SPEECH RECOGNITION – A REVIEW

Abhishek Chand[1], Subhojeet Chakraborty[2], Abhinava Anand[3], Swapnil Dhande[4]

[1]*Student, Department of Computer Engineering, NBN Sinhgad School of Engineering, Maharashtra, India*
[2] *Student, Department of Computer Engineering, NBN Sinhgad School of Engineering, Maharashtra, India*
[3] *Student, Department of Computer Engineering, NBN Sinhgad School of Engineering, Maharashtra, India*
[4] *Student, Department of Computer Engineering, NBN Sinhgad School of Engineering, Maharashtra, India*

ABSTRACT

Deep learning methodologies have had a major impact on performance across a wide variety of machine learning tasks, and speech recognition is no exception. We describe a set of deep learning techniques that proved to be particularly successful in achieving performance gains in word error rate on a popular large vocabulary conversational speech recognition benchmark task. We compare a set of speech recognition models and find out which one among them prove to show the least error rate.

Keyword: - *Speech Recognition, Deep Learning, Neural Networks, Recurrent Neural Networks, Long Short Term Memory Neural Network.*

1. Introduction

Neural networks have a long history in speech recognition, usually in combination with hidden Markov models. Given that speech is an inherently dynamic process, it seems natural to consider recurrent neural networks (RNNs) as an alternative model.

There has been an exponential increase in papers describing more variations of deep learning configurations. The most frustrating aspect of this work is that only a small fraction of these papers show comparisons on a common benchmark. Most papers report results on what are considered either "toy" problems or proprietary data sets, making it impossible to assess the relative merits of different techniques.

The following shows the working of different neural network models and further concludes which one have shown the best lowest error rate related to certain experiments performed by researchers.

2. BASIC NEURAL NETWORK CONFIGURATION FOR SPEECH RECOGNITION

Given an input sequence x = (x1,...,xT), a standard recurrent neural network (RNN) computes the hidden vector sequence h = (h1,...,hT) and output vector sequence y = (y1,...,yT) by iterating the following equations from t = 1 to T:

$$h_t = \mathcal{H}\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right) \quad (1)$$

$$y_t = W_{hy}h_t + b_y \quad (2)$$

where the W terms denote weight matrices (e.g. Wxh is the input-hidden weight matrix), the b terms denote bias vectors (e.g. bh is hidden bias vector) and H is the hidden layer function.

A basic block diagram of a speech recognition system is shown in Figure 1. It consists of a feature extraction process, an acoustic model (AM), an LM, and a speech recognition decoder.

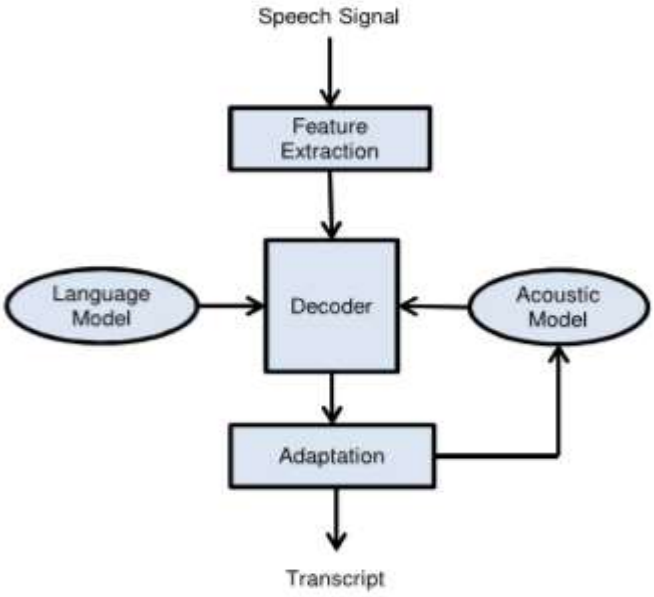

Figure 1. Traditional speech recognition system

The feature extraction process extracts features thought to be important for speech recognition from the basic speech signal. Features are spectrally based and extracted from the signal at 100 times per second. We can therefore represent the output of the feature extraction system as a matrix S whose rows can be associated with frequency slices and whose columns can be associated with time slices.

3. RECURRENT NEURAL NETWORKS

A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed cycle. This allows it to exhibit dynamic temporal behavior. Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

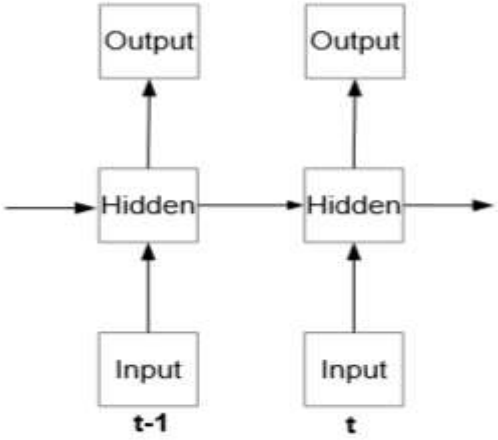

Figure 2. RNN Architecture unfolded in time

In RNN language modeling, the conditional word probabilities P (wt|wt−1, ht−2) are calculated as follows:

$$p(w_t = i|w_{t-1}, h_{t-2}) = \frac{\exp(y_t^i)}{\sum_{j=1}^{N} \exp(y_t^j)}$$

where yi t represents the ith element of the output vector yt. Here, each output target corresponds to a word in the vocabulary. The probability of a word sequence W = w1,w2,⋯ ,wT is calculated by multiplying conditional word probabilities, given as

$$P(W) = \prod_{t=1}^{T} p(w_t|w_{t-1}, h_{t-2})$$

The advantage of RNN language models as compared to word or class n-gram models and feedforward neural networks is that they do not restrict the history to the preceding n−1 words.

4. BIDIRECTIONAL RECURRENT NEURAL NETWORKS

Bidirectional RNNs exploit both the past and future context by processing the input data in both directions. Figure 3 shows a bidirectional RNN architecture unfolded in time for two time steps. As shown in the figure, bidirectional RNNs compute a forward hidden layer hF t by iterating through the input sequence from t = 1,...,T, and a backward hidden layer hB t by iterating through the input sequence from t = T,...,1. These two hidden layers are combined into a single output layer using the following equations:

$$\begin{aligned} h_t^F &= \tanh(W_{xh}^F x_t + W_{hh}^F h_{t-1}^F + b_h^F) \\ h_t^B &= \tanh(W_{xh}^B x_t + W_{hh}^B h_{t+1}^B + b_h^B) \\ y_t &= W_{hy}^F h_t^F + W_{hy}^B h_t^B + b_y \end{aligned}$$

Note that in language modeling, the output from the last time step is the input for the next time step. With bidirectional models, this causes circular dependencies to arise when combining probabilities from multiple time steps. Unlike with unidirectional models, multiplying individual conditional probabilities P (wt|wt −1, ht − 2, wt + 1, ht + 2) from a bidirectional language model does not compute a true likelihood, but rather a pseudo likelihood. Still, it is straightforward to optimize the pseudo likelihood of training data rather than its likelihood during model training, and this is in fact what we do.

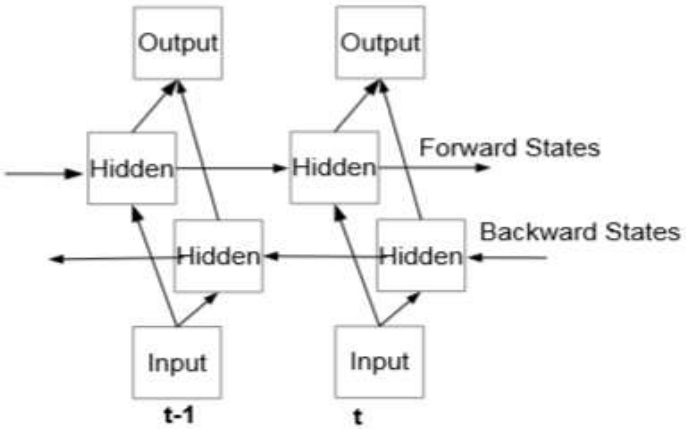

Figure 3. Bidirectional RNN architecture unfolded in time.

5. LONG SHORT-TERM MEMORY NEURAL NETWORKS

Even though RNNs potentially utilize arbitrarily long histories, in practice the effective context length of an RNN is quite limited. Long Short-Term Memory neural networks were proposed to remedy this limitation. An LSTM neural network replaces the nonlinear units in the hidden layer of an RNN with memory blocks containing memory cells for storing values; and multiplicative gates for reading (output), writing(input), and resetting(forget) these values. A memory cell can be used to store information for long periods, and gates collect activations from both inside and outside a memory block to update a memory cell's value. The LSTM equations are given as follows:

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

where it, ft, ot and ct represent the input gate, forget gate, output gate and cell activation vectors at time t, respectively. The matrices W∗∗ denote the weight matrices between various layers, gates, and cells; e.g., Wxi represents the weight matrix between the input layer and the input gates. The gate and cell bias terms are denoted as bi, bf, bo and bc; and σ(·) is the logistic sigmoid function. For language modeling, after computing ht, conditional word probabilities P(wt|wt−1,ht−2) are calculated.

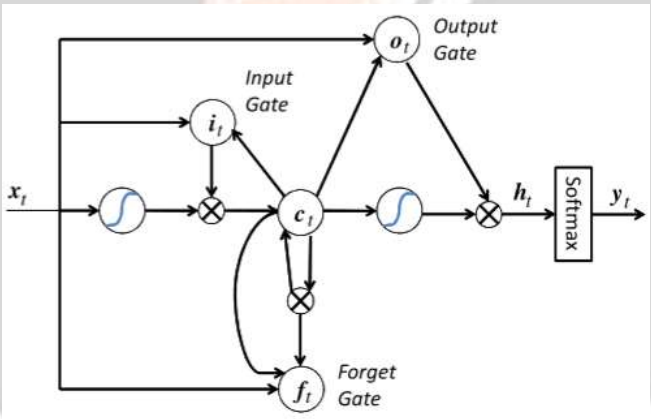

Figure 4. LSTM memory block with input, output, and forget gates

6. COMPARISON OF DIFFERENT NETWORKS

Table 1 compares recognition results across the different types of networks on the SWB corpus. A number of observations can be made. First, it is obvious that increasing the amount of training data from 300 hours to 2,000 hours clearly improves performance for all networks under consideration. Second, the sequence-training objective function clearly produces better performance for speech than the cross-entropy objective function. Third, the attempts to capture local structure and correlation in the signal (especially the VGG, RNN, and LSTM networks) are all successful in driving down the error rate. Whether the best approach is deliberate engineering of the receptive field via use of VGG—or learning this implicitly via a complex recurrent network such as the LSTM network—remains unclear. The results suggest that the LSTMs are better able to take advantage of more data than the VGG networks. Also note that sequence training produces larger accuracy gains for VGG networks than for LSTMs because the latter models already incorporate sequence information due to their memorization capabilities. Another point is important to make. First, the CNN and VGG networks operate directly on the S matrix—the basic time-frequency representation of the feature vector. The DNN, DNN-ivector, RNN, and LSTM can all operate on the X matrix—a matrix of transformed features after applying various types of speaker normalization processes. Unlike for CNN and VGG, the DNN features do not need to be topographical; therefore, they can have more sophisticated

features for speaker adaptation such as i-vectors. This adaptation consists in having a low-dimensional identity vector (or i-vector) that characterizes the speaker as input to the DNN. We could be losing as much as 0.6% absolute in WER for CNNs by not performing speaker normalization (some simple transformations are applied, but nothing as effective as for the other models). This is clearly an area of open research for these types of networks.

System	300 hours training data		2,000 hours training data	
	Cross-entropy	Sequence	Cross-entropy	Sequence
GMM	14.5			
DNN	14.1	12.5		
CNN	13.2	11.8	12.6	10.4
DNN-i	13.2	11.9	11.7	10.3
VGG	11.8	10.5	10.2	9.4
Unfolded RNN-i	12.7	11.3	11.5	9.9
LSTM-i	10.8	10.6	9.5	9.0
RNN-maxout			10.4	9.3

Table 1. Word error rates (%) for various deep learning architectures on the Hub5'00 SWB test set.

7. CONCLUSIONS

It can be seen that significant performance improvements can be obtained in speech recognition through the use of deep learning methods. However, perhaps because speech has been worked on for many years, the basic baselines based on years of development of non-deep-learning (i.e., shallow) techniques presented a "high bar" to entry. We have shown that by applying more complex deep learning architectures that take into account some of the basic properties of time-frequency correlations of speech can significant gains be seen relative to existing baselines. An open question for us also concerns whether gains continue to accrue by significantly increasing the amounts of training data to 20,000 hours of speech and higher or whether the current structures saturate and even more complex structures need to be investigated and developed, for example, extensions of Deep Learning to Conditional Random Fields. More work is needed to achieve human performance, and no speech recognition scientists will rest until they achieve or beat this lofty goal.

8. REFERENCES

[1] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, Yoshua Bengio,2017): A NETWORK OF DEEP NEURAL NETWORKS FOR DISTANT SPEECH RECOGNITION

[2] Abdulghani Ali Ahmed and Nurul Amirah Abdullah, "Real Time Detection of Phishing Websites" 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (X. Chen, A. Ragnil, J. Vasilakes, X. Liu, K. Knilll, MJ.F. Gales, 2017): Recurrent neural network language models for keyword search.

[3] Sri Harsha Dumpala, Sunil Kumar Kopparapu, 2017): Improved Speaker Recognition System for Stressed Speech using Deep Neural Networks.

[4] (Zhiyuan Tang, Dong Wang, Zhiyong Zhang, 2016): Recurrent neural network training with dark knowledge transfer

[5] (G.SaonM.Picheny,2017): Recent advances in conversational speech recognition using convolutional and recurrent neural networks.

[6] (Ebru Arisoy, Abhinav Sethy, Bhuvana Ramabhadran, Stanley Chen, 2015): BIDIRECTIONAL RECURRENT NEURAL NETWORK LANGUAGE MODELS FOR AUTOMATIC SPEECH RECOGNITION