 6**. WER on Broadcast News, 400 hrs

### 4.2. Switchboard

#### *4.2.1. Experimental Setup*

Second, we explore CNNs performance on 300 hours of conversational American English telephony data from the Switchboard corpus. Development is done on the `Hub5'00` set, while testing is done on the `rt03` set, where we report performance separately on the Switchboard (`SWB`) and Fisher (`FSH`) portions of the set.

The GMM systems are trained using the same methods used for Broadcast News, namely using speaker-adaptation with VTLN and fMLLR, followed by feature and model-space discriminative training with the BMMI criterion. Results are reported after MLLR. The GMMs use 8,260 quinphone states and 372K Gaussians. Similar to the Switchboard experiments in [2], the pre-trained DNN hybrid system use the same fMLLR features and 8,260 states as the GMM system described above, with an 11-frame context ($\pm 5$) around the current frame, and use six hidden layers each containing 2,048 sigmoidal units. The DNN hybrid system is pre-trained, followed by cross-entropy and sequence-training. The CNN-based feature system is trained with VTLN-warped mel-FB features. Two convolutional layers have 424 hidden units, four fully connected layers have 2,048 hidden units, and the softmax layer has 512 output targets. Again, the number of parameters of the CNN matches that of the DNN. No pre-training is performed, only cross-entropy and sequence-training. Again, after 40-dimensional features are extracted with PCA, GMM ML training is done followed by discriminative training, and then MLLR at test time.

#### *4.2.2. Results*

Table 7 shows the performance of the CNN-based features compared to both DNNs and GMM/HMMs. Note that we only include results for a Hybrid DNN. Using speaker-independent LDA features, we found on SWB that the hybrid DNN and DNN-based features had the same performance, roughly 13.3% on `Hub5'00`. In addition, from the BN results in Table 6, we see that the hybrid and DNN-based features have similar performance. We take these results to justify using the Hybrid DNN model as a strong and acceptable baseline. The CNN-based features offer between a 13-33% relative improvement over the GMM/HMM system, and between a 4-7% relative improvement over the hybrid DNN. Again, this confirms that across a wide variety of LVCSR tasks, CNNs are better than DNNs.

| | `Hub5'00` | `rt03` | |
|---|---|---|---|
| model | `SWB` | `FSH` | `SWB` |
| Baseline GMM/HMM | 14.5 | 17.0 | 25.2 |
| Hybrid DNN | 12.2 | 14.9 | 23.5 |
| CNN-based Features | **11.5** | **14.3** | **21.9** |

**Table 7**. WER on Switchboard, 300 hrs

## 5. CONCLUSIONS

In this paper, we explored how to make CNNs a more powerful model for speech tasks compared to DNN. Specifically, we empirically determined that having 2 convolutional and 4 fully connected layers and using a pooling strategy of 3 is optimal for CNNs. In addition, we found that the best locally correlated feature set for CNNs is vtln-warped mel-FB with delta+double-delta. We then explored the behavior of neural network features extracted from CNNs on a 400-hr BN and 300-hr SWB task, showing that CNNs offer between a 13-30% relative improvement over GMMs, and a 4-12% relative improvement over DNNs.

## 6. ACKNOWLEDGEMENTS

The authors would like to thank Hagen Soltau, George Saon and Stanley Chen for their contributions towards the IBM toolkit and recognizer utilized in this paper. Also, thank you to Etienne Marcheret for his help with discriminative training.

## 7. REFERENCES


[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable Minimum Bayes Risk Training of Deep Neural Network Acoustic Models Using Distributed Hessian-free Optimization," in *Proc. Interspeech*, 2012.

[3] F. Seide, G. Li, and D. Yu, "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks," in *Proc. Interspeech*, 2011.

[4] N. Jaitly, P. Nguyen, A. W. Senior, and V. Vanhoucke, "Application Of Pretrained Deep Neural Networks To Large Vocabulary Speech Recognition," in *Proc. Interspeech*, 2012.

[5] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition," vol. 20, no. 1, pp. 30–42, 2012.

[6] Y. LeCun and Y. Bengio, "Convolutional Networks for Images, Speech, and Time-series," in *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based Learning Applied to Document Recognition," in *Proceedings of the IEEE*, 1998.

[8] Y. LeCun, F. Huang, and L. Bottou, "Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting," in *Proc. CVPR*, 2004.

[9] S. Lawrence, "Face Recognition: A Convolutional Neural-Network Approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[10] 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.

[11] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A. Mohamed, "Making Deep Belief Networks Effective for Large Vocabulary Continuous Speech Recognition," in *Proc. ASRU*, 2011.

[12] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-Encoder Bottleneck Features Using Deep Belief Networks," in *Proc. ICASSP*, 2012.

[13] A. Mohamed, G. Hinton, and G. Penn, "Understanding how Deep Belief Networks Perform Acoustic Modelling," in *ICASSP*, 2012.

[14] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. IEEE Workshop on Spoken Language Technology*, 2010.

[15] B. Kingsbury, "Lattice-Based Optimization of Sequence Classification Criteria for Neural-Network Acoustic Modeling," in *Proc. ICASSP*, 2009.