SPEECH RECOGNITION SYSTEM USING DEEP NEURAL NETWORK

Taslima Akter (takter@iu.edu), Khandokar Md. Nayem (knayem@iu.edu)
Indiana University, Bloomington

ABSTRACT

We present an automatic speech recognition system developed using end-to-end deep learning. The traditional speech systems usually rely on laboriously engineered processing pipelines and also tend to perform poorly in noisy environments. Our architecture is much more simpler than them and directly learns function that is robust to background noise, reverberation, or speaker variation. Therefore we do not need to hand-designed these components and also do not need a phoneme dictionary. Deep learning models CNNs, RNNs and DNNs are complementary in their modeling capabilities, as CNNs are good at reducing frequency variations, RNNs are good at modeling spatial dependencies, and DNNs are appropriate for mapping features to a more separable space. In this project, we take advantage of the complementarity of CNNs, RNNs and DNNs by combining them into one unified CRNN architecture. Our system, provides state-of-the-art results on the widely studied TIMIT corpus and in noisy environments as well.

Index Terms— Recurrent Neural Networks, Convolutional Recurrent Neural Networks, Connectionist Temporal Classification, LSTM, GRU

1. INTRODUCTION

In automatic speech recognition (ASR), neural networks have been used for a long time along with hidden Markov models [1][2]. But recently they have gained enormous popularity with the advancement of deep neural networks [3][4]. The main factors behind this popularity are: the deeper networks making them more powerful, by initializing the weights sensibly and using much faster hardware helps to train deep neural networks effectively, and lastly using a larger number of (context-dependent) output units improve their performance. Further improvements over DNNs have been obtained with alternative types of neural network architectures, including Convolutional Neural Networks (CNNs) [5] and Recurrent Neural Networks (RNNs) [6][7]. CNNs, RNNs and DNNs are individually limited in their modeling capabilities, and we believe that speech recognition performance can be improved by combining these networks in a unified framework.

In traditional speech systems many heavily engineered processing stages are used, including specialized input features, acoustic models, and HMMs. Domain experts invest a great deal of effort tuning their features and models to improve these pipelines. In this project as the first step we tried to adopt the approach of Deep Speech [8] which applies deep learning end-to-end using Recurrent Neural Networks (RNNs). It takes advantage of the capacity provided by deep learning systems to learn from large datasets to improve the overall performance. The model is trained end-to-end to produce transcriptions and thus, with sufficient data and computing power, can learn robustness to noise or speaker variation on its own.

The main contribution of our project is a neural network model named Convolutional Recurrent Neural Network (CRNN) which is a combination of CNN and RNN [9]. Shi et.al [9] proposed this model for image based sequence recognition system. The novelty of our project is we use this model in automatic speech recognition system. To the best of our knowledge no other works used this approach in recognizing speech till now. For sequence-like objects, CRNN possesses advantages over conventional neural network models: 1) It doesn't require detailed annotations and can be directly learned from sequence labels (for instance, words) 2) It doesn't require specialized components for speaker adaptation or noise filtering 3) Like RNN, it is able to produce a sequence of labels; 4) It is unconstrained to the lengths of sequence-like objects, requiring only height normalization in both training and testing phases; 5) It has much less parameters and consumes less storage space. In our project we implement both RNN and CRNN models and compare their performances. In the rest of this report, we will introduce the key ideas behind our speech recognition system.

2. MODELS

In this section, we describe the cost function we use for the unlabeled frame sequence of audio. Then we explain the structure of our experimental models and their parameters.

2.1. Connectionist Temporal Classification (CTC)

In sequence learning task like speech recognition, audio signal is transcribed into words or sub-word units. For this task, we require pre-segmented training data, and post-processing to transform the outputs into label sequences. In TIMIT corpus, wav files are annotated into word level. But we need framewise (which phoneme is uttered when) labeling. This

kind of labeling is tedious and time-consuming. So we consider the label on full length audio (unsegmented) and try to label this unsegmented data sequences (audio) with *Connectionist Temporal Classification* [10]. CTC transforms the network outputs into a conditional probability distribution over label sequences. The network can then be used as classifier by selecting the most probable labeling for a given input sequence.

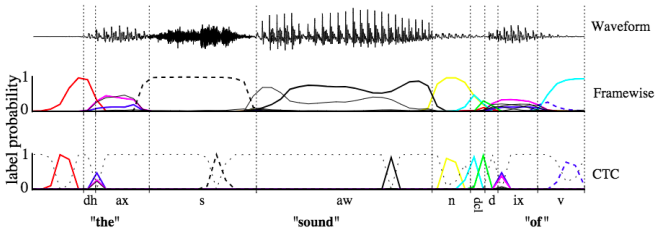

Fig. 1: Framewise & CTC networks classifying speech signal

For an input sequence $\mathbf{x}$ of length T, define a recurrent neural network with m inputs, n outputs and weight vector w as a continuous map $\mathcal{N}_w$: $(\mathbb{R}^m)^T \mapsto (\mathbb{R}^m)^T$. Let $\mathbf{y} = \mathcal{N}_\mathbf{w}(\mathbf{x})$ be the sequence of network outputs, and denote by ${y^t}_k$ the activation of output unit k at time t. Then ${y^t}_k$ is interpreted as the probability of observing label k at time t , which defines a distribution over the set ${L'}^T$ of length T sequences over the alphabet $L' = L \cup \text{blank}$,

$$p(\pi|\mathbf{x}) = \prod_{t-1}^{T} {y^t}_{\pi_t}, \forall_\pi \in {L'}^T$$

The next step is to define a many-to-one map $\mathcal{B} : {L'}^T \mapsto L^{\leq T}$, where $L^{\leq T}$ is the set of possible labellings (i.e. the set of sequences of length less than or equal to T over the original label alphabet L). This is done this by simply removing all blanks and repeated labels (e.g.$\mathcal{B}(aab) = \mathcal{B}(aaabb) = aab$). Intuitively, this corresponds to outputting a new label when the network switches from predicting no label to predicting a label, or from predicting one label to another (c.f. the CTC outputs in figure 1). Finally, the conditional probability of a given labeling $\mathbf{I} \in L^{\leq T}$ as the sum of the probabilities of all the paths corresponding to it,

$$p(\mathbf{I}|\mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1}(I)} p(\pi|\mathbf{x})$$

2.2. Recurrent Neural network (RNN)

Our basic model is a recurrent neural network (RNN) trained to ingest speech features and generate English text transcriptions. Let a single utterance x and label y be sampled from a training set $\mathcal{X} = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}, \dots\}$. Each utterance, $x^{(i)}$ is a time-series of length $T(i)$ where every time-slice is a vector of audio features, $x_t^{(i)}, t = 1, \dots, T^{(i)}$. We use Mel-frequency cepstral coefficients (MFCC) [11] as our features, so $x_{t,p}^{(i)}$ denotes the power of the p^{th} frequency bin in the audio frame at time t . The goal of our RNN is to convert an input sequence x into a sequence of character probabilities for the transcription y, with $\hat{y}_t = \mathbf{P}(c_t|x)$, where $c_t \in \{a, b, c, \dots, z, space, blank\}$.

As model architecture, we mostly adopt structure of deep speech [8] but in a concise way. Our RNN model use first 13 coefficients of mfcc feature vector. These input features feed into a 2 stacked rnn layer. We experimented with the cell type (e.g. basic RNN, LSTM, GRU). Then a Bi-directional RNN layer. Finally, a dense layer with logit activation function for output probability $\mathbf{P}(c_t|x)$.

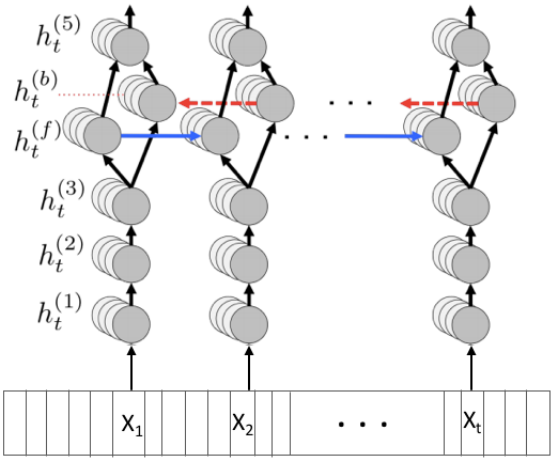

Fig. 2: Structure of our RNN model and notation.

Once we have computed a prediction, we compute CTC loss [12, 13] $\mathcal{L}(\hat{y}, y)$ to measure error in prediction, as described in section 2.1. In training, we can evaluate the gradient $\nabla_{\hat{y}}\mathcal{L}(\hat{y}, y)$ with respect to the network outputs given the ground-truth character sequence y. And we use momentum 0.9 for training.

2.3. Convolutional Recurrent Neural network (CRNN)

In RNN model, we pass the mfcc feature vector to rnn cells and model eventually learn the important structure in the input features. Another way is to use a CNN layer to learn the structure in the feature vector and use that for RNN layer [9]. This CRNN model are faster converging since CNN layer already extracted the important relation and structure hidden in feature, also CNN layer can be parallelized.

CNN layers are now quite common in extracting feature from 2D structure (e.g. image) [14]. For audio, we experimented with both mfcc feature vectors and spectrogram of the audio. We make a 2D feature vector for an audio where each row is a feature value for all time-stamp and each column is all feature in a specific time-stamp. To make feature vector of same shape, we add padding. Structure of the convolution

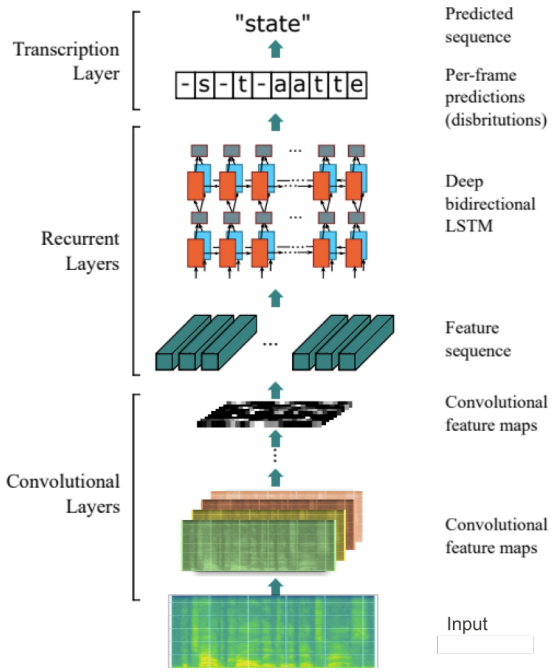

Fig. 3: The CRNN network architecture consists of three parts: 1) convolutional layers extract feature sequence from the input image; 2) recurrent layers predict a label distribution for each frame; 3) transcription layer translates the per-frame predictions into the final label sequence.

layer is shown in figure 4. Map-to-Sequence acts as the bridge between convolutional layers and recurrent layers. This layer simply convert the feature maps to feature sequences.

2.4. Language Model

When trained from large quantities of labeled speech data, the RNN/CRNN model tend to produce readable character-level transcriptions. Many times the most likely character sequence predicted by the model is exactly correct without external language constraints. But many error made because of phonetically plausible renderings of language. Few words occur rarely or never appear in our training set. In practice, this is hard to avoid: training on all possible word of a language is infeasible. Therefore, we add a NLTK N-gram language model since these models are easily trained from huge unlabeled text corpora.

3. EXPERIMENTS

3.1. Data Preprocessing

We use TIMIT corpus (total 6300 sentences, 10 sentences spoken by each of 630 speakers from 8 major US dialect regions). We divide this data in 75%, 5%, 20% percentile for respectively train, validation and test.

Type	Configurations
Bidirectional LSTM/GRU	# hidden units: 265
Convolution	#maps:512, k:2x2, s:1, p:0
Map-to-Sequence	-
Dropout	Param: 0.25
MaxPooling	Window: 2x2, s:2
Convolution	#maps:64, k:5x5, s:1, p:1
Convolution	#maps:64, k:5x5, s:1, p:1
Dropout	Param: 0.25
MaxPooling	Window: 2x2, s:2
Convolution	#maps:32, k:3x3, s:1, p:1
Convolution	#maps:32, k:3x3, s:1, p:1
Input	Wx500 (mfcc/spectogram)

Fig. 4: Network configuration summary. First row is the top layer. 'k', 's' and 'p' stand for kernel size, stride and padding size respectively.

To make this model robust and workable for real-life environment, we synthetically add 10 different types of noises (Babble, Cafe, Car, Factory, Machine gun, Plane, Restaurant, SSN, Tank, White Noise) in 5 different mixture signal-noise ratio (SNR) levels, -3dB, 0dB, 3dB, 6dB, 9dB. So we make the data size, $(6300 + 6300 \times 10 \times 5) = 321,300$. Same ratio percentiles are used here too. These audios are normalized and then feed into RNN or CNN layer.

Here we want to note that most of the state-of-art speech recognizers are trained on million size dataset (which are not free to use) and suitable for home environment (very insignificant noise). Whereas our model is designed to work in noisy environment too.

3.2. Result

We calculate the L2 distance norm as error function. So we also decode the sentences for the test dataset. We do this for clean TIMIT corpus and noisy version. As noisy dataset is too large and it takes long time to run. So in times of running, we take a subsample of that and run experiments on that.

Figure 5 shows the performance (L2 norm) of our model. We experiment with both LSTM and GRU cells. When only one directional layer is used, LSTM layer performs well than GRU which is tend to stuck in local minima. But LSTM net takes more time to converge than GRU. In bi-directional model, LSTM layers also give almost same performance like GRU. So we choose GRU because of its faster convergence. Now CRNN models are performing better than all other. But the cost is more parameters to optimize. Convergence time is a bit greater than the RNN model. But we think that is because we do not design the code to take full advantages of the

Model	Dataset	Feature	L2 Norm (at converge)
2 LSTM layers	TIMIT	Mfcc	2.87×10^3
	Noisy TIMIT		2.45×10^3
	TIMIT	Spectrogram	2.8×10^3
	Noisy TIMIT		2.54×10^3
2 GRU layer, 1 Bidirectional layer	TIMIT	Mfcc	2.65×10^3
	Noisy TIMIT		2.57×10^3
	TIMIT	Spectrogram	2.68×10^3
	Noisy TIMIT		2.5×10^3
CNN, 2 GRU layer, 1 Bidirectional layer	TIMIT	Mfcc	2.43×10^3
	Noisy TIMIT		2.556×10^3
	TIMIT	Spectrogram	2.39×10^3
	Noisy TIMIT		2.41×10^3
CNN, 2 GRU layer, 1 Bidirectional layer	TIMIT	Mfcc	2.46×10^3
	Noisy TIMIT		2.5×10^3
	TIMIT	Spectrogram	2.41×10^3
	Noisy TIMIT		2.43×10^3

Fig. 5: Performance comparisons of RNNs and CRNNs.

parallelism property of CNN. We notice that noisy results are quite similar of clean one, sometimes better. That supports our hypothesis that adding noise will force the model to learn the speech properly. So we think if we train the model for the full size dataset, we can get robust speech recognizer. We also tried our models with spectrogram feature. The performance is almost as good as mfcc. So using deep models, we can use only the spectrogram of an audio.

L2 norm may not reflect the true power of the model. If we inspect the decoded text and compare them with the true annotation we can see how well our model performs. In figure 6, we can see that our model fails for long sentences and complex words. But CRNN can detect the labels almost in good accuracy. So this actually supports our hypothesis that CRNN with noisy data performs well in real-life environment.

4. CHALLENGES AND FUTURE WORKS

In this project we only worked with English corpus but one of our goals was to apply our speech recognition model on other language corpus, specially on Bengali. But Bengali audio corpus with transcriptions are not freely available. We tried to contact with some personnel who had done some works with Bengali corpus by manually created them but could not manage the corpus. We aim to create our own Bengali corpus and apply speech recognition algorithm on that in future as this is a time consuming task. Also very few works have been done to recognize speech for mixed language (code switching) specially with Bengali and English languages due to unavailability of Bengali corpora. Recognizing mixed speech on two or more languages can be considered as future task. Another obvious next step is to extend the system to large vocabulary speech recognition.

Model Output	Decoded Transcript
Good Example	
eve n a simple voc e bulari contains som bols	Even a simple vocabulary contains symbols.
aw shut uup he said	Aw, shut up, he said.
high way nd freeway mean the same thing	Highway and freeway mean the same thing.
jon cleans selfish for a living	John cleans shellfish for a living.
Bad Examples	
D n t ask me to ca an o lu rag l kat that	Don't ask me to carry an oily rag like that.
is el c so on the other hand, would en que ss n stng h the regulars	His election, on the other hand, would unquestionably strengthen the regulars.
the fish beg n to leap fon t ly on the surface of the s mll kee	The fish began to leap frantically on the surface of the small lake.
the small bu put the wom mp on the kk	The small boy put the worm on the hook.
calum makes boy and te s ng	calcium makes bones and teeth strong

Fig. 6: Decoded transcripts of CRNN model.

5. CONCLUSION

In this report, we have presented two end-to-end deep learning-based speech systems: recurrent neural networks [8] and convolutional recurrent neural networks [9] which give state-of-the-art results in speech recognition on the TIMIT database in two challenging scenarios: clear, conversational speech and speech in noisy environments. We believe this approach will continue to improve as we capitalize on increased computing power and dataset sizes in the future.

6. REFERENCES

[1] Qifeng Zhu, Barry Chen, Nelson Morgan, and Andreas Stolcke, "Tandem connectionist feature extraction for conversational speech recognition," in *International Workshop on Machine Learning for Multimodal Interaction*. Springer, 2004, pp. 223–231.

[2] Herve A Bourlard and Nelson Morgan, *Connectionist speech recognition: a hybrid approach*, vol. 247, Springer Science & Business Media, 2012.

[3] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[4] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on*. IEEE, 2013, pp. 8614–8618.

[6] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[7] Haşim Sak, Andrew Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth annual conference of the international speech communication association*, 2014.

[8] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[9] Baoguang Shi, Xiang Bai, and Cong Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017.

[10] Mohammed Waleed Kadous et al., *Temporal classification: Extending the classification paradigm to multivariate time series*, University of New South Wales, 2002.

[11] Bin Zhen, Xihong Wu, Zhimin Liu, and Huisheng Chi, "On the importance of components of the mfcc in speech and speaker recognition," in *Sixth International Conference on Spoken Language Processing*, 2000.

[12] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[13] Haşim Sak, Andrew Senior, Kanishka Rao, Ozan Irsoy, Alex Graves, Françoise Beaufays, and Johan Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4280–4284.

[14] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*. IEEE, 2014, pp. 512–519.