A DEEP CONVOLUTIONAL NEURAL NETWORK USING HETEROGENEOUS POOLING FOR TRADING ACOUSTIC INVARIANCE WITH PHONETIC CONFUSION

Li Deng[1], Ossama Abdel-Hamid[2], and Dong Yu[1]
[1]Microsoft Research, One Microsoft Way, Redmond, WA, USA
[2]York University, Toronto, ON Canada
{deng|dongyu}@microsoft.com; ossama@cse.yorku.ca

ABSTRACT

We develop and present a novel deep convolutional neural network architecture, where heterogeneous pooling is used to provide constrained frequency-shift invariance in the speech spectrogram while minimizing speech-class confusion induced by such invariance. The design of the pooling layer is guided by domain knowledge about how speech classes would change when formant frequencies are modified. The convolution and heterogeneous-pooling layers are followed by a fully connected multi-layer neural network to form a deep architecture interfaced to an HMM for continuous speech recognition. During training, all layers of this entire deep net are regularized using a variant of the "dropout" technique. Experimental evaluation demonstrates the effectiveness of both heterogeneous pooling and dropout regularization. On the TIMIT phonetic recognition task, we have achieved an 18.7% phone error rate, lowest on this standard task reported in the literature with a single system and with no use of information about speaker identity. Preliminary experiments on large vocabulary speech recognition in a voice search task also show error rate reduction using heterogeneous pooling in the deep convolutional neural network.

Index Terms— convolution, heterogeneous pooling, deep, neural network, invariance, discrimination, formants

1. INTRODUCTION

The deep neural network (DNN) is an emerging technology that has recently demonstrated dramatic success in speech feature extraction and recognition, scaling very well from small [8][26][27][28] to medium [3][4][19][36] and to large [2][6][17] [21][34][32][37] tasks. (For recent reviews on the use of neural networks in speech recognition, see [29][17]). Some related DNN architectures have also demonstrated effectiveness in speech understanding and (small scale) image recognition tasks [7][9][35]. For larger scale image recognition and computer vision with high variability, a convolutional structure is often needed. Incorporation of convolution and subsequent pooling into a neural network gives rise to a Convolutional Neural Network (CNN) [24][25]. Stacking a CNN with a fully connected DNN or with one or more CNNs gives rise to a deep CNN [1][22][23]. The deep CNN has been shown to achieve a strong success for image recognition [5][22], similar to the success achieved by the DNN on speech recognition.

For images, the convolutional structure followed by pooling in a CNN is a natural way to embed translation invariance --- an object can be located at different places in an image while maintaining the same class identity of the object. However, for speech that is represented as a 2D "image" or spectrogram over time and frequency, things are different. This is because the same spectral pattern that is present in separate frequency bands (or at different temporal locations) would mean a different sound class. In other words, the simple convolution-pooling operation in the CNN, while introducing "translation" invariability either in frequency (or in time or in both), would cause confusion among speech classes and reduce the discrimination ability. This fundamental difference between the image and speech tasks motivated us to analyze the error patterns of phonetic recognition obtained by the deep CNN architectures typically used for image classification, and to design a new architecture, the heterogeneous-pooling CNN or HP-CNN.

Like image recognition, it is also desirable to derive invariant features by normalizing and reducing the variability in acoustic patterns --- e.g., the frequency shift due to vocal tract differences across speakers or due to the contextual effects on the formant-frequency changes [10][11] --- associated with the same speech class. However, the challenge is to strike the right balance between the extent of invariance (via convolution/pooling) and possible speech-class confusion when the shift becomes too large. The HP-CNN described in this paper is aimed at achieving such a tradeoff.

This paper is organized as follows. In Section 2, we provide motivations for the development of the HP-CNN based on error analysis on the results of a CNN with a fixed or homogeneous pooling size for all convolutional feature maps. Details of the HP-CNN are described in Section 3, highlighting the new set of hyper-parameters not present in any previous CNN and the principles by which they are determined. The roles of domain knowledge of speech are discussed. Section 4 is devoted to describing the "dropout" technique recently published in [18], which we modified and used to regularize the HP-CNN and the higher-layer fully-connected DNN used in our experiments. Experimental evaluation of the HP-CNN-DNN and dropout technique is presented in Section 5, reporting the best result in the literature on the standard TIMIT phone recognition task. We discuss related work in Section 6 and conclude the paper in Section 7.

2. EFFECTS OF CNN'S POOLING SIZE ON PHONETIC CONFUSION

To achieve the intended invariance to limited frequency shift via convolution and pooling (while retaining the discrimination ability), we use (scaled) filter-banks or spectrograms as the input feature for the CNN. Compared with MFCCs which have been most popular for speech recognition, spectrogram features not only retain more information (despite possibly redundant or irrelevant information for the recognition task), but also enable the use of the convolution and pooling operations to represent some typical

speech invariance and variability expressed explicitly in the frequency domain. An example of such speech variability is formant undershooting (or overshooting) caused by casual (or forced) speaking styles that have been mathematically represented by hidden dynamic or trajectory models of speech [11][12][13] [14].

As a baseline, we first explored a primitive CNN studied in [1], where the pooling size was fixed across all different convolutional feature maps. A larger pooling size enforces a greater degree of invariance in frequency shift but this also carries greater risk of being unable to distinguish among different speech sounds with similar formant frequencies. We have conducted detailed error analysis on the effects of the CNN's pooling size. When a fixed pooling size increases from one to 12, we found increasing confusions among the phones whose major formant frequencies are close to each other; in the meantime, the discrimination among phones whose formant frequencies are at extreme values tends to improve. This observation and analysis motivated the development of a new type of pooling strategy in DNNs, which we describe below.

3. HETEROGENEOUS POOLING IN THE CNN

The basic CNN structure we use has the following three characteristics: 1) input locality: We learn a set of filters, each of which receives the input from a local range of frequencies; 2) weight sharing: Each filter shifts along the frequency axis while computing the output with tied filter weights (this is mathematically equivalent to the ubiquitous convolution operation in DSP); and 3) max pooling or sub-sampling: High-level features with lower resolution are produced by the CNN. A combination of these characteristics endows the CNN with invariant properties for the input acoustic patterns that shift along the frequency axis. The extent to which the invariance is represented depends on the crucial parameter of the pooling size associated with 3) above.

We have extensively explored the use of a fixed pooling size for all convolutional layers corresponding to a full set of feature maps. As discussed in Section 2, any given pooling size corresponds to a tradeoff between the desired invariance over a range of frequency shift and the undesirable phonetic confusion caused by having similar but distinct phones' formants fall within the range. A natural way to take advantage of this tradeoff to the benefit of overall phonetic discrimination is to apply different or heterogeneous pooling sizes to various subsets of the full feature maps. We call this type of CNN as Heterogeneous-Pooling CNN, or HP-CNN. Figure 1 illustrates an HP-CNN, with two sets of pooling sizes, P_1 (of value 2) and P_2 (of value 3) shown, corresponding to N_1 and N_2 maps, respectively, in the convolution layer.

In general, the number of different pooling sizes can be much larger than two, constrained by the total number of feature maps. A general HP-CNN(m) is characterized by the following hyper-parameter set: [P_1, N_1; P_2, N_2; … , P_m, N_m]. This gives the total number of feature maps in the convolutional layer: $N = \sum_{i=1}^{m} N_i$.

The optimal choice of the above hyper-parameters' values is determined by the convolution filter design and, more importantly, by the nature of the phonetic space expressed in scaled frequency in accordance with the input filter-bank features. For example, for highly fluent speech with a faster speaking rate, the formant space of speech acoustics tends to shrink [11][31]. Thus, a larger number (N_i) of feature maps, which gives more presentational power analogous to more hidden units in the fully-connected DNN, should be given to lower pooling size P_i in order to bias the tradeoff towards a lesser degree of invariance and a higher degree of discrimination. This type of domain knowledge on speech acoustics has been incorporated into the design of hyper-parameter values in our experiments reported in Section 5.

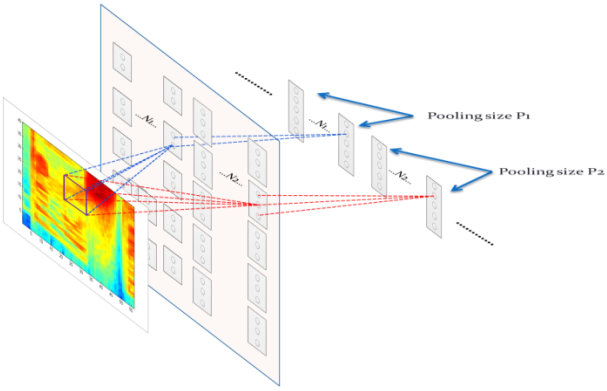

Figure 1. *Illustration of convolution and heterogeneous-pooling layers in a HP-CNN, followed by a fully connected DNN (not shown here). Note the different columns represent separate feature maps corresponding to the same input vector, not separate time frames.*

The HP-CNN described above effectively converts the spectrogram features of speech into a higher-level representation at the heterogeneous pooling layer. This HP-CNN output, which is equipped with partial within-class invariance, is followed by a subsequent fully-connected DNN to form a deep-CNN architecture that can be used for speech recognition after an interface to an HMM in the same manner as described in [3][4][26].

4. REGULARIZING HP-CNN BY "DROPOUT"

Recently, a regularization procedure called "dropout" [18] significantly improved image recognition and phone recognition accuracy by randomly omitting half of the hidden units in each layer of a DNN during training while doubling the size of each layer. During run time, the effect is efficiently compensated by scaling down the DNN weights. This regularization mechanism lies in its ability to prevent "co-adaptation" in which a feature detector is only helpful in the context of other feature detectors. Like different hidden units in a DNN which tend to co-adapt each other, different feature maps and different hidden units within the same map in the HP-CNN also co-adapt. Hence, the HP-CNN is expected to benefit from the dropout technique for its regularization. Due to the weight constraints in the convolution layer and the heterogeneous nature of the pooling layer in the HP-CNN, co-adaptation among the hidden units within and across feature maps in the HP-CNN would behave differently from those in the DNN. Our experimental results in Section 5 have shown more significant improvement in phone recognition accuracy than reported in [18] on the same task of phone recognition.

In contrast to applying dropout for the DNN in the TIMIT task as reported in [18], we found that applying dropout to input filter-bank features for the HP-CNN has not been effective. Therefore, we apply dropout only to the hidden units in the deep HP-CNN, including those in both the convolution and pooling layers, as well as in all the DNN layers on top of the HP-CNN's pooling layer.

Further, in contrast to [18], we found that the dropout rate for both the HP-CNN and DNN needs to be significantly smaller than 0.5 reported in [18] in order to make it effective in achieving low recognition errors. Too large a dropout rate not only drastically slows down the convergence in training but also leads to higher recognition errors despite increases in the number of hidden units as suggested in [18]. A typical effective value of the dropout rate is between 0.05 and 0.25 for the deep HP-CNN with *N=100* (the size of the convolutional feature maps) and with 2000 hidden units per layer in the DNN on top of the HP-CNN.

5. EXPERIMENTAL EVALUATION

5.1. Experimental setup and HP-CNN-DNN training

Reported in this paper are mainly the results of TIMIT's standard phonetic recognition task as used in [1][12][27][26][28][33]. Speech feature vectors are generated by a Fourier-transform-based filter-bank, and include 40 coefficients distributed on a Mel scale, plus their first and second temporal derivatives. Standard setups are used, and in particular, we report the 39-folded-class results of phonetic recognition using the 192 core test set sentences. The targets of 183 mono-phone states are obtained by using a tri-phone HMM model to generate state-level forced alignments. No phone segment information provided in the TIMIT database is used.

The training objective is the standard, frame-level cross-entropy, simpler than the full-sequence objective in our earlier work [27]. For training the HP-CNN followed by a DNN with three fully-connected hidden layers (HP-CNN-DNN), we find near perfect correlation between the frame error rate and the objective, as shown in Figure 2.

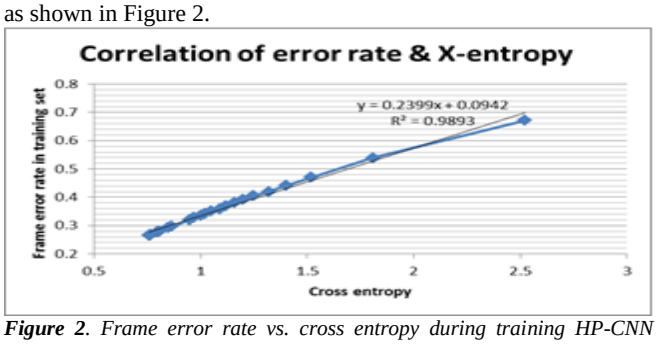

Figure 2*. Frame error rate vs. cross entropy during training HP-CNN followed by a DNN with three fully connected layers. Cross entropy is the objective function for optimization, correlating well with training error rate.*

5.2. Phonetic recognition results

In Table 1, we summarize the phone recognition error rates for several deep networks, including the HP-CNN-DNN with and without using dropout regularization. Note that with a fixed pooling size P in CNN-DNN, the error rates vary substantially from P=1 to P=12, and P=6 gives the lowest error rate. The HP-CNN-DNN uses a distributed P from 1 to m=12, and gives significantly lower errors. The HP-CNN-DNN with dropout regularization achieves the lowest published error rate of 18.7% on this same task. We note that in [33] the same 18.7% error rate was reported by exploiting multiple systems and using additional information about speaker identity for adaptation. Our single deep HP-CNN-DNN system does not make use of any information about speaker identity as in the standard evaluation protocol. All the CNNs shown in Table 1 have the same structure (e.g., the same number of convolutional feature maps) except for the differences in the pooling layer.

For the HP-CNN-DNN configuration which produced the lowest error rate, we plot the learning curve (for the training error rate) in blue and phone recognition accuracy for the development (dev) or validation set (in red) and test set (in green) in Figure 3. The stopping criterion is determined solely by the behavior in the dev set, following the same procedure as described in [26]. In Figure 4, we show the confusion matrix of 39 merged-phone classes after dynamic programming based decoding. Further, we present in Figure 5 the normalized values of the diagonal elements in the confusion matrix (i.e., correct phone recognition) for several deep networks in Table 1, including the best-performing one (in light blue). Publications of such detailed results are expected to benefit future research into more advanced techniques. (Related error analysis was performed in comparing DNN and hidden trajectory systems in 2009, which ignited further research into DNN; see [15][16]).

Deep Networks	Phone Error Rate
DNN (fully connected)	22.3%
CNN-DNN; P=1	21.8%
CNN-DNN; P=12	20.8%
CNN-DNN; P=6 (fixed P, optimal)	20.4%
CNN-DNN; P=6 (add dropout)	19.9%
CNN-DNN; P=1:m (HP, m=12)	**19.3%**
CNN-DNN; above (add dropout)	**18.7%**

Table 1: *TIMIT core test set phone recognition error rate comparisons.*

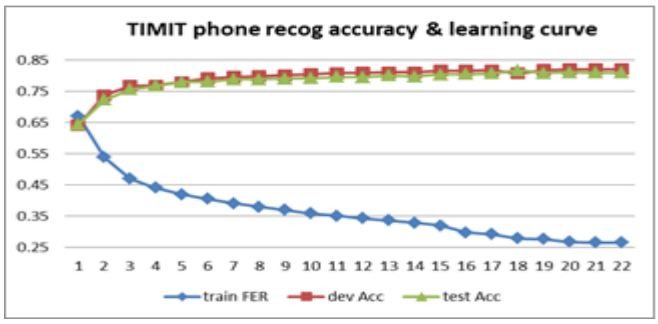

Figure 3*. Frame error rate in* training *(blue) and phone recognition accuracy of dev set (red) and core test set (green) as a function of training epochs.*

Our more recent, preliminary experiments extending the TIMIT task to large vocabulary speech recognition in a voice search task have shown error rate reduction from 32.4% to 30.1% after incorporating heterogeneous pooling in the otherwise identical deep CNN system.

6. RELATION TO OTHER WORK

The work presented in this paper has focused on the motivation and construction of the HP-CNN. This network makes a flexible tradeoff between invariance of speech patterns expressed in the frequency domain and discrimination of speech classes. The earlier work of [1] adopted fixed or homogeneous pooling, lacking such flexibility.

While the present study is related to recent work on image recognition where "tiled" CNN was proposed [23], they differ from each other in two main aspects. In the tiled CNN, weights are shared over the entire image. Hence the total number of different sets of weights depends on the tile size. The HP-CNN reported in this paper uses local weight sharing appropriate for speech, and it uses completely different sets of filters for each pooling node.

Further, the design of the pooling takes into account discrimination of classes in the HP-CNN, not so in the tiled CNN.

The dropout regularization technique used in this work is a variant of that published in [18]. The main difference is that we apply dropout in both the convolution and fully-connected layers while the original dropout technique [18] was applied to only the fully-connected layers. Experimentally we found that the best dropout rate is between 0.05 to 0.25 for the typical size of the network, and [18] reported only the results with the dropout rate of 0.5 incurring much longer training time (reflected by many more training epochs) than ours.

One major motivation of the HP-CNN is to handle the acoustic variability in frequency, in common with the feature normalization technique exploited in [28]. The key difference is that our method integrates such variability normalization and speech class discrimination into a single framework in learning, while the study of [28] separates out feature normalization and the deep net learning. We report much stronger results than [28] on the same evaluation task.

7. SUMMARY AND DISCUSSION

In the work reported in this paper, we have developed a novel deep learning architecture, an HP-CNN followed by a DNN. We motivate the HP-CNN by domain knowledge of speech pertaining to the phonetic space expressed in the formant-frequency distributions among distinct phonemes, as well as to how the phonetic space would shrink as the speaking style becomes more casual. The HP-CNN is also motivated by the error analysis carried out on the behavior of CNNs with a varying but fixed pooling size across all convolutional features maps. We use a weighted mix of pooling sizes in the HP-CNN to devise a strategy for trading between within-class invariance and between-class discrimination. This strategy reduces the TIMIT core test set's phone recognition error rate to 19.3% from 20.4% obtained with the optimal but single fixed pooling size. After regularizing the CNN using a variant of the "dropout" technique, the error rate of the HP-CNN-DNN drops further to 18.7%, from 19.9% with the same dropout but without heterogeneous pooling. Note that all the error analysis and domain knowledge of speech leading to the fundamental concept of the HP-CNN have been based on the invariance-vs.-discrimination interpretation of the convolution and pooling operations in the CNN. All this has been made possible only after a change from the use of MFCCs to spectrogram-like features, supporting the basic tenet of deep learning: back to more primitive features while letting machine learning to automatically discover the appropriate high-level features.

We are currently extending the application of the HP-CNN-DNN to larger, real-world tasks, where we expect a greater need for trading invariance with confusion due to the freer speaking style and hence stronger shrinking in the phonetic space.

```
---------------------- Overall Results ----------------------
SENT: %Correct=1.04 [H=2, S=190, N=192]
WORD: %Corr=83.98, Acc=81.33 [H=6158, D=338, S=837, I=194, N=7333]
------------------------ Confusion Matrix -------------------------
       a   a   a   a   a   b   c   d   d   d   e   e   e   f   g   h   i   i   j   k   l   m   n   n   o   o   p   p   r   s   s   t   t   u   u   v   w   y   z
       a   e   h   w   y       h       h   x   h   r   y               h   h   y   h                       g   w   y       a               h       h   h   w
                                                                                                                   u                                   Del [ %c / %e]
  aa 190   0  10   2   7   0   0   0   0   0   2   0   0   0   1   0   0   0   0   3   1   0   0   1   1   0   0   2   0   0   0   0   0   0   0   0   8 [86.4/0.4]
  ae   3  63   2   3   2   0   0   0   0   0  15   0   3   0   0   0   2   0   0   0   0   0   1   0   1   0   0   1   0   1   0   0   0   0   0   0   0   8 [64.9/0.5]
  ah   8   0 218   1   2   0   0   1   0   0   8   8   0   0   0   0  35   0   0   0   0   0   0   0   7   0   0   1   1   0   0   0   0   2   3   0   0   0   0  26 [73.9/1.1]
  aw   0   2   0  24   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0 [80.0/0.1]
  ay   5   0   4   1  71   0   0   0   0   0   1   0   3   0   0   0   1   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   2 [81.6/0.2]
   b   0   0   0   0   0 115   0   0   1   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   8   0   0   0   0   0   0   0   0   2   0   0   5 [90.6/0.2]
  ch   0   0   0   0   0   0  32   0   0   0   0   0   0   0   0   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0   1   3   0   0   0   0   0   0   1 [82.1/0.1]
   d   0   0   0   0   0   0   0  79   5   2   0   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  13   0   0   0   1   0   0   0  10 [76.7/0.3]
  dh   0   0   0   0   0   1   0   4 100   3   0   0   0   0   0   1   0   0   0   1   3   1   1   0   0   0   2   0   0   0   0   0   1   1   0   0   2   0   0   2   5 [81.3/0.3]
  dx   0   0   0   0   0   0   0   0   5   1  76   0   0   0   0   0   0   0   0   0   1   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   5 [89.4/0.1]
  eh   1   8  15   0   1   0   0   0   0   0   0 127   0   2   0   0   0  24   1   0   0   0   0   0   0   1   0   0   0   0   1   0   0   0   0   1   0   0   0   0   7 [69.8/0.8]
  er   0   0   3   0   1   0   0   0   0   0   1 192   0   0   0   0   0  11   0   0   0   0   0   0   0   0   0   0   0   0  20   0   0   0   0   0   1   0   0   0   0   5 [83.8/0.5]
  ey   0   1   0   0   1   0   0   0   0   0   2   0  97   0   0   0   0   6   6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1 [85.8/0.2]
   f   0   0   0   0   0   1   0   0   0   0   0   0   0 119   0   0   0   0   0   0   0   0   0   0   0   0   0   2   2   0   1   0   0   2   0   0   3   0   0   0   1 [91.5/0.2]
   g   0   0   1   0   0   1   0   3   0   0   0   0   0   0  54   0   0   0   0   3   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   3 [85.7/0.1]
  hh   0   0   0   0   0   0   1   0   0   2   0   0   0   0   0  59   0   0   0   0   0   0   1   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0  13 [92.2/0.1]
  ih   0   1  32   0   0   0   0   0   0   0  14  10   5   0   0   0 448  23   0   0   0   0   0   0   4   0   0   3   0   1   1   0   0   2   5   1   0   1   0  29 [81.3/1.4]
  iy   0   0   1   0   0   0   0   0   0   0   0   0   6   0   0   0  11 214   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0   0   3   0   5 [89.9/0.3]
  jh   0   0   0   0   0   0   1   1   0   0   0   0   0   0   0   0   0   0  36   0   0   0   0   0   0   0   0   0   0   1   0   1   0   0   0   0   0   0   1   1 [87.8/0.1]
   k   1   0   0   0   0   0   0   1   0   0   0   0   0   0   2   0   0   0   0 151   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   8 [96.2/0.1]
   l   3   0   1   2   0   0   1   0   3   0   0   0   0   1   0   0   0   0   0   0 244   2   0   0   9   0   2   0   3   0   0   1   0   0   0   1   2   0   0  16 [88.7/0.4]
   m   0   0   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2 179   8   0   0   0   1   1   0   0   0   0   0   0   0   0   2   0   0   9 [91.8/0.2]
   n   0   0   1   0   0   2   0   2   1   9   0   1   1   1   0   0   0   2   0   0   0   0   8 324   7   0   0   0   3   1   1   0   0   0   0   0   0   0   0  22 [89.3/0.5]
  ng   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   0   0   0   0   3   5  40   0   0   0   0   1   0   0   0   0   0   0   0   0   0   1 [78.4/0.2]
  ow  10   0   2   2   0   0   0   0   0   0   0   1   1   0   0   0   2   0   0   0   4   0   0   0  64   0   0   1   0   0   0   0   0   0   1   0   0   0   0   1 [72.7/0.3]
  oy   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   2   0   0   0   1   0   0   0   0  12   0   0   0   0   0   0   0   0   0   0   0   0   0   0 [75.0/0.1]
   p   0   0   0   0   0   5   0   0   1   0   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0   0 117   0   0   0   0   4   0   0   0   1   0   0   0   5 [89.3/0.2]
 pau   0   0   4   0   0   0   0   0   0   3   4   0   0   0   0   0   1   0   0   0   0   1   1   0   1   1   0 1486   0   1   0   1   0   0   0   2   0   0   0  73 [98.7/0.3]
   r   0   0   1   0   0   0   0   0   0   0   0  10   0   0   0   0   0   0   0   1   0   0   1   0   0   1   0   1   0 225   0   0   1   0   0   0   2   0   0   0  27 [92.6/0.2]
   s   0   0   0   0   0   0   6   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 294   0   1   0   0   0   0   0   0  17   3 [92.5/0.3]
  sh   0   0   0   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   2   0   2  68   0   0   0   0   0   0   1   0   0 [88.3/0.1]
   t   0   0   0   0   0   0   1   3   4   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0   1   0 162   2   0   0   0   0   0  12 [92.6/0.2]
  th   0   0   0   0   0   0   0   0   3   0   0   0   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   2   0   3  26   0   0   0   0   0   0 [68.4/0.2]
  uh   0   0   0   3   0   0   0   0   0   0   0   1   1   0   0   0   0   6   0   0   0   0   1   0   0   0   0   1   0   0   0   1   0   0   0  11   2   0   0   0   0   2 [40.7/0.2]
  uw   0   0   0   0   0   0   0   0   0   0   0   1   4   0   0   0   0   0   5   3   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   1  59   0   0   0   0 [79.7/0.2]
   v   0   0   0   0   0   0   4   0   0   2   0   0   0   0   0   1   0   0   1   0   0   0   0   3   0   0   0   1   0   2   2   0   0   0   0   0   0   0   0   0  70   0   0   0   7 [81.4/0.2]
   w   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   1   1 130   0   0   8 [95.6/0.1]
   y   0   0   0   0   0   1   0   0   0   0   0   1   0   0   0   0   0   0   0   2   0   0   0   0   1   0   0   1   0   1   0   1   0   0   0   0   0   0   0  35   0   7 [81.4/0.1]
   z   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   0   2   0  25   1   0   1   0   0   0   0   0 147   2 [82.1/0.4]
 Ins  10   3  13   1   4   2   2  12   5   1   4   4   0   2   1   6  14   5   2   6   0   1   5   1   4   1   3  49   9   4   1   3   1   0   3   1   4   6   1
```

Figure 4*. Phone confusion matrix (including Deletion and Insertion) in core test set produced by HTK's HResults tool for the standard 39-class phonetic recognition task of TIMIT.*

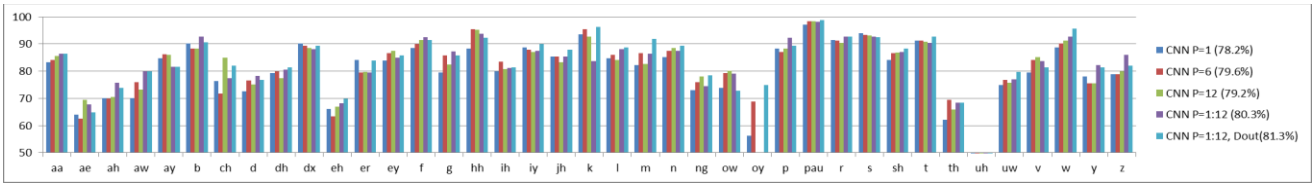

Figure 5. *Phone classification accuracy for each of the 39 merged classes in the TIMIT core test set for five different pooling and training methods. The last one (marked with P=1:12 Dout) uses dropout regularization during training.*

8. REFERENCES

[1] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," ICASSP, 2012.

[2] X. Chen, A. Eversole, G. Li, D. Yu, and F. Seide, "Pipelined back-propagation for context-dependent deep neural networks," Interspeech, 2012.

[3] G. Dahl, D. Yu, L. Deng. "Large vocabulary continuous speech recognition with context-dependent DBN-HMMs," ICASSP, 2011.

[4] G. Dahl, D. Yu, L. Deng, and A. Acero. "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition." IEEE Trans. Speech and Audio Proc., vol. 20, no. 1, pp. 30 – 42, 2012.

[5] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng. "Large scaled distributed deep networks," NIPS, 2012.

[6] L. Deng, J. Li, J. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero. "Recent advances of deep learning for speech research at Microsoft," ICASSP, 2013.

[7] L. Deng, D. Yu, and J. Platt. "Scalable stacking and learning for building deep architectures," ICASSP, 2012.

[8] L. Deng, M. Seltzer, D. Yu, A. Acero, A. Mohamed, and G. Hinton, "Binary coding of speech spectrograms using a deep auto-encoder," Interspeech, 2010.

[9] L. Deng, G. Tur, X. He, and D. Hakkani-Tur, "Use of kernel deep convex networks and end-to-end learning for spoken language understanding," IEEE SLT, 2012.

[10] L. Deng and J. Ma, "Spontaneous speech recognition using a statistical coarticulatory model for the vocal tract resonance dynamics," J. Acoust.Soc.Am., vol. 108, pp. 3036-3048, 2000.

[11] L. Deng, D. Yu, and A. Acero. "A bidirectional target filtering model of speech coarticulation: Two-stage implementation for phonetic recognition," IEEE Transactions on Audio and Speech Processing, vol. 14, pp. 256-265, 2006.

[12] L. Deng, D. Yu, and A. Acero. "Structured speech modeling," IEEE Trans. on Audio, Speech and Language Processing, vol. 14, no. 5, pp. 1492-1504, 2006.

[13] L. Deng and D. Yu. "Use of differential cepstra as acoustic features in hidden trajectory modeling for phonetic recognition," ICASSP, 2007.

[14] L. Deng and D. O'Shaughnessy, SPEECH PROCESSING --- A Dynamic and Optimization-Oriented Approach, Publisher: Marcel Dekker Inc., June 2003.

[15] L. Deng, D. Yu, and G. Hinton. "Deep Learning for Speech Recognition and Related Applications" NIPS Workshop, 2009 http://nips.cc/Conferences/2009/Program/event.php?ID=1512

[16] L. Deng, G. Hinton, and B. Kingsbury. "New types of deep neural network learning for speech recognition and related applications: An overview," ICASSP, 2013.

[17] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. "Deep neural networks for acoustic modeling in speech recognition," IEEE Signal Processing Magazine, Vol. 29, No. 6, pp. 82-97, Nov., 2012.

[18] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, & R. Salakhutdinov. "Improving neural networks by preventing co-adaptation of feature detectors," arXiv: 1207.0580v1, 2012.

[19] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," IEEE Trans. Pattern Analysis and Machine Intelligence (special issue of Learning Deep Architectures), 2013, to appear.

[20] N. Jaitly, P. Nguyen, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," Interspeech, 2012.

[21] B. Kingsbury, T. N. Sainath, and H. Soltau. "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization," Interspeech, 2012.

[22] A. Krizhevsky Ilya Sutskever G. Hinton. "ImageNet classification with deep convolutional neural networks," NIPS, 2012.

[23] Q. Le, J. Ngiam, Z. Chen, D. Chia, W. Pang, and A. Ng. "Tiled convolutional neural networks," NIPS, 2010.

[24] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition," Proceedings of the IEEE, pp. 2278–2324, 1998.

[25] Y. LeCun, F. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," Proc. IEEE CVPR, 2004.

[26] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," IEEE Trans. on Audio, Speech, and Language Processing," Vol. 20, no. 1, pp. 14–22, 2012.

[27] A. Mohamed, D. Yu, and L. Deng. "Investigation of full-sequence training of deep belief networks for speech recognition," Interspeech, 2010.

[28] A. Mohamed, T. Sainath, G. Dahl, B. Ramabhadran, G. Hinton, M. Picheny. "Deep belief nets using discriminative features for phone recognition," ICASSP, 2011.

[29] N. Morgan. "Deep and wide: Multiple layers in automatic speech recognition," IEEE Trans. on Audio, Speech, and Language Processing, vol. 20, no. 1, pp. 7-13, 2012.

[30] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Ng, "Multimodal deep learning," ICML, 2011.

[31] M. Pitermann, "Effect of speaking rate and contrastive stress on formant dynamics and vowel perception," J. Acoust. Soc. Am., vol. 107, pp. 3425–3437, 2000.

[32] T. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition", Proc. ASRU, pp. 30-35, 2011.

[33] T. Sainath, D. Nahamoo, D. Kanevsky, B. Ramabhadran, "Enhancing exemplar-based posteriors for speech recognition tasks," Interspeech, 2012.

[34] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," Interspeech, 2011.

[35] G. Tur, L. Deng, D. Hakkani-Tur, and X. He, "Towards deeper understanding: Deep convex networks for semantic utterance classification," ICASSP, 2012.

[36] D. Yu, L. Deng, and G. Dahl, "Roles of pretraining and fine-tuning in context-dependent DNN-HMMs for real-world speech recognition," NIPS Workshop, 2010.

[37] D. Yu, L. Deng, and F. Seide. "The deep tensor neural network with applications to large vocabulary speech recognition," IEEE Trans. Audio, Speech, and Lang. Proc. vol. 21, no. 2, pp. 388-396, Feb, 2013.