A REPORT ON

IDENTIFICATION OF HUMAN SOUNDS USING NEURAL NETWORK

BY

Komal Gupta 2015B5A3330G

AT

Indian Institute of Science, Bangalore

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

June, 2017

A REPORT ON

IDENTIFICATION OF HUMAN SOUNDS USING NEURAL NETWORK

BY

Komal Gupta	2015B5A3330G	M.Sc. Physics B.E. Electrical & Electronics

AT
Indian Institute of Science, Bangalore

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

June, 2017

Acknowledgement

I would like to express my sincere gratitude towards Prof. Anurag Kumar, Director of IISc for allowing the institute to be a part of the Practice School program. I also wish to thank my mentor, Prof. Manoj Varma, for giving me the opportunity to do this wonderful project on neural networks and for his continual guidance and supervision throughout the course of the project. In addition to this, I would like to extend my sincerest thanks to our faculty in charge for PS1, Dr. Satya Sudhakar Yedlapalli, for his constant encouragement and support during the program.

Abstract

Neural network is a biologically inspired programming paradigm modelled upon the brain. A neural network enables a computer to learn from observational data and hence solve a multitude of problems which are too difficult for conventional computers, such as image recognition, speech recognition and language processing. This project aims to create a neural network that can be used to classify human and non-human sounds. MATLAB is used to create, train and test the network. However, a neural network cannot be 100% accurate in the classification. The goal is to choose a set of parameters that minimize the error in classification.

Contents

List of Figures

1 Introduction

Recognizing a human sound and being able to differentiate it from a non-human sound is a very easy problem for us humans. In fact, we do it every day and during every conversation. One can argue that it is because on our shoulders rests the most powerful supercomputer ever - tuned by millions of years of evolution - our brain. But for conventional computers, this seemingly simple problem poses a challenge.

Since conventional computers fail to help us here and hence are off the table, we need a new programming paradigm, one which can adapt to the problem it has been asked to solve. Neural Network is one such paradigm, which is modelled upon the human brain. It enables a computer to learn from observational data, and hence can solve problems which are too difficult for conventional computers, such as image recognition, speech recognition and natural language processing.

This project aims to create a neural network which can correctly distinguish between human and non-human sounds and classify them accordingly. The next two chapters explain the choice of input parameters and the construction of the network. The fourth chapter showcases the results obtained when additional tests were performed on the network. Finally, the results are discussed in the fifth chapter. The sixth chapter concludes the report.

2 Defining Input Parameters

The Neural Networks Toolbox of MATLAB, which is used extensively in this project, requires input data in a matrix format. The comparison between two sound waves can be made by comparing several features and hence there are several ways in which the input matrix to the neural network can be chosen. This section explains how this choice is made and also lists various alternatives to the chosen features.

2.1 Choosing basis for comparison

A sound wave can be characterized by three quantities as perceived by humans: pitch, quality and loudness. These are related to physical characteristics of the sound waves, namely frequency, waveform and intensity respectively. In this project, the comparison has been made on the basis of frequency and amplitude (or intensity, since intensity is amplitude squared) of the sound waves. The range of frequencies is divided into 100 bins, and the mean amplitude (in decibels) is extracted for each bin of each sound.

2.2 Choosing features

The feature used in this project is the mean value of amplitudes in each bin of each sound. However, other features such as variance, difference between maximum and minimum amplitude etc. could have been chosen as well. This choice of feature is empirical, i.e. it depends on the problem at hand. No particular feature is certain to give best results in all cases.

2.3 Extracting feature sets

So far, it has been established that the input matrix to the neural network toolbox will contain mean values of amplitudes in bins. However, not all the bins are equally apt to be fed as inputs. Hence, in order to maximize the accuracy of the network, the 100 bins are divided into two groups: good feature set and poor feature set. This distinction is made by plotting the cumulative distribution functions (CDFs) of the mean amplitudes across all sounds, for each bin. CDFs of corresponding bins of human and non-human sounds are plotted together (i.e. on the same axes), and 100 such plots (one for each bin) are collected.

The following diagrams show two such plots, one from each feature set.

Figure 1: Example: Good feature set (Bin 1)

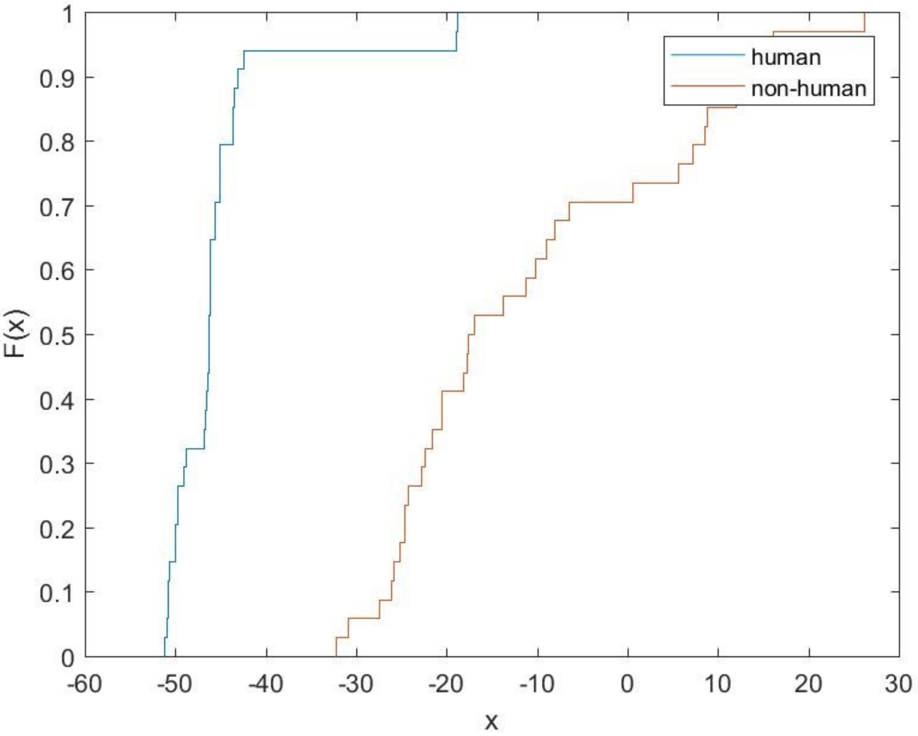
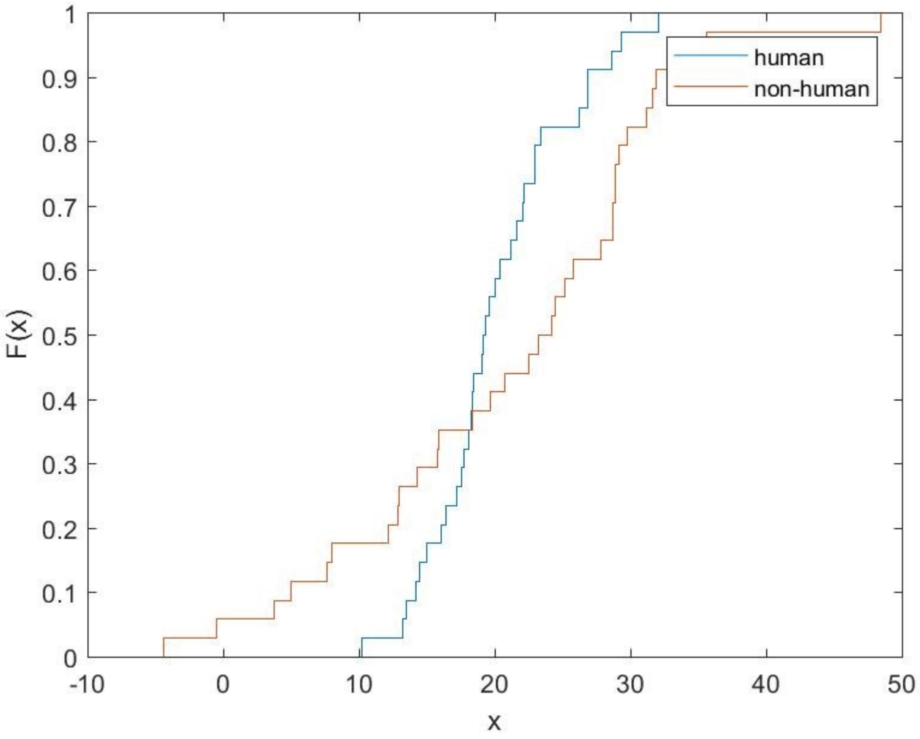


Figure 2: Example: Poor feature set (Bin 43)



The distinction between good and poor feature set is made by looking at the separation between the cumulative distribution curves of human and non-human sounds. In figure 1, the two curves do not intersect and can be seen as separated to a great degree. Bins for which the two curves are separated constitute the good feature set, of which bin 1 is an example. On the contrary, in figure 2 the two curves intersect. Bin 43 therefore constitutes poor feature set.

For a training data set of 108 audio clips (54 human sounds and 54 non-human sounds), 22 bins were regarded as good feature set, and the rest 78 bins were regarded as poor feature set. A 108 x 22 matrix[1] containing was then used to train the neural network. The next chapter explains how the network is created and tested.

[1] The entry in i$^{\text{th}}$ row and j$^{\text{th}}$ column of the matrix is the mean amplitude of j$^{\text{th}}$ bin (of good feature set) of i$^{\text{th}}$ clip.

3 Creating the Network

This chapter describes the construction of the network and its testing after the first step is completed. This process is repeated several times with different sets of input parameters[2] in order to determine the average accuracy and also to find out the best set of parameters for this problem. Furthermore, the network is also trained in a similar way using the poor feature set so as to compare the accuracy attained by the two sets. Since the problem is based on classification, all the processes hereafter (including construction of the network) involve the Neural Pattern Recognition application of MATLAB.

3.1 Construction

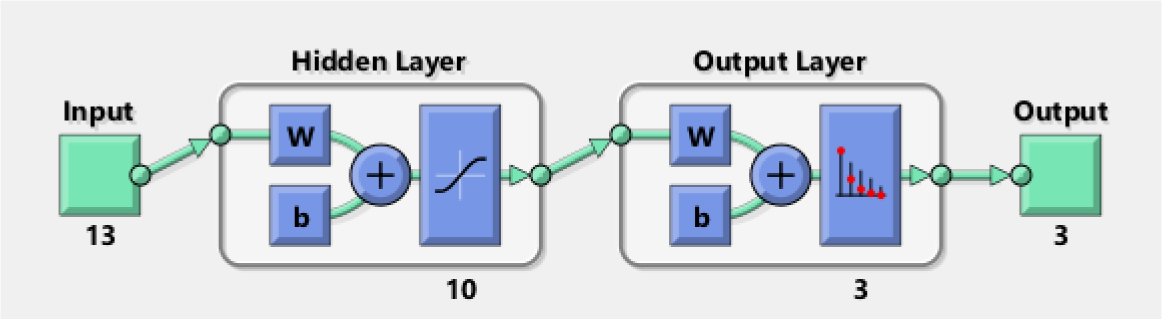
Before the network is created, the samples are randomly divided into three kinds:

- Training: These are presented to the network during training, and the network is adjusted according to its error.
- Validation: These are used to measure network generalization, and to halt training when the generalization stops improving.
- Testing: These have no effect on training and so provide an independent measure of network performance during and after training.

Next, the size of the network is adjusted by changing the number of neurons in its hidden layer. The following diagram shows a neural network with 10 hidden neurons:

[2] Input parameters include no. of neurons in hidden layer and the percentage of data split into training, validation and testing sets.

Figure 3: Example: Neural Network



After setting up these parameters the network is created by the application. Next section describes the training of the network.

3.2 Training

For each set of parameters, the network is trained repeatedly until the percent error[3] in classification is minimum[4]. Training multiple times generates different results due to different initial conditions and sampling. Once the minimum error is achieved, the network is tested as explained in the next section.

3.3 Testing

Once the network has been trained, it is tested using a new set of audio clips it has never encountered before, in order to determine its accuracy. The testing data set used in this project involved 20 audio clips consisting of 10 human sounds and 10 non-human sounds. The network was tested again and again with different input parameters and the accuracy for each run was recorded. The following chapter summarizes the results obtained after 24 repetitions of the process with different input parameters for each run.

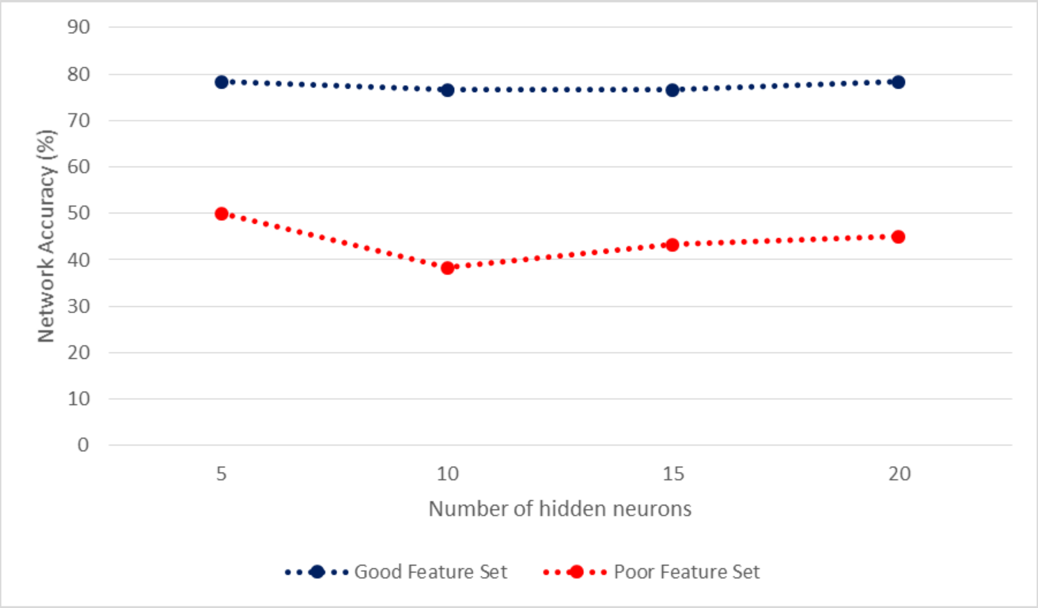
[3] Indicates the fraction of samples that were misclassified during initial testing.

[4] This is done purely by inspection.

4 Results

This section summarizes the results obtained after training the neural network. The network was created and trained 24 times in total, with different input parameters for each run. The following plot shows the variation in percentage error with number of hidden neurons for both good and bad feature sets. The distribution of samples was also changed and different percentages of the initial data set (consisting of 108 clips) were chosen for training, validation and testing each time in order to get an average value of accuracy of the network.

Figure 4: Network Performance



As can be seen from the above plot, the accuracy is maximum for the neural network with 5 hidden neurons in both the cases. The hidden layer does most of the decision-making in a neural network classifier, and one might falsely presume that increasing the number of neurons in this layer will improve the accuracy of the network. However, the results provide a sharp contradiction to this notion.

For the network trained with good feature set, the accuracy remains more or less the same with change in number of hidden neurons, and the average accuracy was found out to be 77.5%. On the other hand, for the network trained with poor feature set, the accuracy decreases to a minimum when the number of hidden neurons is 10, and later increases. The average accuracy in this case was found out to be 44.17%. As expected, the accuracy is greater for the network trained with good feature set.

Several inferences regarding the choice of feature set and the optimum size of the network can be drawn from these results which are explained in the next chapter.

5 Discussion

This chapter explains the results obtained in the previous section. The first observation from the plot is that the accuracy of the network trained with good feature set is higher than the one trained with poor feature set. This justifies the task of segregating the 100 bins into groups of good and bad features before using them to train the network. It also proves that not all the regions in the frequency spectrum of a human and a non-human clip can be compared with each other. The good feature set contains the regions in frequency spectra where the human and non-human sounds differ from each other to a greater degree and hence gives more accurate results, whereas the poor feature set contains regions where the two types of sounds do not differ greatly and hence gives less accurate results than the former.

It is also observed that the accuracy for neural network with 5 hidden neurons is maximum in both cases which then decreases when the number of hidden neurons is increased to 10. The accuracy slowly increases again after increasing the number of hidden neurons to 15 and later 20. This shows that some sort of saturation point of error is reached at around 10 hidden neurons, where the error is maximum and the accuracy is minimum. However, this trend is more obvious in the network trained with poor feature set.

These results show that a small neural network with about 5 hidden neurons can be employed to solve a classification problem such as this one accurately. Aforementioned network can thus be used in a small robot or a device with low computing power to perform the task of classifying objects.

6 Conclusion

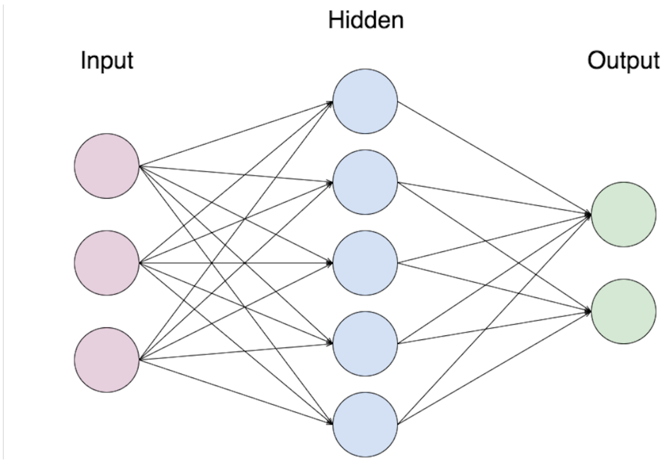
This project demonstrates that a small neural network can be used to solve classification problems accurately, and can be used in devices with low computing power, with an accuracy of around 80%. There were, however, several shortcomings which lead to a reduced accuracy of the network. While collecting training data for this project, several factors which affect the input of the neural network were not taken into consideration. This includes demographic factors such as age, race, gender and nationality of the speakers. Another significant influence is the background noise present in many audio clips which was not removed before using them. By taking these elements into consideration and also by increasing the size of the training data set, the accuracy of the network can be further improved.

Appendix: Neural Network

A neural network is a programming paradigm inspired by the nervous system in living organisms. It consists of vast number of artificial neurons, which are mathematical functions modelled upon biological neurons. Each neuron takes several binary inputs and produces a single binary output. Each input has a particular weight attached to it, which reflects the importance of the input in determining the output. The neurons also have an associated threshold value, which decides whether the output will be 0 or 1. By varying the weights and the threshold, we can get different models of decision-making.

A typical neural network consists of 3 or more layers, as can be seen in the following diagram:

Figure 5: Neural Network Architecture



The input layer receives information from the outside world. The neurons in this layer make simple decisions by weighing the input evidence. The hidden layers do most of the decision-making, by weighing up results from the previous layer. In more complicated networks, there can be multiple hidden layers. Finally, the output layer signals the response of the network to the inputs. In this way, sophisticated decisions can be made by a neural network.

Bibliography

- Michael A. Nielsen, “Neural Networks and Deep Learning”, Determination Press, 2015

- “Neural Network Toolbox Documentation – MathWorks India”. *https://www.mathworks.com/help/nnet*

- “Neural Networks in JavaScript”. *https://blog.webkid.io/neural-networks-in-javascript/*

- “Characteristics of Sound”. *http://personal.cityu.edu.hk/~bsapplec/characte.htm*

- “How do you characterize sounds?” *http://www.dosits.org/science/sound/characterizesound/*