Age Estimation based on Human Voice using Deep Neural Networks

Luca Arrotta

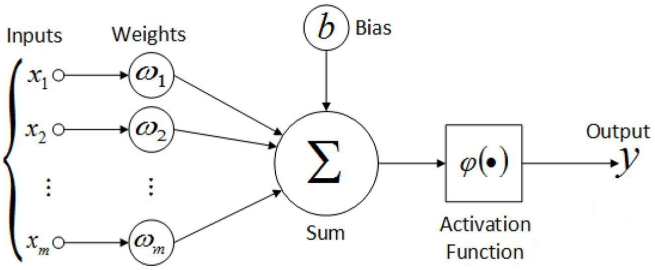


Fig. 1: An example of artificial neuron.

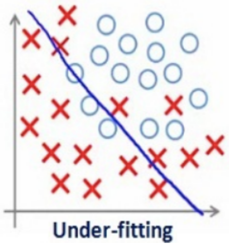


Fig. 2: Example of underfitting in binary classification.

1 Introduction

In this work, a deep neural network was trained in order to estimate the age of a person, based on his voice. The used dataset provides audio files which contain recorded human voices. From each of these files, 24 features are extracted and used to train the neural network. The trained neural network reached a test accuracy equals to 94.48%.

2 Neural Networks

Artificial Neural Networks (ANN) are computational models inspired by the biological neural network [1]. An ANN is a collection of layers. In each layer there can be several nodes called artificial neurons. Each artificial neuron (Figure 1) computes a linear combination of its inputs and gives it to an activation function. The output of the activation function becomes the input of other neurons, connected to the first one.

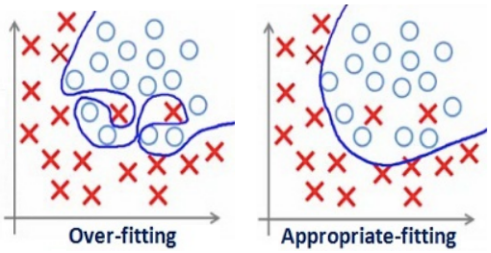


Fig. 3: Examples of overfitting and appropriate fitting in binary classification.

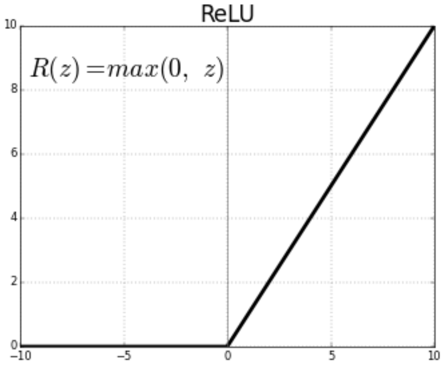


Fig. 4: The ReLU activation function.

2.1 Activation Functions

Activation functions are used to introduce non-linearity in the neural network. Without the activation functions, the neural network would be a linear model, so it would be too simple and it would generate an underfitting problem (see Figures 2 and 3) [2]. ReLU is the most used activation function of these years. It maps negative values to zero and keeps the positive values (see Figure 4). This activation function allows a faster training of the model [3].

2.2 Fully Connected Layer

In a fully connected layer, each neuron is connected to all the neurons of the adjacent layers. Tipically, the last fully connected layer use the Softmax activation function, whose output is a value between 0 and 1 for each class that the neural network has to recognize. The sum of these values is equal to 1.

L. Arrotta, Intelligent Systems, A/A 2018-2019, Universitá degli Studi di Milano, Via Celoria 18, Milan, Italy
E-mail: luca.arrotta@studenti.unimi.it

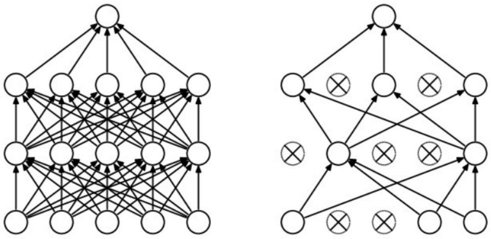


Fig. 5: Left: the neural network without the usage of the dropout technique. Right: the same network, using the dropout technique.

2.3 Dropout

We can use the dropout technique in order to reduce the overfitting problem. When we apply the dropout to a layer, during the training stage, some of the neurons of that layer will be randomly chosen and disabled (Figure 5). In this way, we can reduce the dependency between the neurons of the same layer and so we increase the robustness of the neural network.

2.4 Batch Normalization [4]

We define Internal Covariate Shift as the change in the distributions of layers' inputs. This is a problem, because the layers need to continuously adapt to the new distribution. To improve the training, we seek to reduce the internal covariate shift, by fixing the distribution of the layer inputs. In this way, we can also improve the training speed. In particular, Batch Normalization transform the mean of the input to 0 and the variance of the input to 1.

3 Dataset

The dataset used in this work is the Mozilla Common Voice dataset [5], an open and publicly available dataset of voices.

The corpus is split into several parts. The subsets with "valid" in their name are audio clips that have had at least 2 people listen to them, and the majority of those listeners say the audio matches the text. The subsets with "invalid" in their name are clips that have had at least 2 listeners, and the majority say the audio does not match the clip. All other clips have "other" in their name. In this work, only the "valid" and the "other" subsets are used.

Each subset of data has a corresponding csv file. Each row of a csv file represents a single audio clip, and contains the following information:

- filename: relative path of the audio file
- text: supposed transcription of the audio
- up_votes: number of people who said audio matches the text
- down_votes: number of people who said audio does not match text
- age: age of the speaker (teens, twenties, thirties, fourties, fifties, sixties, seventies, eighties)
- gender: gender of the speaker (male, female, other)
- accent: accent of the speaker (us, australia, ...)

The audio clips for each subset are stored as mp3 files in folders with the same naming conventions as it's corresponding csv file. So, for instance, all audio data from the valid train set will be kept in the folder "cv-valid-train" alongside the "cv-valid-train.csv" metadata file.

4 Development

4.1 Feature Extraction [6] [7]

These are the 24 features used in order to estimate the age of the human voices:

- Gender,
- Spectral Centroid,
- Spectral Spread (or Spectral Bandwidth),
- Spectral Rolloff,
- Mel Frequency Cepstral Coefficients (MFCCs), which are 20 features.

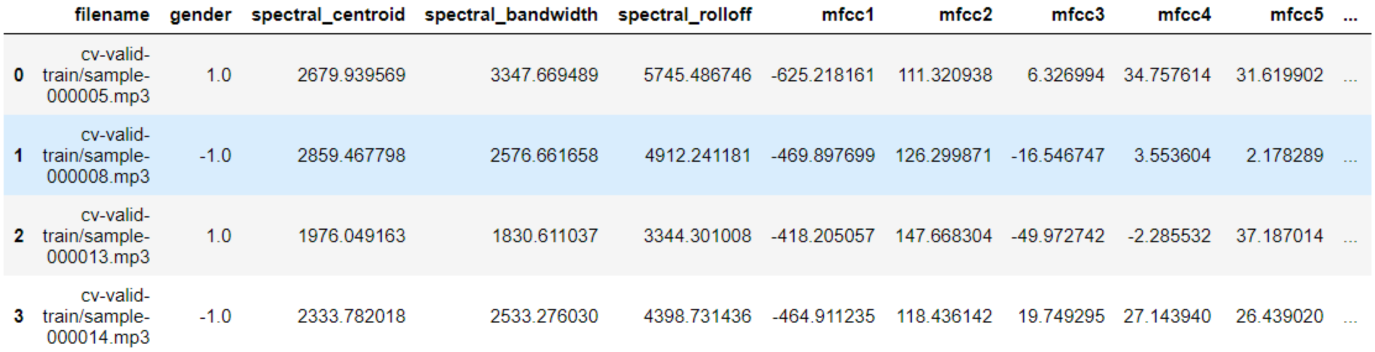
The gender information is taken directly from the dataset csv files.

The Spectral Centroid is the barycenter of the spectrum. It is computed considering the spectrum as a distribution, which values are the frequencies and the probabilities to observe these are the normalized amplitudes.

The Spectral Spread is the variance of the distribution previously defined. It is commonly associated with the bandwidth of the signal.

The Spectral Rolloff point is the frequency so that 95% of the signal energy is contained below this frequency. It is correlated to the noise cutting frequency.

The Mel Frequency Cepstral Coefficients (MFCCS) is a robust technique for speech feature extraction. It represents the shape of the spectrum with very few coefficients (20 coefficients in this work). The cepstrum is the Fourier Transform of the logarithm of the spectrum. The Mel-cepstrum is the cepstrum computed on the Mel bands instead of the Fourier spectrum. The Mel bands are based on the Mel frequency scale, which is linear at low frequencies (below 1000 Hz) and logarithmic at high frequencies (above 1000 Hz). The use of the Mel scale allows to take better into account the mid-frequencies part of the signal. The MFCC are the coefficients of the Mel cepstrum.



	filename	gender	spectral_centroid	spectral_bandwidth	spectral_rolloff	mfcc1	mfcc2	mfcc3	mfcc4	mfcc5	...
0	cv-valid-train/sample-000005.mp3	1.0	2679.939569	3347.669489	5745.486746	-625.218161	111.320938	6.326994	34.757614	31.619902	...
1	cv-valid-train/sample-000008.mp3	-1.0	2859.467798	2576.661658	4912.241181	-469.897699	126.299871	-16.546747	3.553604	2.178289	...
2	cv-valid-train/sample-000013.mp3	1.0	1976.049163	1830.611037	3344.301008	-418.205057	147.668304	-49.972742	-2.285532	37.187014	...
3	cv-valid-train/sample-000014.mp3	-1.0	2333.782018	2533.276030	4398.731436	-464.911235	118.436142	19.749295	27.143940	26.439020	...

Fig. 6

4.2 Dataset management

First of all, in this work all the mp3 files were converted in wav files in order to use specific libraries which work only with wav files. This convertion is developed in the ”Converter” jupyter notebook.

All the other work was developed in the ”Age Estimation based on Human Voice” jupyter notebook. After the conversion, new csv files were created starting from the csv files of the dataset. The problem is that the dataset is split into several parts. So, the idea is to create a single dataframe which includes all the useful information about the audio files that we want to use to train and test the neural network. For each useful audio file, we consider the path of the file, the 24 features and its label (the age). Actually, in this dataframe is memorized the path of the mp3 version of each audio file. A specific function was developed in order to obtain the path of the wav version of the file, starting from the mp3 version. In Figure 6 is shown part of the previously described dataframe, which contains information about 143170 audio files.

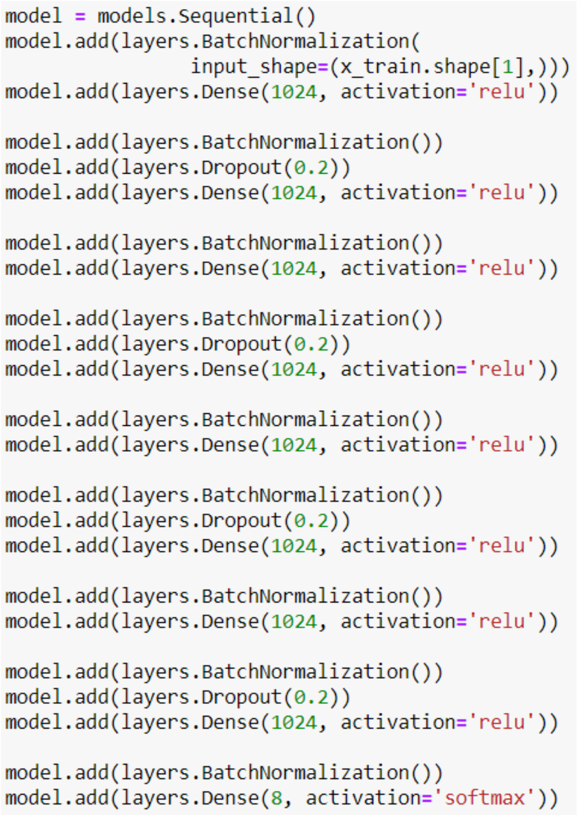
The features are extracted using the librosa library [8] and then they are scaled with the StandardScaler function of the sklearn library [9]. The values about the gender are not scaled, in this way we will have always three possible values for the gender feature: -1 for ”male”, +1 for ”female”, 0 for ”other”.

4.3 Classification and Results

The artificial neural network was developed with the keras library [10]. The architecture of the neural network which led to the best results is shows in Figure 7. Note that, in keras, a ”Dense” layer is a ”Fully Connected” layer and that 8 is the number of possible labels (so the output layer has 8 neurons).

The dataframe is split into Training (80%), Test (10%) and Validation set (10%). To train the model, five training stages with different batch sizes were used.

In order to obtain the best performances, in this work were tried combinations of different architectures and batch sizes.



```
model = models.Sequential()
model.add(layers.BatchNormalization(
                input_shape=(x_train.shape[1],)))
model.add(layers.Dense(1024, activation='relu'))

model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.2))
model.add(layers.Dense(1024, activation='relu'))

model.add(layers.BatchNormalization())
model.add(layers.Dense(1024, activation='relu'))

model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.2))
model.add(layers.Dense(1024, activation='relu'))

model.add(layers.BatchNormalization())
model.add(layers.Dense(1024, activation='relu'))

model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.2))
model.add(layers.Dense(1024, activation='relu'))

model.add(layers.BatchNormalization())
model.add(layers.Dense(1024, activation='relu'))

model.add(layers.BatchNormalization())
model.add(layers.Dropout(0.2))
model.add(layers.Dense(1024, activation='relu'))

model.add(layers.BatchNormalization())
model.add(layers.Dense(8, activation='softmax'))
```

Fig. 7: The architecture developed with keras library which led to the best results.

5 Results

In this section, we talk about the best model obtained in this work. The associated notebook was saved in the ”best_model.html” file.

After the first 26 epochs with batch size equals to 128, the model reaches a test accuracy equals to 93.03%. The training and validation accuracies trend after these 26 epochs is shown in Figure 8.

After another epoch with batch size equals to 256, the model reaches a test accuracy equals to 94.15%. The training and validation accuracies trend after this second training stage is shown in Figure 9.

A third training stage with batch size equals to 512

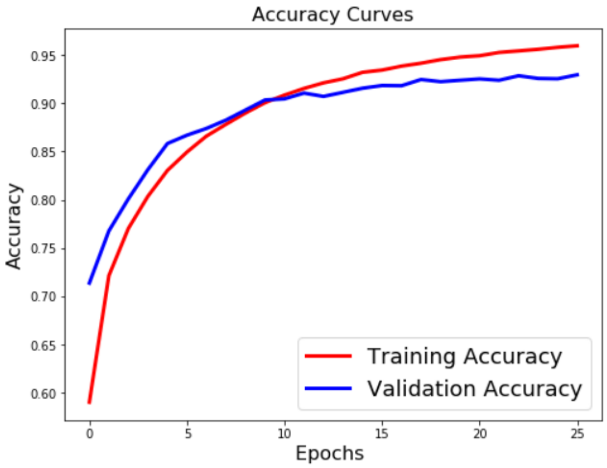


Fig. 8: Training and validation accuracies trend after the first training stage.

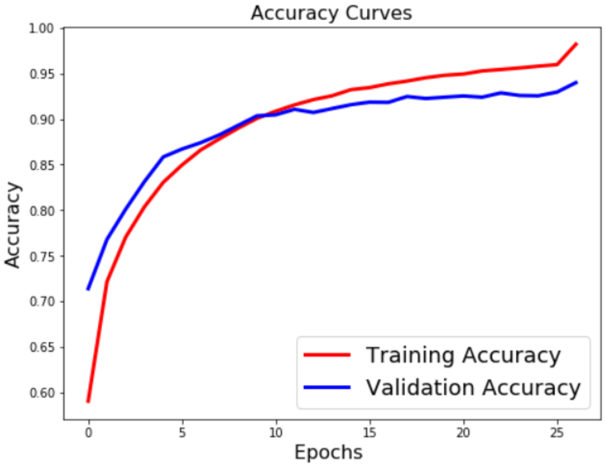


Fig. 9: Training and validation accuracies trend after second training stage.

didn’t improve the performances of the model. Neither a fourth training stage with batch size equals to 1024 improved the performances of the model.

After another epoch with batch size equals to 2048, the model reaches a test accuracy equals to 94.48%. The training and validation accuracies trend after the last training stage is shown in Figure 10. The training and validation losses trend after the last training stage is shown in Figure 11.

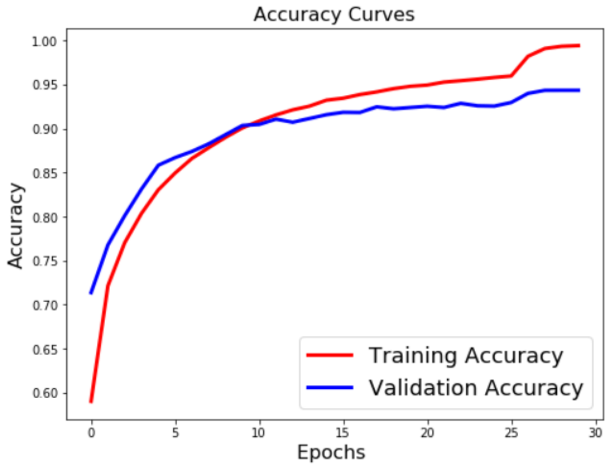


Fig. 10: Training and validation accuracies trend after the last training stage.

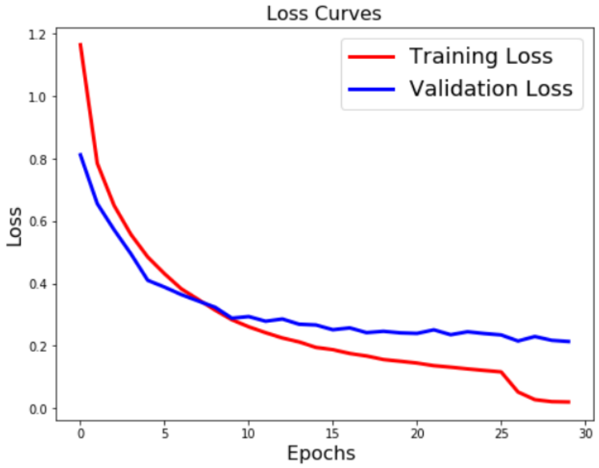


Fig. 11: Training and validation losses trend after the last training stage.

References

[1] Artificial neural networks as models of neural information processing. [Online]. Available: https://www.frontiersin.org/research-topics/4817/artificial-neural-networks-as-models-of-neural-information-processing

[2] Deep convolutional neural networks for image classification: A comprehensive review. [Online]. Available: https://www.mitpressjournals.org/doi/full/10.1162/neco_a_00990

[3] G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for lvcsr using rectified linear units and dropout,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8609–8613.

[4] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.

[5] Common voice dataset. [Online]. Available: https://voice.mozilla.org/en/datasets

[6] A large set of audio features for sound description. [Online]. Available: http://recherche.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf

[7] Feature extraction techniques for speech recognition. [Online]. Available: https://pdfs.semanticscholar.org/a53b/d73a611ab74e986f080bc9e8a12d3505fe20.pdf

[8] Librosa. [Online]. Available: https://librosa.github.io/librosa/

[9] Sklearn. [Online]. Available: https://scikit-learn.org/stable/

[10] Keras. [Online]. Available: https://keras.io/