**Speaker Recognition using Neural Networks and MFCC Features**

*Hamza Zamani1, Taishi Kato1, David Rosenwasser1*

1 Department of Electrical and Computer Engineering, University of California, Los Angeles

hwzamani@g.ucla.edu, taishikato10@ucla.edu, dbrosenwasser@gmail.com

# Alternative Approach

## Neural Network Implementation

Recent proliferation of neural networks in both academia and industry have made usage of them widely available and simple to implement for research and commercial uses. The popularity of these classifiers have also helped discover various applications of these algorithms in interesting and diverse fields.

Neural networks offer an attractive solution for their ability to classify data in a particularly high dimensional feature space. Rather than statistical pattern matching, the neural networks invoke supervised learning to develop complex functions to model behavior, i.e. universal function approximation [1].



Figure : *Binary classification neural network with a feature vector of 4 parameters with a single hidden layer.*

### Architecture

Though implementation of a neural network is relatively straight forward, unfortunately there does not currently exist a method to determine *a priori* the optimal architecture of the network. Parameters such as the number of hidden layers, number of neurons per layer, etc. are all subject to user tuning and is highly dependent on the nature of the input data. An example of a simple neural network is depicted in Figure 1. These open-loop parameters along with the possibility of the algorithm settling to some local minima [2] can further make optimization and tuning difficult thereby making optimal architecture selection difficult and time consuming.

Given the training data sets, a network consisting of three hidden layers with 30 neurons each produced the best EER results out of the numerous configurations tested. Figure 2 depicts a top level structure of the network.



Figure : *Block diagram representation of implemented network*

The selection of the input layer size was selected on the feature vector chosen for testing. Once the feature vector size was determined, the number of input layer neurons was automatically created through the Neural Network tool in MATLAB. Next, the hidden layers consist of three layer of 30 neurons each and finally a single neuron output layer was configured for the binary classification of the speaker data. A single neuron output layer, rather than a 2 neuron output layer was chosen since fewer parameters and computation is needed.

### Training

Implementation of a neural network for speech recognition has been research rigorously, but many are employed to classify using test-dependent speech [3] or text-independent speech with a closed speaker set, some achieving 100% classification accuracy [4]. Other techniques include using a convolutional neural net to classify the spectrogram of a text-dependent closed set utterance [5]. The unique challenge for this project is the mixture of text-dependent and text-independent sample sets along with an open speaker set for the final classification.

An additional constraint is the ratio of excitatory classes in comparison of the overall training set. Table 1 shows the ratio of training vectors that have zero labels in comparison to vectors containing one labels.

|  |  |  |  |
| --- | --- | --- | --- |
| Training | Vector Size | Excitatory | % Excitatory |
| READ | 9730 | 147 | 1.51 % |
| PHONE | 11175 | 150 | 1.34 % |

Table : Ratio of excitatory targets in read and phone training sets

The nature of the training set allows for the neural network to classify all vectors as non-match speakers thereby artificially achieving a classification accuracy of 98.49% and 98.66% for read and phone training sets respectively given the training set. In reality network performs relatively poorly given any excitatory test conditions. The issue of obtaining enough training data of all class types is an added difficulty of correctly training the classifier. Though not implemented for this design, there are methods to balance the training set which are further detailed in the next section.

The Receiver Operating Characteristic (ROC) plot and confusion matrix for the net trained with read data are shown in Figure 3 and Figure 4 while performance of the net trained with the phone data set are shown in Figure 5 and Figure 6.

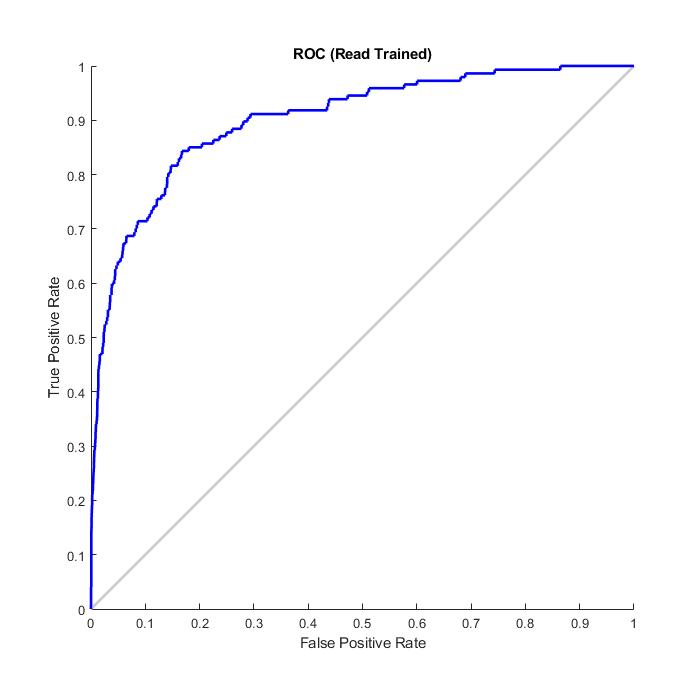


Figure : *ROC (read training set)*

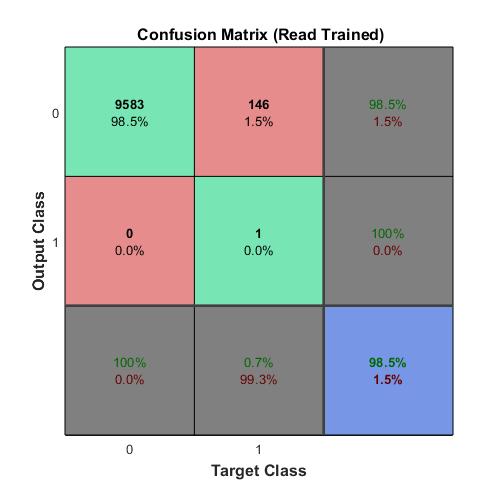


Figure : *Confusion Matrix (read training set)*

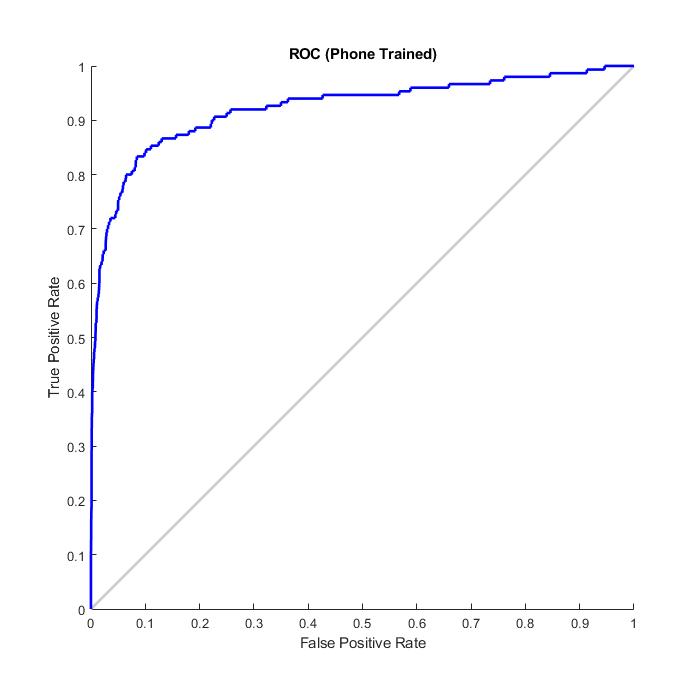


Figure : *ROC (phone training set)*

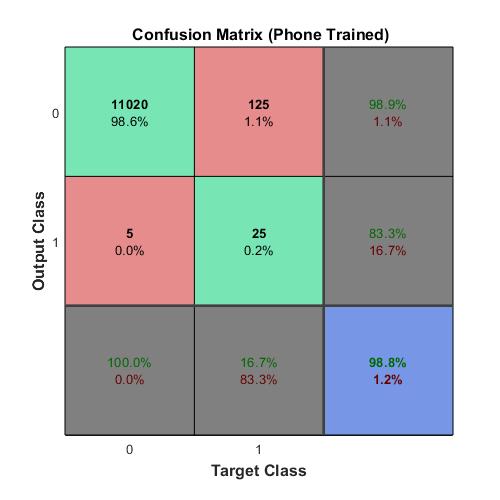


Figure : *Confusion matric (phone training set)*

It can be seen that due to the lack of balance in the classes both confusion matrices deceptively report exceptionally high performance. This miscalculation can be easily disproven by simply inputting a feature vector that corresponds to a matching speaker set. Notice for both outputs the misclassified values almost exactly match the number of matching classes presented in the read and phone training sets.

### Results

After implementation and tuning of the network, the results achieved were subpar and therefore this method was not selected for the final implementation. The calculated EER is shown in Table 2.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Read | Phone | Mismatch |
| Train Read | 30.98% | 29.94% | 46.67% |
| Train Phone | 21.67% | 26.67% | 47.59% |

Additionally, since we must train the network in order for it to classify the run times of the design as a whole also negatively impacts the metrics used for selection of a neural net for the finalized design. As the feature set grows, the longer the network takes to train to converge. Run times for both training sets are collected in Table 3. Notice the relative fast runtimes of the scripts. The speed of the runtime points towards a neural network that is not truly training correctly to the data presented. Given the same feature set, the KNN was clearly able to outperform the neural net.

|  |  |  |
| --- | --- | --- |
|  | Read | Phone |
| Mean Runtime (sec) | 16.54 | 18.01 |

Table : *Mean script runtime for both training sets*

Additional challenges were encountered simply due to the nature of the training sets and structure. Since requirements outlined two separate training tasks, a neural net architecture that performed well with the read training set would not necessarily perform well with phone training sets and vice versa. This forces the architecture configuration to be adequate for both types of training sets rather than optimized for one. If a single architecture that favoured one training set heavily, it would perform poorly given speech samples of the phone variety.

As mentioned in the previous section there are methods to balance the training data to include a more even distribution of classes to allow for proper training of the neural network. Other than collecting more matching cases future implementations could balance the data by pruning the overrepresented class, i.e. speech instances where the speaker is not the same, while keeping all the set where the speaker is the same [6]. Another possible method would be to artificially inject noise, or slightly perturb existing sets of data (with matching speakers) and add those to the training set, or more simple use mean shifting MFCC values

Though the KNN implementation outperformed the neural network given the project conditions, with an enhanced trainset and further tuning it is plausible that the neural network could outperform the KNN especially if enhanced high-dimensional features are extracted from the speech data.

# References

1. Hornik, Kurt et al. “Multilayer feedforward networks are universal approximators.” *Neural Networks* 2 (1989): 359-366.
2. D. Ringach and Y. Baram, "A Diffusion Mechanism for Obstacle Detection from Size-Change Information" in *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 14, no. 01, pp. 76-80, 1994.
3. E. Variani, X. Lei, E. McDermott, I. L. Moreno and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, 2014, pp. 4052-4056.
4. Zhenhao Ge, Ananth N. Iyer, Srinath Cheluvaraja, Ram Sundaram: “Neural Network Based Speaker Classification and Verification Systems with Enhanced Features”, 2017; [http://arxiv.org/abs/1702.02289 arXiv:1702.02289].
5. Salehghaffari, H., “Speaker Verification using Convolutional Neural Networks”, 2018; [http://arxiv.org/abs/1803.05427 arXiv:1803.05427].
6. K. R. Farrell, R. J. Mammone and K. T. Assaleh, "Speaker recognition using neural networks and conventional classifiers," in *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 194-205, Jan. 1994.