ATLANTIS PRESS

Advances in Computer Science Research, volume 58

Modeling, Simulation and Optimization Technologies and Applications (MSOTA 2016)

Voice Gender Recognition Using Deep Learning

Mucahit Buyukyilmaz[1,*] and Ali Osman Cibikdiken[2]

[1]Necmettin Erbakan University, Advanced Computation and Data Analysis Laboratory, Konya, Turkey

[2]Necmettin Erbakan University, Department of Computer Engineering, Konya, Turkey

[*]Corresponding author

***Abstract*—In this article, a Multilayer Perceptron (MLP) deep learning model has been described to recognize voice gender. The data set have 3,168 recorded samples of male and female voices. The samples are produced by using acoustic analysis. An MLP deep learning algorithm has been applied to detect gender-specific traits. Our model achieves 96.74% accuracy on the test data set. Also the interactive web page has been built for recognition gender of voice.**

Keywords-deep learning; voice recognition; multilayer perceptron networks

I. INTRODUCTION

Acoustic analysis of the voice depend upons parameter settings specific to sample characteristics such as intensity, duration, frequency and filtering [1]. The acoustic properties of the voice and speech can be used to detect gender of speaker. warbleR R package is designed for acoustic analysis. The data set which have acoustic parameters can be obtained with this analysis. The data set can be trained with different machine learning algorithms. In this paper, MLP has been used to obtain model. The results have been compared with related work. A web page has been designed to detect the gender of voice by using obtained model.

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 "Table I".

TABLE I. ACCURACY OF MODELS FOR RECOGNITION VOICES.

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

The pre-processed WAV files have been saved into a CSV file. The CSV file is contained 3168 rows and 21 columns. There are features and the classification of male or female in these 21 columns.

B. Software Libraries

Python; is an interpreted, interactive, object-oriented, dynamic type, easy to learn and open source programming language. Python combines remarkable power with very clear syntax [12] .

Keras; "is a high-level neural networks library, written in Python and capable of running on top of either TensorFlow or Theano" [13].

TensorFlow™ is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them [14]. TensorFlow's flexible architecture allows you to use GPU or CPU to mainly conducting machine learning and deep neural networks research, but other domains can be adapted easily.

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.

Django is free and open source high-level Python Web framework that encourages rapid development and clean, pragmatic design. Django is reassuringly secure, exceedingly scalable and was designed to help developers create applications quickly as possible [16].

warbleR is a package designed to streamline acoustic analysis in R. This package allows users to collect open-access acoustic data or input their own data into a workflow that facilitates automated spectrographic visualization and acoustic measurements.

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

IV. MULTILAYER PERCEPTRON NETWORKS

Deep feedforward networks, or Multilayer Perceptron (MLP) networks, are used in supervised learning problems. These problems have a training set of input-output. The network must produce a model to find the dependency between them. An MLP is one of typical deep learning algorithms. It uses a supervised learning technique called backpropagation for training the network [17]. An MLP network contains a set of input layers, one or more hidden layers of computation nodes, and an output layer of nodes.

An MLP function f is, f: RD→RL, where D is the size of input vector x and L is the size of the output vector f(x). This function can be represented in matrix notation:

$$f(x) = G(b^{(2)} + W^{(2)}(s(^{b(1)} + W^{(1)}x))), \quad (1)$$

with bias vectors b(1), b(2); weight matrices W(1), W(2); activation functions G and s. Wi(1) represents the weights from the input units to the i-th hidden unit [18]. Generally, a function tanh is chosen for activation function, with tanh(x)=(ex-e-x)/(ex+e-x) [19].

V. METHOD

All training, test and prediction codes have been written by using Python libraries. Data have been loaded from csv file into Numpy arrays with built-in Python libraries. Data set has been loaded from csv file into 2 dimension Python array. Each row has 20 parameters and 1 label. The array has been shuffled randomly. It has been splitted to 5 chunks. First 4 chunk has 633 data but last has 636 data. Also last column of data, which is label, has been converted integer as 0 for male and 1 for female and added to Python array to 5 chunks.

5-Fold cross validation has been used and average score has been obtained. Training and test loop have been run 5 times. On each run different chunk has been used for test, other chunks are concatenated to Numpy array and used for training. On each loop, 20% of data has been used for test and 10% of data has been used for validation. Keras has been used top of Tensorflow and has been configured to use GPU.

1 input layer, 4 hidden layers and 1 output layer have been used to build our model. Input layer has 20 inputs and connected to first hidden layer which has 64 perceptrons. Second and third hidden layers have each 256 perceptrons. Forth hidden layer has 64 perceptrons. The output layer has 2 perceptrons. Softmax activation function conducted in output layer to obtain the categorical distribution of the result for labels. Dropout 0.25 has been applied between each hidden layers. Dropout consists of randomly setting a fraction of input units to 0 at each update during training time. In this way, it helps to prevent overfitting.

Nadam optimization algorithm in Keras has been used to train our model. The learning rate has been chosen 0.001. This gave us slow learning but it prevents us to miss minimum. By choosing lower learning rate our model has been trained with 150 epochs. Total training time is around 100-120 sec for each fold. Several loss function has been tested with our model and Kullback–Leibler divergence [20] algorithm has been chosen which gave best performance and accuracy. The model achieved 96.74% accuracy on the test data set. The result is showed in "Table III" for test data set.

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 gender of uploaded voice.

A website has been developed by using Django framework:

https://www.konya.edu.tr/acdal/projects/deep-learning-voice-gender-detection

Model has been built same as training part. After compiling the model saved HDF5 file has been loaded and model weights have been set up. User can upload wav or mp3 file on web browser. Mp3 files convert to wav format. Rpy2 library has been used to run R code inside Django. After load and conversation the file, filename passed to R code by using rpy2. Voice file has been readed as data frame and passed specan function of warbleR library. Specan function return 22 parameters about loaded file. Chosen 20 parameters have been succeed to predict result using our model. Results is taken by Django and showed to user. All computations have been performed in Advanced Computing and Data Analysis Laboratory (ACDAL), Necmettin Erbakan University, Konya.

VI. Conclusion

The model obtained in paper show us that we can use acoustic properties of the voices and speech to detect the voice gender. MLP has been used to obtain the model for classification from data set which have the parameters of voice samples. A larger data set of voice samples can be minimized incorrect classifications from intonation. The web page has been published to develop the model from loaded examples about male and female voice samples.

Acknowledgment

This work is supported in part by the Necmettin Erbakan University, BAP Coordination Office.

References

[1] A.P. Vogel, P. Maruff, P. J. Snyder, J.C. Mundt, Standardization of pitch-range 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/voice-gender/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.