
Feature Learning in Deep Neural Networks – Studies on Speech Recognition Tasks

Dong Yu, Michael L. Seltzer, Jinyu Li¹, Jui-Ting Huang¹, Frank Seide²

Microsoft Research, Redmond, WA 98052

¹Microsoft Corporation, Redmond, WA 98052

²Microsoft Research Asia, Beijing, P.R.C.

{dongyu, mseltzer, jinyuli, jthuang, fseide}@microsoft.com

Abstract

Recent studies have shown that deep neural networks (DNNs) perform significantly better than shallow networks and Gaussian mixture models (GMMs) on large vocabulary speech recognition tasks. In this paper, we argue that the improved accuracy achieved by the DNNs is the result of their ability to extract discriminative internal representations that are robust to the many sources of variability in speech signals. We show that these representations become increasingly insensitive to small perturbations in the input with increasing network depth, which leads to better speech recognition performance with deeper networks. We also show that DNNs cannot extrapolate to test samples that are substantially different from the training examples. If the training data are sufficiently representative, however, internal features learned by the DNN are relatively stable with respect to speaker differences, bandwidth differences, and environment distortion. This enables DNN-based recognizers to perform as well or better than state-of-the-art systems based on GMMs or shallow networks without the need for explicit model adaptation or feature normalization.

1 Introduction

Automatic speech recognition (ASR) has been an active research area for more than five decades. However, the performance of ASR systems is still far from satisfactory and the gap between ASR and human speech recognition is still large on most tasks. One of the primary reasons speech recognition is challenging is the high variability in speech signals. For example, speakers may have different accents, dialects, or pronunciations, and speak in different styles, at different rates, and in different emotional states. The presence of environmental noise, reverberation, different microphones and recording devices results in additional variability. To complicate matters, the sources of variability are often nonstationary and interact with the speech signal in a nonlinear way. As a result, it is virtually impossible to avoid some degree of mismatch between the training and testing conditions.

Conventional speech recognizers use a hidden Markov model (HMM) in which each acoustic state is modeled by a Gaussian mixture model (GMM). The model parameters can be discriminatively trained using an objective function such as maximum mutual information (MMI) [1] or minimum phone error rate (MPE) [2]. Such systems are known to be susceptible to performance degradation when even mild mismatch between training and testing conditions is encountered. To combat this, a variety of techniques has been developed. For example, mismatch due to speaker differences can be reduced by Vocal Tract Length Normalization (VTLN) [3], which nonlinearly warps the input feature vectors to better match the acoustic model, or Maximum Likelihood Linear Regression (MLLR) [4], which adapt the GMM parameters to be more representative of the test data. Other techniques such as Vector Taylor Series (VTS) adaptation are designed to address the mismatch caused by environmental noise and channel distortion [5]. While these methods have been

successful to some degree, they add complexity and latency to the decoding process. Most require multiple iterations of decoding and some only perform well with ample adaptation data, making them unsuitable for systems that process short utterances, such as voice search.

Recently, an alternative acoustic model based on deep neural networks (DNNs) has been proposed. In this model, a collection of Gaussian mixture models is replaced by a single context-dependent deep neural network (CD-DNN). A number of research groups have obtained strong results on a variety of large scale speech tasks using this approach [6–13]. Because the temporal structure of the HMM is maintained, we refer to these models as CD-DNN-HMM acoustic models.

In this paper, we analyze the performance of DNNs for speech recognition and in particular, examine their ability to learn representations that are robust to variability in the acoustic signal. To do so, we interpret the DNN as a joint model combining a nonlinear feature transformation and a log-linear classifier. Using this view, we show that the many layers of nonlinear transforms in a DNN convert the raw features into a highly invariant and discriminative representation which can then be effectively classified using a log-linear model. These internal representations become increasingly insensitive to small perturbations in the input with increasing network depth. In addition, the classification accuracy improves with deeper networks, although the gain per layer diminishes. However, we also find that DNNs are unable to extrapolate to test samples that are substantially different from the training samples. A series of experiments demonstrates that if the training data are sufficiently representative, the DNN learns internal features that are relatively invariant to sources of variability common in speech recognition such as speaker differences and environmental distortions. This enables DNN-based speech recognizers to perform as well or better than state-of-the-art GMM-based systems without the need for explicit model adaptation or feature normalization algorithms.

The rest of the paper is organized as follows. In Section 2 we briefly describe DNNs and illustrate the feature learning interpretation of DNNs. In Section 3 we show that DNNs can learn invariant and discriminative features and demonstrate empirically that higher layer features are less sensitive to perturbations of the input. In Section 4 we point out that the feature generalization ability is effective only when test samples are small perturbations of training samples. Otherwise, DNNs perform poorly as indicated in our mixed-bandwidth experiments. We apply this analysis to speaker adaptation in Section 5 and find that deep networks learn speaker-invariant representations, and to the Aurora 4 noise robustness task in Section 6 where we show that a DNN can achieve performance equivalent to the current state of the art without requiring explicit adaptation to the environment. We conclude the paper in Section 7.

2 Deep Neural Networks

A deep neural network (DNN) is conventional multi-layer perceptron (MLP) with many hidden layers (thus deep). If the input and output of the DNN are denoted as x and y , respectively, a DNN can be interpreted as a directed graphical model that approximates the posterior probability $p_{y|x}(y = s|x)$ of a class s given an observation vector x , as a stack of $(L + 1)$ layers of log-linear models. The first L layers model the posterior probabilities of hidden binary vectors h^ℓ given input vectors v^ℓ . If h^ℓ consists of N^ℓ hidden units, each denoted as h_j^ℓ , the posterior probability can be expressed as

$$p^\ell(h^\ell|v^\ell) = \prod_{j=1}^{N^\ell} \frac{e^{z_j^\ell(v^\ell) \cdot h_j^\ell}}{e^{z_j^\ell(v^\ell) \cdot 1} + e^{z_j^\ell(v^\ell) \cdot 0}}, \quad 0 \leq \ell < L$$

where $z^\ell(v^\ell) = (W^\ell)^T v^\ell + a^\ell$, and W^ℓ and a^ℓ represent the weight matrix and bias vector in the ℓ -th layer, respectively. Each observation is propagated forward through the network, starting with the lowest layer ($v^0 = x$). The output variables of each layer become the input variables of the next, i.e. $v^{\ell+1} = h^\ell$. In the final layer, the class posterior probabilities are computed as a multinomial distribution

$$p_{y|x}(y = s|x) = p^L(y = s|v^L) = \frac{e^{z_s^L(v^L)}}{\sum_{s'} e^{z_{s'}^L(v^L)}} = \text{softmax}_s(v^L). \quad (1)$$

Note that the equality between $p_{y|x}(y = s|x)$ and $p^L(y = s|v^L)$ is valid by making a mean-field approximation [14] at each hidden layer.

In the DNN, the estimation of the posterior probability $p_{y|x}(y = s|x)$ can also be considered a two-step deterministic process. In the first step, the observation vector x is transformed to another feature vector v^L through L layers of non-linear transforms. In the second step, the posterior probability $p_{y|x}(y = s|x)$ is estimated using the log-linear model (1) given the transformed feature vector v^L . If we consider the first L layers fixed, learning the parameters in the softmax layer is equivalent to training a conditional maximum-entropy (MaxEnt) model on features v^L . In the conventional MaxEnt model, features are manually designed [15]. In DNNs, however, the feature representations are jointly learned with the MaxEnt model from the data. This not only eliminates the tedious and potentially erroneous process of manual feature extraction but also has the potential to automatically extract invariant and discriminative features, which are difficult to construct manually.

In all the following discussions, we use DNNs in the framework of the CD-DNN-HMM [6–10] and use speech recognition as our classification task. The detailed training procedure and decoding technique for CD-DNN-HMMs can be found in [6–8].

3 Invariant and discriminative features

3.1 Deeper is better

Using DNNs instead of shallow MLPs is a key component to the success of CD-DNN-HMMs. Table 1, which is extracted from [8], summarizes the word error rates (WER) on the Switchboard (SWB) [16] Hub5’00-SWB test set. Switchboard is a corpus of conversational telephone speech. The system was trained using the 309-hour training set with labels generated by Viterbi alignment from a maximum likelihood (ML) trained GMM-HMM system. The labels correspond to tied-parameter context-dependent acoustic states called senones. Our baseline WER with the corresponding discriminatively trained traditional GMM-HMM system is 23.6%, while the best CD-DNN-HMM achieves 17.0%—a 28% relative error reduction (it is possible to further improve the DNN to a one-third reduction by realignment [8]).

We can observe that deeper networks outperform shallow ones. The WER decreases as the number of hidden layers increases, using a fixed layer size of 2048 hidden units. In other words, deeper models have stronger discriminative ability than shallow models. This is also reflected in the improvement of the training criterion (not shown). More interestingly, if architectures with an equivalent number of parameters are compared, the deep models consistently outperform the shallow models when the deep model is sufficiently wide at each layer. This is reflected in the right column of the table, which shows the performance for shallow networks with the same number of parameters as the deep networks in the left column. Even if we further increase the size of an MLP with a single hidden layer to about 16000 hidden units we can only achieve a WER of 22.1%, which is significantly worse than the 17.1% WER that is obtained using a $7 \times 2k$ DNN under the same conditions. Note that as the number of hidden layers further increases, only limited additional gains are obtained and performance saturates after 9 hidden layers. The $9 \times 2k$ DNN performs equally well as a $5 \times 3k$ DNN which has more parameters. In practice, a tradeoff needs to be made between the width of each layer, the additional reduction in WER and the increased cost of training and decoding as the number of hidden layers is increased.

3.2 DNNs learn more invariant features

We have noticed that the biggest benefit of using DNNs over shallow models is that DNNs learn more invariant and discriminative features. This is because many layers of simple nonlinear processing can generate a complicated nonlinear transform. To show that this nonlinear transform is robust to small variations in the input features, let’s assume the output of layer $l - 1$, or equivalently the input to the layer l is changed from v^ℓ to $v^\ell + \delta^\ell$, where δ^ℓ is a small change. This change will cause the output of layer l , or equivalently the input to the layer $l + 1$ to change by

$$\delta^{\ell+1} = \sigma(z^\ell(v^\ell + \delta^\ell)) - \sigma(z^\ell(v^\ell)) \approx \text{diag}(\sigma'(z^\ell(v^\ell))) (w^\ell)^T \delta^\ell.$$

Table 1: Effect of CD-DNN-HMM network depth on WER (%) on Hub5’00-SWB using the 309-hour Switchboard training set. DBN pretraining is applied.

$L \times N$	WER	$1 \times N$	WER
$1 \times 2k$	24.2	–	–
$2 \times 2k$	20.4	–	–
$3 \times 2k$	18.4	–	–
$4 \times 2k$	17.8	–	–
$5 \times 2k$	17.2	1×3772	22.5
$7 \times 2k$	17.1	1×4634	22.6
$9 \times 2k$	17.0	–	–
$5 \times 3k$	17.0	–	–
–	–	$1 \times 16k$	22.1

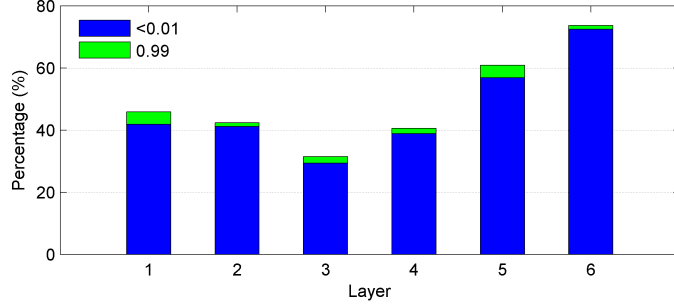


Figure 1: Percentage of saturated activations at each layer

The norm of the change $\delta^{\ell+1}$ is

$$\begin{aligned}
 \|\delta^{\ell+1}\| &\approx \|\text{diag}(\sigma'(z^\ell(v^\ell)))(w^\ell)^T \delta^\ell\| \\
 &\leq \|\text{diag}(\sigma'(z^\ell(v^\ell)))(w^\ell)^T\| \|\delta^\ell\| \\
 &= \|\text{diag}(v^{\ell+1} \circ (1 - v^{\ell+1})) (w^\ell)^T\| \|\delta^\ell\|
 \end{aligned} \tag{2}$$

where \circ refers to an element-wise product.

Note that the magnitude of the majority of the weights is typically very small if the size of the hidden layer is large. For example, in a $6 \times 2k$ DNN trained using 30 hours of SWB data, 98% of the weights in all layers except the input layer have magnitudes less than 0.5.

While each element in $v^{\ell+1} \circ (1 - v^{\ell+1})$ is less than or equal to 0.25, the actual value is typically much smaller. This means that a large percentage of hidden neurons will not be active, as shown in Figure 1. As a result, the average norm $\|\text{diag}(v^{\ell+1} \circ (1 - v^{\ell+1})) (w^\ell)^T\|_2$ in (2) across a 6-hr SWB development set is smaller than one in all layers, as indicated in Figure 2. Since all hidden layer values are bounded in the same range of $(0, 1)$, this indicates that when there is a small perturbation on the input, the perturbation shrinks at each higher hidden layer. In other words, features generated by higher hidden layers are more invariant to variations than those represented by lower layers. Note that the maximum norm over the same development set is larger than one, as seen in Figure 2. This is necessary since the differences need to be enlarged around the class boundaries to have discrimination ability.

4 Learning by seeing

In Section 3, we showed empirically that small perturbations in the input will be gradually shrunk as we move to the internal representation in the higher layers. In this section, we point out that the

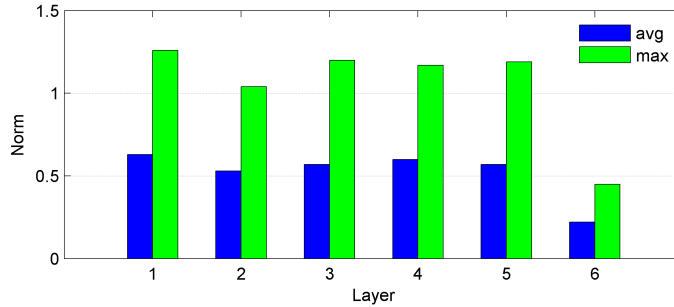


Figure 2: Average and maximum $\|\text{diag}(v^{\ell+1} \circ (1 - v^{\ell+1}))(w^\ell)^T\|_2$ across layers on a $6 \times 2k$ DNN

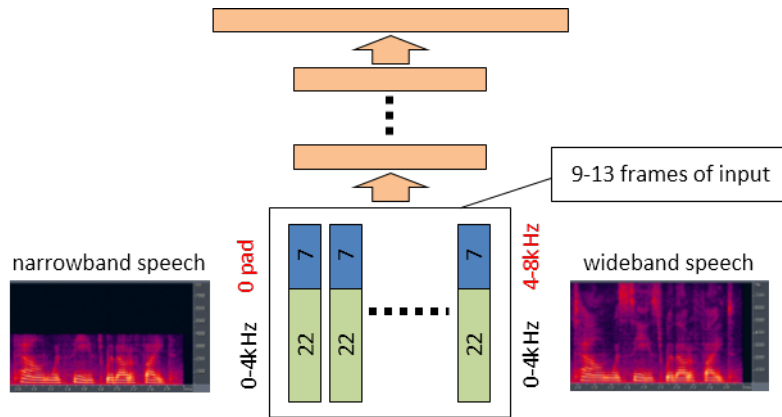


Figure 3: Illustration of mixed-bandwidth speech recognition using a DNN

above result is only applicable to small perturbations around the training samples. When the test samples deviate significantly from the training samples, DNNs cannot accurately classify them. In other words, DNNs must see examples of representative variations in the data during training in order to generalize to similar variations in the test data.

We demonstrate this point using a mixed-bandwidth ASR study. Typical speech recognizers are trained on either narrowband speech signals, recorded at 8 kHz, or wideband speech signals, recorded at 16 kHz. It would be advantageous if a single system could recognize both narrowband and wideband speech, i.e. mixed-bandwidth ASR. One such system was recently proposed using a CD-DNN-HMM [17]. In that work, the following DNN architecture was used for all experiments. The input features were 29 mel-scale log filter-bank outputs together with dynamic features. An 11-frame context window was used generating an input layer with $29 \cdot 3 \cdot 11 = 957$ nodes. The DNN has 7 hidden layers, each with 2048 nodes. The output layer has 1803 nodes, corresponding to the number of senones determined by the GMM system.

The 29-dimensional filter bank has two parts: the first 22 filters span 0–4 kHz and the last 7 filters span 4–8 kHz, with the center frequency of the first filter in the higher filter bank at 4 kHz. When the speech is wideband, all 29 filters have observed values. However, when the speech is narrowband, the high-frequency information was not captured so the final 7 filters are set to 0. Figure 3 illustrates the architecture of the mixed-bandwidth ASR system.

Experiments were conducted on a mobile voice search (VS) corpus. This task consists of internet search queries made by voice on a smartphone. There are two training sets, VS-1 and VS-2, consisting of 72 and 197 hours of wideband audio data, respectively. These sets were collected during

Table 2: WER (%) on wideband (16k) and narrowband (8k) test sets with and without narrowband training data.

training data	16 kHz VS-T	8 kHz VS-T
16 kHz VS-1 + 16 kHz VS-2	27.5	53.5
16 kHz VS-1 + 8 kHz VS-2	28.3	29.3

different times of year. The test set, called VS-T, has 26757 words in 9562 utterances. The narrow band training and test data were obtained by downsampling the wideband data.

Table 2 summarizes the WER on the wideband and narrowband test sets when the DNN is trained with and without narrowband speech. From this table, it is clear that if all training data are wideband, the DNN performs well on the wideband test set (27.5% WER) but very poorly on the narrowband test set (53.5% WER). However, if we convert VS-2 to narrowband speech and train the same DNN using mixed-bandwidth data (second row), the DNN performs very well on both wideband and narrowband speech.

To understand the difference between these two scenarios, we take the output vectors at each layer for the wideband and narrowband input feature pairs, $h^\ell(x_{wb})$ and $h^\ell(x_{nb})$, and measure their Euclidean distance. For the top layer, whose output is the senone posterior probability, we calculate the KL-divergence in nats between $p_{y|x}(s_j|x_{wb})$ and $p_{y|x}(s_j|x_{nb})$. Table 3 shows the statistics of d_l and d_y over 40,000 frames randomly sampled from the test set for the DNN trained using wideband speech only and the DNN trained using mixed-bandwidth speech.

Table 3: Euclidean distance for the output vectors at each hidden layer (L1-L7) and the KL divergence (nats) for the posteriors at the top layer between the narrowband (8 kHz) and wideband (16 kHz) input features, measured using the wideband DNN or the mixed-bandwidth DNN.

		wideband DNN		mixed-band DNN	
layer	dist	mean	variance	mean	variance
L1	Eucl	13.28	3.90	7.32	3.62
L2		10.38	2.47	5.39	1.28
L3		8.04	1.77	4.49	1.27
L4		8.53	2.33	4.74	1.85
L5		9.01	2.96	5.39	2.30
L6		8.46	2.60	4.75	1.57
L7		5.27	1.85	3.12	0.93
Top	KL	2.03	-	0.22	-

From Table 3 we can observe that in both DNNs, the distance between hidden layer vectors generated from the wideband and narrowband input feature pair is significantly reduced at the layers close to the output layer compared to that in the first hidden layer. Perhaps what is more interesting is that the average distances and variances in the data-mixed DNN are consistently smaller than those in the DNN trained on wideband speech only. This indicates that by using mixed-bandwidth training data, the DNN learns to consider the differences in the wideband and narrowband input features as irrelevant variations. These variations are suppressed after many layers of nonlinear transformation. The final representation is thus more invariant to this variation and yet still has the ability to distinguish between different class labels. This behavior is even more obvious at the output layer since the KL-divergence between the paired outputs is only 0.22 in the mixed-bandwidth DNN, much smaller than the 2.03 observed in the wideband DNN.

5 Robustness to speaker variation

A major source of variability is variation across speakers. Techniques for adapting a GMM-HMM to a speaker have been investigated for decades. Two important techniques are VTLN [3], and feature-space MLLR (fMLLR) [4]. Both VTLN and fMLLR operate on the features directly, making their application in the DNN context straightforward.

Table 4: Comparison of feature-transform based speaker-adaptation techniques for GMM-HMMs, a shallow, and a deep NN. Word-error rates in % for Hub5'00-SWB (relative change in parentheses).

adaptation technique	GMM-HMM 40 mix	CD-MLP-HMM 1×2k	CD-DNN-HMM 7×2k
speaker independent	23.6	24.2	17.1
+ VTLN	21.5 (-9%)	22.5 (-7%)	16.8 (-2%)
+ {fMLLR/fDLR}×4	20.4 (-5%)	21.5 (-4%)	16.4 (-2%)

VTLN warps the frequency axis of the filterbank analysis to account for the fact that the precise locations of vocal-tract resonances vary roughly monotonically with the physical size of the speaker. This is done in both training and testing. On the other hand, fMLLR applies an affine transform to the feature frames such that an adaptation data set better matches the model. In most cases, including this work, ‘self-adaptation’ is used: generate labels using unsupervised transcription, then re-recognize with the adapted model. This process is iterated four times. For GMM-HMMs, fMLLR transforms are estimated to maximize the likelihood of the adaptation data given the model. For DNNs, we instead maximize cross entropy (with back propagation), which is a discriminative criterion, so we prefer to call this transform feature-space Discriminative Linear Regression (fDLR). Note that the transform is applied to individual frames, prior to concatenation.

Typically, applying VTLN and fMLLR jointly to a GMM-HMM system will reduce errors by 10–15%. Initially, similar gains were expected for DNNs as well. However, these gains were not realized, as shown in Table 4 [9]. The table compares VTLN and fMLLR/fDLR for GMM-HMMs, a context-dependent ANN-HMM with a single hidden layer, and a deep network with 7 hidden layers, on the same Switchboard task described in Section 3.1. For this task, test data are very consistent with the training, and thus, only a small amount of adaptation to other factors such as recording conditions or environmental factors occurs. We use the same configuration as in Table 1 which is speaker independent using single-pass decoding.

For the GMM-HMM, VTLN achieves a strong relative gain of 9%. VTLN is also effective with the shallow neural-network system, gaining a slightly smaller 7%. However, the improvement of VTLN on the deep network with 7 hidden layers is a much smaller 2% gain. Combining VTLN with fDLR further reduces WER by 5% and 4% relative, for the GMM-HMM and the shallow network, respectively. The reduction for the DNN is only 2%. We also tried transplanting VTLN and fMLLR transforms estimated on the GMM system into the DNN, and achieved very similar results [9].

The VTLN and fDLR implementations of the shallow and deep networks are identical. Thus, we conclude that to a significant degree, the deep neural network is able to learn internal representations that are invariant with respect to the sources of variability that VTLN and fDLR address.

6 Robustness to environmental distortions

In many speech recognition tasks, there are often cases where the despite the presence of variability in the training data, significant mismatch between training and test data persists. Environmental factors are common sources of such mismatch, e.g. ambient noise, reverberation, microphone type and capture device. The analysis in the previous sections suggests that DNNs have the ability to generate internal representations that are robust with respect to variability seen in the training data. In this section, we evaluate the extent to which this invariance can be obtained with respect to distortions caused by the environment.

We performed a series of experiments on the Aurora 4 corpus [18], a 5000-word vocabulary task based on the Wall Street Journal (WSJ0) corpus. The experiments were performed with the 16 kHz multi-condition training set consisting of 7137 utterances from 83 speakers. One half of the utterances was recorded by a high-quality close-talking microphone and the other half was recorded using one of 18 different secondary microphones. Both halves include a combination of clean speech and speech corrupted by one of six different types of noise (street traffic, train station, car, babble, restaurant, airport) at a range of signal-to-noise ratios (SNR) between 10-20 dB.

The evaluation set consists of 330 utterances from 8 speakers. This test set was recorded by the primary microphone and a number of secondary microphones. These two sets are then each corrupted by the same six noises used in the training set at SNRs between 5-15 dB, creating a total of 14 test sets. These 14 test sets can then be grouped into 4 subsets, based on the type of distortion: none (clean speech), additive noise only, channel distortion only, and noise + channel. Notice that the types of noise are common across training and test sets but the SNRs of the data are not.

The DNN was trained using 24-dimensional log mel filterbank features with utterance-level mean normalization. The first- and second-order derivative features were appended to the static feature vectors. The input layer was formed from a context window of 11 frames creating an input layer of 792 input units. The DNN had 7 hidden layers with 2048 hidden units in each layer and the final softmax output layer had 3206 units, corresponding to the senones of the baseline HMM system. The network was initialized using layer-by-layer generative pre-training and then discriminatively trained using back propagation.

In Table 5, the performance obtained by the DNN acoustic model is compared to several other systems. The first system is a baseline GMM-HMM system, while the remaining systems are representative of the state of the art in acoustic modeling and noise and speaker adaptation. All used the same training set. To the authors' knowledge, these are the best published results on this task.

The second system combines Minimum Phone Error (MPE) discriminative training [2] and noise adaptive training (NAT) [19] using VTS adaptation to compensate for noise and channel mismatch [20]. The third system uses a hybrid generative/discriminative classifier [21] as follows. First, an adaptively trained HMM with VTS adaptation is used to generate features based on state likelihoods and their derivatives. Then, these features are input to a discriminative log-linear model to obtain the final hypothesis. The fourth system uses an HMM trained with NAT and combines VTS adaptation for environment compensation and MLLR for speaker adaptation [22]. Finally, the last row of the table shows the performance of the DNN system.

Table 5: A comparison of several systems in the literature to a DNN system on the Aurora 4 task.

Systems	distortion				AVG
	none (clean)	noise	channel	noise + channel	
GMM baseline	14.3	17.9	20.2	31.3	23.6
MPE-NAT + VTS [20]	7.2	12.8	11.5	19.7	15.3
NAT + Derivative Kernels [21]	7.4	12.6	10.7	19.0	14.8
NAT + Joint MLLR/VTS [22]	5.6	11.0	8.8	17.8	13.4
DNN (7×2048)	5.6	8.8	8.9	20.0	13.4

It is noteworthy that to obtain good performance, the GMM-based systems required complicated adaptive training procedures [19, 23] and multiple iterations of recognition in order to perform explicit environment and/or speaker adaptation. One of these systems required two classifiers. In contrast, the DNN system required only standard training and a single forward pass for classification. Yet, it outperforms the two systems that perform environment adaptation and matches the performance of a system that adapts to both the environment and speaker.

Finally, we recall the results in Section 4, in which the DNN trained only on wideband data could not accurately classify narrowband speech. Similarly, a DNN trained only on clean speech has no ability to learn internal features that are robust to environmental noise. When the DNN for Aurora 4 is trained using only clean speech examples, the performance on the noise- and channel-distorted speech degrades substantially, resulting in an average WER of 30.6%. This further confirms our earlier observation that DNNs are robust to modest variations between training and test data but perform poorly if the mismatch is more severe.

7 Conclusion

In this paper we demonstrated through speech recognition experiments that DNNs can extract more invariant and discriminative features at the higher layers. In other words, the features learned by

DNNs are less sensitive to small perturbations in the input features. This property enables DNNs to generalize better than shallow networks and enables CD-DNN-HMMs to perform speech recognition in a manner that is more robust to mismatches in speaker, environment, or bandwidth. On the other hand, DNNs cannot learn something from nothing. They require seeing representative samples to perform well. By using a multi-style training strategy and letting DNNs to generalize to similar patterns, we equaled the best result ever reported on the Aurora 4 noise robustness benchmark task without the need for multiple recognition passes and model adaptation.

References

- [1] L. Bahl, P. Brown, P.V. De Souza, and R. Mercer, “Maximum mutual information estimation of hidden markov model parameters for speech recognition,” in *Proc. ICASSP*, Apr, vol. 11, pp. 49–52.
- [2] D. Povey and P. C. Woodland, “Minimum phone error and i-smoothing for improved discriminative training,” in *Proc. ICASSP*, 2002.
- [3] P. Zhan et al., “Vocal tract length normalization for lvcsr,” Tech. Rep. CMU-LTI-97-150, Carnegie Mellon Univ, 1997.
- [4] M. J. F. Gales, “Maximum likelihood linear transformations for HMM-based speech recognition,” *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.
- [5] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, “HMM Adaptation Using Vector Taylor Series for Noisy Speech Recognition,” in *Proc. of ICSLP*, 2000.
- [6] D. Yu, L. Deng, and G. Dahl, “Roles of pretraining and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition,” in *Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.
- [7] G.E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pretrained deep neural networks for large vocabulary speech recognition,” *IEEE Trans. Audio, Speech, and Lang. Proc.*, vol. 20, no. 1, pp. 33–42, Jan. 2012.
- [8] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using context-dependent deep neural networks,” in *Proc. Interspeech*, 2011.
- [9] F. Seide, G. Li, X. Chen, and D. Yu, “Feature engineering in context-dependent deep neural networks for conversational speech transcription,” in *Proc. ASRU*, 2011, pp. 24–29.
- [10] D. Yu, F. Seide, G. Li, and L. Deng, “Exploiting sparseness in deep neural networks for large vocabulary speech recognition,” in *Proc. ICASSP*, 2012, pp. 4409–4412.
- [11] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, “An application of pretrained deep neural networks to large vocabulary conversational speech recognition,” Tech. Rep. Tech. Rep. 001, Department of Computer Science, University of Toronto, 2012.
- [12] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A. r. Mohamed, “Making deep belief networks effective for large vocabulary continuous speech recognition,” in *Proc. ASRU*, 2011, pp. 30–35.
- [13] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Large vocabulary continuous speech recognition with context-dependent dbn-hmms,” in *Proc. ICASSP*, 2011, pp. 4688–4691.
- [14] L. Saul, T. Jaakkola, and M. I. Jordan, “Mean field theory for sigmoid belief networks,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 61–76, 1996.
- [15] D. Yu, L. Deng, and A. Acero, “Using continuous features in the maximum entropy model,” *Pattern Recognition Letters*, vol. 30, no. 14, pp. 1295–1300, 2009.
- [16] J. Godfrey and E. Holliman, *Switchboard-1 Release 2*, Linguistic Data Consortium, Philadelphia, PA, 1997.
- [17] J. Li, D. Yu, J.-T. Huang, and Y. Gong, “Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM,” in *Proc. SLT*, 2012.
- [18] N. Parihar and J. Picone, “Aurora working group: DSR front end LVCSR evaluation AU/384/02,” Tech. Rep., Inst. for Signal and Information Process, Mississippi State University.
- [19] O. Kalinli, M. L. Seltzer, J. Droppo, and A. Acero, “Noise adaptive training for robust automatic speech recognition,” *IEEE Trans. on Audio, Sp. and Lang. Proc.*, vol. 18, no. 8, pp. 1889–1901, Nov. 2010.
- [20] F. Flego and M. J. F. Gales, “Discriminative adaptive training with VTS and JUD,” in *Proc. ASRU*, 2009.
- [21] A. Ragni and M. J. F. Gales, “Derivative kernels for noise robust ASR,” in *Proc. ASRU*, 2011.
- [22] Y.-Q. Wang and M. J. F. Gales, “Speaker and noise factorisation for robust speech recognition,” *IEEE Trans. on Audio Speech and Language Proc.*, vol. 20, no. 7, 2012.
- [23] H. Liao and M. J. F. Gales, “Adaptive training with joint uncertainty decoding for robust recognition of noisy data,” in *Proc. of ICASSP*, Honolulu, Hawaii, 2007.