ISSN: 2393-8528

IJICSE
Open Access

Contents lists available at www.ijicse.in

International Journal of Innovative Computer Science & Engineering

Volume 5 Issue 4; July-August 2018; Page No. 01-11

Speech Emotion Recognition using Convolutional Neural Network with Recurrent Neural Network Architecture, End to End architecture, Real time Recognition And Noise Robust approach for Recognition.

Manav Mehra[1], Shreya Dwivedi[2]

[1]B.M.S College of Engineering

bm15cs055@bmsce.ac.in

[2]S.R.M Institute of Science and Technology.

shreyadwivedi15@gmail.com

ABSTRACT

This article reviews the different approaches that are available for the process of emotion recognition using speech. Firstly, we talk about the problems in emotion recognition such as choosing an appropriate speech database, then identifying various features and finally selecting an appropriate classification model. the model reviewed in this approach uses 13 MFCC and acceleration constants as features, a CNN and a Long Short Term Memory for classification. The second approach reviewed in this article uses an approach that is based on adaptively trained Very Deep Convolutional Residual network that highlights Cluster adaptive Training and Factor aware Training. The third approach overcomes the problems of losing some speech recognition information while extracting the feature first and then classifying the emotion hence resulting in reduced accuracy by using the end-to-end recognition approach. Finally it reviews the process of real-time speech recognition investigating high level descriptors and i-vectors.

Keywords: Neural networks, Speech Emotion Recognition, MFCC, CNN, LSTM, residual learning, high-level descriptors, i-vectors.

INTRODUCTION

Emotion recognition has been one of the most important research directions in the field of artificial intelligence. Professor Minsky presented the notion- "Allow the computer to have the emotional capacity". In the early 90s, The Technology Media Lab of Massachusetts Institute fabricated an "emotional Editor" to detect various emotional signals. It applied body's physiological signals, faces preliminary recognition signal, voice signals to recognize a diversity of emotions and make fitting simple reaction respectively. Emotion recognition from speech has prominent applications in the speech-processing systems. The basics of emotion recognition can be summarized into three stages: feature extraction, feature selection, and emotion recognition. The steps towards building of an emotion recognition system are, an emotional speech corpora is implemented then emotion specific features are extracted from those speeches and in a concluding step a classification model is used to recognize the emotions. Choice of features and size of the database plays a vital role in recognition scheme.

In several aspects, the prosodic features derived from parameters like, loudness, energy contours, and speaking rate have been used. Better performance has been attained by using short-term acoustic features, such as

Corresponding author: Manav Mehra

mel-frequency cepstral coefficients (MFCCs), logarithm frequency power coefficients (LFPCs), and modulation spectral features (MSFs). Recently, most of the emotion recognition models have been based on the maximum likelihood Bayes (MLB) and linear discriminate classification (LDC). However, artificial neural networks (ANNs) , support vector machines (SVMs) , decision trees, K-nearest neighbor (KNN), Gaussian mixture models (GMMs), hidden Markov models (HMMs), and Bayesian networks have also been used for emotion recognition.

The main challenge of emotion recognition is different length of each speech, MFCC feature extraction method works in a sliding window method i.e it sets a 25ms frame over the speech signal and compute 13 cepstral coefficient from each frame (those are used as features). Now depending upon various length MFCC returns different number of frames. As a result we have different number of features which is not acceptable. Thus a preprocessing step to make each speech signal of equal length is done. We have used CNN-LSTM architecture as our classifier. Output from CNN is treated as a input of LSTM network.This is an end to end technique of emotion recognition . Later in this paper we have also introduced end-to-end speech emotion recognition system which is based on neural network without feature extraction and real time technique of emotion detection for achieving a higher performance in this two fixed-dimensional speech representations are introduced:

high-level descriptors and i-vectors.Both approaches are proposed to convert variable length speech utterances into fixed-dimensional vectors with a noise robust approach.

MODEL

I. Convolutional Neural Network with RNN Architecture:

Feature Extraction - We have used mel frequency cepstral coefficient (MFCC)

method for feature extraction. Proper choice of features play a very important part for emotion recognition.

Mel Frequency Cepstral Coefficient: Mel frequency cepstral coefficients are computed on the basis of human hearing ability. In Mel frequency cepstral coefficients (MFCC) method, two types of filter are used. Some filter are spaced linearly at low frequency below 1 kHz and other are spaced logarithmically at high frequency above 1 kHz. MFCC feature extraction process consists of a few steps as discussed below,

Pre-emphasis: Pre-emphasis is required to increase signal energy. In this process, speech signal is passed through a filter which increase the energy of signal. This increment of energy level gives more information.

Framing: In this process, speech sample is segmented into 20-40 ms frames. The length of human voice may vary, so for fixing the size of speech this processes is necessary. Although the speech signal is non-stationary in nature (i.e.frequency can be changed over the time period), but for a short duration of time, signal behave like a stationary signal.

Windowing: After framing process, the windowing process is performed. Windowing function reduce the signal discontinuities at the start and end of each frame. In this process, frame is shifted with a 10ms span. That means each frame contains some overlapping portion of previous frame.

Fast Fourier Transform (FFT): FFT is used to generate the frequency spectrum of each frame. Each sample of each frame converted from time domain to frequency domain by the FFT. FFT is used to find all frequencies present in the particular frame.

Mel scale filter bank: This is a set of 20-30 triangular filters applied to each frame. The mel scale filter bank identify how much energy exists in a particular frame. The mathematical equation to convert the normal frequency f to the Mel scale m is as follows,

$$m = 2595 \log\left(1 + \frac{f}{700}\right)$$

Log energy computation: After getting the filter bank energy of each frame, log function is applied to them. It is also inspired by human hearing perception. Human does not listen loud volume on a linear scale. If the volume of the sound is high, human ear can not recognize large variations in energy. Log energy computation gives those features for which human can listen clearly.

Discrete Cosine Transformation (DCT): In the final step DCT is calculated of the

log filter bank energies. We have used 25ms frames with 10ms of sliding. We have also used 26 band pass filters. From each frame we computed 13 MFCC features.

We have also calculated energy within a frame. After getting 13 MFCC features, we

also computed 13 velocity components and 13 acceleration components by

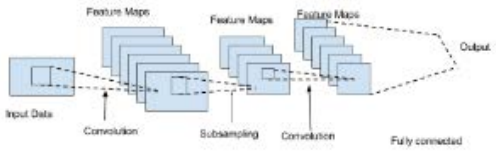
calculating time derivatives of energy and MFCC.

$$\Delta C_m(t) = \sum_{\tau=-M}^{M} \tau C_m(t+\tau) / \sum_{\tau=-M}^{M} \tau^2$$

Classifier - A classification system is an approach to set each speech to a proper emotion class according to the extracted features from speech. There are different classifiers available for emotion recognition. We have used one dimension CNN with LSTM for classification.

CNN:

Convolutional neural network or CNN consist of several layer of convolution. In CNN, nonlinear activation function for example rectified linear unit (ReLU) or Sigmoid function are used to the result. In neural networks nodes of input layer are connected with the nodes of hidden layer and those hidden layer nodes are fully connected with nodes of output layer. In CNN, convolution are applied on input layer to generate the output. Each part of input are convoluted by different filters and combining them we get the final result. In CNN, there is a pooling layer and the purpose of these layer is sub sampling the input from a specified filter. A common pooling technique is max pooling. In max pooling a maximum value is selected from each filter. Pooling layer reduce the size of input . Max pooling layer can also perform over a window instead of performing the whole matrix. After getting the output from CNN a neural network like Multilayer Perceptron (MLP) or Recurrent Neural Network (RNN) or Long Short Term Memory (LSTM) network can be used for training.



LSTM:

While classifying a set of temporal data, Recurrent Neural Network (RNN) architectures have outperformed other classifiers in temporal data classification.RNNs exploit the temporal relations present in a sequence of data, thus will be very effective for speech signals. Unfortunately though the working principle of RNN is very promising it has a major drawback. It is referred as problem of Vanishing Gradient. It happens because RNN can unfold itself quite deep in time (depending on the length of input vector). If the input sequence is very long (which is a common scenario for emotion recognition using speech) RNN does not produce satisfactory result. A solution to this problem is to keep the error term always 1. This is known as Constant Error Carousel (CEC). This can be implemented by a variation of simple RNN node known as Long Short Term Memory (LSTM). The aim of using this new node structure is to use CEC by the help of different gate units as shown in Fig.

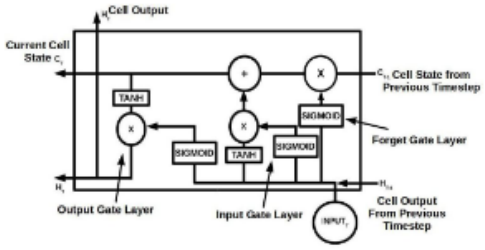


Fig. LSTM Node as a memory cell Unlike RNN one Node of the LSTM network is known as Memory Cell. Using the gates enables one to remove or add information to the cell state. This incorporates the ability to remember or forget information about states in a time frame that appeared long back in past. The cell states and cell output at T th time step are represented by C T and H forget gate layer resets the states of the cell. Input gate layer decides how much of the input will affect the current cell state. The output gate layer decides how much output will affect the rest of the network. In this way LSTM eliminates the problem of Vanishing Error Gradient. This improvisation enables us to use Recurrent Neural Network (with LSTM) to train even longer sequences.

II. End to End:

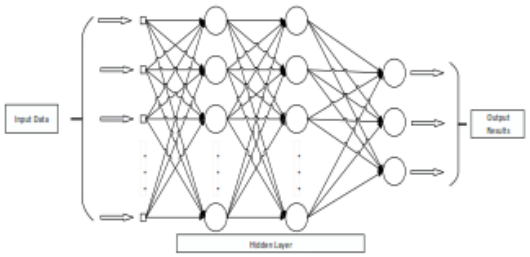
Neural network arch -

Artificial neural network (ANN) is a non-linear dynamic system, whose distinguishing feature is the distributed memory and parallel processing of information. The network system consists of a large number of neurons connections with adjustable connection weights. There are features such as massively parallel processing, distributed information storage, good self-organized learning ability and so on. Back propagation (BP) is a supervised learning algorithms. It can do approximation of arbitrary functions in theory. And it has a strong nonlinear map ability with basic structure by nonlinear element. The layer's number, units' number, learning factors and other parameters can be set according to the specific circumstances. It has broad application prospects in optimization, signal processing and pattern recognition, intelligent control, fault diagnosis and many other areas. BP algorithm consists of two processes: Forward Propagation: signal input from input layer upon hidden layer, spread to the output layer. Back Propagation: the errors from the output back to the input layer, through gradient descent algorithm to adjust the weights and bias values.

$$h^{(l)} = g\left(W^{(l)}h^{(l-1)} + b^{(l)}\right) \quad (1)$$

$$y_k = P(k|x) = \frac{exp\left(W_k^{(l)}h^{(l-1)} + b_k^{(l)}\right)}{\sum_j exp\left(W_j^{(l)}h^{(l-1)} + b_j^{(l)}\right)} \quad (2)$$

In "(1)", □(□) stands for any hidden l and their ID, g stands for activate function, □(□) expresses the weight matrix of this layer that needs to be updated, □(□) indicates the offset of the hidden layer. We use standard SoftMax classifier in the output ("(2)"), where x represents the input. The overall architecture is shown in Figure.



Optimization:

The traditional neural network often has the problem of low fitting rate, low feedback efficiency and poor network convergence. We improve the network performance through interlayer optimization and the use of optimized back propagation algorithm.

A. Layer Optimization The network will be overfitting when layer number or node number is too large or the network be trained too much for classic artificial neural networks. In order to ensure the training effect and prevent overfitting, we use Dropout optimized algorithm. Dropout means during the training period t some of the hidden layer nodes randomly do not work. Those who do not work can be temporarily considered not part of the network structure, but its weight is retained and it can work when the next sample input. There is no guarantee that every 2 hidden nodes will appear at the same time when you train the network to updates

weights. For the reason that the renewal of the weights no longer depends on the co-operation of the implied nodes with fixed relationships, and also prevents some features from being effective only under other specific characteristics. Dropout can be considered as a model of the average. For each sample input into the network, the corresponding network structure is different, but all of these different networks structure share the weight of the hidden node. On the other hand, the activation function of the Sigmoid (Logistic-Sigmoid, Tanh-Sigmoid) is regarded as the core in the traditional neural network. From a mathematical viewpoint, nonlinear Sigmoid function has a good effect on the characteristics of the signal space mapping, because large signal gain on center area and small signal gain on both sides. Currently, a clear goal of deep learning is to isolate key factors from the data variable. The original data (mainly natural data) usually surround height-intensive feature. The reason is that these eigenvectors are interrelated, and a small key factor may entangle a bunch of features. However, if the complex relationships between features can be solved, they are converted to sparse features which are more likely to be linearly separable, or have less dependency on nonlinear mapping mechanisms. ReLu made the network can introduce their own sparseness. This practice is equivalent to pretraining without supervised learning. At this level, ReLu narrowed the gap between unsupervised learning and supervised learning.

B. Algorithm Optimization of Finding Optimum In the training process, it is difficult to guarantee the balance between the optimal speed and the authenticity when the gradient is found by BP to find the optimal point. Generally, the stochastic gradient descent method and the bulk gradient descent method are used: Bulk gradient drop (BGD) solution is as follows: Evaluating derivatives of J( ) for at back propagation can get each corresponds to gradient:

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}(y^i - h_\theta(x^i))x_j^i$$

Because it is a function to minimize risk, we get the gradient of each parameter to update each one:

$$\theta_j' = \theta_j + \frac{1}{m}\sum_{i=1}^{m}(y^i - h_\theta(x^i))x_j^i$$

Can be noted from the above formula, it got to be a global solution. But at each iteration step, it used to train all of the data, then the iteration speed will be very slow if m is large. Stochastic gradient descent (SGD) method: Loss function can be written as follows. The loss function corresponds to the size of each sample of training set, but batch gradient descent corresponds to all training examples:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}(y^i - h_\theta(x^i))^2$$

Evaluating loss function for each sample can get the derivative corresponding gradient to update :

$$\theta_j' = \theta_j + (y^i - h_\theta(x^i))x_j^i$$

Stochastic gradient descent is updated by training each sample every iteration. If the sample size is large (for example, hundreds of thousands), then you may only use thousands to the optimal solution. However, stochastic gradient descent has more noises than batch gradient descent method. Stochastic gradient descent won't toward to the right direction of overall optimization each iteration. In order to ensure the speed and effectiveness of the algorithm at the same time, we use a small batch of random gradient drop hair. Each input is a collection from the entire input set of random collection. Define the adjustable constant C as the number of inputs for each iteration.

III. Real Time:

Fixed-Dimensional Speech Representations - Two fixed-dimensional speech representations are introduced . Both of two approaches are proposed to convert variable length speech utterances into fixed-dimensional vectors.

A. High-level Descriptors High-level descriptors (HLDs), also called as function-

als, are the statistical features derived from LLDs, which are the high-level features . Because HLDs are related to LLDs, we simply introduce the LLDs firstly. LLD is defined as a parameter computed from a short time frame of an speech signal at time t. Therefore, LLDs are at frame level, which are estimated frame-byframe. Reference summarizes fifteen types of LLDs. They are time domain descriptors, energy, spectrum, spectral descriptors, autocorrelation, cepstrum, linear prediction, formants, perceptual linear prediction, cepstral features, pitch, F0 harmonics, voice quality, tonal features, and non-linear vocal tract model features. These features are demonstrated the relation with emotion. For example, reference show that the overall speech energy and energy distribution across the frequency spectrum are affected by the emotional arousal state of the speaker. The acoustic intensities of happiness and anger become higher while that of sadness and disgust turn to be lower. However, LLDs are frame-based features, which is varied according to the segment length. Some machine learning tools are only suitable for vector space features. HLDs are proposed to avoid the dependency of segment length. Specifically, the statistical function is used to map each type of LLDs into a single value. Assume that we have 5 types of LLDs and use mean as the statistical function, we will obtain a mean value for each type of LLDs. These 5 types of LLDs are converted into a 5-dimensional HLDs vector. Commonly used statistical functions are means, moments, extreme values, percentiles, etc. Because there are many types of LLDs and statistical functions, the dimension of HLDs become large after the combination of different types of LLDs and statistical functions.

B. I-vectors The i-vector approach was originally proposed for speaker recognition. The idea of i-vector is based on the theory of joint factor analysis (JFA). Instead of modeling the speaker- and channel-spaces separately, it model the total variability space to represent all possible variability. The benefits of i-vector approach are mapping the variable-duration utterances into low dimensional vectors and including all possible variability such as speaker and emotion variation. Specifically, given an utterance of emotion e, the emotion-dependent GMM-supervector m is written as:

$$m_e = m + \mathbf{T}\mathbf{w}_e$$

where m is the GMM-supervector of the universal background model (UBM) which is emotion-independent, T is a low-rank total variability matrix, and the posterior mean of the latent factor w is defined as i-vector. The posterior mean of latent factor w e is given by:

$$\phi_e = \mathbf{L}_e^{-1}\mathbf{T}^{\mathsf{T}}\boldsymbol{\Sigma}^{(b)^{-1}}\tilde{\mathbf{F}}_e \qquad (2)$$

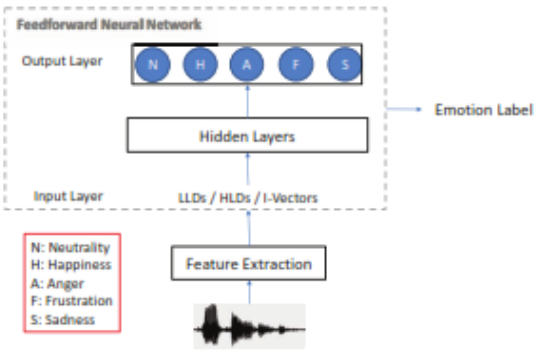
where

$$\mathbf{L}_e = \mathbf{I} + \mathbf{T}^{\mathsf{T}}\boldsymbol{\Sigma}^{(b)^{-1}}\mathbf{N}_e\mathbf{T} \qquad (3)$$

is a precision matrix and I is the identity matrix. N is zero-order statistics. ˜ is centered first-order BaumWelch statistics. Σ (b) F e is a covariance matrix modeling the residual variability not captured by the total variability matrix T. In practice, we substitute this matrix by the covariance matrices of the UBM. The posterior mean ϕ (Eq. 2) is the i-vector representing the emotion.

C. Neural Network Based Speech Emotion Recognition

Figure



shows the framework of speech emotion recognition in this paper. It is mainly composed by two important modules: feature extraction and classifier. For feature extraction module, we explore three types of features: LLDs, HLDs, and i-vectors. This paper

adopts the feed-forward neural network (NN) as the classifier. It consists of input layer, several hidden layers, and output layer. Each layer contains a fixed number of nodes and has a linear transform and a nonlinear activation function. In this paper, we use sigmoid as nonlinear activation function for hidden layer. The input features are affine transformed by weight matrix and bias first, then go through the activation function to form the output of 1st hidden layer which is then forward further to the subsequent layers till the output layer of neural network. The number of hidden node in output layer is equal to the number of emotion classes (N). Each node is corresponding to one emotion class and training label varies from 1 to N. Then, the output of NN is normalized by output activation function – softmax to obtain class probabilities. The cross entropy between the true class labels and the output of the softmax is selected as the cost function of NN classifier. Five emotion classes are considered in this paper. They are neutrality, happiness, anger, frustration, and sadness. Given a test segment, LLDs/HLDs/I-Vectors are firstly extracted and then fed into neural network to obtain the posterior scores for five emotion class. The emotion class with the maximum posterior score will be considered as the emotion label of this test segment.

IV. Noise Robust:

Systems still performance in noisy environments is poor (i.e scenarios with additive noise) and a magnified degradation has been observed under the distant (far-field) talking condition. The low Signal to Noise ratio in these noisy conditions makes Deep Neural Networks more susceptible to the mismatch problem. Therefore noise robustness is still a critical problem for adoption of ASR real scenarios. A lot many technologies have been proposed to manage the difficulty of mismatch between training and testing in a noisy scenario.

Those methods can be grouped majorly into two categories 1. feature enhancement on the front-end (denoising or dereverberation), attempts to removing noise at the signal level 2. acoustic modeling with adaptation on the back-end. Adaptation methods update the model parameters to better fit the unseen condition rather than denoise the features. Many techniques have been developed to adapt Deep neural networks.For instance , transformation based adaptation is an indispensable category in DNN adaptation, e.g., linear input network (LIN) ,feature discriminative linear regression (fDLR), and linear output network (LON) . The transformation could also be employed at a non-linear layer such as in LHUC . Another way of implementation is factor aware training, in which auxiliary features representing the non-speech variability are incorporated into DNNs. Features like i-vectors , environment features, speaker codes and bottleneck are used]. Cluster adaptive training (CAT) has also been developed for Deep Neural Networks: bases are fabricated for DNNs to represent non-speech characteristics.

We focus on the technologies on the back-end to improve the robustness of ASR systems. These technologies can be divided into two approaches. The first one is exploring more robust acoustic models, which can inherently limit the mismatch between training and testing. The second one is model adaptation, constructed based on the new model structure, which can further improve the system performance in noisy conditions. For the more robust acoustic model, we focus on the convolutional neural network, which has been explored for acoustic model and yields a lower word error rate (WER) than standard fully connected feed-forward DNNs in many tasks],. Recently have designed very deep CNNs for speech recognition and gotten a significant WER reduction on telephone speech. Furthermore, in, our previous works have shown that VDCNNs particularly show noise robustness superior to other models in noisy scenarios and have also revealed some natural properties of VDCNNs. In this work, we introduce batch normalization and residual learning into our previous VDCNN structure. We discover that using these

methods can further improve model robustness and reduce the mismatch in noisy scenarios. Various design aspects of the architecture are investigated in detail for the noisy scenarios. In addition, some new adaptation techniques are developed based on this new very deep convolutional residual network (VDCRN). The first one is factor aware training (FAT). Typically, the auxiliary vector is concatenated with the input feature in DNNs or RNNs. Considering the different properties of the normal spectrum feature (e.g., FBANK) and auxiliary feature (e.g., i-vector), a parallel joint-learning structure is usually designed when applying factor aware training in CNN. In this work, a unified framework is proposed: an adaptation neural network is learned to convert auxiliary features to a factor specific bias vector. Then, this specific bias vector could be added to any layers of a VDCRN including convolution layer and fully connected layer. The second technique developed in this paper is cluster adaptive training (CAT). We previously implemented CAT for DNN, whereas in this work we further explore a similar idea to use the filter or feature map as the basis for doing the CAT for VDCRN. Moreover, a factorized CAT structure is designed to incorporate multiple sources of nonspeech variability into one complete model. A comprehensive investigation and an in-depth analysis of all those technologies are performed in this paper. Experimental results on the noisy Aurora4, CHiME-4 and AMI meeting transcription tasks show that applying the proposed techniques can obtain promising per- formance improvements. The remainder of this paper is organized as follows. In Section II the conventional CNN-HMM hybrid system and structure of the VDCNN are first revisited, then the new very deep convolutional residual network (VDCRN), which shows better noise robustness, is presented. In Section III, we introduce factor aware training and cluster adaptive training based on VDCRN to alleviate the mismatch between the training and testing.

VERY DEEP CONVOLUTIONAL RESIDUAL NETWORK

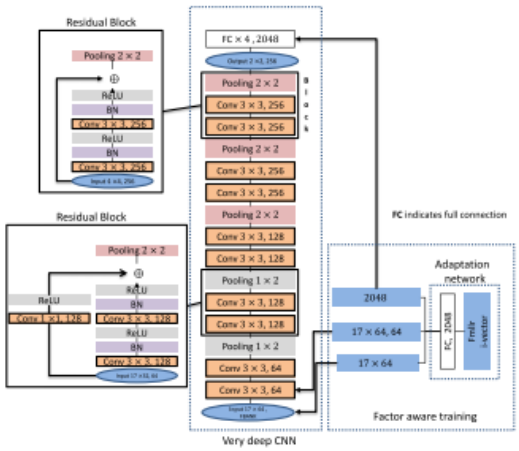
A. Convolutional Neural Network The convolutional neural network (CNN) has shown better performance than the traditional DNN in many speech recognition tasks. Recall that the inputs and outputs of each convolutional layer are several feature maps. Each feature map is a two-dimensional matrix. In speech recognition, the feature map at the input layer is a time-frequency map. Convolution is an operation that applies a filter to the feature map. The result of a convolution operation is still a feature map. We use $\otimes$ to represent it. A convolution layer consists of #outchannel × #inchannel filters. The i-th output feature map of layer l is given by

$$\mathbf{o}_i^l = \sigma\left(\sum_{j=1}^{N} W_{i,j}^l \otimes \mathbf{o}_j^{l-1} \oplus b_i^l\right)$$

where o l i and o l−1 j are feature maps in the current layer l and previous layer l − 1 respectively. W is the filter between input feature map j and output feature map i at layer l. b l i,j is a bias applied to the whole feature map, $\oplus$ indicates each element in the feature map plus the same scalar b activation function, which is typically a sigmoid or ReLU. N is the number of output feature maps. A pooling layer is a layer that performs down-sampling on the feature maps of the previous layer. In this work, max-pooling is used.

B. Very Deep Convolutional Neural Network Recently very deep CNNs, which have many more convolution layers, have been successfully used in speech recognition and particularly greatly outperformed DNN, RNN and shallow CNN models in noisy scenarios. Our previous work shows the following key principles for designing very deep CNNs for speech recognition: Rather than using a large filter as in shallow CNNs (9 × 9 or 3 × 4 in a smaller filter is used in a VDCNN. What's more, zero padding is used before the convolution operation so the feature map resolutions can be preserved. In this way, it is

possible to increase the number of convolutional layers. Compared to computer vision tasks, the size of the input feature map in speech recognition is relatively small. In addition to the adjustment of the size of filters and pooling, the input size needs to be enlarged to allow more convolution and pooling operations. So a 17 × 64 feature map is used as model input, i.e., a 17 frames context window with 64-dimension FBANK feature. For very deep CNNs, a pooling layer is added after at least two convolutional layers. The size of the output feature map at the top convolutional layer is relatively small, 2 × 2, which can reduce the model parameters. Moreover, to achieve a better trade-off within model complexity and size, the number of feature maps is increased gradually: which will only be doubled after some pooling layers. The full structure of the VDCNN is shown in the middle block of Fig. (enclosed with the blue dotted line). It contains 5 blocks separated by the pooling operation, and each block contains two convolutional layers and one pooling layer. There are 4 FC (fully connected) layers with 2048 nodes in each layer after the convolutional part. The model configuration, such as



C. Very Deep Convolutional Residual Network Based on the VDCNNs, further extensions are developed in this work to better train the model with increased depth (a similar idea is also explored in other recent works). Batch normalization (BN) and residual learning are mainly incorporated, which are motivated by the great success of ResNets in the image community. Batch Normalization: BN is an additional layer to normalize the mean and variance of a feature map within a mini-batch before the activation function is applied. After introducing BN, the gradients are well-behaved which allows the use of a larger learning rate and the training of deeper and more complicated networks. Furthermore, using BN can reduce the non-speech variability because it normalizes the activations over different frames. Residual Learning: Residual learning is proposed to ease the training of very deep neural networks. The key idea is to use a residual block (denoted as res-block for simplicity), which incorporates BNand skip connections across several convolutional layers. The very deep convolutional residual network used in this work is shown in Fig.. A residual block replaces each block in previous VDCNN (two convolutional layers with one pooling layer in VDCNNs). Two kinds of res-block are designed here, as shown in the left part of Fig.. Since the number of feature maps doubles in the first, second and fourth blocks, a convolutional layer with a 1 × 1 filter is applied in the corresponding skip connection to double the number of feature maps, as shown at the bottom left of Fig.. For the blocks with the same number of feature maps as the previous block, e.g.,the third and fifth block, a direct skip connection can be used, as shown at the top left of Fig. . In this work, we use this new structure for noise robust speech recognition, and we named our proposed model very deep convolutional residual network (VDCRN). The following experiments reveal that this new model particularly shows greater robustness than our previous VDCNN under noisy conditions, leading to a significant mismatch reduction within the clean and noisy data.

Conclusion

This paper gives a brief idea about the different approaches that can be employed for the purpose of Emotion recognition from speech. The technique that uses a combination of Convolutional Neural Network

and Long Short Term Memory provides an efficient mechanism for the said purpose and has shown promising results in performance. It can also be said that a larger data set enhances the performance of the network. The approach based on VDCRNs are more robust to noisy conditions and can be made adaptive using CAT and FAT, these adapted versions show significant improvement in performance. We also review the procedures incorporating high-level descriptors and i-vectors used for real time emotion recognition. The end-to-end emotion recognition approach uses a neural network to which the original voice is fed for independent learning.

References

1. Van Bezooijen R, Otto SA, Heenan TA. Recognition of vocal expressions of emotion: A three-nation study to identify universal characteristics. Journal of Cross-Cultural Psychology,1983,14(4):387−406.
2. Tolkmitt FJ, Scherer KR. Effect of experimentally induced stress on vocal parameters. Journal of Experimental Psychology Human Perception Performance, 1986,12(3):302−313.
3. Cahn JE. The generation of affect in synthesized speech. Journal of the American Voice Input/Output Society, 1990,8:1−19.
4. Yang B, Lugger M (2010) Emotion recognition from speech signals using new harmony features. Signal Process 90:1415–1423.
5. Lee C-C, Mower E, Busso C, Lee S, Narayanan S (2009) Emotion recognition using a hierarchical binary decision tree approach. In: The proceedings of Interspeech, pp 320–323.
6. Lo ´pez-Co ´zar R, Silovsky J, Kroul M (2011) Enhancement of
7. S. Basu, N. Jana, A. Bag, M. Mahadevappa, J. Mukherjee, S. Kumar, and R. Guha. Emotion recognition based on physiological signals using valence-arousal model. In Image Information Processing (ICIIP), 2015 Third International Conference on, pages 50–55. IEEE, 2015.
8. S. Basu, A. Bag, M. Mahadevappa, J. Mukherjee, and R. Guha. Affect detection in normal groups with the help of biological markers. In India Conference (INDICON), 2015 Annual IEEE, pages 1–6. IEEE, 2015.
9. R. Elbarougy and M. Akagi. Cross-lingual speech emotion recognition system based on a three-layer model for human perception. pages 1–10,2013.
10. S. Basu, A. Bag, M. Aftabuddin, Md. Mahadevappa, J. Mukherjee, and R. Guha. Effects of emotion on physiological signals. In 2016 IEEE Annual India Conference (INDICON), pages 1–6, Dec 2016.
11. M. El Ayadi, M. S. Kamel, and F. Karray, “Survey on speech emotion recognition: Features, classification schemes, and databases,” Pattern Recognition, vol. 44, no. 3, pp. 572–587, 2011.
12. C. N. Anagnostopoulos, T. Iliou, and I. Giannoukos, “Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011,” Artificial Intelligence Review, pp. 1–23, 2012.
13. F. Eyben, Real-Time Speech and Music Classification by Large Audio Feature Space Extraction. Springer, 2016.
14. N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” IEEE Trans. on Audio, Speech, and Language Processing, vol. 19, no. 4, pp. 788–798, May 2011.
15. G. Hinton et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal Process. Mag., vol. 29, no. 6, pp. 82–97, Nov. 2012.
16. F. Seide, G. Li, X. Chen, and D. Yu, “Feature engineering in context dependent deep neural networks for conversational speech transcription,” in Proc. IEEE Automat. Speech Recognit. Understanding Workshop, 2011, pp. 24–29.

17. F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in Proc. INTERSPEECH, 2011, pp. 437–440.
18. G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pretrained deep neural networks for large-vocabulary speech recognition," IEEE Trans. Audio, Speech, Lang., Process., vol. 20, no. 1, pp. 30–42, Jan. 2012.
19. Y. Wang and M. J. Gales, "Speaker and noise factorization for robust speech recognition," IEEE Trans. Audio, Speech, Lang., Process., vol. 20, no. 7, pp. 2149–2158, Sep. 2012.
20. T. Hain et al., "Transcribing meetings with the AMIDA systems," IEEE Trans. Audio, Speech, Lang., Process., vol. 20, no. 2, pp. 486–498, Feb. 2012.
21. Adaptive Very Deep Convolutional Residual Network for Noise Robust Speech Recognition Tian Tan , Student Member, IEEE,YanminQian , Member, IEEE, Hu Hu, Ying Zhou ,WenDing , and Kai Yu, Senior Member, IEEEEnd-to-End Speech Emotion Recognition Based on Neural Network Bing Zhu, Wenkai Zhou
22. 2017 17th IEEE International Conference on Communication Technology Yutian Wang, Hui Wang,Juan Juan Cai
23. Proceedings of the 2nd International Conference on Communication and Electronics Systems (ICCES 2017) IEEE Xplore Compliant - Part Number:CFP17AWO-ART, ISBN:978-1-5090-5013-0 Emotion Recognition from Speech using Convolutional Neural Network with Recurrent Neural Network Architecture Saikat Basu Jaybrata Chakraborty ,Md. Aftabuddin
24. S. Xue, O. Abdel-Hamid, H. Jiang, and L. Dai, "Direct adaptation of hybrid DNN/HMM model for fast speaker adaptation in LVCSR based on Speaker Code," in Proc. Int. Conf. Acoust., Speech, Signal Process., 2014, pp. 6339–6343.
25. H. Huang and K. C. Sim, "An investigation of augmenting speaker representations to improve speaker normalisation for DNN-Based speech recognition," in Proc. Int. Conf. Acoust., Speech, Signal Process., 2015, pp. 4610–4614.
26. T. Tan et al., "Speaker-aware training of LSTM-RNNS for acoustic modeling," in Proc. Int. Conf. Acoust., Speech, Signal Process., Shanghai, China, Mar. 2016, pp. 5280–5284.