An Investigation of Deep Neural Network Architectures for Language Recognition in Indian Languages

by

Mounika KV, Sivanand a, Lakshmi H R, Suryakanth V Gangashetty, Anil Kumar Vuppala

in

The 43rd International Symposium on Computer Architecture (ISCA)

Seoul, South Korea

Report No: IIIT/TR/2016/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2016

INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

An Investigation of Deep Neural Network Architectures for Language Recognition in Indian Languages

Mounika K V, Sivanand Achanta, Lakshmi H R, Suryakanth V Gangashetty, and Anil Kumar Vuppala

Speech and Vision Laboratory, IIIT Hyderabad, India
{mounika.kv,sivanand.a}@research.iiit.ac.in, {svg,anil.vuppala}@iiit.ac.in

Abstract

In this paper, deep neural networks are investigated for language identification in Indian languages. Deep neural networks (DNN) have been recently proposed for this task. However many architectural choices and training aspects that have been made while building such systems have not been studied carefully. We perform several experiments on a dataset consisting of 12 Indian languages with a total training data of about 120 hours in evaluating the effect of such choices.

While DNN based approach is inherently a frame based one, we propose an attention mechanism based DNN architecture for utterance level classification there by efficiently making use of the context. Evaluation of models were performed on 30 hours of testing data with 2.5 hours for each language. In our results, we find that deeper architectures outperform shallower counterparts. Also, DNN with attention mechanism outperforms the regular DNN models indicating the effectiveness of attention mechanism.

Index Terms: deep neural network, attention mechanism, language identification

1. Introduction

Language recognition (LR) refers to the task of recognizing the language from a spoken utterance [1]. While the spoken utterance has underlying lexical, speaker, channel, environment and other such variations, the goal is to recognize the identity of the language invariant to these factors. The LR technology plays a key role in many applications such as multilingual speech recognition [2], in security [3], and call-routing [4].

Approaches using phonotactic and prosodic features derived from the speech signals have been explored for LR in [5][6][7][8]. Lately, following the trend in speaker recognition, i-vector based approaches have proven to be successful for LR [9]. The i-vector based approach uses Gaussian mixture models (GMMs) for acoustic modeling. The use of deep neural networks (DNN) in acoustic modeling for speech recognition instead of traditional GMMs has resulted in significant performance gains [10]. This has led researchers in the LR community to explore i-vector extraction using DNNs for acoustic modeling. Application of DNNs in this way for LR has been recently investigated in [11][12].

Broadly, there are two ways in which neural networks can be used for LR: (1) to get posteriors and use a back-end classifier (i.e., as an acoustic model), and (2) as an end-to-end LR system. In [11], the authors proposed the use of convolutional neural networks for posterior extraction and then trained the i-vector based LR systems. They have reported improvements in noisy conditions over the conventional i-vector based systems. In [12], a single DNN acoustic model for both speaker and language recognition tasks has been trained and significant gains have been achieved in both the tasks simultaneously. On the other hand, an end-to-end LR system using 8-layer DNN was explored in [13] and excellent performance improvements over i-vector baseline on the standard NIST LRE 2009 dataset have been reported. In this work, we propose to further explore DNN and a modified DNN architecture for end-to-end LR task.

The paper is organized as follows: In section 2, we describe the relation to prior work and outline the contributions of this work. In section 3, a detailed description of the database used for our experiments is given followed by proposed method in section 4. The experiments and results are detailed in section 5. Finally, the conclusions and scope for future work are presented in section 6 and 7 respectively.

2. Relation to Prior Work

Neural networks for LR in Indian languages have been proposed earlier in [14]. There are two aspects in which this study differs from the current trend.

- Features : Prosodic Vs. Spectral
- Model : Shallow Vs. Deep

In this paper our approach is similar in spirit to [13], i.e., we perform end-to-end DNN based LR experiments using spectral features. Given large amounts of training data per language ($>$ 10 hrs), the DNN based approach outperforms the i-vector based approaches as was reported in [13]. However, one drawback of the DNN based systems is that, the decision is taken at every frame and the context used is fixed whilst language id is usually assigned to a whole utterance. To better capture the temporal context and to do utterance wise classification, recently, recurrent neural network (RNN) based LR has been proposed [15]. This technique uses long short-term memory (LSTM) [16] cells as RNN units which have been shown to be effective in memorizing long temporal context. The LSTM-RNNs are trained using back-propagation through time, with targets set for every 5 frames.

However, the LSTM-RNNs are computationally intensive to train and also because of the sequential nature of RNNs they are not parallelizable. Although computation can be reduced using simple RNNs [17] instead of advanced RNN units like LSTMs, the sequential nature is still preserved. In this paper, we propose a recently introduced architecture in [18] for alleviating the above problems for LR. In [18], authors proposed a feed-forward deep neural network with attention for solving some memory problems involving pathological long range dependencies. We refer to this architecture as DNN with attention

http://dx.doi.org/10.21437/Interspeech.2016-910

(DNN-WA) in the rest of the paper. The advantage of DNN-WA is that while it is able to memorize, it is also parallelizable because of the strictly feed-forward architecture with no recurrent connections. We have explored this architecture in the context of LR for classifying entire utterance rather than emitting a frame-level decision and finally combining the decisions as will be done for a DNN. This also takes into account the problem of using contextual information for LR. In addition, as will be shown later in section 5, this architecture allows us to see what part of the input sequence plays a crucial role in making the decision. The attention mechanism allows us to peep into the input feature frames that are more important for LR.

To summarize the contributions of this work, firstly we have explored DNN-WA [18] architecture for utterance level LR and secondly investigation of effect of depth of DNNs for LR has been carried out.

3. Database

Our dataset consists of 12 Indian languages. The details of the structure of the data including the number of speakers per language in train and test sets are given in Table 1. The three columns under the train/test represent total speech data in number of hours, number of male and female speakers respectively. The style of speech data is read speech and has been recorded at 16 kHz sampling rate. For the purposes of our experiments, each sound file has been sliced into chunks of around 5s both in the training and testing datasets.

Table 1: Description of speech corpus used

Language	Train			Test		
	#Hrs	#M	#F	#Hrs	#M	#F
Assamese	12.40	22	11	1.94	3	3
Bengali	9.91	24	35	1.53	15	15
Gujarati	9.71	115	75	2.18	37	36
Hindi	10.96	41	28	3.23	16	19
Kannada	10.08	21	16	0.99	10	4
Malayalam	10.08	7	6	3.07	9	7
Manipuri	5.31	5	6	2.50	3	3
Marathi	7.84	74	31	2.47	17	15
Odiya	9.81	31	31	2.45	9	9
Punjabi	15.43	2	9	3.78	2	1
Telugu	10.43	21	21	3.15	4	4
Urdu	10.80	56	18	3.27	16	5

One unique challenge in building end-to-end LR systems on Indian language dataset is that most of the phonemes overlap amongst several languages. For instance, Telugu, Malayalam and Kannada being from the same language family have similar phonemes. The same can be said to be true for Assamese and Bengali. The geographical proximity also plays a role. Hence LR can be quite challenging on this dataset.

Samples from the database can be heard online [1].

4. DNN-WA

In this section we describe the architecture of DNN-WA (see Fig. 1). This is a simple DNN equipped with attention mechanism. Attention mechanism has been inspired from the one proposed in [19] for neural machine translation. The modification was done such that the attention is computed just by using the input feature vectors as opposed to using both input and output feature vectors as in [19].

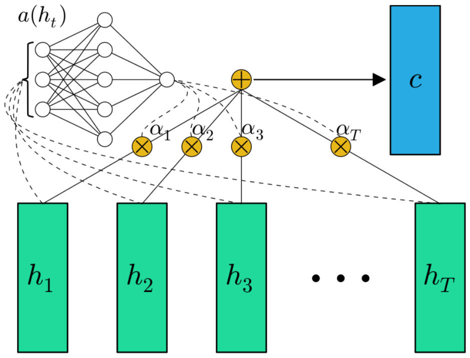

Figure 1: Deep neural network with attention model [18]

Given an input sequence, $X = \{x_1, x_2, \cdots, x_T\}$, a hidden layer representation, $H = \{h_1, h_2, \cdots, h_T\}$, is computed by forward pass through regular DNN and attention is computed on this hidden features.

The attention mechanism $a(h_t)$ shown in Fig. 1, is computed using a single layer perceptron and then a *softmax* operation is performed to normalize the values between zero and one.

$$\begin{aligned} H &= [h_1 \; h_2 \; \cdots \; h_T] \\ \gamma &= tanh(W_a H + b_a) \\ \alpha &= softmax(\gamma) \end{aligned} \tag{1}$$

In the above equations, α is referred to as *attention vector*, and W_a, b_a are the parameters of the attention network optimized along with other parameters of the network using back-propagation algorithm.

The context vector is computed from the attention vector as

$$c = H\alpha \tag{2}$$

The output is computed by transforming the context vector c using output layer weights U followed by *softmax* operation.

$$y = softmax(Uc + b_o) \tag{3}$$

Where b_o is the output layer bias. Note that for the entire input utterance X only a single decision vector y is predicted.

The depth of neural network before/after the attention mechanism can be varied. In this paper we have used 1 and 3 hidden layers before the attention to understand the effect of depth. Increasing the number of hidden layers after the context vector has not been investigated here and is left as part of future work. The total number of layers for DNN-WA is the number of hidden layers before the context plus the additional output layer.

5. Experiments and Results

We extract 39-dimensional MFCCs (13 static + Δ +$\Delta\Delta$) from each of the 5s chunks. DNNs were trained using a mini-batch stochastic gradient descent (SGD) with classical momentum.

[1] https://goo.gl/YSnSoe

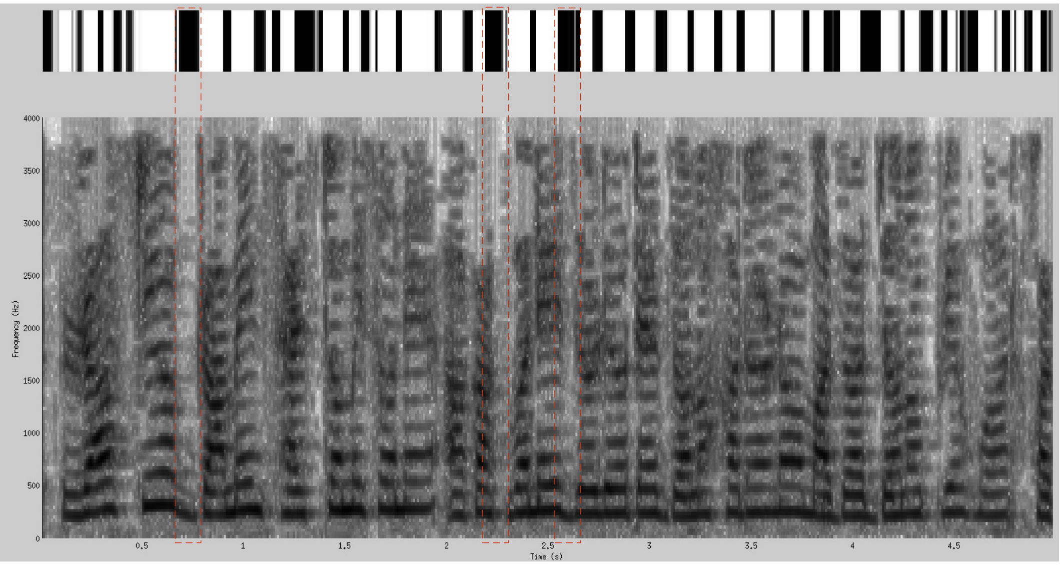

Figure 2: An example spectrogram with attention

Table 2: Equal error rate (EER in %)

Language	Ass	Ben	Guj	Hin	Kan	Mal	Man	Mar	Odi	Pun	Tel	Urd	Average
DNN_{2l}	4.82	12.28	15.58	23.42	16.57	13.18	6.95	17.67	8.85	3.87	5.74	8.41	11.45
DNN_{4l}	6.07	7.68	12.90	17.96	14.67	17.85	7.28	12.57	6.21	5.65	4.90	6.14	9.99
DNN_{6l}	5.05	8.42	17.35	20.28	15.48	14.72	7.28	15.59	10.57	4.57	5.65	10.31	11.27
$DNN-WA_{2l}$	6.82	6.91	10.79	27.49	8.00	17.14	7.57	8.71	5.72	5.51	5.27	8.89	9.90
$DNN-WA_{4l}$	5.66	6.73	7.33	24.44	6.83	11.70	4.81	8.67	6.73	5.49	5.05	7.86	$\mathbf{8.44}$

Normalized initialization proposed in [20] was used to initialize our networks. We adjust the hyper-parameters using a validation set. Inorder for the comparison between DNN and DNN-WA to be fair, we have randomized only the sequences presented to the networks while the frames within a given sequence were not randomized. The mini-batch size was equal to the length of the sequence given as input to the network.

The input layer has 39 linear units, while the output layer is *softmax* with 12 units. Rectified linear units (ReLU/R) [21] were used as the activation functions in the hidden layers.

Table 3: DNN architectures

# layers	Architecture
2	700R 500R
4	700R 500R 200R 100R
6	700R 500R 200R 100R 50R 25R

All the networks are trained to minimize the cross-entropy loss over the entire training set. For DNN, the frame based output were averaged before taking the final decision for the utterance while for DNN-WA there was no necessity to do this as can be seen from Eq. 3. We use equal error rate (EER) as performance metric, when considering only scores of each individual language.

The first set of experiments were performed to determine the depth of the architecture that is best suited for LR. The number of units used in each layer are presented in Table 3 for various depths. We can see from the Table 2 that architecture with depth of 4 layers performs significantly better than architecture with 2 layers.

Next we examine the DNN-WA architecture performance. The number of hidden layers before the attention layer was 1 and 3. We can clearly see form the Table 2 that the DNN-WA outperforms the regular DNN (rows 1 vs. 4 and 2 vs. 5) in most of the languages and on the average (last column). This could be attributed to the fact that implicitly model captures context and integrates information before taking the final decision. In DNN-WA also having more number of hidden layers before the attention mechanism helps as can be seen from the bottom two rows of Table 2. A plot of confusion matrix obtained using the best architecture (DNN-WA with 4 layers) is shown in Fig. 3.

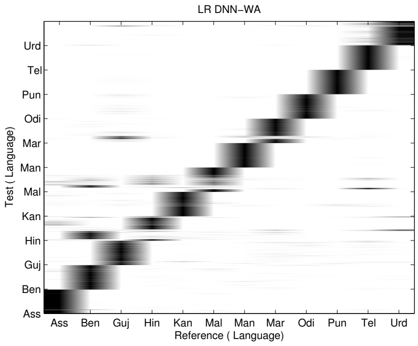

Figure 3: Confusion matrix for DNN-WA model with 4 layers

5.1. Analysis of attention mechanism

One feature of attention mechanism in neural networks is that it helps us analyze how the input is being attended to while making the decision. A plot of speech signal spectrum (from Assamese language) and the attention vector (α in Eq. 1) weights are shown in Fig 2. The spectrum is plotted only till 4KHz for clarity. The black regions in the attention plot above the spectrum implies higher values (or *attention*) for the respective frames in the input. It is interesting to note from the figure that the attention is largely towards sections of the signal where the transitions take place as is indicated at three places by the red dashed rectangles.

The code for replicating the experiments is available online [2]. All our experiments were run on NVIDIA Geforce GTX-660 graphics card.

6. Summary and Conclusions

In this paper, we have introduced the DNN-WA for performing the utterance level LR. Our results indicate that the proposed attention architecture is well suited for the task of LR. The integration of hidden layer features using the attention mechanism has resulted in effective usage of context. A preliminary qualitative analysis of attention mechanism revealed that transition regions in the signals have more discriminative information for LR.

In the conventional DNN based LR, we have performed experiments to determine optimal depth for the dataset. Our results are consistent with previous findings [15].

7. Scope for Future Work

Using fusion mechanism to further improve the results by exploiting the complementary nature of the two architectures is currently being investigated into. The analysis of varying the utterance length during test time and its effect on the performance of DNN-WA architecture has to be studied.

In addition to directly using the context vector from the DNN-WA model for classification, the utterance level representation can be stacked to the regular frame-wise feature vector. This is similar to the way bottle-neck features are used. However the key difference is that while bottleneck features change from frame to frame, the proposed context vector remains same throughout the utterance. This approach will be explored as part of future work.

8. Acknowledgements

The authors would like to thank Science & Engineering Research Board (SERB) for funding "Language Identification in Practical Environments (YSS/2014/000933)" project. The second author also likes to acknowledge TCS for partially funding his PhD.

9. References

[1] Y. K. Muthusamy, E. Barnard, and R. A. Cole, "Reviewing automatic language identification," *IEEE Signal Processing Magazine*, vol. 11, no. 4, pp. 33–41, Oct 1994.

[2] B. Ma1, C. Guan, H. Li, and C.-H. Lee, "Multilingual speech recognition with language identification," in *Proc. ICSLP*, 2002.

[3] H. Li, B. Ma, and C. H. Lee, "A vector space modeling approach to spoken language identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 271–284, Jan 2007.

[4] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, "Language identification: a tutorial," *Circuits and Systems Magazine,IEEE*, vol. 11, no. 2, pp. 82–108, 2011.

[5] M. A. Zissman and K. M. Berkling, "Automatic language identification," *Speech Communication*, vol. 35, no. 1, pp. 115–124, 2001.

[6] M. A. Zissman *et al.*, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, p. 31, 1996.

[7] A. E. Thymé-Gobbel and S. E. Hutchins, "On using prosodic cues in automatic language identification," in *Proc. ICSLP*, 1996.

[8] J.-L. Rouas, J. Farinas, F. Pellegrino, and R. André-Obrecht, "Modeling prosody for language identification on read and spontaneous speech," in *Proc. ICASSP*, 2003, pp. I–40.

[9] N. Dehak, P. A. Torres-Carrasquillo, D. A. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction." in *Proc. INTERSPEECH*, 2011, pp. 857–860.

[10] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath, "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.

[11] Y. Lei, L. Ferrer, A. Lawson, M. McLaren, and N. Scheffer, "Application of convolutional neural networks to language identification in noisy conditions," in *Proc. Odyssey-14, Joensuu, Finland*, 2014.

[12] F. Richardson, D. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," in *Proc. INTERSPEECH*, 2015, pp. 1146–1150.

[13] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *Proc. ICASSP*, 2014, pp. 5337–5341.

[14] M. Leena, K. Srinivasa Rao, and B. Yegnanarayana, "Neural network classifiers for language identification using phonotactic and prosodic features," in *Proc. of International Conference on Intelligent Sensing and Information Processing*, 2005, pp. 404–408.

[15] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and P. J. Moreno, "Automatic language identification using long short-term memory recurrent neural networks." in *Proc. INTERSPEECH*, 2014, pp. 2155–2159.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] S. Achanta, T. Godambe, and S. V. Gangashetty, "An investigation of recurrent neural network architectures for statistical parametric speech synthesis," in *Proc. INTERSPEECH*, 2015, pp. 2524–2528.

[18] C. Raffel and D. P. W. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *CoRR*, vol. abs/1512.08756, 2015. [Online]. Available: http://arxiv.org/abs/1512.08756

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2014.

[20] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. International conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[21] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. Hinton, "On rectified linear units for speech processing," in *Proc. ICASSP*, May 2013, pp. 3517–3521.

[2] https://goo.gl/q4dvKF