# Speaker Recognition Using Deep Neural Networks

**Harald Stiff**
Royal Institute of Technology
hstiff@kth.se

**Carl Jernbäcker**
Royal Institute of Technology
carljer@kth.se

## Abstract

This report covers the problem of speaker recognition using deep neural networks. Several unsupervised and supervised machine learning algorithms have been applied to this field with various success. In this report you will read about how the neural networks were implemented and how they performs when classifying the 112 speakers of the TIDIGIT database. Also the feature extracting process is described showing how the MFCC features and mel filter bank features are extracted. Results show that using mel filter bank features and MFCC features as input a peak accuracy of 39% and 26% was achieved respectively for classifying all of the speakers.

## 1 Introduction

Speaker recognition is an interesting topic, not only because of the demand of different biometric footprints but also in the context of validating new machine learning algorithms. Speaker recognition falls into two categories, speaker identification and speaker verification. Speaker verification concerns the problem of verifying if an utterance belongs to a certain person or more formally a specific template. The system is not concerned about details that differ one incorrect voice from another since it is a binary classifier. Systems like this are of use in biometric verification enabling users to unlock anything from doors to cellphones using their voices. To add more security a combination of text based verification might also be used forcing the user to say a passphrase in order to not be rejected by the system.

In contrast to the verification task, speaker identification concerns the problem of identifying a voice from an utterance. The main difference here is that the system is capable labeling $N$ speakers instead of just one in the speaker verification task. These systems can be applied more generally, for instance airport employees do not have access to the same secured locations of an airport. Introducing a speaker identifier lets the system unlock user specific locations which only the identified speaker has access to. For maximum security the system is pre trained and knows the speakers it should identify. More sophisticated systems are also able to identify new unlabeled speakers. This is an important issue for human like robots that constantly learns and identifies speakers to enable it to know how to interact with the speaker.

### 1.1 Objective

This project is part of the DT2119 course at KTH. The objective of the project is to investigate how good Deep Neural Networks (DNNs) are at identifying speakers from extracted features from their speech signals. In this report the theory of the DNNs used in the speaker recognition system is presented. Moreover, the preprocessing maneuvers applied to the data are shown and explained. Lastly the performance of the trained DNNs are shown with respect to different parameter settings and accuracy measurements.

## 2 Background

Speaker identification is split into two categories, supervised speaker recognition and unsupervised speaker recognition. The latter revolves around clustering data using various segmentation techniques. One straight forward way is to apply the K-means clustering algorithm Wilpon and Rabiner [1985] to extracted speech features using $N$ centroids to define the different speakers. A method that has reached more popularity is the use of Gaussian mixture models Reynolds et al. [2000] where several independent Gaussian models are used to model the distributions of each speaker. An i-vector approach is also commonly used as by Dehak et al. [2011]. This method involves extracting a large feature vector from an entire utterance. However, it suffers from the drawback that its robustness decreases drastically with reduced amount of data. Also, since the feature vector is computed after an utterance fast real time applications can not be implement. Lately, the implementation of deep neural networks in speech classification tasks such as language identification Gonzalez-Dominguez et al. [2015] has increased. By training neural networks on frame level feature vectors the network can perform any wanted speech classification task. The main advantage of this approach is that speech signals can be classified at 10 ms frames which makes it possible for real time applications. If a more accuract system is wanted more feature vectors can be used in the classification. This introduces a new design parameter which is a drawback between speed and accuracy. A further improvement to the neural network model using raw acoustic feature vectors is to use phonetically based features which have been shown in Tang et al. [2018] to improve the performance of language identification systems.

## 3 Tools

All of the code is done in Python, the following modules of the python library are used in the code. Below all versions and modules that were used are listed.

- Python 3.6.2
  - Numpy 1.14.3
  - Tensorflow 1.6.0
  - Keras 2.1.6
  - pysndfile 1.1.0
  - sklearn 0.0

## 4 Dataset

The neural networks of this paper were trained on the TIDIGITS database by Leonard and Doddington [1993] which was designed for speaker independent digit recognition experiments. Due to the large amount of speakers and the amount of utterances of each speaker it fits well into being used for training deep neural networks in the context of speaker recognition. TIDIGITS contains of utterances of one or more digits. Only one speaker is present in each recording and the recordings are free from any substantial noise.

## 5 Method

### 5.1 Artificial Neural Networks

Consider the $N$-layered artificial neural network in figure 1 which maps the input feature vector $\boldsymbol{x}$ to the output vector $\hat{\boldsymbol{y}}$ by

$$f(\boldsymbol{x};\Theta) = \text{SoftMax}(\boldsymbol{W}^N\sigma(\boldsymbol{W}^{N-1}\sigma(\dots\sigma(\boldsymbol{W}^1\boldsymbol{x}+\boldsymbol{b}^1))+\boldsymbol{b}^{N-1})+\boldsymbol{b}^N) \tag{1}$$

where $\Theta = \{\boldsymbol{W}^i, \boldsymbol{b}^i\}_{i=1}^N$ contains the parameters of the network and $\sigma$ is an element wise nonlinear activation function of choise.

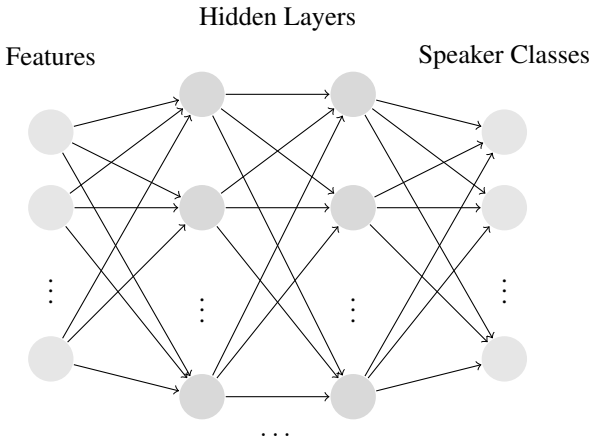


Figure 1: An artificial neural network.

We use the mini-batch cross entropy loss function

$$J(\mathcal{B}, \Theta) = \frac{1}{n} \sum_{i=1}^{n} -\log(\boldsymbol{y}_i^T \hat{\boldsymbol{y}}_i) \tag{2}$$

as a measure of how close $\hat{\boldsymbol{y}}$ predicts $\boldsymbol{y}$ where $\mathcal{B} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$ is a stochastically chosen $n$ sized batch of features and their corresponding correct onehot encoded output labels. To optimize the network's ability to map the input speech feature vector $\boldsymbol{x}_i$ to its corresponding speaker class $\boldsymbol{y}_i$ the parameters in $\Theta$ are trained using the ADAM[1] optimizer presented by Kingma and Ba [2014] provided the gradients

$$\left\{ \frac{\partial J(\mathcal{B}, \Theta)}{\partial \boldsymbol{W}^i}, \frac{\partial J(\mathcal{B}, \Theta)}{\partial \boldsymbol{b}^i} \right\}_{i=1}^{N}. \tag{3}$$

We define the single feature accuracy as the fraction between the number of correctly classified feature vectors and the total number of classifications. Moreover we define the $m$ voted multiple feature accuracy the same way but where the predictions are calculated with

$$\hat{\boldsymbol{y}} = \frac{1}{m} \sum_{k=1}^{m} f(\boldsymbol{x}_k, \Theta) \tag{4}$$

where each feature vector $\boldsymbol{x}_k$ belongs to the same speaker class. In this case $m$ feature vectors are used for one classification.

### 5.2 Feature Extraction

The features used as inputs for the neural network are the Mel-Frequency Cepstral Coefficients (MFCCs) and the mel filterbank features. The standard implementation of computing these features is shown below in the flow chart in figure 2. Each rectangle represents one computational step.

[1]The learning rate is set to $0.001$ in the adam optimizer

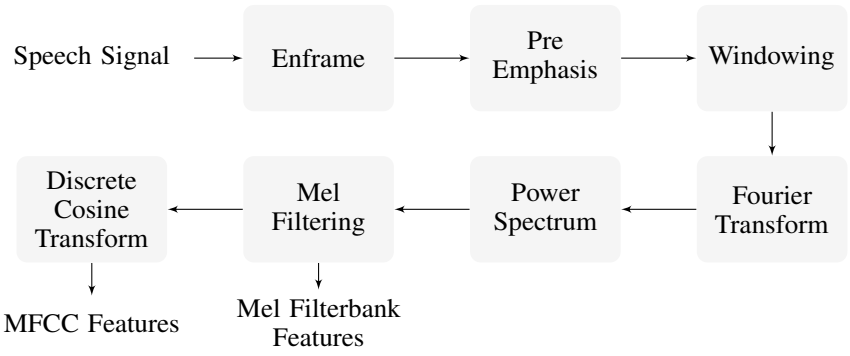


Figure 2: Computational flow to get the feature vectors.

#### 5.2.1 Enframe

Enframing of the speech signal involves decomposing it into 20 ms frames $x_i(n)$ with an overlap of 10 ms between each frame. The frames are stacked into a matrix which is further processed.

#### 5.2.2 Pre Emphasis

A pre emphasis filter is applied to each frame $x_i(n)$ to reduce the effect of the radiation of the speakers lips in the signal. The output of the filter is

$$y_i(n) = x_i(n) - 0.97x_i(n-1). \tag{5}$$

#### 5.2.3 Windowing

A hamming window

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{M-1}\right) \tag{6}$$

is applied to each frame to reduce the aliasing effect where $M$ is the number of elements in each frame.

#### 5.2.4 Fourier Transform

To transform the frames into the frequency domain a discrete fourier transform is applied to each frame. The discrete Fourier transformation of a function $f$ is

$$F(k) = \sum_{n=0}^{N-1} f(n)\, e^{-j2\pi kn/N} \tag{7}$$

where $F(k)$ denotes the Fourier transformed signal. A Fourier transformed signal shows what frequency components made up the original signal $f$.

#### 5.2.5 Power Spectrum

For a given signal, the power spectrum $S(k)$ shows how the power of the signal is dissipated within different frequencies which is applied to each frame where

$$S(k) = |F(k)|^2. \tag{8}$$

#### 5.2.6 Logarithmic Mel Filterbank Features

A Mel filterbank consists of triangular filters $T_i(k)$, $i = 1, 2, ..., M$ where

$$T_i(k) = \begin{cases} 0, & k < f(i-1) \\ \frac{k-f(i-1)}{f(i)-f(i-1)}, & f(i-1) \leqslant k \leqslant f(i) \\ \frac{f(i+1)-k}{f(i+1)-f(i)}, & f(i) \leqslant k \leqslant f(i+1) \\ 0, & k > f(i+1) \end{cases} \tag{9}$$

and

$$f(i) = 700(10^{i/2595} - 1). \tag{10}$$

By letting

$$\boldsymbol{T} = \begin{bmatrix} T_1(0) & T_1(1) & \dots & T_1(N-1) \\ T_2(0) & T_2(1) & \dots & T_2(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ T_M(0) & T_M(1) & \dots & T_M(N-1) \end{bmatrix} \quad \text{and} \quad \boldsymbol{s} = \begin{bmatrix} S(0) \\ S(1) \\ \vdots \\ S(N-1) \end{bmatrix} \tag{11}$$

the logarithmic mel filter bank features $\boldsymbol{m}$ are computed from a power spectrum $\boldsymbol{s}$ with

$$\boldsymbol{m} = \log \boldsymbol{T}\boldsymbol{s} \tag{12}$$

where the log function is applied element wise. By applying the filter bank information from critical frequency bands are extracted from the power spectrum.

#### 5.2.7 Discrete cosine transformation

To further reduce the dimensionality of the feature vector a discrete cosine transform

$$c_j = \sqrt{\frac{2}{N}} \sum_{i=1}^{N} m_i \cos\left(\frac{j\pi(i-0.5)}{N}\right) \tag{13}$$

is applied to the filter bank vector $\boldsymbol{m}$ where the coefficients $c_j$, $j = 1, 2, ..., 13$ are used for the MFCC feature vector.

### 5.3 DNN Settings

The Neural Network trained in a number of configurations, the number of hidden layers varied between 1 and 7 layers, and the activation function was always a rectified linear unit[2] Batch size was set to 256 and epochs was set to 40 during all the runs. For each hidden layer there was 256 hidden nodes in the network.

### 5.4 Pre Processing

To apply DNNs in the context of speaker recognition labeled data has to be obtained. MFCC and mel filter bank feature matrices were extracted from each utterance in the TIDIGITs dataset. Each row of features in the matrices were labeled with the speaker who made the utterance. $90\%$ of the features from each utterance were placed in the training set whereas the rest were placed in the validation set. The features in the training and validation set were extracted from random time instances of each utterance making sure that the validation set does not only contain the end silence of each of the recordings in the dataset. Global normalization was used instead of normalizing the features for each speaker at a time. The reason behind this is to make sure that the speaker dependent features does not get removed during the normalization process.

### 5.5 Multiple feature classification

When classifying using multiple features the soft max classification for each feature vector is regarded as one vote. When using $n$ feature vectors, the mean of the $n$ soft max classifications is regarded as the $n$:th vote. When the mean of the $n$ soft max vectors is calculated the argmax is taken on the vector the find the classified output class.

## 6 Results

Several measurements of accuracy can be made. Even though the networks are trained to predict the speaker corresponding to a single feature vector in reality speaker recognition systems often make use

[2]The rectified linear unit (ReLU) is the most used activation function, $f(z)$ is zero when $z$ is less than zero and $f(z)$ is equal to $z$ when $z$ is above or equal to zero, so the function is defined in the range $[0, \infty)$.

of more features when classifying a speaker. Each feature vector is extracted from 20 ms of speech, a good system would make use of several predictions from different feature vectors of an utterance to classify the speaker. The results of this section are partitioned into single feature accuracy results and then the results of how multiple features improve the accuracy of the system are presented.

### 6.1 Single Feature Accuracy

In tables 1-2 the peak test data accuracy and cross entropy loss are displayed for different networks with $n$ hidden layers of size 256 trained on different data. Rows with gender M, F, M&F correspond to networks trained on only male speakers, female speakers and both male and female speakers respectively. The difference between the tables are the features used in training where table 1 displays the results of networks trained on the mel filter bank features and 2 display the results of networks trained on the MFCC features. Accuracy corresponds to percentage of feature vectors that were classified to the correct speaker.

Table 1: Mel Filter Bank Features

| Model | h-layers | gender | acc | loss |
|---|---|---|---|---|
| 1 | 1 | M | 29.9 | 2.73 |
| 2 | 1 | F | 34 | 2.56 |
| 3 | 1 | M&F | 28.3 | 3.03 |
| 4 | 3 | M | 43.4 | 2.18 |
| 5 | 3 | F | 46 | 2.06 |
| 6 | 3 | M&F | 39.3 | 2.5 |
| 7 | 5 | M | 43 | 2.24 |
| 8 | 5 | F | 46 | 2.08 |
| 9 | 5 | M&F | 37.2 | 2.6 |
| 10 | 7 | M | 41.5 | 2.33 |
| 11 | 7 | F | 45 | 2.18 |
| 12 | 7 | M&F | 36.1 | 2.68 |

Table 2: MFCC Features

| Model | h-layers | gender | acc | loss |
|---|---|---|---|---|
| 1 | 1 | M | 23 | 2.99 |
| 2 | 1 | F | 21 | 3.12 |
| 3 | 1 | M&F | 18 | 3.52 |
| 4 | 3 | M | 33 | 2.57 |
| 5 | 3 | F | 30 | 2.73 |
| 6 | 3 | M&F | 25.6 | 3.1 |
| 7 | 5 | M | 34 | 2.55 |
| 8 | 5 | F | 31 | 2.71 |
| 9 | 5 | M&F | 26,4 | 3.1 |
| 10 | 7 | M | 33 | 2.60 |
| 11 | 7 | F | 30 | 2.73 |
| 12 | 7 | M&F | 24.9 | 3.18 |

In figures 3 the confusion matrices for classifications made by models 6 and 9 in tables 1-2 are shown respectively. Also, in Figure 4 the accuracy and cost is plotted against the epochs of training using model 5 in Table 1.

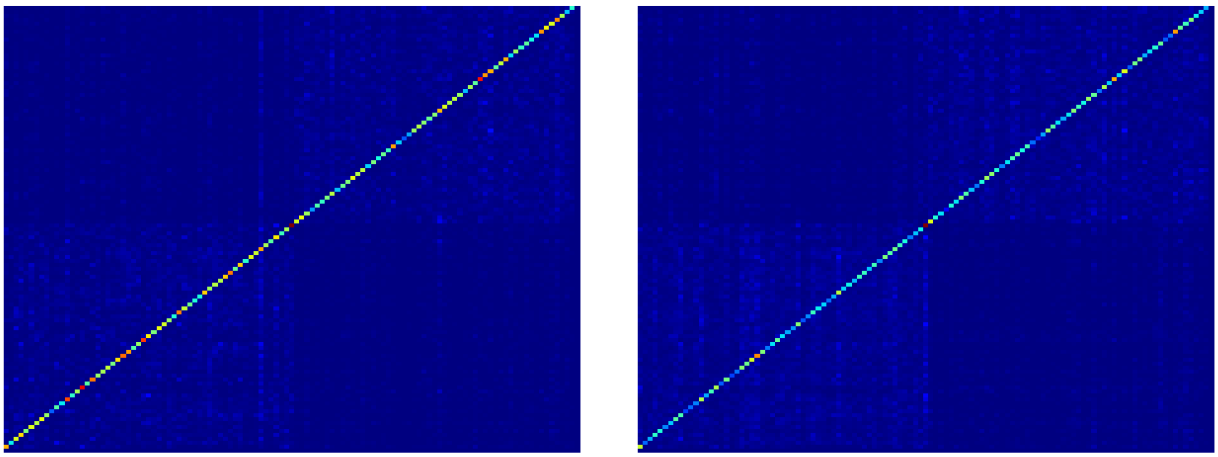

Figure 3: Confusion matrix computed for classifications made by models 6 (left) and 9 (right) in Tables 1-2 respectively.

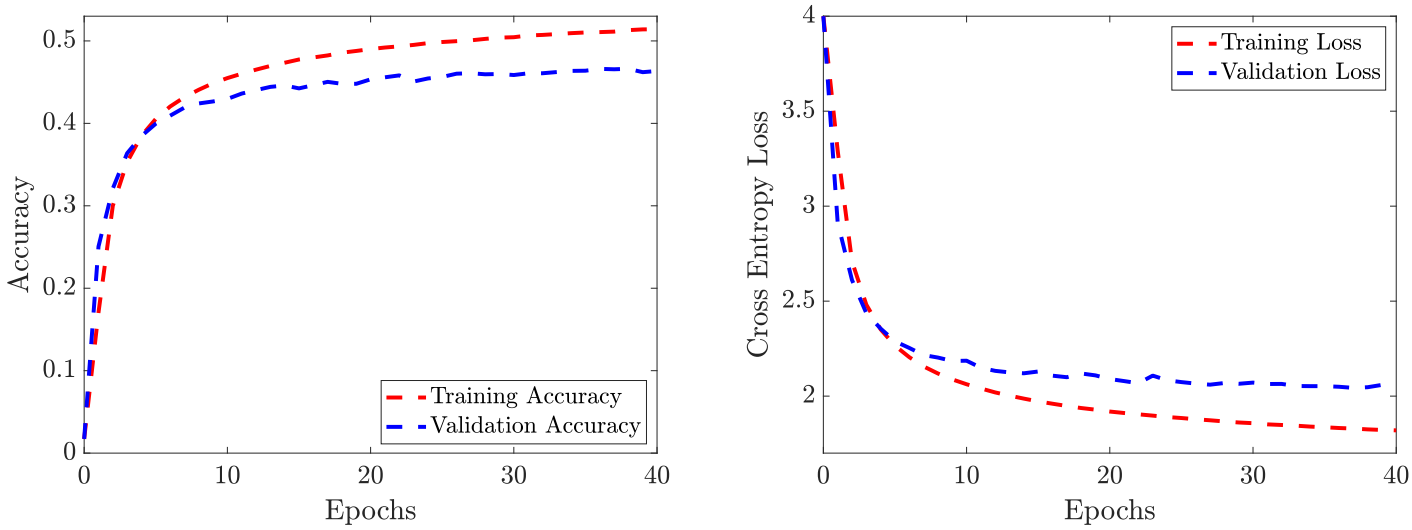


Figure 4: Loss and accuracy graph for model 5 in Table 1.

### 6.2 Multiple Feature Accuracy

In figure 6 the accuracy of models 6 and 9 in Tables 1-2 are plotted against the number of feature vector used for the classification. Moreover, figures 5 are cluster plots of predictions $\hat{\boldsymbol{y}}$ of model 6 using a different amount of votes. The reduced dimensionality is performed with principle component analysis of 50 components that are later reduced to two dimensions with the t-SNE algorithm developed by Maaten and Hinton [2008]. Only predictions of features belonging to ten males and ten females $M_i, W_i, i = 1, 2, ..., 10$ are shown in the cluster plots for increased clarity.

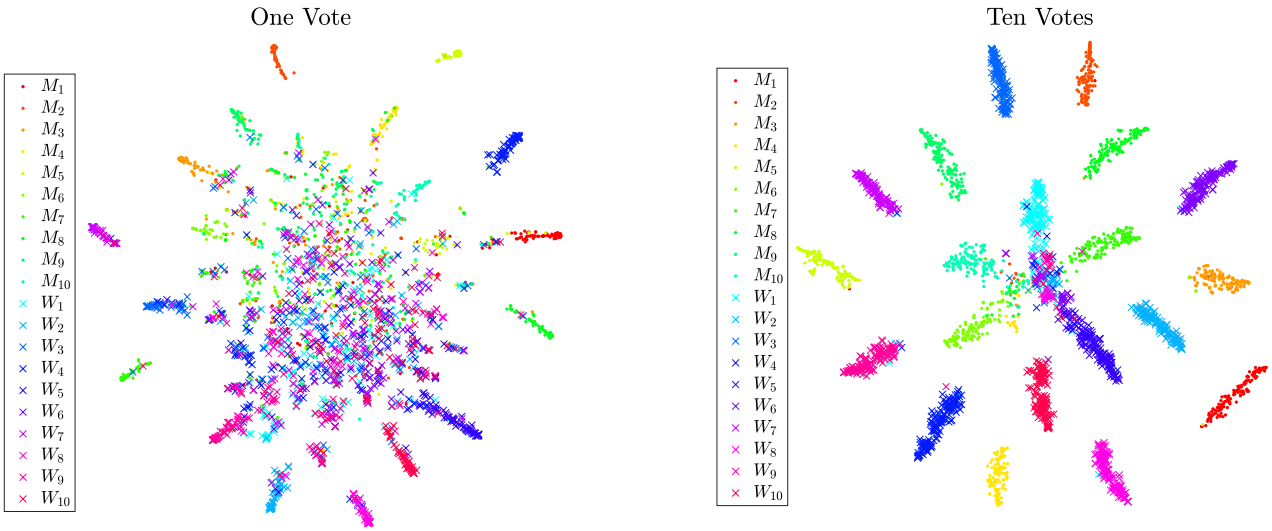


Figure 5: Two clusters showing predictions $\hat{\boldsymbol{y}}$ using one vote (left) and ten votes (right) and model 6 in Table 1.

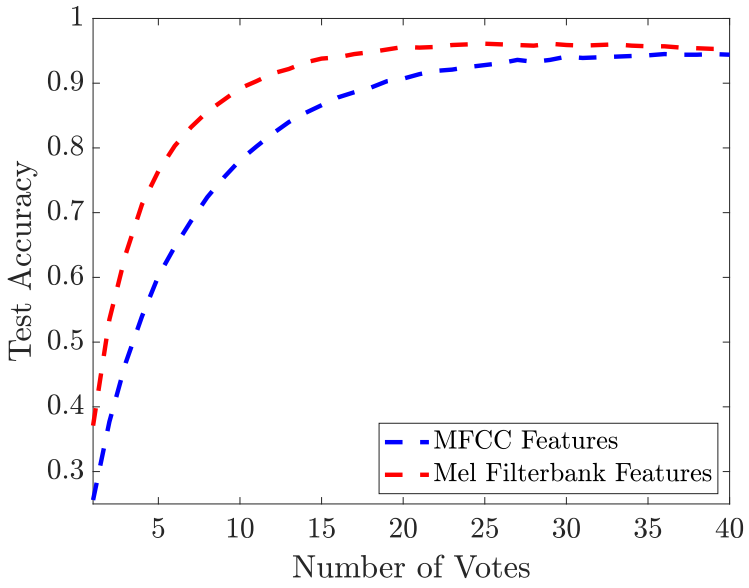


Figure 6: Accuracy over number of votes for model 6 in Table 1 (red) and model 9 in Table 2 (blue).

## 7 Discussion

When looking at the model depicted in figure 6, we see that the model converges to an accuracy of around 90% after 20 votes. We can also see that the characteristics of the model changes as well after introducing voting, when comparing figure 5 (left) which has only one vote with figure 5 (right), we see that introducing votes makes the classes more separable. There still exist some overlapping of the classes in figure 5 (right), by refining methods it might be possible to separate it even further.

Looking at figure 3 (left and right), the first and second quadrant contained utterances from male speakers, and so the third and forth quadrant contained utterances from female speakers. You can see between the genders there are fewer miss classifications than there are within the same gender. This could be true due to that the male frequency register is distinguishable from the female register, and therefore separates the two classes to such a degree that the are almost no overlapping when training on both male and female voices. The same pattern is true for both methods but more prevalent in lmfcc due to its lower accuracy. You can from figure 3 (left) and figure 3 (right) deduce that the miss classification are present over all the speakers to all the speakers within each gender (quadrant one and three), this could be the result of the short utterance that the model has to categories the speaker on. Given 20 ms of utterances are hard to distinguish.

In figure 4 you can see that after 40 epochs the accuracy and loss has not yet converged, both have leveled out but not fully reached its peak performance. This was true for all the different configurations, if one would superimpose to training data for each of the tests almost all the models followed the same pattern. When we discovered that 40 epochs would not be enough for convergence a decision was made not to run more epochs, reason for this was time, we did not have enough time to do 100 epochs or more for each model. Future experiment should probably run for more epochs than we did to see how to performance would change.

When getting the results posted in table 1 and table 2, the input data to was of equivalent size of 20 ms of recording before transforming it to the frequency domain. In figure 4 this accuracy is depicted for a female speaker with input data equivalent size of 20 ms, when comparing it to figure 6 we see an sharp increase in test accuracy. This indicated that in further experiments having larger windows might be beneficial. Due to the accuracy for all test increase when adding layers to a certain degree tends to indicate that the data is not linearly separable and is prone to over fit when adding excessive layers.

## References


N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak. Language recognition via i-vectors and dimensionality reduction. In *Twelfth annual conference of the international speech communication association*, 2011.

J. Gonzalez-Dominguez, I. Lopez-Moreno, P. J. Moreno, and J. Gonzalez-Rodriguez. Frame-by-frame language identification in short utterances using deep neural networks. *Neural Networks*, 64: 49–58, 2015.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL `http://arxiv.org/abs/1412.6980`.

R. G. Leonard and G. Doddington. Tidigits speech corpus. *Texas Instruments, Inc*, 1993.

L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (Nov):2579–2605, 2008.

D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.

Z. Tang, D. Wang, Y. Chen, L. Li, and A. Abel. Phonetic temporal neural model for language identification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1): 134–144, 2018.

J. Wilpon and L. Rabiner. A modified k-means clustering algorithm for use in isolated work recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(3):587–594, 1985.