Voice Gender Recognition Using Deep Learning

Mandeep Shishodia
Btech(Cse).Graphic era hill university (NAAC)
Dhersun, India
mandeepsingh245301@gmail.com

***Abstract*— This article describes a multilayer perceptron deep learning model for gender recognition. The dataset consists of 3,168 male and female voice samples recorded through acoustic analysis. We have used an MLP algorithm to detect gender specific traits. The test data set achieved 96.74%**

Keywords-deep learning; voice recognition; multilayer perceptron networks

I. INTRODUCTION

Depending on the individual features of the sample, including Duration, Filtering, Intensity, and Frequency, the acoustic analysis relies on voice parameter settings. Through analyzing the acoustic attributes of speech, the speaker's gender can be determined. The warbler r package provides the means to execute acoustic analysis. Extracting the dataset of acoustic characteristics can be achieved through this examination. To secure the model used in this study, we applied several machine learning algorithms to the data set. Ultimately, MLP was the algorithm that proved most effective and was used to construct the model. Following a thorough analysis of comparable work, we've developed a web page that deploys the obtained model for determining gender based on voice..

II. RELATED WORK

Becker [2] used a frequency-based baseline model, logistic regression model [3], classification and regression tree (CART) model [4], random forest model [5], boosted tree model [6], Support Vector Machine (SVM) model [7], XGBoost model [8], stacked model [9] for recognition of voices data set [10].According to used models, the results are showed in "TableI".

Accuracy (%)		
Model	***Train***	***Test***
Frequency-based baseline	61	59
Logistic regression	72	71
CART	81	78
Random forest	100	87
Boosted tree	91	84
SVM	96	85
XGBoost	100	87
Stacked	100	89

III. DATA SET AND SOFTWARE LIBRARIES

A. Data Set

Each voice sample format is a .WAV file. The .WAV format files have been pre-processed for acoustic analysis using the specan function by the WarbleR R package [11]. A specan function measures 22 acoustic parameters on acoustic signals. These parameters are showed in "Table II".

TABLE II. MEASURED ACOUSTIC PROPERTIES.

Acoustic Properties	
Properties	***Description***
duration	length of signal
meanfreq	mean frequency (in kHz)
sd	standard deviation of frequency
median	median frequency (in kHz)
Q25	first quantile (in kHz)
Q75	third quantile (in kHz)
IQR	interquantile range (in kHz)
skew	skewness
kurt	kurtosis
sp.ent	spectral entropy

sfm	spectral flatness
mode	mode frequency
centroid	frequency centroid
peakf	peak frequency
meanfun	average of fundamental frequency measured across acoustic signal
minfun	minimum fundamental frequency measured across acoustic signal
maxfun	maximum fundamental frequency measured across acoustic signal
meandom	average of dominant frequency measured across acoustic signal
mindom	minimum of dominant frequency measured across acoustic signal
maxdom	maximum of dominant frequency measured across acoustic signal
dfrange	range of dominant frequency measured across acoustic signal
modindx	modulation index

In this CSV file, nestled within 3168 rows and 21 columns, lies the results of pre-processed WAV files. Among these 21 columns, there exists a classification of male or female as well as several intricate features.

Libraries of software are an important resource for programmers. They contain pre-existing code that can be used to create new applications and streamline development. By implementing a library, programmers can save time and effort, as well as ensuring the functionality of their applications. These libraries are constantly being updated and improved upon by developers around the world. It is important for programmers to stay up-to-date with these developments and find the best libraries for their specific needs. Using open-source libraries can also allow for collaboration with other developers and a greater exchange of knowledge and ideas.

An open source programming language, Python is dynamic and object-oriented with interpreted functionality. Its syntax is easy to learn while still offering considerable power [12].

"Top of TensorFlow or Theano," Keras, a "high-level neural networks library," is written in Python [13].

Using data flow graphs, numerical computation open source software library TensorFlow™ utilizes nodes representing mathematical operations and graph edges for multidimensional data arrays. The library is versatile, allowing for both GPU and CPU usage. Although it is primarily used for machine learning and deep neural network research, adaptation to other domains is easily achievable. [14]..

NumPy is the open source fundamental package for scientific computing with Python. It contains powerful capabilities such as N-dimensional array objects, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities [15]. By using Numpy arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases. Keras uses Numpy for input data types.

warbleR is a software package designed to simplify acoustic analysis in R. This software package allows users to collect explicit acoustic data or import their own data into the workflow to facilitate the use of spectral visualization and acoustic measurement..

Rpy2 is a Python package to provide interface to run R code embedded in a Python process.

IV. LITERATIRE SURVEY

Deep feedforward networks, or MLP, are common in supervised learning problems where a training set of input-output pairs is provided. The network must establish the link between them and the overall structure. For this task, a popular deep learning algorithm is MLP. A back-propagation method is utilized for supervised learning during training. The MLP includes input units, at least two intermediary layers of computation nodes, and outcome nodes.

As it pertains to RD and $x \rightarrow f(x)$, an MLP function f can be written as f: a function of n vectors where D = n and L = n. In matrix form, the function is represented as follows: f is a function of n vectors.

(1) G[b(2) + W(2)[s(xb(1) + w(1)x)]] is represented by f(x).

Matrices Wi(1) and W2 are part of the equation with biased vectors b1 and b2 and the activation function s [18]. The go-to activation function is 'tanh' which computes as (1/2)*((exp(x)-exp(-x))/(exp(x)+exp(-x))).

V. METHOD

In Python, one can create training, test, and prediction codes with ease. Data extraction from csv files is made possible with libraries like Numpy, which transforms information into a 2D array. This array, with 20 parameters and a label for every row, is then randomized and split into five segments. The first four segments include 633 data points, while the last one contains 636. The fifth chunk contained a python array that included a new column of data. This column was the last one and consisted of labels that were converted into integer format - 0 for male and 1 for female. Utilizing 5-fold cross validation, an average score has been calculated. The process involved a training and test loop that ran for five cycles. Throughout each cycle, a different chunk was chosen to serve as the test set while the remainder were added into a Numpy array for training purposes. During each iteration, 20% of the data was designated for validation and another 20% for testing. GPU was implemented for keras, which was used in conjunction with tensorflow.

Our design features a unique construction consisting of a single input layer, a series of four hidden layers, and a solitary output layer. Specifically, the input layer boasts 20 separate inputs which then connect to the initial hidden layer encompassing 64 perceptrons. Interestingly, both the second

and third hidden layers each contain 256 perceptrons. Continuing on, the fourth hidden layer is compromised of 64 perceptrons. Finally, the output layer is made up of only 2 perceptrons. We employed softmax activation within the output layer to obtain the distribution of results, categorizing them according to labels. Additionally, we inserted a dropout of 0.25 between every hidden layer in our model. By randomly setting an appropriate subset of input units to 0 during each training update, dropout effectively reduces the risk of overfitting..

Our model was trained using the Keras Nadam optimization algorithm. We opted for a learning rate of 0.001 in order to prevent missing the minimum despite slower learning. With this rate, we conducted 150 epochs to train our model. Each fold took about 100-120 seconds to complete the training. While we experimented with multiple loss functions, we ultimately chose the Kullback-Leibler divergence [20] algorithm as it provided us with the best performance and accuracy. Our model was able to achieve 96.74% accuracy on the test data set, as demonstrated in "Table III"..

TABLE III. RESULTS OF TEST DATA

Test Data Set		
Gender	***Correct***	***Incorrect***
Male	1553	31
Female	1512	72
Total	3065	103

Model weights has been saved to HDF5 file on each fold by using Keras. Best weight file has been chosen by fold accuracy. Chosen model weights have been used on website to predict The construction of the model mimicked that of its training component. Upon compilation, the model's HDF5 file was retrieved and its weights initialized. A user could input a wav or mp3 file to the web interface; in the latter case, the file was first converted to a wav format. With Rpy2's library, R code could run through Django. The filename was sent to the R code via rpy2 following file conversion. Once the file was read and processed, the warbleR library's specan function produced 22 parameters regarding the file contents. Our model has successfully relied on 20 parameters to predict outcomes that are then displayed to the user through Django.

VI. CONCLUSION

By analyzing the acoustic properties of speech, the model presented in the paper reveals that identifying the gender of a speaker is possible. Utilizing MLP, a model was developed by examining the voice parameters within a data set. Preventing inaccurate classifications caused by pitch variations can be achieved through an increase in the volume of voice samples within a data set. In order to construct the model utilizing male and female voice samples, a website has been launched.

VII. Experimental Result

A set of trials are conducted for assessing the benefactions, which include studying the effectiveness of uprooted features, assessing different literacy ways, and assaying the three natural optimizers used for point selection. This section also shows the datasets, experimental parameters, and settings and also presents the evaluation of the presented benefactions. Experimental Settings. A standard dataset of artificial voices from the study in is used. The dataset consists of 20 languages. Each language has 16 voice samples of eight lines for each gender. The artificial voice is a signal mathematically produced for regenerating the time and spectral characteristics of the mortal speech. These artificial voices have bandwidth between 100 Hz and 8 kHz, which significantly affects the performance of direct and nonlinear telecommunication systems. Mistalk is often used to measure the purpose and content of speech. A continuously operating (i.e. lag-free) channel is sufficient to measure the product. The advantage of lying is that it is easier to create and has fewer differences from real speech. Phonetic feature effects and relationships. To find out which features are best suited to create the best product, it is necessary to examine the relationship between features to see how they are related. Four features were considered in this study, including MFCC, Chroma, Mel, and Tonnetz. The relationship between features is illustrated in the figure, showing that a scatterplot represents the correlation of different types of features. Each graph has a linear regression equation used to generate eigenvalues. Moreover, the R2 correlation coefficient is also obvious. R2 is a statistical test used to determine how well a linear regression model fits real data. This means that if the R2 value is close to 1, the profiles fit the regression line well and there is no difference in their results in the test form, or vice versa, it negatively affects the label.

REFERENCES

[1] A.P. Vogel, P. Maruff, P. J. Snyder, J.C. Mundt, Standardization of pitchrange settings in voice acoustic analysis, Behavior Research Methods, v.41, n.2, p.318-324, 2009.

[2] K. Becker, “Identifying the Gender of a Voice using Machine Learning”, 2016, *unpublished*.

[3] J. M. Hilbe, Logistic Regression Models, CRC Press, 2009.

[4] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, CRC Press, 1984.

[5] L. Breiman, “Random forests”, Machine Learning, Springer US, 45:5– 32, 2001.

[6] J.H. Friedman, Stochastic Gradient Boosting, 1999.

[7] C. Cortes, V. Vapnik, “Support-vector networks”, Machine Learning, 20 (3): 273–297, 1995.

[8] J.H. Friedman, Greedy Function Approximation: A Gradient Boosting Machine, 1999.

[9] L. Breiman, “Stacked regressions”, Machine Learning, Springer US, 45:5–32, 2001.

[10] Dataset, https://raw.githubusercontent.com/primaryobjects/voicegender/master/voice.csv

[11] M. Araya-Salas, G. Smith-Vidaurre, warbleR: an R package to streamline analysis of animal acoustic signals. Methods Ecol Evolution, 2016, *doi:10.1111/2041-210X.12624*.

[12] Python, https://docs.python.org/3/faq/general.html

[13] Keras, Chollet, François, 2015, https://github.com/fchollet/keras

[14] M. Abadi, A. Agarwal, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from *tensorflow.org*.

[15] S. van der Walt, S.C. Colbert, G. Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering, 13, 22-30, 2011, *doi:10.1109/MCSE.2011.37*

[16] Django, https://djangoproject.com

[17] D.E. Rumelhart, G.E. Hinton, R.J.Williams, Learning representations by back-propagating errors, Nature 323: 533-536, 1986.

[18] http://deeplearning.net/tutorial/_sources/mlp.txt

[19] S. Haykin, Neural Networks: A Comprehensive Foundation (2 ed.). Prentice Hall, 1998.

[20] S. Kullback, R.A. Leibler, On information and sufficiency, Annals of Mathematical Statistics. 22 (1): 79–86, 1951.