We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,500
Open access books available

119,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the
TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter 2

Convolutional Neural Networks for Raw Speech Recognition

Vishal Passricha and Rajesh Kumar Aggarwal

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.80026

Abstract

State-of-the-art automatic speech recognition (ASR) systems map the speech signal into its corresponding text. Traditional ASR systems are based on Gaussian mixture model. The emergence of deep learning drastically improved the recognition rate of ASR systems. Such systems are replacing traditional ASR systems. These systems can also be trained in end-to-end manner. End-to-end ASR systems are gaining much popularity due to simplified model-building process and abilities to directly map speech into the text without any predefined alignments. Three major types of end-to-end architectures for ASR are attention-based methods, connectionist temporal classification, and convolutional neural network (CNN)-based direct raw speech model. In this chapter, CNN-based acoustic model for raw speech signal is discussed. It establishes the relation between raw speech signal and phones in a data-driven manner. Relevant features and classifier both are jointly learned from the raw speech. Raw speech is processed by first convolutional layer to learn the feature representation. The output of first convolutional layer, that is, intermediate representation, is more discriminative and further processed by rest convolutional layers. This system uses only few parameters and performs better than traditional cepstral feature-based systems. The performance of the system is evaluated for TIMIT and claimed similar performance as MFCC.

Keywords: ASR, attention-based model, connectionist temporal classification, CNN, end-to-end model, raw speech signal

1. Introduction

ASR system has two important tasks—phoneme recognition and whole-word decoding. In ASR, the relationship between the speech signal and phones is established in two different steps [1]. In the first step, useful features are extracted from the speech signal on the basis of

IntechOpen

© 2018 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

prior knowledge. This phase is known as information selection or dimensionality reduction phase. In this, the dimensionality of the speech signal is reduced by selecting the information based on task-specific knowledge. Highly specialized features like MFCC [2] are preferred choice in traditional ASR systems. In the second step, discriminative models estimate the likelihood of each phoneme. In the last, word sequence is recognized using discriminative programming technique. Deep learning system can map the acoustic features into the spoken phonemes directly. A sequence of the phoneme is easily generated from the frames using frame-level classification.

Another side, end-to-end systems perform acoustic frames to phone mapping in one step only. End-to-end training means all the modules are learned simultaneously. Advanced deep learning methods facilitate to train the system in an end-to-end manner. They also have the ability to train the system directly with raw signals, i.e., without hand-crafted features. Therefore, ASR paradigm is shifting from cepstral features like MFCC [2], PLP [3] to discriminative features learned directly from raw speech. End-to-end model may take raw speech signal as input and generates phoneme class conditional probabilities as output. The three major types of end-to-end architectures for ASR are attention-based method, connectionist temporal classification (CTC), and CNN-based direct raw speech model.

Attention-based models directly transcribe the speech into phonemes. Attention-based encoder-decoder uses the recurrent neural network (RNN) to perform sequence-to-sequence mapping without any predefined alignment. In this model, the input sequence is first transformed into a fixed length vector representation, and then decoder maps this fixed length vector into the output sequence. Attention-based encoder-decoder is much capable of learning the mapping between variable-length input and output sequences. Chorowski and Jaitly proposed speaker-independent sequence-to-sequence model and achieved 10.6% WER without separate language models and 6.7% WER with a trigram language model for Wall Street Journal dataset [4]. In attention-based systems, the alignment between the acoustic frame and recognized symbols is performed by attention mechanism, whereas CTC model uses conditional independence assumptions to efficiently solve sequential problems by dynamic programming. Attention model has shown high performance over CTC approach because it uses the history of the target character without any conditional independence assumptions.

Another side, CNN-based acoustic model is proposed by Palaz et al. [5–7] which processes the raw speech directly as input. This model consists of two stages: feature learning stage, i.e., several convolutional layers, and classifier stage, i.e., fully connected layers. Both the stages are learned jointly by minimizing a cost function based on relative entropy. In this model, the information is extracted by the filters at first convolutional layer and modeled between first and second convolutional layer. In classifier stage, learned features are classified by fully connected layers and softmax layer. This approach claims comparable or better performance than traditional cepstral feature-based system followed by ANN training for phoneme recognition on TIMIT dataset.

This chapter is organized as follows: In Section 2, the work performed in the field of ASR is discussed with the name of related work. Section 3 covers the various architectures of ASR. Section 4 presents the brief introduction about CNN. Section 5 explains CNN-based direct raw speech recognition model. In Section 6, available experimental results are shown. Finally, Section 7 concludes this chapter with the brief discussion.

2. Related work

Traditional ASR system leveraged the GMM/HMM paradigm for acoustic modeling. GMM efficiently processes the vectors of input features and estimates emission probabilities for each HMM state. HMM efficiently normalizes the temporal variability present in speech signal. The combination of HMM and language model is used to estimate the most likely sequence of phones. The discriminative objective function is used to improve the recognition rate of the system by the discriminatively fine-tuned methods [8]. However, GMM has a shortcoming as it shows inability to model the data that is present on the boundary line. Artificial neural networks (ANNs) can learn much better models of data lying on the boundary condition. Deep neural networks (DNNs) as acoustic models tremendously improved the performance of ASR systems [9–11]. Generally, discriminative power of DNN is used for phoneme recognition and, for decoding task, HMM is preferred choice. DNNs have many hidden layers with a large number of nonlinear units and produce a very large number of outputs. The benefit of this large output layer is that it accommodates the large number of HMM states. DNN architectures have densely connected layers. Therefore, such architectures are more prone to overfitting. Secondly, features having the local correlations become difficult to learn for such architectures. In [12], speech frames are classified into clustered context-dependent states using DNNs. In [13, 14], GMM-free DNN training process is proposed by the researchers. However, GMM-free process demands iterative procedures like decision trees, generating forced alignments. DNN-based acoustic models are gaining much popularity in large vocabulary speech recognition task [10], but components like HMM and n-gram language model are same as in their predecessors.

GMM or DNN-based ASR systems perform the task in three steps: feature extraction, classification, and decoding. It is shown in **Figure 1**. Firstly, the short-term signal s_t is processed at time “t” to extract the features x_t. These features are provided as input to GMM or DNN acoustic model which estimates the class conditional probabilities $P_e(i|x_i)$ for each phone class $i \in \{1, \ldots, I\}$. The emission probabilities are as follows:

$$p_e(x_t|i) \propto \frac{p(x_t|i)}{p(x_t)} = \frac{P(i|x_t)}{p(i)} \quad \forall_i \in i, \ldots, I \tag{1}$$

The prior class probability $p(i)$ is computed by counting on the training set.

DNN is a feed-forward NN containing multiple hidden layers with a large number of hidden units. DNNs are trained using the back-propagation methods then discriminatively fine-tuned for reducing the gap between the desired output and actual output. DNN-/HMM-based hybrid systems are the effective models which use a tri-phone HMM model and an n-gram language model [10, 15]. Traditional DNN/HMM hybrid systems have several independent components that are trained separately like an acoustic model, pronunciation model, and

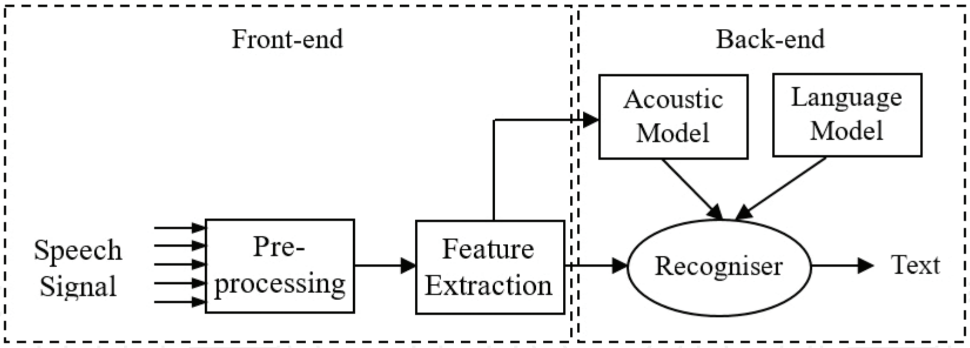

Figure 1. General framework of automatic speech recognition system.

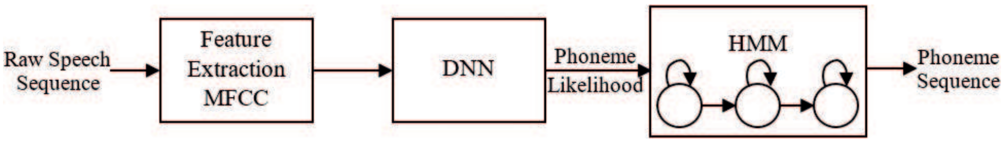

Figure 2. Hybrid DNN/HMM phoneme recognition.

language model. In the hybrid model, the speech recognition task is factorized into several independent subtasks. Each subtask is independently handled by a separate module which simplifies the objective. The classification task is much simpler in HMM-based models as compared to classifying the set of variable-length sequences directly. **Figure 2** shows the hybrid DNN/HMM phoneme recognition model.

On the other side, researchers proposed end-to-end ASR systems that directly map the speech into labels without any intermediate components. As the advancements in deep learning, it has become possible to train the system in an end-to-end fashion. The high success rate of deep learning methods in vision task motivates the researchers to focus on classifier step for speech recognition. Such architectures are called deep because they are composed of many layers as compared to classical “shallow” systems. The main goal of end-to-end ASR system is to simplify the conventional module-based ASR system into a single deep learning framework. In earlier systems, divide and conquer approaches are used to optimize each step independently, whereas deep learning approaches have a single architecture that leads to more optimal system. End-to-end speech recognition systems directly map the speech to text without requiring predefined alignment between acoustic frame and characters [16–24]. These systems are generally divided into three broad categories: attention-based model [19–22], connectionist temporal classification [16–18, 25], and CNN-based direct raw speech method [5–7, 26]. All these models have a capability to address the problem of variable-length input and output sequences.

Attention-based models are gaining much popularity in a variety of tasks like handwriting synthesis [27], machine translation [28], and visual object classification [29]. Attention-based models directly map the acoustic frame into character sequences. However, this model differs from other machine translation tasks by requesting much longer input sequences. This model generates a character based on the inputs and history of the target character. The

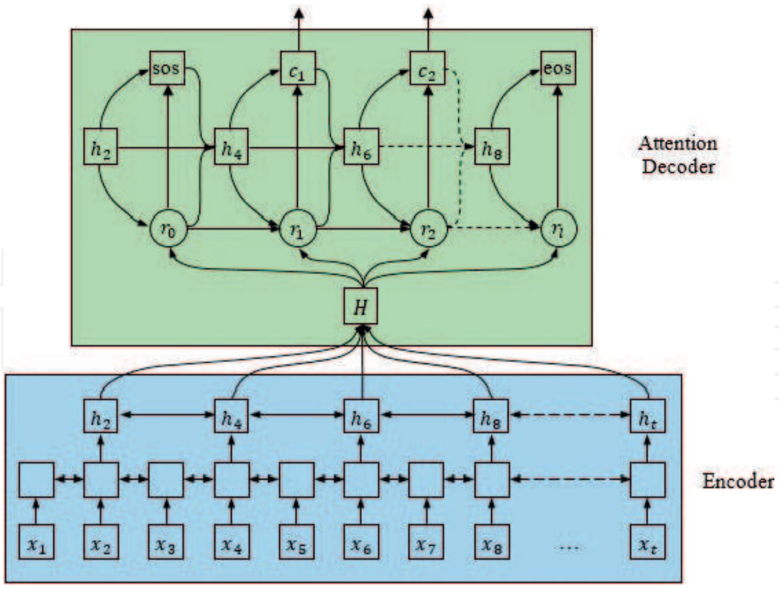

Figure 3. Attention-based ASR model.

attention-based models use encoder-decoder architecture to perform the sequence mapping from speech feature sequences to text as shown in **Figure 3**. Its extension, i.e., attention-based recurrent networks, has also been successfully applied to speech recognition. In the noisy environment, these models' results are poor because the estimated alignment is easily corrupted by noise. Another issue with this model is that it is hard to train from scratch due to misalignment on longer input sequences. Sequence-to-sequence networks have also achieved many breakthroughs in speech recognition [20–22]. They can be divided into three modules: an encoding module that transforms sequences, attention module that estimates the alignment between the hidden vector and targets, and decoding module that generates the output sequence. To develop successful sequence-to-sequence model, the understanding and preventing limitations are required. The discriminative training is a different way of training that raises the performance of the system. It allows the model to focus on most informative features with the risk of overfitting.

End-to-end trainable speech recognition systems are an important application of attention-based models. The decoder network computes a matching score between hidden states generated by the acoustic encoder network at each input time. It processes its hidden states to form a temporal alignment distribution. This matching score is used to estimate the corresponding encoder states. The difficulty of attention-based mechanism in speech recognition is that the feature inputs and corresponding letter outputs generally proceed in the same order with only small deviations within word. However, the different length of input and output sequences makes it more difficult to track the alignment. The advantage of attention-based mechanism is that any conditional independence assumptions (Markov assumption) are not required in this mechanism. Attention-based approach replaces the HMM with RNN to perform the sequence prediction. Attention mechanism automatically learns alignment between the input features and desired character sequence.

CTC techniques infer the speech-label alignment automatically. CTC [25] was developed for decoding the language. Firstly, Hannun et al. [17] used it for decoding purpose in Baidu's deep speech network. CTC uses dynamic programming [16] for efficient computation of a strictly monotonic alignment. However, graph-based decoding and language model are required for it. CTC approaches use RNN for feature extraction [28]. Graves et al. [30] used its objective function in deep bidirectional long short-term memory (LSTM) system. This model successfully arranges all possible alignments between input and output sequences during model training, not on the prior.

Two different versions of beam search are adopted by [16, 31] for decoding CTC models. **Figure 4** shows the working architecture of the CTC model. In this, noisy and not informative frames are discarded by the introduction of the blank label which results in the optimal output sequence. CTC uses intermediate label representation to identify the blank labels, i.e., no output labels. CTC-based NN model shows high recognition rate for both phoneme recognition [32] and LVCSR [16, 31]. CTC-trained neural network with language model offers excellent results [17].

End-to-end ASR systems perform well and achieve good results, yet they face two major challenges. First is how to incorporate lexicons and language models into decoding. However, [16, 31, 33] have incorporated lexicons for searching paths. Second, there is no shared experimental platform for the purpose of benchmark. End-to-end systems differ from the traditional system in both aspects: model architecture and decoding methods. Some efforts were also made to model the raw speech signal with little or no preprocessing [34]. Palaz et al. [6] showed in his study that CNN [35] can calculate the class conditional probabilities from raw speech signal as direct input. Therefore, CNNs are the preferred choice to learn features from the raw speech. Two stages of learned feature process are as follows: initially, features are learned by the filters at first convolutional layer, and then learned features are modeled by second and higher-level convolutional layers. An end-to-end phoneme sequence

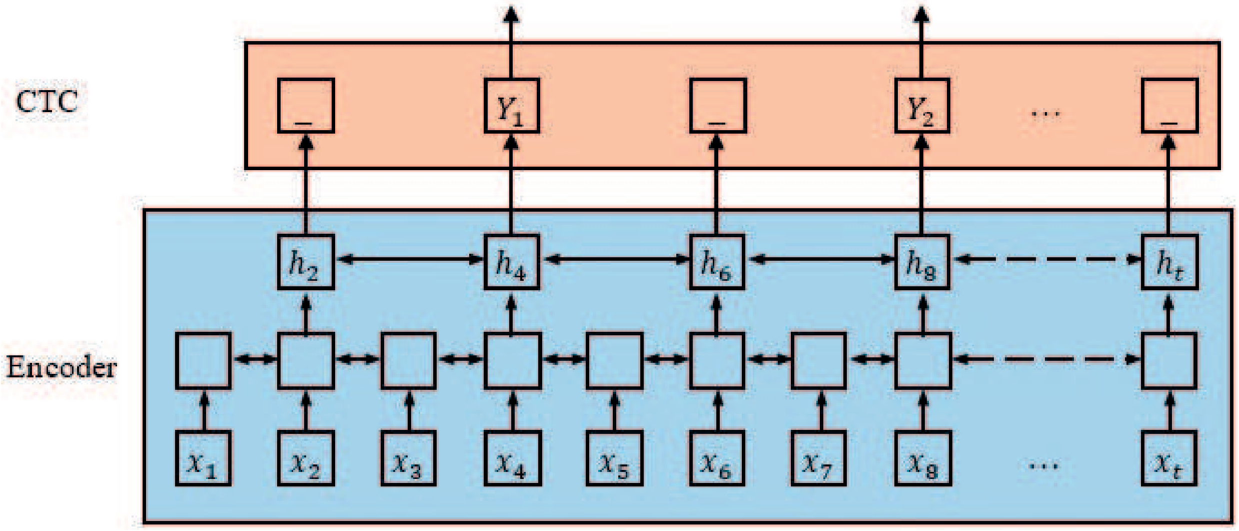

Figure 4. CTC model for speech recognition.

recognizer directly processes the raw speech signal as inputs and produces a phoneme sequence. The end-to-end system is composed of two parts: convolutional neural networks and conditional random field (CRF). CNN is used to perform the feature learning and classification, and CRFs are used for the decoding stage. CRF, ANN, multilayer perceptron, etc. have been successfully used as decoder. The results on TIMIT phone recognition task also confirm that the system effectively learns the features from raw speech and performs better than traditional systems that take cepstral features as input [36]. This model also produces good results for LVCSR [7].

IntechOpen

3. Various architectures of ASR

In this section, a brief review on conventional GMM/DNN ASR, attention-based end-to-end ASR, and CTC is given.

3.1. GMM/DNN

ASR system performs sequence mapping of T-length speech sequence features, $X = \left\{X_t \in \mathbb{R}^D | t = 1, \ \ldots, T\right\}$, into an N-length word sequence, $W = \{w_n \in v | n = 1, \ldots, N\}$ where X_t represents the D-dimensional speech feature vector at frame t and w_n represents the word at position n in the vocabulary, v.

The ASR problem is formulated within the Bayesian framework. In this method, an utterance is represented by some sequence of acoustic feature vector X, derived from the underlying sequence of words W, and the recognition system needs to find the most likely word sequence as given below [37]:

$$\hat{W} = \arg\max_{w} p(W|X) \tag{2}$$

In Eq. (2), the argument of $p(W|X)$, that is, the word sequence W, is found which shows maximum probability for given feature vector, X. Using Bayes' rule, it can be written as

$$\hat{W} = \arg\max_{w} \frac{p(X|W)p(W)}{p(X)} \tag{3}$$

IntechOpen

In Eq. (3), the denominator $p(X)$ is ignored as it is constant with respect to W. Therefore,

$$\hat{W} = \arg\max_{w} p(X|W)p(W) \tag{4}$$

where $p(X|W)$ represents the sequence of speech features and it is evaluated with the help of acoustic model. $p(W)$ represents the prior knowledge about the sequence of words W and it is determined by the language model. However, current ASR systems are based on a hybrid

HMM/DNN [38], which is also calculated using Bayes' theorem and introduces the HMM state sequence S, to factorize $p(W|X)$ into the following three distributions:

$$\arg \max_{w \in v^*} p(W|X) \tag{5}$$

$$= \arg \max_{w \in v^*} \sum_S p(X|S, W) p(S|W) p(W) \tag{6}$$

$$\approx \arg \max_{w \in v^*} \sum_S p(X|S), p(S|W) p(W) \tag{7}$$

where $p(X|S)$, $p(S|W)$, and $p(W)$ represent acoustic, lexicon, and language models, respectively. Equation (6) is changed into Eq. (7) in a similar way as Eq. (4) is changed into Eq. (5).

3.1.1. Acoustic models $p(X|S)$

$p(X|S)$ can be further factorized using a probabilistic chain rule and Markov assumption as follows:

$$p(X|S) = \prod_{t=1}^{T} p(x_t|x_1, \ldots, x_{t-1}, S) \tag{8}$$

$$\approx \prod_{t=1}^{T} p(x_t|s_t) \propto \prod_{t=1}^{T} \frac{p(s_t|x_t)}{p(s_t)} \tag{9}$$

In Eq. (9), framewise likelihood function $p(x_t|s_t)$ is changed into the framewise posterior distribution $\frac{p(s_t|x_t)}{p(s_t)}$ which is computed using DNN classifiers by pseudo-likelihood trick [38]. In Eq. (9), Markov assumption is too strong. Therefore, the contexts of input and hidden states are not considered. This issue can be resolved using either the recurrent neural networks (RNNs) or DNNs with long-context features. A framewise state alignment is required to train the framewise posterior which is offered by an HMM/GMM system.

3.1.2. Lexicon model $p(S|W)$

$p(S|W)$ can be further factorized using a probabilistic chain rule and Markov assumption (first order) as follows:

$$p(S|W) = \prod_{t=1}^{T} p(s_t|s_1, \ldots, s_{t-1}, W) \tag{10}$$

$$\approx \prod_{t=1}^{T} p(s_t|\ s_{t-1}, W) \tag{11}$$

An HMM state transition represents this probability. A pronunciation dictionary performs the conversion from w to HMM states through phoneme representation.

IntechOpen

IntechOpen

3.1.3. *Language model $p(W)$*

Similarly, $p(W)$ can be factorized using a probabilistic chain rule and Markov assumption ((m-1) th order) as an m-gram model, i.e.,

$$p(W) = \prod_{n=1}^{N} p(w_n | w_1, \dots, w_{n-1}) \tag{12}$$

$$\approx \prod_{n=1}^{N} p(w_n | w_{n-m-1}, \dots, w_{n-1}) \tag{13}$$

The issue of Markov assumption is addressed using recurrent neural network language model (RNNLM) [39], but it increases the complexity of decoding process. The combination of RNNLMs and m-gram language model is generally used and it works on a rescoring technique.

3.2. Attention mechanism

The approach based on attention mechanism does not make any Markov assumptions. It directly finds the posterior $p(C|X)$, on the basis of a probabilistic chain rule:

$$p(C|X) = \underbrace{\prod\nolimits_{l=1}^{L} p(c_l | c_1, \dots, c_{l-1}, X)}_{\triangleq p_{att}(C/X)} \tag{14}$$

where $p_{att}(C|X)$ represents an attention-based objective function. $p(c_l | c_1, \dots, c_{l-1}, X)$ is obtained by

$$\boldsymbol{h}_t = Encoder(X), \tag{15}$$

$$a_{lt} = \begin{cases} ContentAttention\left(\boldsymbol{q}_{l-1}, \boldsymbol{h}_t\right) \\ LocationAttention\left(\{a_{l-1}\}_{t=1}^{T}, \boldsymbol{q}_{l-1}, \boldsymbol{h}_t\right) \end{cases}, \tag{16}$$

$$\boldsymbol{r}_l = \sum_{t=1}^{T} a_{lt} \boldsymbol{h}_t, \tag{17}$$

$$p(c_l | c_1, \dots, c_{l-1}, X) = Decoder\ \left(\boldsymbol{r}_l, \boldsymbol{q}_{l-1}, c_{l-1}\right) \tag{18}$$

Eq. (15) represents the encoder and Eq. (18) represents the decoder networks. a_{lt} represents the soft alignment of the hidden vector, $\boldsymbol{h}_t$. Here, $\boldsymbol{r}_l$ represents the weighted letter-wise hidden vector that is computed by weighted summation of hidden vectors. Content-based attention mechanism with or without convolutional features are shown by $ContentAttention(.)$ and $LocationAttention(.)$, respectively.

3.2.1. *Encoder network*

The input feature vector X is converted into a framewise hidden vector, $\boldsymbol{h}_t$ using Eq. (15). The preferred choice for an encoder network is BLSTM, i.e.,

$$Encoder(X) \triangleq BLSTM_t(X) \tag{19}$$

It is to be noted that the computational complexity of the encoder network is reduced by subsampling the outputs [20, 21].

3.2.2. Content-based attention mechanism

$ContentAttention(.)$ is shown as

$$e_{lt} = g^T \tanh\left(Lin\left(\boldsymbol{q}_{l-1}\right) + LinB(\boldsymbol{h}_t)\right) \tag{20}$$

$$a_{lt} = Softmax\left(\{e_{lt}\}_{t=1}^{T}\right) \tag{21}$$

g represents a learnable parameter. $\{e_{lt}\}_{t=1}^{T}$ represents a T-dimensional vector. $\tanh(.)$ and $Lin(.)$ represent the hyperbolic tangent activation function and linear layer with learnable matrix parameters, respectively.

3.2.3. Location-aware attention mechanism

It is an extended version of content-based attention mechanism to deal with the location-aware attention. If $a_{l-1} = \{a_{l-1}\}_{t=1}^{T}$ is replaced in Eq. (16), then $LocationAware(.)$ is represented as follows:

$$\{\boldsymbol{f}_t\}_{t=1}^{T} = \boldsymbol{\mathcal{K}} * \boldsymbol{a}_{l-1} \tag{22}$$

$$e_{lt} = g^T \tanh\left(Lin\left(\boldsymbol{q}_{l-1}\right) + Lin(\boldsymbol{h}_t) + LinB\left(\boldsymbol{f}_t\right)\right) \tag{23}$$

$$a_{lt} = softmax\left(\{e_t\}_{t=1}^{T}\right) \tag{24}$$

Here, * denotes 1-D convolution along the input feature axis, t, with the convolution parameter, $\boldsymbol{\mathcal{K}}$, to produce the set of T features $\{f_t\}_{t=1}^{T}$.

3.2.4. Decoder network

The decoder network is an RNN that is conditioned on previous output C_{l-1} and hidden vector $\boldsymbol{q}_{l-1}$. LSTM is preferred choice of RNN that represented as follows:

$$Decoder(.) \triangleq softmax(LinB(LSTM_l(.))) \tag{25}$$

$LSTM_l(.)$ represents uniconditional LSTM that generates hidden vector $\boldsymbol{q}_l$ as output:

$$\boldsymbol{q}_l = LSTM_l\left(\boldsymbol{r}_l, \boldsymbol{q}_{l-1}, c_{l-1}\right) \tag{26}$$

$\boldsymbol{r}_l$ represents the concatenated vector of the letter-wise hidden vector; c_{l-1} represents the output of the previous layer which is taken as input.

3.2.5. Objective function

The objective function of the attention model is computed from the sequence posterior

$$p_{att}(C|X) \approx \prod_{l=1}^{L} p\left(c_l|c_1^*, \dots, c_{l-1}^*, X\right) \triangleq p_{att}^*(C|X) \tag{27}$$

where c_l^* represents the ground truth of the previous characters. Attention-based approach is a combination of letter-wise objectives based on multiclass classification with the conditional ground truth history $c_l^*, \dots, c_{l-1}^*$ in each output l.

3.3. Connectionist temporal classification (CTC)

The CTC formulation is also based on Bayes' decision theory. It is to be noted that L-length letter sequence,

$$C' = \{< b >, c_1, < b >, c_2, < b >, \dots, c_L, < b >\} = \left\{c_l' \in \mathcal{U} \cup \{< b >\} | l = 1, \dots, 2L+1\right\} \tag{28}$$

In C', c_l' is always $'' < b > ''$ and letter when l is an odd and an even number, respectively. Similar as DNN/HMM model, framewise letter sequence with an additional blank symbol

$$Z = \{z_t \in \mathcal{U} \cup \{< b >\} | t = 1, \dots, T\} \tag{29}$$

is also introduced. The posterior distribution, $p(C|X)$, can be factorized as

$$p(C|X) = \sum_z p(C|Z, X) p(Z|X) \tag{30}$$

$$\approx \sum_z p(C|Z).p(Z|X) \tag{31}$$

Same as Eq. (3), CTC also uses Markov assumption, i.e., $p(C|Z, X) \approx p(C|Z)$, to simplify the dependency of the CTC acoustic model, $p(Z|X)$, and CTC letter model, $p(C|Z)$.

3.3.1. CTC acoustic model

Same as DNN/HMM acoustic model, $p(Z|X)$ can be further factorized using a probabilistic chain rule and Markov assumption as follows:

$$p(Z|X) = \prod_{t=1}^{T} p(z_t|z_1, \dots, z_{t-1}, X) \tag{32}$$

$$\approx \prod_{t=1}^{T} p(z_t|\, X) \tag{33}$$

The framewise posterior distribution, $p(z_t|X)$ is computed from all inputs, X, and it is directly modeled using bidirectional LSTM [30, 40]:

$$p(z_t|X) = Softmax(LinB(\boldsymbol{h}_t)), \tag{34}$$

$$\boldsymbol{h}_t = BLSTM_t(X) \tag{35}$$

where $Softmax(.)$ represents the softmax activation function. $LinB(.)$ is used to convert the hidden vector, $\boldsymbol{h}_t$, to a $(|\mathcal{U}| + 1)$ dimensional vector with learnable matrix and bias vector parameter. $BLSTM_t(.)$ takes full input sequence as input and produces hidden vector $(\boldsymbol{h}_t)$ at t.

3.3.2. CTC letter model

By applying Bayes' decision theory probabilistic chain rule and Markov assumption, $p(Z|X)$ can be written as

$$p(C/Z) = \frac{p(Z/C)p(C)}{p(Z)} \tag{36}$$

$$= \prod_{t=1}^{T} p(z_t|z_1, \ldots, z_{t-1}, C)\frac{p(C)}{p(Z)} \tag{37}$$

$$\approx \prod_{t=1}^{T} p(z_t|z_{t-1}, C)\frac{p(C)}{p(Z)} \tag{38}$$

where $p(z_t|z_{t-1}, C)$ represents state transition probability. $p(C)$ represents letter-based language model, and $p(Z)$ represents the state prior probability. CTC architecture incorporates letter-based language model. CTC architecture can also incorporate a word-based language model by using letter-to-word finite state transducer during decoding [18]. The CTC has the monotonic alignment property, i.e.,

when $z_{t-1} = c'_m$, then $z_t = c'_l$ where $l \geq m$.

Monotonic alignment property is an important constraint for speech recognition, so ASR sequence-to-sequence mapping should follow the monotonic alignment. This property is also satisfied by HMM/DNN.

3.3.3. Objective function

The posterior, $p(C|X)$, is represented as

$$p(C|X) \approx \underbrace{\sum\nolimits_z \prod\nolimits_{t=1}^{T} p(z_t|z_{t-1}, C)p(z_t|X)}_{\triangleq p_{ctc}(C/X)} . \frac{p(C)}{p(Z)} \tag{39}$$

Viterbi method and forward-backward algorithm are dynamic programming algorithm which is used to efficiently compute the summation over all possible Z. CTC objective function $p_{CTC}(C|X)$ is designed by excluding the $p(C)/p(Z)$ from Eq. (23).

The CTC formulation is also same as HMM/DNN. The minute difference is that Bayes' rule is applied to $p(C|Z)$ instead of $p(W|X)$. It has also three distribution components like HMM/DNN, i.e., framewise posterior distribution, $p(z_t|X)$; transition probability, $p(z_t|z_{t-1}, C)$; and letter model, $p(C)$. It also uses Markov assumption. It does not fully utilize the benefit of end-to-end ASR, but its character output representation still possesses the end-to-end benefits.

4. Convolutional neural networks

CNNs are the popular variants of deep learning that are widely adopted in ASR systems. CNNs have many attractive advancements, i.e., weight sharing, convolutional filters, and pooling. Therefore, CNNs have achieved an impressive performance in ASR. CNNs are composed of multiple convolutional layers. **Figure 5** shows the block diagram of CNN. LeCun and Bengio [41] describe the three states of convolutional layer, i.e., convolution, pooling, and nonlinearity.

Deep CNNs set a new milestone by achieving approximate human level performance through advanced architectures and optimized training [42]. CNNs use nonlinear function to directly process the low-level data. CNNs are capable of learning high-level features with high complexity and abstraction. Pooling is the heart of CNNs that reduces the dimensionality of a feature map. Maxout is widely used nonlinearity and has shown its effectiveness in ASR tasks [43, 44].

Pooling is an important concept that transforms the joint feature representation into the valuable information by keeping the useful information and eliminating insignificant information. Small frequency shifts that are common in speech signal are efficiently handled using pooling. Pooling also helps in reducing the spectral variance present in the input speech. It maps the input from p adjacent units into the output by applying a special function. After the element-wise nonlinearities, the features are passed through pooling layer. This layer executes the downsampling on the feature maps coming from previous layer and produces the new feature maps with a condensed resolution. This layer drastically reduces the spatial dimension of input. It serves the two main purposes. The first is that the amount of parameters or weight is reduced by 65%, thus lessening the computational cost. The second is that it controls the overfitting. This term refers to when a model is so tuned to the training examples.

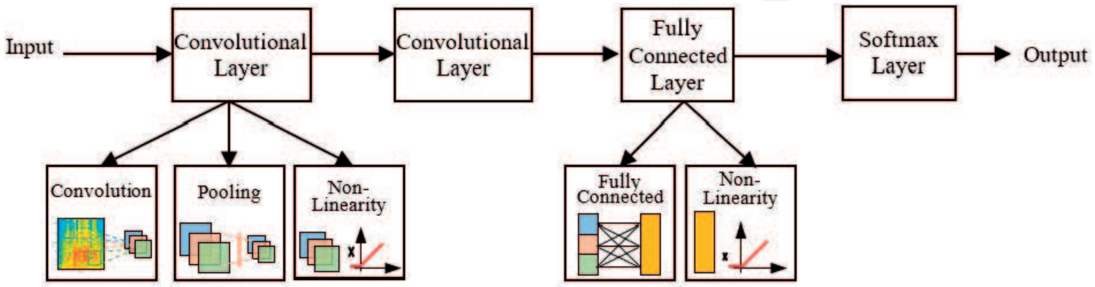

Figure 5. Block diagram of convolutional neural network.

5. CNN-based end-to-end approach

A novel acoustic model based on CNN is proposed by Palaz et al. [5] which is shown in **Figure 6**. In this, raw speech signal is segmented into input speech signal $s_t^c = \{s_{t-c}, \dots, s_t, \dots, s_{t+c}\}$ in the context of 2c frames having spanning window w_{in} milliseconds. First convolutional layer learns the useful features from the raw speech signal, and remaining convolutional layers further process these features into the useful information. After processing the speech signal, CNN estimates the class conditional probability, i.e., $P(i/s_t^c)$, which is used to calculate emission scaled-likelihood $P(s_t^c/i)$. Several filter stages are present in the network before the classification stage. A filter stage is a combination of convolutional layer, pooling layer, and a nonlinearity. The joint training of feature stage and classifier stage is performed using the back-propagation algorithm.

The end-to-end approach employs the following understanding:

1. Speech signals are non-stationary in nature. Therefore, they are processed in a short-term manner. Traditional feature extraction methods generally use 20–40 ms sliding window size. Although in the end-to-end approach, short-term processing of signal is required. Therefore, the size of the short-term window is taken as hyperparameter which is automatically determined during training.

2. Feature extraction is a filter operation because its components like Fourier transform, discrete cosine transform, etc. are filtering operations. In traditional systems, filtering is applied on both frequency and time. So, this factor is also considered in building convolutional layer in end-to-end system. Therefore, the number of filter banks and their parameters are taken as hyperparameters that are automatically determined during training.

3. The short-term processing of speech signal spread the information across time. In traditional systems, this spread information is modeled by calculating temporal derivatives and contextual information. Therefore, intermediate representation is supplied to classifier and calculated by taking long time span of input speech signal. Therefore, w_{in}, the size of input window, is taken as hyperparameter, which is estimated during training.

The end-to-end model estimates $P(i/s_t^c)$ by processing the speech signal with minimal assumptions or prior knowledge.

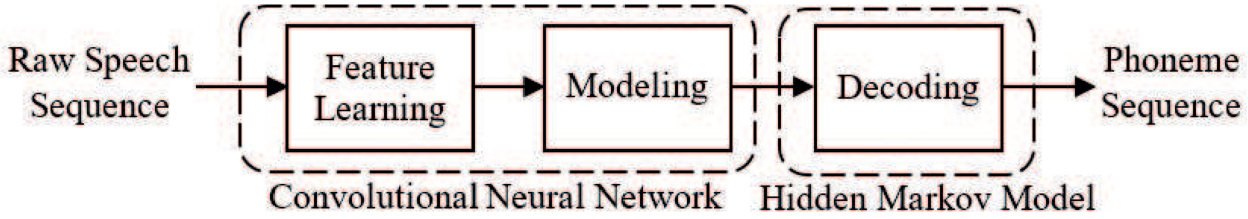

Figure 6. CNN-based raw speech phoneme recognition system.

6. Experimental results

In this model, a number of hyperparameters are used to specify the structure of the network. The number of hidden units in each hidden layer is very important; hence, it is taken as hyperparameter. w_{in} represents the time span of input speech signal. kW represents the kernel and temporal window width. dW represents the shift of temporal window. kW_{mp} represents max-pooling kernel width and dW_{mp} represents the shift of max-pooling kernel. The value of all hyperparameters is estimated during training based on frame-level classification accuracy on validation data. The range of hyperparameters after validation is shown in **Table 1**.

The experiments are conducted for three convolutional layers. The speech window size (w_{in} is taken 250 ms with a shift of temporal window (dW) 10 ms. **Table 2** shows the comparison of existing end-to-end speech recognition model in the context of PER. The results of the experiments conducted on TIMIT dataset for this model are compared with already existing techniques, and it is shown in **Table 3**. The main advantages of this model are that it uses only few parameters and offers better performance. It also increases the generalization capability of the classifiers.

Hyperparameter	Units	Range
Input window size (w_{in})	ms	100–700
Kernel width of the first ConvNet layer (kW_1)	Samples	10–90
Kernel width of the nth ConvNet layer (kW_n)	Samples	1–11
Number of filters per kernel (d_{outt})	Filters	20–100
Max-pooling kernel width (kW_{mp})	Frames	2–6
Number of hidden units in the classifier	Units	200–1500

Table 1. Range of hyperparameter for TIMIT dataset during validation.

End-to-end speech recognition model	PER (%)
CNN-based speech recognition system using raw speech as input [7]	33.2
Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks [36]	32.4
Convolutional neural network-based continuous speech recognition using raw speech signal [6]	32.3
End-to-end phoneme sequence recognition using convolutional neural networks [5]	27.2
CNN-based direct raw speech model	**21.9**
End-to-end continuous speech recognition using attention-based recurrent NN: First results [19]	18.57
Toward end-to-end speech recognition with deep convolutional neural networks [44]	18.2
Attention-based models for speech recognition [20]	17.6
Segmental recurrent neural networks for end-to-end speech recognition [45]	17.3

Bold value and text represent the performance of the CNN-based direct raw speech model.

Table 2. Comparison of existing end-to-end speech model in the context of PER (%).

Methods	PER (%)
GMM-/HMM-based ASR system [46]	34
CNN-based direct raw speech model	**21.9**
Attention-based models for speech recognition [20]	17.6
Segmental recurrent neural networks for end-to-end speech recognition [45]	17.3
Combining time and frequency domain convolution in convolutional neural network-Based phone recognition [47]	16.7
Phone recognition with hierarchical convolutional deep maxout networks [48]	16.5

Bold value and text represent the performance of the CNN-based direct raw speech model.

Table 3. Comparison of existing techniques with CNN-based direct raw speech model in the context of PER (%).

7. Conclusion

This chapter discusses the CNN-based direct raw speech recognition model. This model directly learns the relevant representation from the speech signal in a data-driven manner and calculates the conditional probability for each phoneme class. In this, CNN as an acoustic model consists of a feature stage and classifier stage. Both the stages are trained jointly. Raw speech is supplied as input to first convolutional layer, and it is further processed by several convolutional layers. Classifiers like ANN, CRF, MLP, or fully connected layers calculate the conditional probabilities for each phoneme class. After that decoding is performed using HMM. This model shows the similar performance as shown by MFCC-based conventional mode.

Author details

Vishal Passricha and Rajesh Kumar Aggarwal*

*Address all correspondence to: rka15969@gmail.com

National Institute of Technology, Kurukshetra, India

References

[1] Rabiner LR, Juang B-H. Fundamentals of Speech Recognition. Englewood Cliffs: PTR Prentice Hall; 1993

[2] Davis SB, Mermelstein P. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. Readings in Speech Recognition. Elsevier; 1990. pp. 65-74

[3] Hermansky H. Perceptual linear predictive (PLP) analysis of speech. The Journal of the Acoustical Society of America. 1990;**87**(4):1738-1752

[4] Chorowski J, Jaitly N. Towards better decoding and language model integration in sequence to sequence models. 2016. arXiv preprint arXiv:161202695

[5] Palaz D, Collobert R, Doss MM. End-to-end phoneme sequence recognition using convolutional neural networks. 2013. arXiv preprint arXiv:13122137

[6] Palaz D, Doss MM, Collobert R. Convolutional neural networks-based continuous speech recognition using raw speech signal. In: Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE; 2015

[7] Palaz D, Collobert R. Analysis of CNN-Based Speech Recognition System Using Raw Speech as Input. In Proceeding of Interspeech 2015 (No. EPFL-Conf-210029); 2015

[8] O'Shaughnessy D. Automatic speech recognition: History, methods and challenges. Pattern Recognition. 2008;**41**(10):2965-2979

[9] Dahl GE, Yu D, Deng L, Acero A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech and Language Processing. 2012;**20**(1):30-42

[10] Hinton G, Deng L, Yu D, Dahl GE, Mohamed AR, Jaitly N, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine. 2012;**29**(6):82-97

[11] Seide F, Li G, Yu D. Conversational speech transcription using context-dependent deep neural networks. In: Twelfth Annual Conference of the International Speech Communication Association; 2011

[12] Abdel-Hamid O, Mohamed AR, Jiang H, Penn G. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); March 2012. IEEE; 2012

[13] Senior A, Heigold G, Bacchiani M, Liao H. GMM-free DNN acoustic model training. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE; 2014

[14] Bacchiani M, Senior A, Heigold G. Asynchronous, online, GMM-free training of a context dependent acoustic model for speech recognition. In: Fifteenth Annual Conference of the International Speech Communication Association; 2014

[15] Gales M, Young S. The application of hidden Markov models in speech recognition. Foundations and Trends® in Signal Processing. 2008;**1**(3):195-304

[16] Graves A, Jaitly N. Towards end-to-end speech recognition with recurrent neural networks. In: International Conference on Machine Learning; 2014

[17] Hannun A, Case C, Casper J, Catanzaro B, Diamos G, Elsen E, et al. Deepspeech: Scaling up end-to-end speech recognition. 2014. arXiv preprint. arXiv preprint arXiv:14125567

[18] Miao Y, Gowayyed M, Metze F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In: Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on. IEEE; 2015

[19] Chorowski J, Bahdanau D, Cho K, Bengio Y. End-to-end continuous speech recognition using attention-based recurrent NN: First results. 2014. arXiv preprint arXiv:14121602

[20] Chorowski JK, Bahdanau D, Serdyuk D, Cho K, Bengio Y. Attention-based models for speech recognition. In: Advances in Neural Information Processing Systems. 2015

[21] Chan W, Jaitly N, Le Q, Vinyals O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In: Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE; 2016

[22] Bahdanau D, Chorowski J, Serdyuk D, Brakel P, Bengio Y. End-to-end attention-based large vocabulary speech recognition. In: Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE; 2016

[23] Lu L, Zhang X, Renais S. On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition. In: Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE; 2016

[24] Chan W, Lane I. On online attention-based speech recognition and joint Mandarin Character-Pinyin training. In: INTERSPEECH; 2016

[25] Graves A, Fernández S, Gomez F, Schmidhuber J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning; ACM; 2006

[26] Golik P, Tüske Z, Schlüter R, Ney H. Convolutional neural networks for acoustic modeling of raw time signal in LVCSR. In: Sixteenth Annual Conference of the International Speech Communication Association. 2015

[27] Graves A. Generating sequences with recurrent neural networks. 2013. arXiv preprint arXiv:13080850

[28] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. 2014. arXiv preprint arXiv:14090473

[29] Mnih V, Heess N, Graves A. Recurrent models of visual attention. In: Advances in Neural Information Processing Systems; 2014

[30] Graves A, Jaitly N, Mohamed AR. Hybrid speech recognition with deep bidirectional LSTM. In: Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE; 2013

[31] Hannun AY, Maas AL, Jurafsky D, Ng AY. First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs. 2014. arXiv preprint arXiv:14082873

[32] Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing; May 2013. IEEE; 2013

[33] Maas A, Xie Z, Jurafsky D, Ng A. Lexicon-free conversational speech recognition with neural networks. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics. Human Language Technologies; 2015

[34] Jaitly N, Hinton G. Learning a better representation of speech soundwaves using restricted boltzmann machines. In: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE; 2011

[35] LeCun Y. Generalization and network design strategies. Connectionism in Perspective. 1989:143-155

[36] Palaz D, Collobert R, Doss MM. Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks. 2013. arXiv preprint arXiv:13041018

[37] Rabiner LR, Juang B-H. Speech recognition: Statistical methods. Encyclopedia of Linguistics. 2006:1-18

[38] Bourlard HA, Morgan N. Connectionist Speech Recognition: A Hybrid Approach. Vol. 247. Springer Science & Business Media; 2012

[39] Mikolov T, Karafiát M, Burget L, Černocký J, Khudanpur S. Recurrent neural network based language model. In: Eleventh Annual Conference of the International Speech Communication Association; 2010

[40] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation. 1997;**9**(8): 1735-1780

[41] LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series. The Handbook of Brain Theory and Neural Networks. 1995;**3361**(10):1995

[42] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems; 2012

[43] Zhang X, Trmal J, Povey D, Khudanpur S. Improving deep neural network acoustic models using generalized maxout networks. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE; 2014

[44] Zhang Y, Pezeshki M, Brakel P, Zhang S, Bengio CLY, Courville A. Towards end-to-end speech recognition with deep convolutional neural networks. 2017. arXiv preprint arXiv: 170102720

[45] Lu L, Kong L, Dyer C, Smith NA, Renals S. Segmental recurrent neural networks for end-to-end speech recognition. In: INTERSPEECH 2016; 8 September 2016. ISCA; 2016

[46] Fauziya F, Nijhawan G. A Comparative study of phoneme recognition using GMM-HMM and ANN based acoustic modeling. International Journal of Computer Applications. 2014:12-16

[47] Toth L. Combining time- and frequency-domain convolution in convolutional neural network-based phone recognition. In: Acoustics, Speech and Signal Processing (ICASSP). 2014 IEEE International Conference on May 4 2014. IEEE; pp. 190-194

[48] Toth L. Phone recognition with hierarchical convolutional deep maxout networks. EURASIP Journal on Audio, Speech, and Music Processing. 2015;**2015**(1):25

IntechOpen

IntechOpen