

000  
001  
002  
003  
004  
005  
006  
007  
008  
009  
010  
011  
012  
013  
014  
015  
016  
017  
018  
019  
020  
021  
022  
023  
024  
025  
026  
027  
028  
029  
030  
031  
032  
033  
034  
035  
036  
037  
038  
039  
040  
041  
042  
043  
044  
045  
046  
047  
048  
049  
050  
051  
052  
053

---

# Deep Learning for Speaker Recognition

---

**Sai Prabhakar Pandi Selvaraj**  
CMU  
spndise@andrew.cmu.edu

**Sandeep Konam**  
CMU  
skonam@andrew.cmu.edu

## Abstract

Automated speaker recognition has become increasingly popular to aid in crime investigations and authorization processes with the advances in computer science. Speaker recognition or broadly speech recognition has been an active area of research for the past two decades. There has been significant improvement in the recognition accuracy due to the recent resurgence of deep neural networks. In this work we built a LSTM based speaker recognition system on a dataset collected from Coursera lectures. We achieved an accuracy of 93%. Prior to applying deep-learning techniques, we tested on a base-line using feed-forward network on a different dataset and achieved an accuracy of 96.48%. Results show that the deep learning network could detect the speakers very well except in cases where there is significant overlap in the speaker's accent and tone. Based on the comparison with previous work we conclude that the variation in LSTM produce significant improvement only in terms of the training speed but very little improvement in performance.

## 1 Introduction

Finding a way to make computers understand the human languages has been a subject of research for a long period of time. Speech recognition is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable format. An area related to speech recognition is speaker recognition. Speaker recognition is easiest explained as the ability to identify who is speaking, based on audio data.

In this we have explored the use of recurrent neural network for speaker recognition. As a baseline and a proof of concept we have tested the gender detection using shallow neural network, which is done using features from dataset containing phonemes. And since phonemes are the fundamental building block of the speech this should serve as a perfect baseline. The recurrent neural network used for this experiment is Fast Long short term memory unit (Fast-LSTM), which avoid many shortcomings of a naive recurrent neural network, for speaker detection on text-independent data.

## 2 Related Work

In the first part section we will look at previous work done in the domain of Gender Identification. Konig and Morgan (1992) extracted 12 Linear Prediction coding Coefficients (LPC) and the energy feature for every 500 ms and used a Multi-Layer Perceptron (MLP) as a classifier for gender detection [1] and obtained good results. Vergin and Farhat (1996) used the first two formants estimated from vowels to classify gender based on a 7 second sentences reporting 85% of classification accuracy on the Air Travel Information System (ATIS) corpus (Hemphill Charles et al., 1990), containing specifically recorded clean speech [2]. Parris and Carey (1996) combined pitch and HMM for gender identification reporting results of 97.3% [3]. Their experiments have been carried out on sentences of 5 seconds from the OGI database. [4] Reports studies on the behavior of specific speech units, such as phonemes, for each gender were carried out. This overview of the existing techniques

for gender identification shows that the reported accuracies are generally based on sentences from 3 to 7 seconds obtained manually. We have implemented Azghadi, S. M. et.al.[5], where speech segments have 1 second length and obtained 96.49% accuracy.

In this part of the section we will focus on recent work using recurrent neural network for speaker recognition, in particular work on long short term memory units. To the author’s knowledge, there are two main work recently in the domain of using LSTM for speaker recognition, [7] also explores the effectiveness of LSTM in this domain, specifically it compares LSTM with simple RNN and DNN. They also address the issues of scalability with LSTM, that is, having deep LSTM based architectures dramatically increases the number of parameters to train. The propose architectures introduces a recurrent projection layer between the LSTM layer (which itself has no recursion) and the output layer. Which decreases their training time considerably. [8] Uses Bidirectional LSTM, with peephole connections on similar dataset as our work and obtained similar results 93% on text-independent dataset.

### 3 RNN and LSTM

In this section we will look at the basic intuition behind RNNs and LSTMs and compare them.

RNNs in contrast to feed forward network are cyclic, and they can be interpreted as having feedback through time steps. This makes them effective in learning sequential information, as this gives them a form of memory across time. Typical RNNs have direct connection from the output of the network in the previous timestep to the current timestep. Unfortunately, this kind of direct connection cannot be effectively used for learning relations in the time sequence spanning a large number of timesteps (say 100). The root of the problem is in the way the networks are trained which is BPTT, having a lot of back propagations causes vanishing and exploding gradients problem which hamper the training over large timesteps. The solution that has proven to give the best results up till now is named Long Short-Term Memory, introduced by Hochreiter and Schmidhuber [9].

A recurrent network can be said to store information in a combination of long- and short-term memory. The short-term memory is formed by the activation of units, containing the recent history of the network. The longterm memory is instead formed by the slowly changing weights of the unit transitions that are holding experience based information about the system. Long Short-Term Memory is an attempt to extend the time that a RNN can hold important information. Since its invention [9] LSTM has gotten improved with several additions to its structure. The enhancements have, as mentioned earlier, been forget gates [10] and peephole connections [11]. The version of LSTM we use does not use peephole connections and is called as Fast-LSTM– since it has lesser number of weights it trains faster. Performance wise the author believes there is a very little variation in the types of LSTM that are used in practice.

Instead of the hidden nodes in a traditional RNN, see Figure 1, an LSTM RNN makes use of something called memory blocks. The memory blocks are recurrently connected units that in themselves hold a network of units. Inside these memory blocks is where the solution to the vanishing gradient problem lies. The memory blocks are made up of a memory cell, an input gate, an output gate and a forget gate, these gates learn when to update the memory, output the information stored in the memory and when to forget the memory respectively. The equations governing the memory unit are shown in Equation 1.

$$\begin{aligned}
 i[t] &= (W[x \rightarrow i]x[t] + W[h \rightarrow i]h[t1] + b[1 \rightarrow i]) \\
 f[t] &= (W[x \rightarrow f]x[t] + W[h \rightarrow f]h[t1] + b[1 \rightarrow f]) \\
 z[t] &= \tanh(W[x \rightarrow c]x[t] + W[h \rightarrow c]h[t1] + b[1 \rightarrow c]) \\
 c[t] &= f[t]c[t1] + i[t]z[t] \\
 o[t] &= (W[x \rightarrow o]x[t] + W[h \rightarrow o]h[t1] + b[1 \rightarrow o]) \\
 h[t] &= o[t]\tanh(c[t])
 \end{aligned} \tag{1}$$

108  
 109  
 110  
 111  
 112  
 113  
 114  
 115  
 116  
 117  
 118  
 119  
 120  
 121  
 122  
 123  
 124  
 125  
 126  
 127  
 128  
 129  
 130  
 131  
 132  
 133  
 134  
 135  
 136  
 137  
 138  
 139  
 140  
 141  
 142  
 143  
 144  
 145  
 146  
 147  
 148  
 149  
 150  
 151  
 152  
 153  
 154  
 155  
 156  
 157  
 158  
 159  
 160  
 161

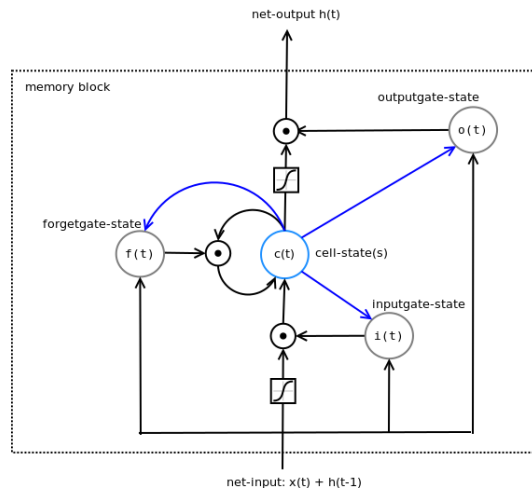


Figure 1: One memory unit of a LSTM. Shown in blue are the peephole connections which we have not used to make the network train faster.

## 4 Experimental Setup

### 4.1 Gender detection

The detection involves two parts, first is the feature extraction and next is our classifying based on a feed forward neural network. In general, female speech has higher pitch (120 - 200 Hz) than male speech (60 - 120 Hz) and could therefore be used to discriminate between men and women. Another important feature is short term acoustic features which describe the spectral components of the audio signal. Fast Fourier Transform can be used to extract the spectral components of the signal. However, such features which are extracted at a short term basis (several ms) have a great variability for the male and female speech and captures phoneme like characteristics which is not required. For the problem of gender classification, we actually need features that do not capture the linguistic information such as words or phonemes. Since, fourier transform can capture the discriminating features we choose this as our only feature.

#### 4.1.1 Dataset

In the work we have used the Harvard-Haskins database of Regularly-Timed Speech. This database contains acoustic data from speakers who were uttering sequences of alternating syllables in an evenly-timed fashion. The database contains acoustic data from 6 speakers (3 male, 3 female). All utterances in this database have 11 syllables, and consist of the syllable /ba/ alternating with another syllable.

The data processed were divided in training (70%), validation (15%) and test (15%) sets randomly. Validation curve was used to decide which iteration to stop during training thus to prevent overfitting.

#### 4.1.2 Feature extraction

Each of the datafiles are about 49 seconds with the sampling frequency of 1000Hz. The data were divided into smaller speech samples (uniformly) of at-most 4seconds (the data were padded with zeros if the sample is smaller than 4s). Fourier transformation was performed using FFT for each of the data (4s). Since the FFT signal for symmetric we discard the second half the signal and scale the absolute value of the signal according to the length of the data, to discount the effect of padding. The training data were then normalized each in the features to the range of [0,1]. The feature vectors are of length 2049.

162 Previous works in gender classification [5] indicate that the MLP with one hidden layer with 11 hid-  
163 den neurons performs the best. We use cross entropy cost function with sigmoid activation function  
164 for the neurons.

165 The project's second part deals with the speaker recognition, which will be done in a data set con-  
166 taining words or sentences the speaker will be speaking. The authors thought it would be better  
167 to work with phoneme data before going to higher level (sentences), since the phonemes are the  
168 fundamental building block of the human speech the algorithm.

### 170 4.1.3 Platform

171  
172 The training was done in Matlab's Neural Network Toolbox using the batch gradient descent method.  
173 Overall training took 163 seconds for 188 epochs in CPU 64-bit i3 processor with 2.2GHz using a  
174 single core.

## 175 4.2 Speaker recognition

### 177 4.2.1 Dataset

178  
179 In this work, we have created dataset from Coursera lectures. Our dataset consists of acoustic data  
180 from 12 male speakers. It was thought that using speakers of the same sex would be a greater  
181 challenge for the network, compared to doing the research with a research base of mixed female and  
182 male speakers. The language spoken on all of the audio books is English. However, some speakers  
183 use a British accent and some American. All the audio files used were studio recorded. Thus, they  
184 would not represent a real life situation with regards to background noise, for example

185 We chose two lectures of 10-12 minutes for each speaker, which amounts to 20-24 minutes of audio  
186 for each speaker. We split the available data to widely adopted 70-15-15 percentage where training  
187 data constitutes 70

### 188 4.2.2 Feature extraction

189  
190 The sound waves must be processed and converted into a set of discrete features that can be used as  
191 input to the LSTM neural network. In this thesis, the features withdrawn from the sound waves were  
192 Mel Frequency Cepstral Coefficients (MFCC) together with their differentials and accelerations, i.e  
193 Delta and Delta-Delta coefficients. By their characteristics, they represent features of speech signals  
194 that are important for the phonetic information. These features are withdrawn from the short time  
195 power spectrum and represent the characteristics of the signal that are emotionally independent.  
196 From every frame, 13 MFCC coefficients were extracted using a set of 26 filter-bank channels. To  
197 better model the behavior of the signal, the differentials and accelerations of the MFCC coefficients  
198 were calculated. All these features were combined into a feature vector of size 39. The feature  
199 vectors served as input to the neural network.

### 200 4.2.3 Network Architecture and Training

201  
202 The speaker recognition work in [2], analyzed the classification and concluded that the network  
203 using two hidden layers with 25 Bidirectional-LSTM units each performed best. We decided to  
204 experiment with same hyperparameters, but decided to change the type of LSTM, we used Fast-  
205 LSTM with same hyperparameter and finally a non-recurrent unit to convert the number of outputs  
206 from 25  $\rightarrow$  12, with a softmax classifier on top of it. The final network is as shown below:

```
207 nn.Sequential @ nn.Recursor @ nn.Sequential {  
208   [input -> (1) -> (2) -> (3) -> (4) -> output]  
209   (1): nn.FastLSTM(39 -> 25)  
210   (2): nn.FastLSTM(25 -> 25)  
211   (3): nn.Linear(25 -> 12)  
212   (4): nn.LogSoftMax  
213 }
```

214  
215 Frameworks like 'nn.Sequential', 'nn.Sequencer' and 'nn.Recursor' [12] are used to facilitate the  
min-batch training. Training was done using a backpropagation length of 20 time steps, with batch-

216  
 217  
 218  
 219  
 220  
 221  
 222  
 223  
 224  
 225  
 226  
 227  
 228  
 229  
 230  
 231  
 232  
 233  
 234  
 235  
 236  
 237  
 238  
 239  
 240  
 241  
 242  
 243  
 244  
 245  
 246  
 247  
 248  
 249  
 250  
 251  
 252  
 253  
 254  
 255  
 256  
 257  
 258  
 259  
 260  
 261  
 262  
 263  
 264  
 265  
 266  
 267  
 268  
 269

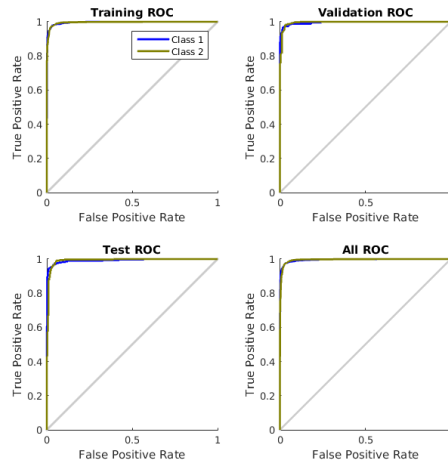


Figure 2: ROC graph of the classifier for gender detection

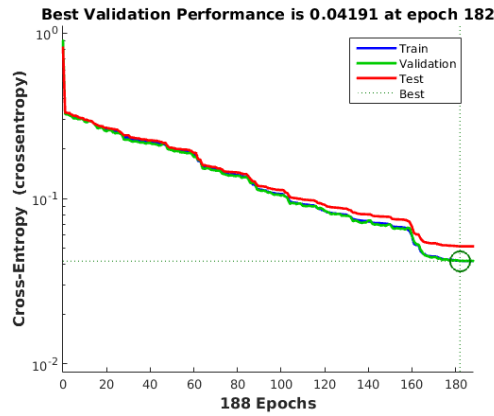


Figure 3: Training, test and validation error vs Epochs. Best model obtained at 182 epoch, for gender detection

size of 12 and a momentum of .9 (high momentum to avoid local minima). The learning rates were set according to the following Equation 2, where  $i$  is the epoch number.

$$LearningRate = 0.005 - i * \frac{0.005 - .000001}{250} \tag{2}$$

#### 4.2.4 Platform

We have built our project using torch7 framework. Computation was done on Intel Core i7-6700HQ CPU with 15.5 GB RAM and NVIDIA GPU of 4GB. Training took approximately 17.5 hrs.

## 5 Results and Discussion

### 5.1 Gender detection

Best Performance was obtained by training with 182 epochs– Figure 3 , and from the test produced 96.48%. From the confusion matrix– Figure 4, it can be seen that the network performs relatively similar for male and female. From ROC curve– Figure 2, it can be inferred that the classifier is close

270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323

	Predicted as Male	Predicted as Female
Actually Male	282	14
Actually Female	8	322

Figure 4: Confusion matrix for gender detection

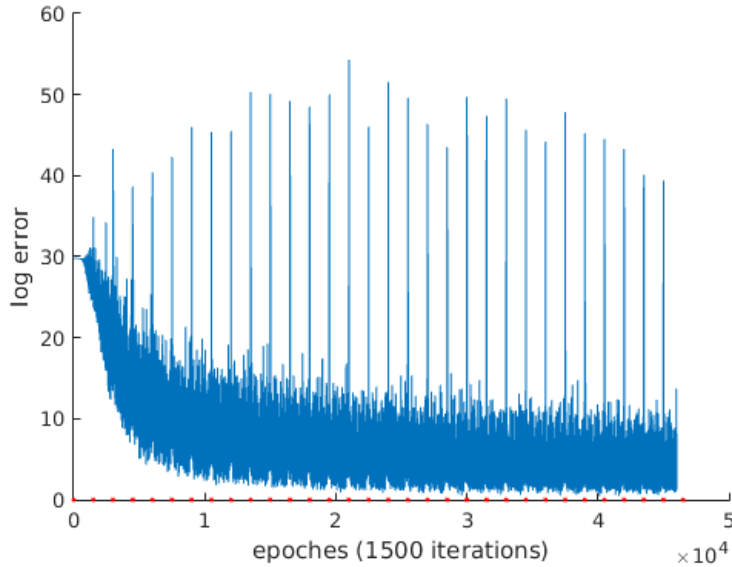


Figure 5: Training error

to perfect. Analyzing the errors of training and validation curve it can be noticed that the classifier during the training has converged. It should be noted that experiments using higher or lower number of neurons in the hidden layer were tested and discarded as, using a higher number of neurons didn't improve the accuracy by expected amount and lowering the number of neurons led to decreased accuracy.

## 5.2 Speaker recognition

Figure 5 shows the training error vs the epochs. From the training it is observable that the network has converged. The big spikes in the training error graphs are exactly one per epoch, so the authors think that this is some small (< 250 ms) non-verbal audio section in one of the datasets.

From the confusion matrix it can be seen that significant confusion occurred between classes 3 and 5 i.e speakers 3 and 5. Speaker 5s audio is confused 892 times with speaker 3s audio and speaker 3s audio is confused 939 times with speaker 5s audio. After listening to their respective voices manually, it is noticed that significant overlap exists in their tone and accent at certain points, hence the significant confusion caused could be justified. Speaker 3 and speaker 5 were **Dr.Gerhard Wickler** from The University of Edinburgh, and **Dr. Magnus Egerstedt** from Georgia Institute of Technology.

## 6 Conclusion

The results we obtained in a text-independent dataset were less ideal than the ones used in [2], even then we obtained similar accuracy, using LSTMs with lesser complexity that is we have used LSTMs without peephole connections while the former used Bidirectional-LSTMs with peephole connections. With which we can conclude that LSTMs are much more capable in performing se-

324	5366	2	163	114	26	10	62	0	28	0	0	229
325	0	5983	0	0	0	0	0	0	2	0	15	0
326	0	0	4668	35	939	0	42	7	9	0	4	287
327	0	9	269	5525	0	0	86	0	40	2	43	26
328	29	0	892	26	4928	8	0	5	12	0	29	71
329	0	0	33	2	26	5935	3	0	0	0	1	0
330	0	1	0	6	0	0	5989	0	4	0	0	0
331	2	51	0	33	1	0	0	5716	167	0	2	28
332	0	19	0	38	0	1	228	18	5687	0	1	8
333	0	4	309	56	0	0	50	0	0	5581	0	0
334	0	2	0	0	42	0	0	0	1	0	5955	0
335	0	0	207	5	85	0	0	0	0	0	13	5690

Figure 6: Confusion matrix for speaker recognition

quantual analysis that what was predicted in [2]. And more explorations on LSTMs have to be done. Hence the authors believe that the state-of-the-art variations in LSTM bring changes only in terms of training time but not a significant improvement in performance.

## References

- [1] Konig, Y. and Morgan, N., GDNN a Gender Dependent Neural Network for Continuous Speech Recognition, International Joint Conference on Neural Networks, 1992. IJCNN, Vol. 2, 7-11, pp. 332-337.
- [2] Rivarol, V., Farhat, A., and OShaughnessy D., Robust Gender-Dependent Acoustic-Phonetic Modelling in Continuous Speech Recognition Based on a New Automatic Male Female Classification, Proc. Fourth International Conference on Spoken Language, 1996. ICSLP 96, Vol. 2, 3-6, pp. 1081-1084.
- [3] Parris, E.S. and Carey, M. J., Language Independent Gender Identification, Proc IEEE IASSP, pp. 685-688.
- [4] Martland, P., Whiteside, S.P., Beet, S.W., and Baghai-Ravaei, Analysis of Ten Vowel Sounds Across Gender and Regional Cultural Accent Proc Fourth International Conference on Spoken Language, 1996. ICSLP 96, Vol. 4, 3-6, pp. 2231-2234.
- [5] Mostafa Rahimi Azghadi, S. and Reza Bonyadi, M. and Shahhosseini, Hamed, Gender Classification Based on FeedForward Backpropagation Neural Network, Artificial Intelligence and Innovations 2007: from Theory to Applications: Proceedings of the 4th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI 2007), Springer US, Boston, MA, 299304.
- [6] Alex Graves. Rnnlib: A recurrent neural network library for sequence learning problems. <http://sourceforge.net/projects/rnnl/>.
- [7] H. Sak, A. Senior, and F. Beaufays, Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, Tech. Rep. 14021128v1 [cs.NE], Feb. 2014, arXiv.
- [8] Larsson, Joel. "Optimizing text-independent speaker recognition using an LSTM neural network." (2014).
- [9] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. 1997.
- [10] F.A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: continual prediction with lstm. In Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470), volume 2, pages 850855 vol.2, 1999. doi: 10.1049/cp:19991218.
- [11] Felix A. Gers, Nicol N. Schraudolph, and Jurgen Schmidhuber. Learning precise timing with lstm recurrent networks. J. Mach. Learn. Res., 3:115143, March 2003. ISSN 1532-4435. doi: 10.1162/153244303768966139. URL <http://dx.doi.org/10.1162/153244303768966139>.
- [12] Lonard, Nicholas, Sagar Waghmare, Yang Wang, and Jin-Hwa Kim. rnn: Recurrent Library for Torch. arXiv preprint arXiv:1511.07889 (2015).