Gender Recognition by Voice

Baek Jihyun
Department of Computer Science, Kyunghee University
bbaek12@gmail.com

***Abstract* - This paper aims to teach the machine to distinguish between male and female through various voice samples. I have tried various methods to get the best results, and finally I will introduce the method that yielded the highest result. The keras library was imported and dropout was used to prevent overfitting. I refer to the code 'Simple NN with Keras' in Kaggle.**

Keyword – deep learning; multilayer perceptron; softmax;

I. INTRODUCTION

People can easily distinguish between male and female voices. However, it is difficult for machine to learn this because voice classification goes beyond just the frequency of the voice and thus requires additional feature detection. I will use deep learning to solve this problem. I will also be analyzing the model that distinguishes the voices based on the training models.

The paper is organized as follows: 'Selected dataset' section describes the dataset I selected to train machine, 'Proposed method' section I took to solve the problem, 'Experiment' section describes the overall result and accuracies I got from the trained model and overall conclusions are mentioned in 'Conclusion' section.

II. SELECTED DATASET

A. Dataset

I will use the dataset provided by Kaggle. The dataset consists of 3,168 recorded voice samples, collected from male and female speakers. The voice samples are pre-processed using warble R package, with an analyzed frequency range of 0hz-280hz (human vocal range).

B. Features

The dataset included following acoustic properties of each voice within the CSV:

1. meanfreq: mean frequency (in kHz)
2. sd: standard deviation of frequency
3. median: median frequency (in kHz)
4. Q25: first quantile (in kHz)
5. Q75: third quantile (in kHz)
6. IQR: interquartile range (in kHz)
7. skew: skewness
8. kurt: kurtosis
9. sp.ent: spectral entropy
10. sfm: spectral flatness
11. mode: mode frequency
12. centroid: frequency centroid
13. peakf: peak frequency (frequency with highest energy)
14. meanfun: average of fundamental frequency measured across acoustic signal
15. minfun: minimum fundamental frequency measured across acoustic signal
16. maxfun: maximum fundamental frequency measured across acoustic signal
17. meandom: average of dominant frequency measured across acoustic signal
18. mindom: minimum of dominant frequency measured across acoustic signal

19. maxdom: maximum of dominant frequency measured across acoustic signal
20. dfrange: range of dominant frequency measured across acoustic signal
21. modindx: modulation index. Calculated as the accumulated absolute difference between adjacent measurements of fundamental frequencies divided by the frequency range
22. label: male or female

C. Existing analysis result

- The accuracy according to the academic journals of the University of Maryland, Baltimore Country:

	Accuracy
Decision tree	0.8659
K nearest neighbour	0.9021
Logistic regression	0.8927
Naive bayes	0.8849
Neural network	0.9306
Perceptron	0.8186
Random forest	0.9006
SVM	0.9259
Gradient boosting	0.9148
Voting	0.9384

Table 1: Accuracy1

- The accuracy provided by the Kaggle:

	Accuracy	
Baseline (always predict male)	50%	50%
Logistic Regression	97%	98%
Cart	96%	97%
Random Forest	100%	98%
SVM	100%	99%
XGBoost	100%	99%

Table 2: Accuracy2

III. EXPERIMENT

A. Proposed method

I propose to train the conventional models to learn the task of classifying male and female voice samples. The models I chose is multilayer perceptron. A neural network with 4 hidden layers each containing 300 nodes. A 'relu' activation function is used at previous three layers and since I see this problem as a multinomial classification with two classes, 'softmax' function is used at last one layer. To optimize the model, 'Adam optimizer' is used with learning rate initialized to 0.0005, and the loss function was 'categorical cross entropy' because the problem is multiclass classification. The batch size is set to 40, and a total of 10 epochs are performed.

B. Techniques

I try the following three techniques to prevent overfitting in the learning process and build a more accurate model

- Train-Test Split

  To detect overfitting, divide the dataset into train datasets and test dataset before train the model. Only the train dataset is used to train the model, and after complete the training, the test dataset is used to evaluate the model. If there is a great difference between the accuracy of the process of learning the model using only the train dataset and the accuracy of evaluating the model using the test dataset after learning, overfitting has occurred. The ratio of the train dataset to the test dataset is divided into 7 to 3 depending on the general case.

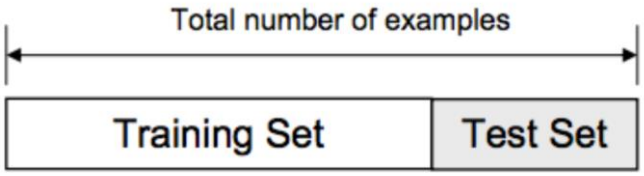

Figure 1 Train-Test Split

- Dropout

  By dropping a certain percentage of nodes for each layer, I can prevent the model from learning too much for training data. I dropped out 0.3 percent of nodes for each layer.

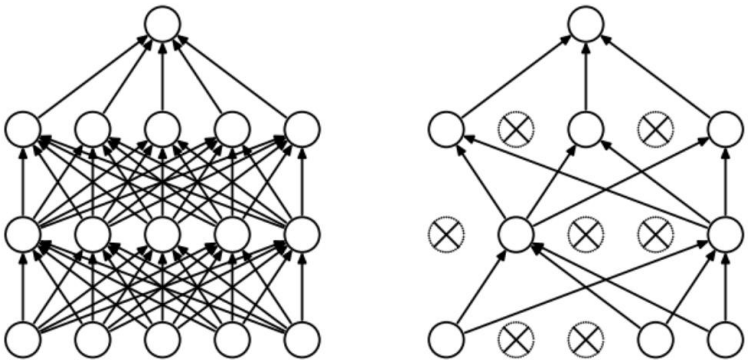

Figure 2 Dropout

- Validation Split

  In training data, some data is left untrained, and after model is learned for each epoch, it is used to validate the model. The validate-loss and validate-accuracy do not directly affect the learning, but they indirectly affect the learning because the trainer adjusts the learning rate, the number of layers, and the batch size through it. I use 0.1 percent of training data as validation data.

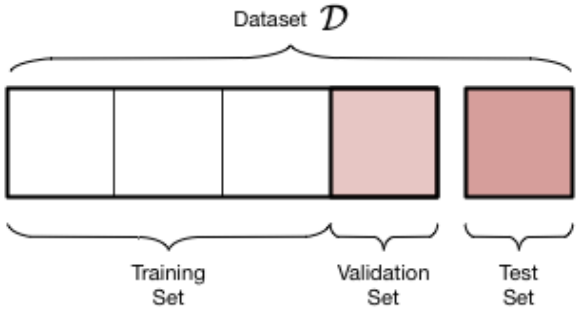

Figure 3 Validation Split

IV. CONCLUSION

After completing 10 epochs, the final training accuracy was 0.9835. As the learning progresses, the loss decreases gradually, and on the contrary the accuracy gradually increases. When the test dataset is input to the learned model, the test accuracy is 0.9737. There is no significant difference from training accuracy, so it can be seen that overfitting does not occur and learning becomes right.

- Train-loss and validate-loss for each epoch are shown in the graph below.

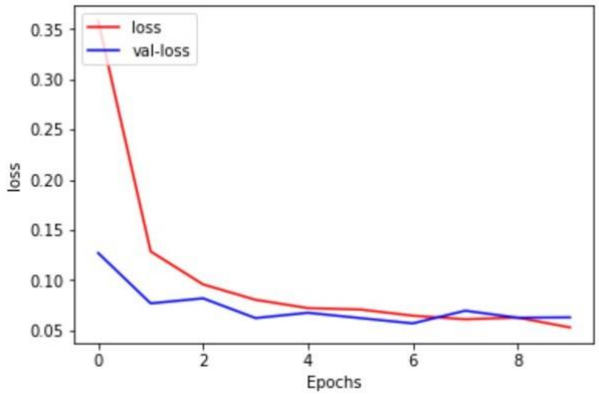

Figure 4 Graph of loss for each epoch

- Train-accuracy and validate-accuracy for each epoch are shown in the graph below.

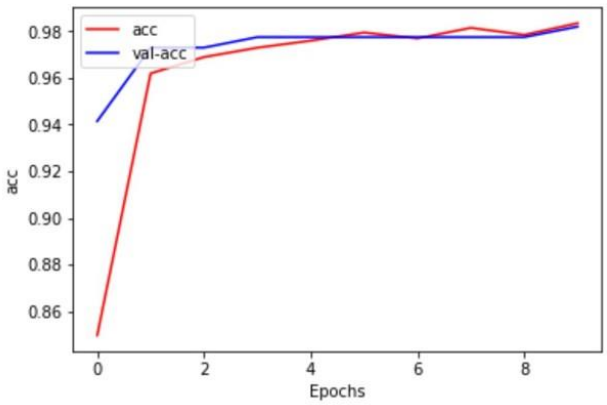

Figure 5 Graph of accuracy for each epoch

In addition, the table below shows ‘Precision’, ‘Recall’ and ‘F1-score’. To illustrate the meaning of each value as ‘Male’, the Precision indicates how much predicted Male is actually Male, and Recall indicates how many Male numbers are expected to be Male. F1-score considers precision and recall at the same time. Since there is not much difference between the values, I can see that trained model is appropriate.

	Precision	Recall	F1-score	Support
Female	0.97	0.98	0.97	474
Male	0.98	0.97	0.97	477
Avg / total	0.97	0.97	0.97	951

Table 3 Table of Precision, Recall and F1-score

V. REFERENCE

1. Bronson, Simple NN with Keras, https://www.kaggle.com/jsultan/simple-nn-with-keras.
2. Vineet Ahirkar and Naveen Bansal, Gender Recognition using Voice, University of Maryland, Baltimore Country.