7

Selected Applications in Speech and Audio Processing

7.1 Acoustic modeling for speech recognition

As discussed in Section 2, speech recognition is the very first successful application of deep learning methods at an industry scale. This success is a result of close academic-industrial collaboration, initiated at Microsoft Research, with the involved researchers identifying and acutely attending to the industrial need for large-scale deployment [68, 89, 109, 161, 323, 414]. It is also a result of carefully exploiting the strengths of the deep learning and the then-state-of-the-art speech recognition technology, including notably the highly efficient decoding techniques.

Speech recognition has long been dominated by the GMM–HMM method, with an underlying shallow or flat generative model of context-dependent GMMs and HMMs (e.g., [92, 93, 187, 293]). Neural networks once were a popular approach but had not been competitive with the GMM–HMM [42, 87, 261, 382]. Generative models with deep hidden dynamics likewise have also not been clearly competitive (e.g., [45, 73, 108, 282]).

Deep learning and the DNN started making their impact in speech recognition in 2010, after close collaborations between academic and

industrial researchers; see reviews in [89, 161]. The collaborative work started in phone recognition tasks [89, 100, 135, 136, 257, 260, 258, 309, 311, 334], demonstrating the power of hybrid DNN architectures discussed in Section 5 and of subsequent new architectures with convolutional and recurrent structure. The work also showed the importance of raw speech features of spectrogram — back from the long-popular MFCC features toward but not yet reaching the raw speech-waveform level [183, 327]. The collaboration continued to large vocabulary tasks with more convincing, highly positive results [67, 68, 94, 89, 161, 199, 195, 223, 323, 353, 399, 414]. The success in large vocabulary speech recognition is in large part attributed to the use of a very large DNN output layer structured in the same way as the GMM–HMM speech units (senones), motivated partially by the speech researchers' desires to take advantage of the context-dependent phone modeling techniques that have been proven to work well in the GMM–HMM framework, and to keep the change of the already highly efficient decoder software's infrastructure developed for the GMM–HMM systems to a minimum. In the meantime, this body of work also demonstrated the possibility to reduce the need for the DBN-like pre-training in effective learning of DNNs when a large amount of labeled data is available. A combination of three factors helped to quickly spread the success of deep learning in speech recognition to the entire speech industry and academia: (1) significantly lowered errors compared with the then-state-of-the-art GMM-HMM systems; (2) minimal decoder changes required to deploy the new DNN-based speech recognizer due to the use of senones as the DNN output; and (3) reduced system complexity empowered by the DNN's strong modeling power. By the ICASSP-2013 timeframe, at least 15 major speech recognition groups worldwide confirmed experimentally the success of DNNs with very large tasks and with the use of raw speech spectral features other than MFCCs. The most notable groups include major industrial speech labs worldwide: Microsoft [49, 89, 94, 324, 399, 430], IBM [195, 309, 311, 307, 317], Google [69, 150, 184, 223], iFlyTek, and Baidu. Their results represent a new state-of-the-art in speech recognition widely deployed in these companies' voice products and services with extensive media coverage in recent years.

In the remainder of this chapter, we review a wide range of speech recognition work based on deep learning methods according to several major themes expressed in the section titles.

7.1.1 Back to primitive spectral features of speech

Deep learning, also referred as representation learning or (unsupervised) feature learning, sets an important goal of automatic discovery of powerful features from raw input data independent of application domains. For speech feature learning and for speech recognition, this goal is condensed to the use of primitive spectral or possibly waveform features. Over the past 30 years or so, largely "hand-crafted" transformations of speech spectrogram have led to significant accuracy improvements in the GMM-based HMM systems, despite the known loss of information from the raw speech data. The most successful transformation is the non-adaptive cosine transform, which gave rise to Mel-frequency cepstral coefficients (MFCC) features. The cosine transform approximately de-correlates feature components, which is important for the use of GMMs with diagonal covariance matrices. However, when GMMs are replaced by deep learning models such as DNNs, deep belief nets (DBNs), or deep autoencoders, such de-correlation becomes irrelevant due to the very strength of the deep learning methods in modeling data correlation. As discussed in detail in Section 4, early work of [100] demonstrated this strength and in particular the benefit of spectrograms over MFCCs in effective coding of bottleneck speech features using autoencoders in an unsupervised manner.

The pipeline from speech waveforms (raw speech features) to MFCCs and their temporal differences goes through intermediate stages of log-spectra and then (Mel-warped) filter-banks, with learned parameters based on the data. An important character of deep learning is to move away from separate design of feature representations and of classifiers. This idea of jointly learning classifier and feature transformation for speech recognition was already explored in early studies on the GMM–HMM based systems; e.g., [33, 50, 51, 299]. However, greater speech recognition performance gain is obtained only recently

in the recognizers empowered by deep learning methods. For example, Mohamed et al. [259], Li et al. [221], and Deng et al. [94] showed significantly lowered speech recognition errors using large-scale DNNs when moving from the MFCC features back to more primitive (Mel-scaled) filter-bank features. These results indicate that DNNs can learn a better transformation than the original fixed cosine transform from the Mel-scaled filter-bank features.

Compared with MFCCs, "raw" spectral features not only retain more information, but also enable the use of convolution and pooling operations to represent and handle some typical speech variability — e.g., vocal tract length differences across speakers, distinct speaking styles causing formant undershoot or overshoot, etc. — expressed explicitly in the frequency domain. For example, the convolutional neural network (CNN) can only be meaningfully and effectively applied to speech recognition [1, 2, 3, 94] when spectral features, instead of MFCC features, are used.

More recently, Sainath et al. [307] went one step further toward raw features by learning the parameters that define the filter-banks on power spectra. That is, rather than using Mel-warped filter-bank features as the input features as in [1, 3, 50, 221], the weights corresponding to the Mel-scale filters are only used to initialize the parameters, which are subsequently learned together with the rest of the deep network as the classifier. The overall architecture of the jointly learned feature generator and classifier is shown in Figure 7.1. Substantial speech recognition error reduction is reported in [307].

It has been shown that not only learning the spectral aspect of the features are beneficial for speech recognition, learning the temporal aspect of the features is also helpful [332]. Further, Yu et al. [426] carefully analyzed the properties of different layers in the DNN as the layer-wise extracted features starting from the lower raw filter-bank features. They found that the improved speech recognition accuracy achieved by the DNNs partially attributes to DNN's ability to extract discriminative internal representations that are robust to the many sources of variability in speech signals. They also show that these representations become increasingly insensitive to small perturbations in

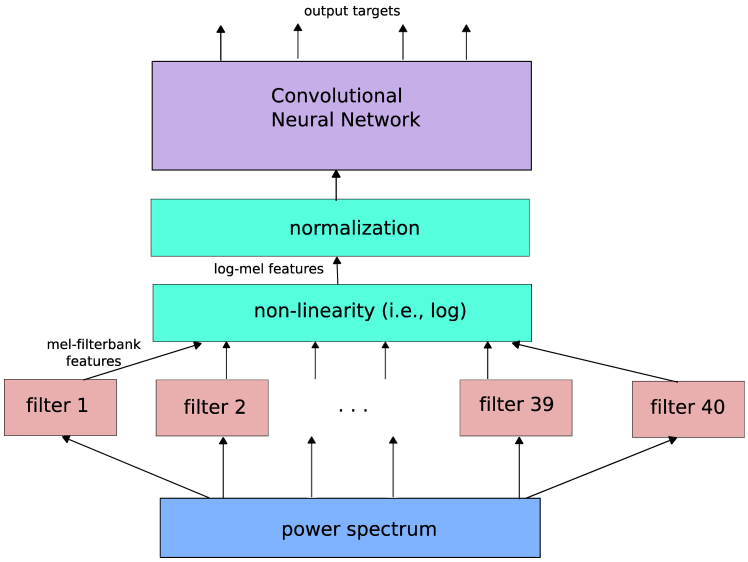


Figure 7.1: Illustration of the joint learning of filter parameters and the rest of the deep network. [after [307], @IEEE].

the input at higher layers, which helps to achieve better speech recognition accuracy.

To the extreme end, deep learning would promote to use the lowest level of raw features of speech, i.e., speech sound waveforms, for speech recognition, and learn the transformation automatically. As an initial attempt toward this goal the study carried out by Jaitly and Hinton [183] makes use of speech sound waves as the raw input feature to an RBM with a convolutional structure as the classifier. With the use of rectified linear units in the hidden layer [130], it is possible, to a limited extent, to automatically normalize the amplitude variation in the waveform signal. Although the final results are disappointing, the work shows that much work is needed along this direction. For example, just as demonstrated by Sainath et al. [307] that the use of raw spectra as features requires additional attention in normalization than MFCCs, the use of speech waveforms demands even more attention in normalization [327]. This is true for both GMM-based and deep learning based methods.

7.1.2 The DNN–HMM architecture versus use of DNN-derived features

Another major theme in the recent studies reported in the literature on applying deep learning methods to speech recognition is two disparate ways of using the DNN: (1) Direct applications of the DNN-HMM architecture as discussed in Section 5.3 to perform speech recognition; and (2) The use of DNNs to extract or derive features, which are then fed into a separate sequence classifier. In the speech recognition literature [42], a system, in which a neural network's output is directly used to estimate the emission probabilities of an HMM, is often called an ANN/HMM hybrid system. This should be distinguished from the use of "hybrid" in Section 5 and throughout this monograph, where a hybrid of unsupervised pre-training and of supervised fine tuning is exploited to learn the parameters of DNNs.

7.1.2.1 The DNN–HMM architecture as a recognizer

An early DNN–HMM architecture [257] was presented at the NIPS Workshop [109], developed, analyzed, and assisted by University of Toronto and MSR speech researchers. In this work, a five-layer DNN (called the DBN in the paper) was used to replace the Gaussian mixture models in the GMM–HMM system, and the monophone state was used as the modeling unit. Although monophones are generally accepted as a weaker phonetic representation than triphones, the DNN–HMM approach with monophones was shown to achieve higher phone recognition accuracy than the state-of-the-art triphone GMM–HMM systems. Further, the DNN results were found to be slightly superior to the then-best-performing single system based on the generative hidden trajectory model (HTM) in the literature [105, 108] evaluated on the same, commonly used TIMIT task by many speech researchers [107, 108, 274, 313]. At MSR, Redmond, the error patterns produced by these two separate systems (the DNN vs. the HTM) were carefully analyzed and found to be very different, reflecting distinct core capabilities of the two approaches and igniting intensive further studies on the DNN–HMM approach described below.

MSR and University of Toronto researchers [67, 68, 414] extended the DNN–HMM system from the monophone phonetic representation of the DNN outputs to the triphone or context-dependent counterpart and from phone recognition to large vocabulary speech recognition. Experiments conducted at MSR on the 24-hour and 48-hour Bing mobile voice search datasets collected under the real usage scenario demonstrate that the context-dependent DNN–HMM significantly outperforms the state-of-the-art GMM-HMM system. Three factors, in addition to the use of the DNN, contribute to the success: the use of tied triphones as the DNN modeling units, the use of the best available tri-phone GMM–HMM to generate the tri-phone state alignment, and the effective exploitation of a long window of input features. Experiments also indicate that the decoding time of a five-layer DNN–HMM is almost the same as that of the state-of-the-art triphone GMM–HMM.

The success was quickly extended to large vocabulary speech recognition tasks with hundreds and even thousands of hours of training set and with thousands of tri-phone states, including the Switchboard and Broadcast News databases, and Google's voice search and YouTube tasks [94, 161, 184, 309, 311, 324]. For example, on the Switchboard benchmark, the context-dependent DNN–HMM (CD-DNN–HMM) is shown to cut error by one third compared to the state-of-the-art GMM–HMM system [323]. As a summary, we show in Table 7.1 some quantitative recognition error rates in relatively early literature produced by the basic DNN–HMM architecture in comparison with those by the previous state-of-the-art systems based on the generative models. (More advanced architectures have produced better results than shown here). Note from sub-tables A to D, the training data are increased approximately one order of magnitude from one task to the next. Not only the computation scales up well (i.e., almost linearly) with the training size, but most importantly the relative error rate reduction increases substantially with increasing amounts of training data — from approximately 10% to 20%, and then to 30%. This set of results highlight the strongly desirable properties of the DNN-based methods, despite the conceptual simplicity of the overall DNN–HMM architecture and some known weaknesses.

Table 7.1: Comparisons of the DNN–HMM architecture with the generative model (e.g., the GMM–HMM) in terms of phone or word recognition error rates. From sub-tables A to D, the training data are increased approximately three orders of magnitudes.

Features	Setup	Error Rates
A: TIMIT Phone recognition (3 hours of training)		
GMM	w. Hidden dynamics	24.8%
DNN	5 layers × 2048	23.0%
B: Voice Search SER (24–48 hours of training)		
GMM	MPE (760 24-mix)	36.2%
DNN	5 layers × 2048	30.1%
C: Switch Board WER (309 hours of training)		
GMM	BMMI (9K 40-mix)	23.6%
DNN	7 layers × 2048	15.8%
D: Switch Board WER (2000 hours of training)		
GMM	BMMI (18K 72-mix)	21.7%
DNN	7 layers × 2048	14.6%

7.1.2.2 The use of DNN-derived features in a separate recognizer

One clear weakness of the above DNN–HMM architecture for speech recognition is that much of the highly effective techniques for the GMM–HMM systems, including discriminative training (in both feature space and model space), unsupervised speaker adaptation, noise robustness, and scalable batch training tools for big training data, developed over the past 20 some years may not be directly applicable to the new systems although similar techniques have been recently developed for DNN–HMMs. To remedy this problem, the "tandem" approach, developed originally by Hermansky et al. [154], has been adopted, where the output of the neural networks in the form of posterior probabilities for phone classes, are used, often in conjunction with the acoustic features to form new augmented input features, in a separate GMM–HMM system.

This tandem approach is used by Vinyals and Ravuri [379] where a DNN's outputs are extracted to serve as the features for mismatched noisy speech. It is reported that DNNs outperform the neural networks with a single hidden layer under the clean condition, but the gains slowly diminish as the noise level is increased. Furthermore, using MFCCs in conjunction with the posteriors computed from DNNs outperforms using the DNN features alone in low to moderate noise conditions with the tandem architecture. Comparisons of such tandem approach with the direct DNN–HMM approach are made by Tüske et al. [368] and Imseng et al. [182].

An alternative way of extracting the DNN features is to use the "bottleneck" layer, which is narrower than other layers in the DNN, to restrict the capacity of the network. Then, such bottleneck features are fed to a GMM–HMM system, often in conjunction with the original acoustic features and some dimensionality reduction techniques. The bottleneck features derived from the DNN are believed to capture information complementary to conventional acoustic features derived from the short-time spectra of the input. A speech recognizer based on the above bottleneck feature approach is built by Yu and Seltzer [425], with the overall architecture shown in Figure 7.2. Several variants of the DNN-based bottleneck-feature approach have been explored; see details in [16, 137, 201, 285, 308, 368].

Yet another method to derive the features from the DNN is to feed its top-most hidden layer as the new features for a separate speech

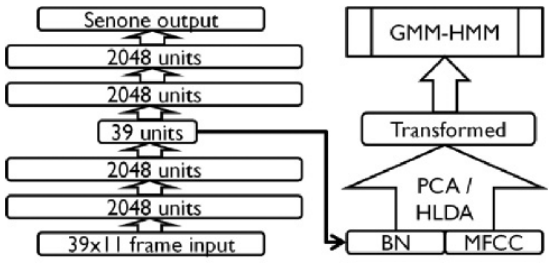


Figure 7.2: Illustration of the use of bottleneck (BN) features extracted from a DNN in a GMM–HMM speech recognizer. [after [425], @IEEE].

recognizer. In [399], a GMM–HMM is used as such a recognizer, and the high-dimensional, DNN-derived features are subject to dimensionality reduction before feeding them into the recognizer. More recently, a recurrent neural network (RNN) is used as the "backend" recognizer receiving the high-dimensional, DNN-derived features as the input without dimensionality reduction [48, 85]. These studies also show that the use of the top-most hidden layer of the DNN as features is better than other hidden layers and also better than the output layer in terms of recognition accuracy for the RNN sequence classifier.

7.1.3 Noise robustness by deep learning

The study of noise robustness in speech recognition has a long history, mostly before the recent rise of deep learning. One major contributing factor to the often observed brittleness of speech recognition technology is the inability of the standard GMM–HMM-based acoustic model to accurately model noise-distorted speech test data that differs in character from the training data, which may or may not be distorted by noise. A wide range of noise-robust techniques developed over past 30 years can be analyzed and categorized using five different criteria: (1) feature-domain versus model-domain processing, (2) the use of prior knowledge about the acoustic environment distortion, (3) the use of explicit environment-distortion models, (4) deterministic versus uncertainty processing, and (5) the use of acoustic models trained jointly with the same feature enhancement or model adaptation process used in the testing stage. See a comprehensive review in [220] and some additional review literature or original work in [4, 82, 119, 140, 230, 370, 404, 431, 444].

Many of the model-domain techniques developed for GMM–HMMs (e.g., model-domain noise robustness techniques surveyed by Li et al. [220] and Gales [119]) are not directly applicable to the new deep learning models for speech recognition. The feature-domain techniques, however, can be directly applied to the DNN system. A detailed investigation of the use of DNNs for noise robust speech recognition in the feature domain was reported by Seltzer et al. [325], who applied the C-MMSE [415] feature enhancement algorithm on the input feature

used in the DNN. By processing both the training and testing data with the same algorithm, any consistent errors or artifacts introduced by the enhancement algorithm can be learned by the DNN–HMM recognizer. This study also successfully explored the use of the noise aware training paradigm for training the DNN, where each observation was augmented with an estimate of the noise. Strong results were obtained on the Aurora4 task. More recently, Kashiwagi et al. [191] applied the SPLICE feature enhancement technique [82] to a DNN speech recognizer. In that study the DNN's output layer was determined on clean data instead of on noisy data as in the study reported by Seltzer et al. [325].

Besides DNN, other deep architectures have also been proposed to perform feature enhancement and noise-robust speech recognition. For example, Mass et al. [235] applied a deep recurrent auto encoder neural network to remove noise in the input features for robust speech recognition. The model was trained on stereo (noisy and clean) speech features to predict clean features given noisy input, similar to the SPLICE setup but using a deep model instead of a GMM. Vinyals and Ravuri [379] investigated the tandem approaches to noise-robust speech recognition, where DNNs were trained directly with noisy speech to generate posterior features. Finally, Rennie et al. [300] explored the use of a version of the RBM, called the factorial hidden RBM, for noise-robust speech recognition.

7.1.4 Output representations in the DNN

Most deep learning methods for speech recognition and other information processing applications have focused on learning representations from input acoustic features without paying attention to output representations. The recent 2013 NIPS Workshop on Learning Output Representations (http://nips.cc/Conferences/2013/Program/event.php?ID=3714) was dedicated to bridging this gap. For example, the Deep Visual-Semantic Embedding Model described in [117], to be discussed more in Section 11) exploits continuous-valued output representations obtained from the text embeddings to assist in the

branch of the deep network for classifying images. For speech recognition, the importance of designing effective linguistic representations for the output layers of deep networks is highlighted in [79].

Most current DNN systems use a high-dimensional output representation to match the context-dependent phonetic states in the HMMs. For this reason, the output layer evaluation can cost 1/3 of the total computation time. To improve the decoding speed, techniques such as low-rank approximation is typically applied to the output layer. In [310] and [397], the DNN with high-dimensional output layer was trained first. The singular value decomposition (SVD)-based dimension reduction technique was then performed on the large output-layer matrix. The resulting matrices are further combined and as the result the original large weight matrix is approximated by a product of two much smaller matrices. This technique in essence converts the original large output layer to two layers — a bottleneck linear layer and a nonlinear output layer — both with smaller weight matrices. The converted DNN with reduced dimensionality is further refined. The experimental results show that no speech recognition accuracy reduction was observed even when the size is cut to half, while the run-time computation is significantly reduced.

The output representations for speech recognition can benefit from the structured design of the symbolic or phonological units of speech as presented in [79]. The rich phonological structure of symbolic nature in human speech has been well known for many years. Likewise, it has also been well understood for a long time that the use of phonetic or its finer state sequences, even with contextual dependency, in engineering speech recognition systems, is inadequate in representing such rich structure [86, 273, 355], and thus leaving a promising open direction to improve the speech recognition systems' performance. Basic theories about the internal structure of speech sounds and their relevance to speech recognition technology in terms of the specification, design, and learning of possible output representations of the underlying speech model for speech target sequences are surveyed in [76] and more recently in [79].

There has been a growing body of deep learning work in speech recognition with their focus placed on designing output representations

related to linguistic structure. In [383, 384], a limitation of the output representation design, based on the context-dependent phone units as proposed in [67, 68], is recognized and a solution is offered. The root cause of this limitation is that all context-dependent phone states within a cluster created by the decision tree share the same set of parameters and this reduces its resolution power for fine-grained states during the decoding phase. The solution proposed formulates output representations of the context-dependent DNN as an instance of the canonical state modeling technique, making use of broad phonetic classes. First, triphones are clustered into multiple sets of shorter bi-phones using broad phone contexts. Then, the DNN is trained to discriminate the bi-phones within each set. Logistic regression is used to transform the canonical states into the detailed triphone state output probabilities. That is, the overall design of the output representation of the context-dependent DNN is hierarchical in nature, solving both the data sparseness and low-resolution problems at the same time.

Related work on designing the output linguistic representations for speech recognition can be found in [197] and in [241]. While the designs are in the context of GMM–HMM-based speech recognition systems, they both can be extended to deep learning models.

7.1.5 Adaptation of the DNN-based speech recognizers

The DNN–HMM is an advanced version of the artificial neural network and HMM "hybrid" system developed in 1990s, for which several adaptation techniques have been developed. Most of these techniques are based on linear transformation of the network weights of either input or output layers. A number of exploratory studies on DNN adaptation made use of the same or related linear transformation methods [223, 401, 402]. However, compared with the earlier narrower and shallower neural network systems, the DNN–HMM has significantly more parameters due to wider and deeper hidden layers used and the much larger output layer designed to model context dependent phones and states. This difference casts special challenges to adapting the DNN–HMM, especially when the adaptation data is small. Here we discuss

representative recent studies on overcoming such challenges in adapting the large-sized DNN weights in distinct ways.

Yu et al. [430] proposed a regularized adaptation technique for DNNs. It adapts the DNN weights conservatively by forcing the distribution estimated from the adapted model to be close to that estimated from those before the adaptation. This constraint is realized by adding Kullback–Leibler divergence (KLD) regularization to the adaptation criterion. This type of regularization is shown to be equivalent to a modification of the target distribution in the conventional backpropagation algorithm and thus the training of the DNN remains largely unchanged. The new target distribution is derived to be a linear interpolation of the distribution estimated from the model before adaptation and the ground truth alignment of the adaptation data. This interpolation prevents overtraining by keeping the adapted model from straying too far from the speaker-independent model. This type of adaptation differs from L2 regularization, which constrains the model parameters themselves rather than the output probabilities.

In [330], adaptation of the DNN was applied not on the conventional network weights but on the hidden activation functions. In this way, the main limitation of current adaptation techniques based on adaptable linear transformation of the network weights in either the input or the output layer is effectively overcome, since the new method only needs to adapt a more limited number of hidden activation functions.

Several studies were carried out on unsupervised or semi-supervised adaptation of DNN acoustic models with different types of input features with success [223, 405].

Most recently, Saon et al. [317] explored a new and highly effective method in adapting DNNs for speech recognition. The method combined I-vector features with fMLLR (feature-domain max-likelihood linear regression) features as the input into a DNN. I-vectors or (speaker) identity vectors are commonly used for speaker verification and speaker recognition applications, as they encapsulate relevant information about a speaker's identity in a low-dimensional feature vector. The fMLLR is an effective adaptation technique developed for GMM–HMM systems. Since I-vectors do not obey locality in frequency, they must be combined carefully with the fMLLR features that obey

locality. The architecture of the multi-scale CNN–DNN was shown to be effective for the combination of these two different types of features. During both training and decoding, the speaker-specific I-vector was appended to the frame-based fMLLR features.

7.1.6 Better architectures and nonlinear units

Over recent years, since the success of the (fully-connected) DNN–HMM hybrid system was demonstrated in [67, 68, 109, 161, 257, 258, 308, 309, 324, 429], many new architectures and nonlinear units have been proposed and evaluated for speech recognition. Here we provide an overview of this progress, extending the overview provided in [89].

The tensor version of the DNN is reported by Yu et al. [421, 422], which extends the conventional DNN by replacing one or more of its layers with a double-projection layer and a tensor layer. In the double-projection layer, each input vector is projected into two nonlinear subspaces. In the tensor layer, two subspace projections interact with each other and jointly predict the next layer in the overall deep architecture. An approach is developed to map the tensor layers to the conventional sigmoid layers so that the former can be treated and trained in a similar way to the latter. With this mapping the tensor version of the DNN can be treated as the DNN augmented with double-projection layers so that the backpropagation learning algorithm can be cleanly derived and relatively easily implemented.

A related architecture to the above is the tensor version of the DSN described in Section 6, also usefully applied to speech classification and recognition [180, 181]. The same approach applies to mapping the tensor layers (i.e., the upper layer in each of the many modules in the DSN context) to the conventional sigmoid layers. Again, this mapping simplifies the training algorithm so that it becomes not so far apart from that for the DSN.

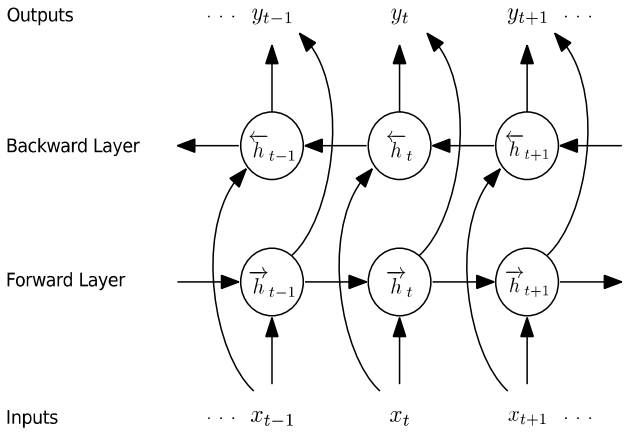
As discussed in Section 3.2, the concept of convolution in time was originated in the TDNN (time-delay neural network) as a shallow neural network [202, 382] developed during early days of speech recognition. Only recently and when deep architectures (e.g. deep convolutional neural network or deep CNN) were used, it has been

found that frequency-dimension weight sharing is more effective for high-performance phone recognition, when the HMM is used to handle the time variability, than time-domain weight sharing as in the previous TDNN in which the HMM was not used [1, 2, 3, 81]. These studies also show that designing the pooling scheme in the deep CNN to properly trade-off between invariance to vocal tract length and discrimination among speech sounds, together with a regularization technique of "dropout" [166], leads to even better phone recognition performance. This set of work further points to the direction of trading-off between trajectory discrimination and invariance expressed in the whole dynamic pattern of speech defined in mixed time and frequency domains using convolution and pooling. Moreover, the most recent studies reported in [306, 307, 312] show that CNNs also benefit large vocabulary continuous speech recognition. They further demonstrate that multiple convolutional layers provide even more improvement when the convolutional layers use a large number of convolution kernels or feature maps. In particular, Sainath et al. [306] extensively explored many variants of the deep CNN. In combination with several novel methods the deep CNN is shown to produce state of the art results in a few large vocabulary speech recognition tasks.

In addition to the DNN, CNN, and DSN, as well as their tensor versions, other deep models have also been developed and reported in the literature for speech recognition. For example, the deep-structured CRF, which stacks many layers of CRFs, have been usefully applied to the task of language identification [429], phone recognition [410], sequential labeling in natural language processing [428], and confidence calibration in speech recognition [423]. More recently, Demuynck and Triefenbach [70] developed the deep GMM architecture, where the aspects of DNNs that lead to strong performance are extracted and applied to build hierarchical GMMs. They show that by going "deep and wide" and feeding windowed probabilities of a lower layer of GMMs to a higher layer of GMMs, the performance of the deep-GMM system can be made comparable to a DNN. One advantage of staying in the GMM space is that the decades of work in GMM adaptation and discriminative learning remains applicable.

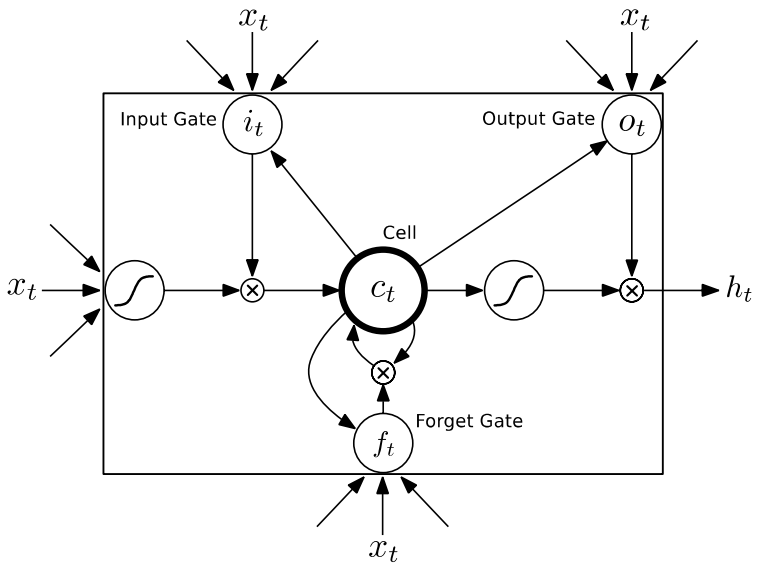
Perhaps the most notable deep architecture among all is the recurrent neural network (RNN) as well as its stacked or deep versions [135, 136, 153, 279, 377]. While the RNN saw its early success in phone recognition [304], it was not easy to duplicate due to the intricacy in training, let alone to scale up for larger speech recognition tasks. Learning algorithms for the RNN have been dramatically improved since then, and much better results have been obtained recently using the RNN [48, 134, 235], especially when the bi-directional LSTM (long short-term memory) is used [135, 136]. The basic information flow in the bi-directional RNN and a cell of LSTM is shown in Figures 7.3 and 7.4, respectively.

Learning the RNN parameters is known to be difficult due to vanishing or exploding gradients [280]. Chen and Deng [48] and Deng and



$$\overrightarrow{h}_t = \mathcal{H}\left(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}\right)$$
$$\overleftarrow{h}_t = \mathcal{H}\left(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right)$$
$$y_t = W_{\overrightarrow{h}y}\overrightarrow{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y$$

Figure 7.3: Information flow in the bi-directional RNN, with both diagrammatic and mathematical descriptions. W's are weight matrices, not shown but can be easily inferred in the diagram. [after [136], @IEEE].



$$
\begin{aligned}
i_t &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right) \\
f_t &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \\
c_t &= f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \\
o_t &= \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right) \\
h_t &= o_t \tanh(c_t)
\end{aligned}
$$

Figure 7.4: Information flow in an LSTM unit of the RNN, with both diagrammatic and mathematical descriptions. W's are weight matrices, not shown but can easily be inferred in the diagram. [after [136], @IEEE].

Chen [85] developed a primal-dual training method that formulates the learning of the RNN as a formal optimization problem, where cross entropy is maximized subject to the condition that the infinity norm of the recurrent matrix of the RNN is less than a fixed value to guarantee the stability of RNN dynamics. Experimental results on phone recognition demonstrate: (1) the primal-dual technique is highly effective in learning RNNs, with superior performance to the earlier heuristic method of truncating the size of the gradient; (2) The use of a DNN to compute high-level features of speech data to feed into the RNN gives much higher accuracy than without using the DNN; and (3) The

accuracy drops progressively as the DNN features are extracted from higher to lower hidden layers of the DNN.

A special case of the RNN is reservoir models or echo state networks, where the output layers are fixed to be linear instead of nonlinear as in the regular RNN, and where the recurrent matrices are carefully designed but not learned. The input matrices are also fixed and not learned, due partly to the difficulty of learning. Only the weight matrices between the hidden and output layers are learned. Since the output layer is linear, the learning is very efficient and with global optimum achievable by a closed-form solution. But due to the fact that many parameters are not learned, the hidden layer needs to be very large in order to obtain good results. Triefenbach et al. [365] applied such models to phone recognition, with reasonably good accuracy obtained.

Palangi et al. [276] presented an improved version of the reservoir model by learning both the input and recurrent matrices which were fixed in the previous model that makes use of the linear output (or readout) units to simplify the learning of only the output matrix in the RNN. Rather, a special technique is devised that takes advantage of the linearity in the output units in the reservoir model to learn the input and recurrent matrices. Compared with the backpropagation through time (BPTT) algorithm commonly used in learning the general RNNs, the proposed technique makes use of the linearity in the output units to provide constraints among various matrices in the RNN, enabling the computation of the gradients as the learning signal in an analytical form instead of by recursion as in the BPTT.

In addition to the recent innovations in better architectures of deep learning models for speech recognition reviewed above, there is also a growing body of work on developing and implementing better nonlinear units. Although sigmoidal and tanh functions are the most commonly used nonlinear types in DNNs their limitations are well known. For example, it is slow to learn the whole network due to weak gradients when the units are close to saturation in both directions. Jaitly and Hinton [183] appear to be the first to apply the rectified linear units (ReLU) in the DNNs to speech recognition to overcome the weakness of the sigmoidal units. ReLU refers to the units in a neural network that use the activation function of $f(x) = \max(0, x)$. Dahl et al. [65]

and Mass et al. [234] successfully applied ReLU to large vocabulary speech recognition, with the best accuracy obtained when combining ReLU with the "Dropout" regularization technique.

Another new type of DNN units demonstrated more recently to be useful for speech recognition is the "maxout" units, which were used for forming the deep maxout network as described in [244]. A deep maxout network consists of multiple layers which generate hidden activations via the maximum or "maxout" operation over a fixed number of weighted inputs called a "group." This is the same operation as the max pooling used in the CNN as discussed earlier for both speech recognition and computer vision. The maximal value within each group is taken as the output from the previous layer. Most recently, Zhang et al. [441] generalize the above "maxout" units to two new types. The "soft-maxout" type of units replace the original max operation with the soft-max function. The second, p-norm type of units used the nonlinearity of $y = \|x\|_p$. It is shown experimentally that the p-norm units with $p = 2$ perform consistently better than the maxout, tanh, and ReLU units. In Gulcehre et al. [138], techniques that automatically learn the p-norm was proposed and investigated.

Finally, Srivastava et al. [350] propose yet another new type of nonlinear units, called winner-take-all units. Here, local competition among neighboring neurons are incorporated into the otherwise regular feedforward architecture, which is then trained via backpropagation with different gradients than the normal one. Winner-take-all is an interesting new form of nonlinearity, and it forms groups of (typically two) neurons where all the neurons in a group are made zero-valued except the one with the largest value. Experiments show that the network does not forget as much as networks with standard sigmoidal nonlinearity. This new type of nonlinear units are yet to be evaluated in speech recognition tasks.

7.1.7 Better optimization and regularization

Another area where significant advances are made recently in applying deep learning to acoustic model for speech recognition is on optimization criteria and methods, as well as on the related

regularization techniques to help prevent overfitting during the deep network training.

One of the early studies on DNNs for speech recognition, conducted at Microsoft Research and reported in [260], first recognizes the mismatch between the desired error rate and the cross-entropy training criterion in the conventional DNN training. The solution is provided by replacing the frame-based, cross-entropy training criterion with the full-sequence-based maximum mutual information optimization objective, in a similar way to defining the training objective for the shallow neural network interfaced with an HMM [194]. Equivalently, this amounts to putting the model of conditional random field (CRF) at the top of the DNN, replacing the original softmax layer which naturally leads to cross entropy. (Note the DNN was called the DBN in the paper). This new sequential discriminative learning technique is developed to jointly optimize the DNN weights, CRF transition weights, and bi-phone language model. Importantly, the speech task is defined in TIMIT, with the use of a simple bi-phone-gram "language" model. The simplicity of the bi-gram language model enables the full-sequence training to carry out without the need to use lattices, drastically reducing the training complexity.

As another way to motivate the full-sequence training method of [260], we note that the earlier DNN phone recognition experiments made use of the standard frame-based objective function in static pattern classification, cross-entropy, to optimize the DNN weights. The transition parameters and language model scores were obtained from an HMM and were trained independently of the DNN weights. However, it has been known during the long history of the HMM research that sequence classification criteria can be very helpful in improving speech and phone recognition accuracy. This is because the sequence classification criteria are more directly correlated with the performance measure (e.g., the overall word or phone error rate) than frame-level criteria. More specifically, the use of frame-level cross entropy to train the DNN for phone sequence recognition does not explicitly take into account the fact that the neighboring frames have smaller distances between the assigned probability distributions over phone class labels. To overcome this deficiency, one can optimize the conditional probability of the

whole sequence of labels, given the whole visible feature utterance or equivalently the hidden feature sequence extracted by DNN. To optimize the log conditional probability on the training data, the gradient can be taken over the activation parameters, transition parameters and lower-layer weights, and then pursue back-propagation of the error defined at the sentence level. We remark that in a much earlier study [212], combining a neural network with a CRF-like structure was done, where the mathematical formulation appears to include CRFs as a special case. Also, the benefit of using the full-sequence classification criteria was shown earlier on shallow neural networks in [194, 291].

In implementing the above full-sequence learning algorithm for the DNN system as described in [260], the DNN weights are initialized using the frame-level cross entropy as the objective. The transition parameters are initialized from the combination of the HMM transition matrices and the "bi-phone language" model scores, and are then further optimized by tuning the transition features while fixing the DNN weights before the joint optimization. Using joint optimization with careful scheduling to reduce overfitting, it is shown that the full-sequence training outperforms the DNN trained with frame-level cross entropy by approximately 5% relative [260]. Without the effort to reduce overfitting, it is found that the DNN trained with MMI is much more prone to overfitting than that trained with frame-level cross entropy. This is because the correlations across frames in speech tend to be different among the training, development, and test data. Importantly, such differences do not show when frame-based objective functions are used for training.

For large vocabulary speech recognition where more complex language models are in use, the optimization methods for full-sequence training of the DNN–HMM are much more sophisticated. Kingsbury et al. [195] reported the first success of such training using parallel, second-order, Hessian-free optimization techniques, which are carefully implemented for large vocabulary speech recognition. Sainath et al. [305] improved and speeded up the Hessian-free techniques by reducing the number of Krylov subspace solver iterations [378], which are used for implicit estimation of the Hessian. They also use sampling

methods to decrease the amount of training data to speed up the training. While the batch-mode, second-order Hessian-free techniques prove successful for full-sequence training of large-scale DNN–HMM systems, the success of the first-order stochastic gradient descent methods is also reported recently [353]. It is found that heuristics are needed to handle the problem of lattice sparseness. That is, the DNN must be adjusted to the updated numerator lattices by additional iterations of frame-based cross-entropy training. Further, artificial silence arcs need to be added to the denominator lattices, or the maximum mutual information objective function needs to be smoothed with the frame-based cross entropy objective. The conclusion is that for large vocabulary speech recognition tasks with sparse lattices, the implementation of the sequence training requires much greater engineering skills than the small tasks such as reported in [260], although the objective function as well as the gradient derivation are essentially the same. Similar conclusions are reached by Vesely et al. [374] when carrying out full-sequence training of DNN–HMMs for large-vocabulary speech recognition. However, different heuristics from [353] are shown to be effective in the training. Separately, Wiesler et al. [390] investigated the Hessian-free optimization method for training the DNN with the cross-entropy objective and empirically analyzed the properties of the method. And finally, Dognin and Goel [113] combined stochastic average gradient and Hessian-free optimization for sequence training of deep neural networks with success in that the training procedure converges in about half the time compared with the full Hessian-free sequence training.

For large DNN–HMM systems with either frame-level or sequence-level optimization objectives, speeding up the training is essential to take advantage of large amounts of training data and of large model sizes. In addition to the methods described above, Dean et al. [69] reported the use of the asynchronous stochastic gradient descent (ASGD) method, the adaptive gradient descent (Adagrad) method, and the large-scale limited-memory BFGS (L-BFGS) method for very large vocabulary speech recognition. Sainath et al. [312] provided a review of a wide range of optimization methods for speeding up the training of DNN-based systems for large speech recognition tasks.

In addition to the advances described above focusing on optimization with the fully supervised learning paradigm, where all training data contain the label information, the semi-supervised training paradigm is also exploited for learning DNN–HMM systems for speech recognition. Liao et al. [223] reported the exploration of using semi-supervised training on the DNN–HMM system for the very challenging task of recognizing YouTube speech. The main technique is based on the use of "island of confidence" filtering heuristics to select useful training segments. Separately, semi-supervised training of DNNs is explored by Vesely et al. [374], where self-training strategies are used as the basis for data selection using both the utterance-level and frame-level confidences. Frame-selection based on per-frame confidences derived from confusion in a lattice is found beneficial. Huang et al. [176] reported another variant of semi-supervised training technique in which multi-system combination and confidence recalibration is applied to select the training data. Further, Thomas et al. [362] overcome the problem of lacking sufficient training data for acoustic modeling in a number of low-resource scenarios. They make use of transcribed multilingual data and semi-supervised training to build the proposed feature front-ends for subsequent speech recognition.

Finally, we see important progress in deep learning based speech recognition in recent years with the introduction of new regularization methods based on "dropout" originally proposed by Hinton et al. [166]. Overfitting is very common in DNN training and co-adaptation is prevalent within the DNN with multiple activations adapting together to explain input acoustic data. Dropout is a technique to limit co-adaptation. It operates as follows. On each training instance, each hidden unit is randomly omitted with a fixed probability (e.g., $p = 0.5$). Then, decoding is done normally except with straightforward scaling of the DNN weights (by a factor of $1 - p$). Alternatively, the scaling of the DNN weights can be done during training [by a factor of $1/(1 - p)$] rather than in decoding. The benefits of dropout regularization for training DNNs are to make a hidden unit in the DNN act strongly by itself without relying on others, and to serve a way to do model averaging of different networks. These benefits are most pronounced when the training data is limited, or when the DNN size is disproportionally large

with respect to the size of the training data. Dahl et al. [65] applied dropout in conjunction with the ReLU units and to only the top few layers of a fully-connected DNN. Seltzer and Yu [325] applied it to noise robust speech recognition. Deng et al. [81], on the other hand, applied dropout to all layers of a deep convolutional neural network, including both the top fully connected DNN layers and the bottom locally connected CNN layer and the pooling layer. It is found that the dropout rate need to be substantially smaller for the convolutional layer.

Subsequent work on applying dropout includes the study by Miao and Metze [243], where DNN-based speech recognition is constrained by low resources with sparse training data. Most recently, Sainath et al. [306] combined dropout with a number of novel techniques described in this section (including the use of deep CNNs, Hessian-free sequence learning, the use of ReLU units, and the use of joint fMLLR and filter-bank features, etc.) to obtain state of the art results on several large vocabulary speech recognition tasks.

As a summary, the initial success of deep learning methods for speech analysis and recognition reported around 2010 has come a long way over the past three years. An explosive growth in the work and publications on this topic has been observed, and huge excitement has been ignited within the speech recognition community. We expect that the growth in the research on deep learning based speech recognition will continue, at least in the near future. It is also fair to say that the continuing large-scale success of deep learning in speech recognition as surveyed in this chapter (up to the ASRU-2013 time frame) is a key stimulant to the large-scale exploration and applications of the deep learning methods to other areas, which we will survey in Sections 8–11.

7.2 Speech synthesis

In addition to speech recognition, the impact of deep learning has recently spread to speech synthesis, aimed to overcome the limitations of the conventional approach in statistical parametric synthesis based on Gaussian-HMM and decision-tree-based model clustering. The goal of speech synthesis is to generate speech sounds directly from text and

possibly with additional information. The first set of papers appeared at ICASSP, May 2013, where four different deep learning approaches are reported to improve the traditional HMM-based statistical parametric speech synthesis systems built based on "shallow" speech models, which we briefly review here after providing appropriate background information.

Statistical parametric speech synthesis emerged in the mid-1990s, and is currently the dominant technology in speech synthesis. See a recent overview in [364]. In this approach, the relationship between texts and their acoustic realizations are modeled using a set of stochastic generative acoustic models. Decision tree-clustered context-dependent HMMs with a Gaussian distribution as the output of an HMM state are the most popular generative acoustic model used. In such HMM-based speech synthesis systems, acoustic features including the spectra, excitation and segment durations of speech are modeled simultaneously within a unified context-dependent HMM framework. At the synthesis time, a text analysis module extracts a sequence of contextual factors including phonetic, prosodic, linguistic, and grammatical descriptions from an input text to be synthesized. Given the sequence of contextual factors, a sentence-level context-dependent HMM corresponding to the input text is composed, where its model parameters are determined by traversing the decision trees. The acoustic features are predicted so as to maximize their output probabilities from the sentence HMM under the constraints between static and dynamic features. Finally, the predicted acoustic features are sent to a waveform synthesis module to reconstruct the speech waveforms. It has been known for many years that the speech sounds generated by this standard approach are often muffled compared with natural speech. The inadequacy of acoustic modeling based on the shallow-structured HMM is conjectured to be one of the reasons. Several very recent studies have adopted deep learning approaches to overcome such deficiency. One significant advantage of deep learning techniques is their strong ability to represent the intrinsic correlation or mapping relationship among the units of a high-dimensional stochastic vector using a generative (e.g., the RBM and DBN discussed in Section 3.2) or discriminative (e.g., the DNN discussed in Section 3.3) modeling

framework. The deep learning techniques are thus expected to help the acoustic modeling aspect of speech synthesis in overcoming the limitations of the conventional shallow modeling approach.

A series of studies are carried out recently on ways of overcoming the above limitations using deep learning methods, inspired partly by the intrinsically hierarchical processes in human speech production and the successful applications of a number of deep learning methods in speech recognition as reviewed earlier in this chapter. In Ling et al. [227, 229], the RBM and DBN as generative models are used to replace the traditional Gaussian models, achieving significant quality improvement, in both subjective and objective measures, of the synthesized voice. In the approach developed in [190], the DBN as a generative model is used to represent joint distribution of linguistic and acoustic features. Both the decision trees and Gaussian models are replaced by the DBN. The method is very similar to that used for generating digit images by the DBN, where the issue of temporal sequence modeling specific to speech (non-issue for image) is by-passed via the use of the relatively large, syllable-sized units in speech synthesis. On the other hand, in contrast to the generative deep models (RBMs and DBNs) exploited above, the study reported in [435] makes use of the discriminative model of the DNN to represent the conditional distribution of the acoustic features given the linguistic features. Finally, in [115], the discriminative model of the DNN is used as a feature extractor that summarizes high-level structure from the raw acoustic features. Such DNN features are then used as the input for the second stage for the prediction of prosodic contour targets from contextual features in the full speech synthesis system.

The application of deep learning to speech synthesis is in its infancy, and much more work is expected from that community in the near future.

7.3 Audio and music processing

Similar to speech recognition but to a less extent, in the area of audio and music processing, deep learning has also become of intense interest

but only quite recently. As an example, the first major event of deep learning for speech recognition took place in 2009, followed by a series of events including a comprehensive tutorial on the topic at ICASSP-2012 and with the special issue at IEEE Transactions on Audio, Speech, and Language Processing, the premier publication for speech recognition, in the same year. The first major event of deep learning for audio and music processing appears to be the special session at ICASSP-2014, titled Deep Learning for Music [14].

In the general field of audio and music processing, the impacted areas by deep learning include mainly music signal processing and music information retrieval [15, 22, 141, 177, 178, 179, 319]. Deep learning presents a unique set of challenges in these areas. Music audio signals are time series where events are organized in musical time, rather than in real time, which changes as a function of rhythm and expression. The measured signals typically combine multiple voices that are synchronized in time and overlapping in frequency, mixing both short-term and long-term temporal dependencies. The influencing factors include musical tradition, style, composer and interpretation. The high complexity and variety give rise to the signal representation problems well-suited to the high levels of abstraction afforded by the perceptually and biologically motivated processing techniques of deep learning.

In the early work on audio signals as reported by Lee et al. [215] and their follow-up work, the convolutional structure is imposed on the RBM while building up a DBN. Convolution is made in time by sharing weights between hidden units in an attempt to detect the same "invariant" feature over different times. Then a max-pooling operation is performed where the maximal activations over small temporal neighborhoods of hidden units are obtained, inducing some local temporal invariance. The resulting convolutional DBN is applied to audio as well as speech data for a number of tasks including music artist and genre classification, speaker identification, speaker gender classification, and phone classification, with promising results presented.

The RNN has also been recently applied to music processing applications [22, 40, 41], where the use of ReLU hidden units instead of logistic or tanh nonlinearities are explored in the RNN. As reviewed in

Section 7.2, ReLU units compute $y = \max(x, 0)$, and lead to sparser gradients, less diffusion of credit and blame in the RNN, and faster training. The RNN is applied to the task of automatic recognition of chords from audio music, an active area of research in music information retrieval. The motivation of using the RNN architecture is its power in modeling dynamical systems. The RNN incorporates an internal memory, or hidden state, represented by a self-connected hidden layer of neurons. This property makes them well suited to model temporal sequences, such as frames in a magnitude spectrogram or chord labels in a harmonic progression. When well trained, the RNN is endowed with the power to predict the output at the next time step given the previous ones. Experimental results show that the RNN-based automatic chord recognition system is competitive with existing state-of-the-art approaches [275]. The RNN is capable of learning basic musical properties such as temporal continuity, harmony and temporal dynamics. It can also efficiently search for the most musically plausible chord sequences when the audio signal is ambiguous, noisy or weakly discriminative.

A recent review article by Humphrey et al. [179] provides a detailed analysis on content-based music informatics, and in particular on why the progress is decelerating throughout the field. The analysis concludes that hand-crafted feature design is sub-optimal and unsustainable, that the power of shallow architectures is fundamentally limited, and that short-time analysis cannot encode musically meaningful structure. These conclusions motivate the use of deep learning methods aimed at automatic feature learning. By embracing feature learning, it becomes possible to optimize a music retrieval system's internal feature representation or discovering it directly, since deep architectures are especially well-suited to characterize the hierarchical nature of music. Finally, we review the very recent work by van den Oord, et al. [371] on content-based music recommendation using deep learning methods. Automatic music recommendation has become an increasingly significant and useful technique in practice. Most recommender systems rely on collaborative filtering, suffering from the cold start problem where it fails when no usage data is available. Thus, collaborative filtering is

not effective for recommending new and unpopular songs. Deep learning methods power the latent factor model for recommendation, which predicts the latent factors from music audio when they cannot be obtained from usage data. A traditional approach using a bag-of-words representation of the audio signals is compared with deep CNNs with rigorous evaluation made. The results show highly sensible recommendations produced by the predicted latent factors using deep CNNs. The study demonstrates that a combination of convolutional neural networks and richer audio features lead to such promising results for content-based music recommendation.

Like speech recognition and speech synthesis, much more work is expected from the music and audio signal processing community in the near future.