Graphic Era Deemed to be University

Dehradun, Uttarakhand

A Mini Project Report

on

Voice based gender identification using deep learning

Submitted by:

SARTHAK SAHAI

B. Tech CSE

Semester IV

2017000

Department of Computer Science and Engineering

JULY,2022

PROBLEM STATEMENT

Gender recognition based on speech signal is a crucial task in the context of content-based multimedia indexing. We demonstrate that the combined performance of features and classifiers outperforms that of any single classifier. Based on these findings, we developed a method for gender identification in multimedia applications. The system employs a group of neural networks containing features relevant to pitch and acoustics. Practical factors like speech continuity and using a variety of experts rather than just one have been demonstrated to increase classification accuracy to 93 percent. For segments of 5 seconds, the categorization accuracy reaches 98.5 percent when applied to a portion of the Switchboard database.

INTRODUCTION

The parameter choices for sample properties including intensity, duration, frequency, and filtering are specific to acoustic analysis of the voice. The gender of the speaker can be determined using the acoustic characteristics of their voice and speech. The R package warble R was created for acoustic analysis. This study can be used to acquire the data set that contains acoustic parameters. Different machine learning algorithms can be used to train the data set. MLP has been utilized in this paper to obtain a model. The outcomes were contrasted with similar research. With the help of the acquired model, a web page has been created to identify the voice's gender. In general, gender identification can be done using a speech and voice recognition system. The human ear serves as a natural voice recognition mechanism. The gender of a voice or piece of speech can be accurately determined by the human ear using factors like frequency and volume. Similarly, by selecting and combining the appropriate features from voice data using a machine learning algorithm, a machine can be trained to perform the same task.

OBJECTIVE

By analyzing speech signals, a technique known as gender recognition can be used to ascertain a speaker's gender category. Using voice signals extracted from a recorded speech, acoustic properties including duration, strength, frequency, and filtering can be acquired. Speech emotion detection, human-machine interaction, telephone call categorization by gender, automatic salutations, muting sounds for a gender, and audio/video categorization with tagging are some applications where gender recognition can be helpful.

MOTIVATION

One of the most important uses of speech signal processing is speech recognition. It is regarded as a well-liked and trustworthy technique for identification. We need to employ effective techniques for speech signal processing to get a reliable and high level of accuracy in speech recognition.

Applications for speech signal processing are numerous. Gender identity is crucial to speech processing. Gender classification has drawn a lot of attention considering the current global security concern. Narrowing down the scope of investigations in crime situations, healthcare systems for vocal fold cyst identification, etc. are some examples of applications. It can also be applied as a feature in a virtual assistant that can identify the gender of the talker.

DATASET

We'll make use of Mozilla's Common Voice Dataset, which is a collection of audio files that users of the Common Voice website have read. Its goal is to make automatic speech recognition training and testing possible. But when I looked at the dataset, I saw that many of the samples had labels in the genre column. As a result, we can perform gender recognition on these tagged samples after extraction.

I did the following to get the dataset ready for gender recognition:

1. I started by only filtering the samples that had labels in the genre field.
2. The dataset was then balanced to include the same number of female samples as male samples, which will prevent the neural network from being overfit on a specific gender.
3. Finally, I extracted a vector of length 128 from each voice sample using the Mel Spectrogram extraction method.

REQUIRED LIBRARIES

- Python is a dynamic type, interpreted, interactive, object-oriented, open source, simple to learn programming language. Python has impressive capability and relatively plain syntax.
- Developed in Python, Keras is a high-level neural network library that can be used with either TensorFlow or Theano.
- An open source software package called TensorFlow™ is used to compute numerical data using data flow graphs. The graph's nodes stand in for mathematical processes, while the graph's edges stand in for the multidimensional data arrays (tensors) that are exchanged between them [14]. Due to TensorFlow's adaptable architecture, it is possible to undertake machine learning and deep neural network research using either a GPU or CPU, though it is also simple to adapt to new domains.

- The open-source foundational Python module for scientific computing is called NumPy. N-dimensional array objects, advanced (broadcasting) functions, facilities for integrating C/C++ and Fortran code, effective linear algebra, Fourier transform, and random number capabilities are only a few of the strong features it has. Any data type can be defined with Numpy. As a result, NumPy can quickly and easily interact with a wide range of databases. Numpy is used by Keras for input data types.
- Pandas is an open-source library designed primarily for working quickly and logically with relational or labelled data. It offers a range of data structures and procedures for working with time series and numerical data. The NumPy library serves as the foundation for this library.
- The measurements and visualizations required for the machine learning workflow can be provided via Tensor Board. It makes it possible to visualize the model graph, follow experiment metrics like loss and accuracy, project embeddings into a lower dimensional space, and do a lot more.

Feedforward Neural Network

The fundamental deep learning models are deep feedforward networks, commonly known as feedforward neural networks or multilayer perceptron (MLPs). A feedforward network's objective is to simulate some function f*. For instance, y = f*(x) transfers an input x to a category y for a classifier. A feedforward network establishes the mapping y = f(x;) and discovers the value of that yields the best function approximation.

The reason these models are named feedforward is because data moves from the input x via the function being evaluated, the calculations necessary to define f, and finally to the output y. The model's outputs cannot be fed back into it because there are no feedback connections. Recurrent neural networks are created when feedforward neural networks are expanded to incorporate feedback connections.

As we know the inspiration behind neural networks are our brains. So let’s see the biological aspect of neural networks.

Visualizing the two images in Fig. 1, where the left image demonstrates how a multilayer neural network can distinguish between various objects by learning various characteristics of those objects at each layer, such as the detection of edges at the first hidden layer and the identification of corners and contours at the second hidden layer. Similar to how different

parts of our bodies have varied functions, the V1 region of the brain recognises edges, corners, and other features.

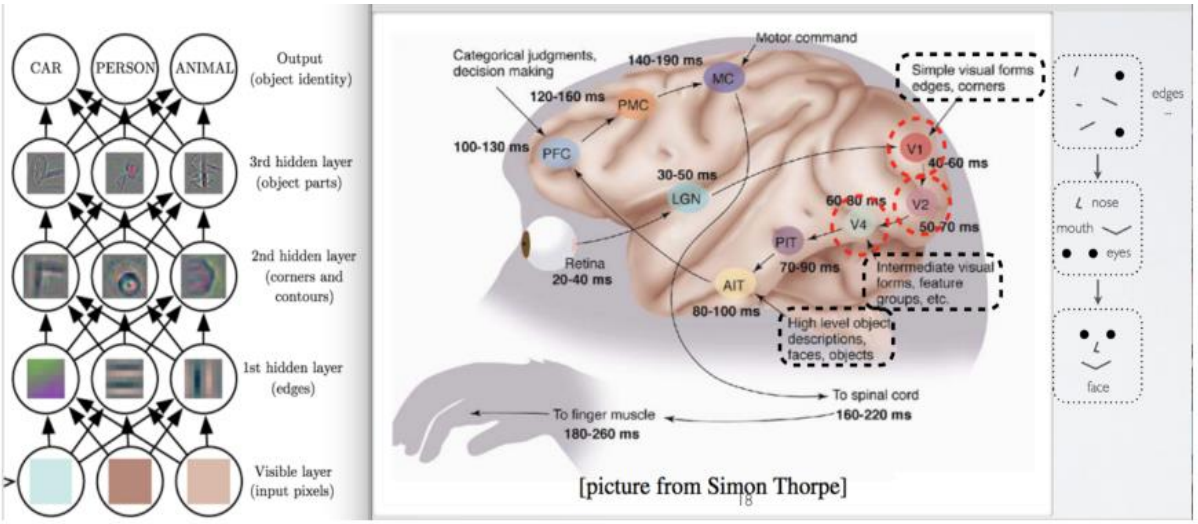

FIGURE 1 :HOW MULTILAYER NEURAL NETWORK IDENTIFY DIFFERENT OBJECT BY LEARNING DIFFERENT CHARACTERISTIC OF OBJECT AT EACH LAYER.

As we can see, the nonlinearity issue can be solved by using neural networks. As shown in Fig. 2, a neural network has trained a model by integrating various distributions into a single acceptable distribution while taking the problem's nonlinearity into consideration.

A highly well-liked machine learning technique by the name of perceptron gave rise to neural networks. Scientist Frank Rosenblatt created perceptron in the 1950s and 1960s as a result of earlier work by Warren McCulloch and Walter Pitts.

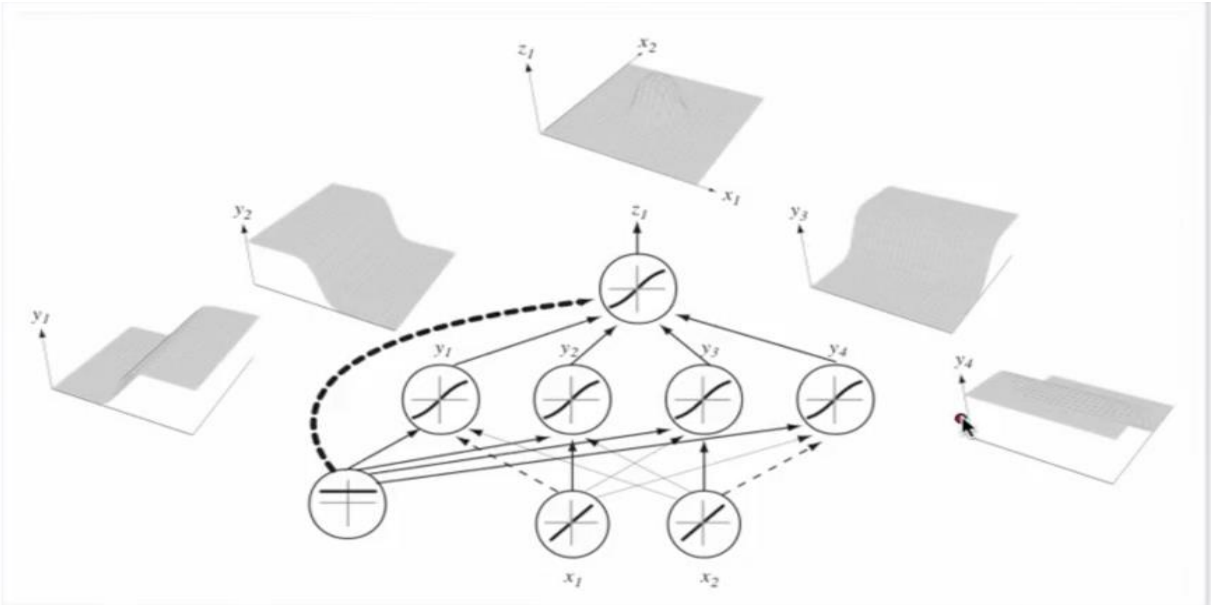

FIGURE 2: DEPICTS THAT DIFFERENT NEURONS IN HIDDEN LAYER, BY COMBINING DIFFERENT DISTRIBUTION COME UP WITH A NEW COMBINED DESIRED DISTRIBUTION.

WHY WE NEED NEURON MODEL?

Let's say we want to use a network of perceptron to learn how to tackle a particular problem. The raw pixel data from a handwritten image of a digit that has been scanned, for instance, could be one of the network's inputs. Additionally, we want the network to pick up weights and biases such that its output accurately categorizes the digit. Consider making a minor tweak to some weight (or bias) in the network to get an idea of how learning might operate. We want the output from the network to only slightly alter in response to this tiny variation in weight. We'll see in a second how this characteristic will facilitate learning.

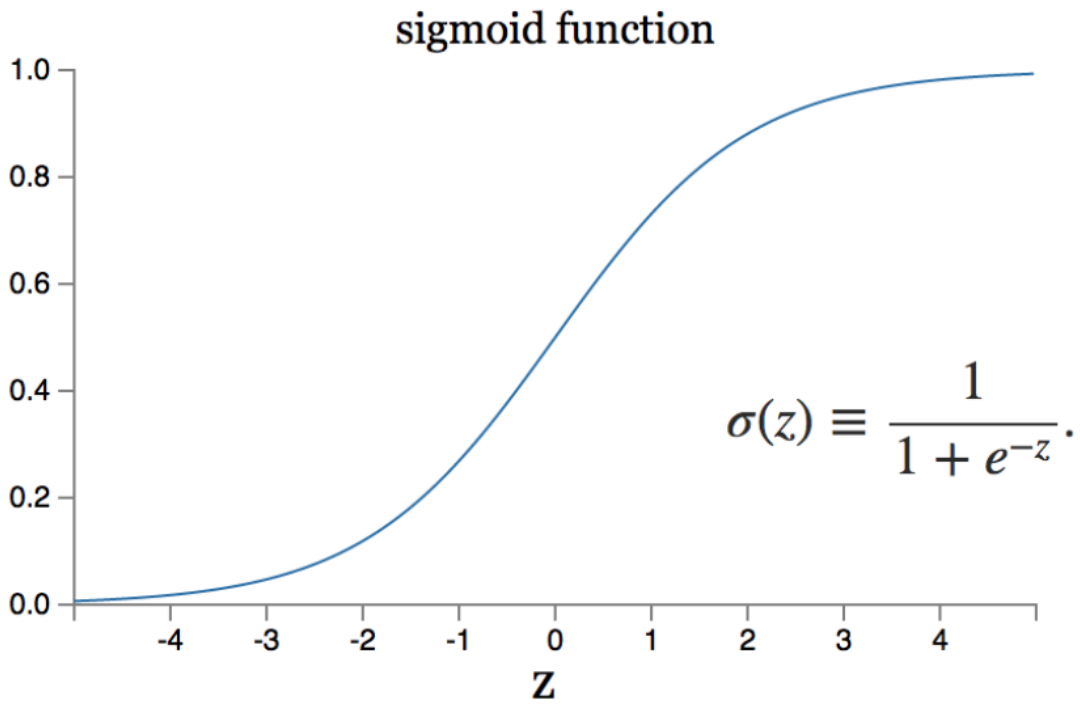

FIGURE 3:SIGMOID FUNCTION. EQUATION OF SIGMOID FUNCTION IS ON RIGHT SIDE.

Let's first examine Fig. 3's sigmoid function behavior. Sigmoidal units saturate over a larger portion of their domain than piecewise linear units do. They saturate to a high value when z is very positive, to a low value when z is very negative, and only substantially respond to input when z is close to 0. Our issue is resolved by this sigmoid function characteristic.

ARCHITECTURE OF NEURAL NETWORK

Input neurons are the neurons that make up the input layer, which is the leftmost layer in this network. The output neurons, or in this instance, just one output neuron, are found in the rightmost or output layer. Since the neurons in the middle layer are neither inputs nor outputs, it is known as a hidden layer.

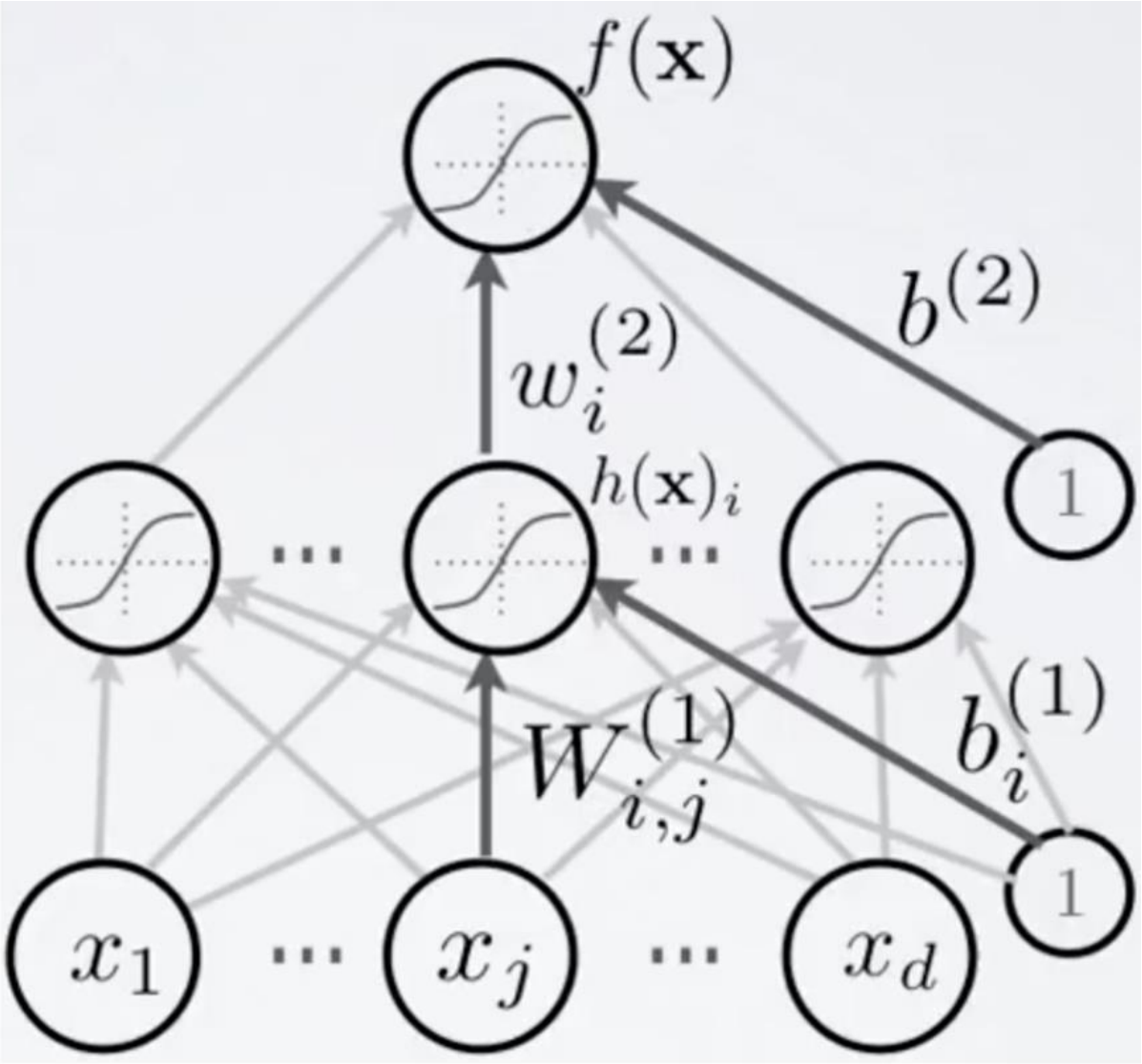

FIGURE 4: MULTILAYER NEURAL NETWORK WITH NOTATIONS.

As shown in the image, wli stands for the weight of the ith neuron in the lth layer. The weight for the link between the jth neuron in the (l1)th layer and the ith neuron in the lth layer is shown by the symbol Wl(i,j). For the bias of the ith neuron, we utilise bi. The activation function is h(x), and we are currently using the sigmoid to do this. The output function is f(x). The universal approximation theorem claims that a network exists that is sufficiently vast to accomplish any level of accuracy we wish, but it does not specify how big this network will be.

Let's look at an illustration of how it functions. In relation to the input of the hidden unit, Figure 5 shows how a network with absolute value rectification generates mirror images of the function computed on top of that hidden unit. To produce mirror replies, each hidden unit specifies where to fold the input space (on both sides of the absolute value nonlinearity). By combining these folding procedures, we are able to create an indefinitely large number of piecewise linear regions that are capable of capturing various recurring patterns and other regular patterns.

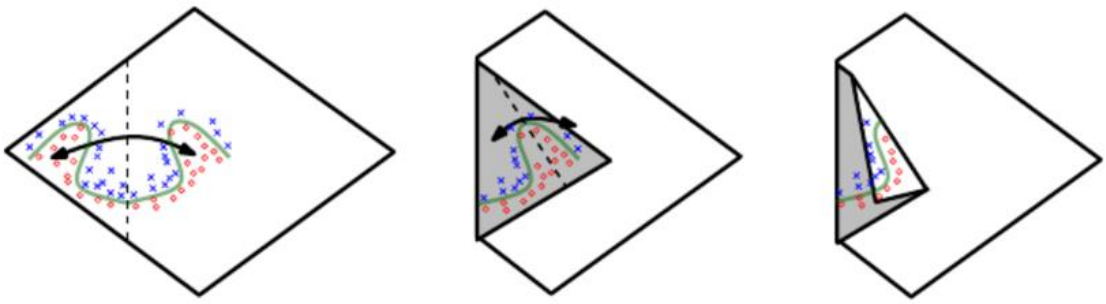

FIGURE 5: BOOK ON DEEP LEARNING (LEFT) EVERY PAIR OF MIRROR POINTS IN AN ABSOLUTE VALUE RECTIFICATION UNIT'S INPUT RESULTS IN THE SAME OUTPUT. THE HYPERPLANE ESTABLISHED BY THE UNIT'S WEIGHTS AND BIAS PROVIDES THE MIRROR AXIS OF SYMMETRY. THE MIRROR COUNTERPART.

RESULT

We are parsing the file path supplied from command lines using the argparse module. The script will begin recording using your default microphone if the file isn't given (using the —file or -f command).

Then, we build the model, load the ideal weights that we previously trained, extract the features of the audio file that was supplied (or recorded), and use model.predict() to obtain the subsequent predictions. Here's an illustration:

```
$ python test.py --file "test-samples/16-122828-0002.wav"
```

```
Result: female
Probabilities:     Male: 20.77%    Female: 79.23%
```

CONCLUSION

We were able to utilize a portion of the enormous Common Voice dataset, extract some pertinent features from it, and use those features to train our neural network. This turned out to be 87.59 percent accurate. We can also enter vocal inputs to verify the model's predictions. As previously indicated, there are numerous real-world uses for this technique. This can also be extended to speech emotion recognition, which in some circumstances can be used to stop bullying. We may also include age recognition to our technique. However, there are a few issues with our specific method, which can be fixed in the following manner.

FUTURE WORK

1. Using noise-reduction methods before signal processing. The MFCC computation algorithm is noise sensitive. As a result, our solution may not work well when there is noise. However, this can be avoided if a noise-reduction technique is employed before calculating the MFCC. Several scientists have also suggested modifying the MFCC algorithm, for example, by increasing the Mel-log-amplitudes to an appropriate power before obtaining the DCT, which will lessen the impact of the signal's low energy components.
2. Utilizing a validation data set will increase accuracy, while early halting will decrease computing complexity.
3. To improve accuracy, switch to a different model architecture like convolution or recurrent nets. Numerous research papers have demonstrated the high accuracy rates of SVM and Xgboost.
4. To determine if the spoken language or accent has an impact on the system.