KTH Royal Institute of Technology
Stockholm

School of Electrical Engineering and
Computer Science

Scalable Machine Learning and Deep Learning - ID2223

Recognizing emotions through spoken sentences with machine learning

Author
Emil Ståhl

Author
Erik Kongpachith

Author
Selemawit Fsha Nguse

January 8, 2022

1 Introduction

In this work, we train six different machine learning models to recognize different emotions through spoken sentences. The dataset used for training is based on the RAVDESS emotional speech audio available at `https://www.kaggle.com/uwrfkaggler/ravdess-emotional-speech-audio`. To achieve this goal, we primarily make use of a convolutional neural networks (CNN), multilayer perceptron(MLP) and Long short-term memory (LSTM). In addition, we created models based on decision trees, random forest and XGBoost. With our different machine learning models, we are able to recognize emotions through spoken sentences with an accuracy of up to 88%. Speech Emotion Recognition (SER) systems is useful in psychiatric diagnosis, lie detection, call centre conversations, customer voice review, voice messages, and many other fields such as accessibility for people with different disabilities.

2 Dataset

The RAVDESS dataset includes around 1,500 audio files from 24 different actors. 12 male and 12 female actors record short audio clips reflecting 8 different emotions where each integer of the audio file corresponds to an emotion.

- 1 = neutral
- 2 = calm
- 3 = happy
- 4 = sad
- 5 = angry
- 6 = fearful
- 7 = disgust
- 8 = surprised.

Each audio file is named in such a way that the seventh character is consistent with the different emotions that they represent.

3 Tools

To achieve the desired results, we make use of the following tools:

- TensorFlow
- Keras
- Numpy
- Pandas
- Scipy
- LibROSA
- Scikit-learn
- Jupyter Notebook

4 Methodology

Here, we present the methodology used and the functionality of our code: First, we define a function to generate MFCC from the audio files. This is done by using librosa. The first step in any automatic speech recognition system is to extract features, i.e. identify the components of the audio signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise, emotion etc. The main point to understand about speech is that the sounds generated by a human are filtered by the shape of the vocal tract including tongue and teeth. This shape determines what sound comes out. If we can determine the shape accurately, this should give us an accurate representation of the phoneme being produced. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and the job of MFCCs is to accurately represent this envelope. The method returns two lists, one containing MFCC values and another containing emotions. Using these lists, we create a numpy array. Finally, we split the data into train and test data which was used to train and evaluate our models. 30% of the dataset is used for testing.

5 Results

This section presents the results of the performed benchmarks of the models. In the table below, we see that the CNN model performed best while LSTM and MLP performed the worst. An explanation of this is given in section 6.

Model	Accuracy
CNN	0.88
XGBoost	0,88
RF	0,79
DT	0.60
LSTM	0.48
MLP	0.48

6 Discussion

This section provides an analysis of the results presented in section 5.

6.1 CNN vs. RNN

As seen in section 5, the CNN model performed better than the LSTM model which performed the worst amongst the six models. Long Short-Term Memory (LSTM) is an RNN architecture specifically designed to address the vanishing gradient problem. However, CNN has been proven in many research papers to perform better than RNN at Speech Emotional Recognition due to application of filters and MaxPooling which leads to elimination of noise (compared to the voice phonemes) in Speech Recognition. CNNs work by reducing an image or speech to its key features and using the combined probabilities of the identified features appearing together to determine a classification. While RNNs (LSTM) deals with only sequential and there is no elimination and filtering which has made RNN fall back compared to CNN in aforementioned tasks.[1] However, the results also depends on network configurations, the way one creates training examples and datasets. A lack of systematic benchmarking of existing methods however creates confusion. There are several studies which shows that LSTM outperforms CNN for speech emotion recognition. Specially LSTM with attention mechanism helps to boost emotion recognition performance, which could be an interesting topic to research in future work.

6.2 Decision Tree

The accuracy, precision, and F-score of the Decision Tree classifier increases when decreasing the number of emotions, it proves that Decision Tree classifier is not suitable for multi-class classification problems and is more efficient for binary class problems.

[1] https://www.researchgate.net/post/Has-CNN-taken-over-RNN-in-Speech-Emotion-Recognition-If-yes-why

6.3 XGBoost

Like random forests, gradient boosting is a set of decision trees. The two main differences are:

- How trees are built: random forests builds each tree independently while gradient boosting builds one tree at a time. This additive model (ensemble) works in a forward stage-wise manner, introducing a weak learner to improve the shortcomings of existing weak learners.
- Combining results: random forests combine results at the end of the process (by averaging or "majority rules") while gradient boosting combines results along the way.

XGBoost uses parallelization, tree pruning, and hardware optimizations which may explain why it performs better than both decision tree and random forest.

6.4 Random Forest

The Random Forest (RF) model performed 24% better than the Decision Tree. A random forest is considered better than a single decision tree because of the fact that this reduces the over-fitting problem which can be produce really inefficient results among various other classification techniques.

6.5 Multilayer Perceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to refer to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons. MLPs are suitable for classification prediction problems where inputs are assigned a class or label. They are also suitable for regression prediction problems where a real-valued quantity is predicted given a set of inputs. It is clearly evident that the CNN converges faster than the MLP model in terms of epochs but each epoch in CNN model takes more time compared to MLP model as the number of parameters is more in CNN model than in MLP model in this example. MLP is now deemed insufficient for modern advanced computer vision tasks. It has the characteristic of fully connected layers, where each perceptron is connected with every other perceptron. However, the disadvantages is that the number of total parameters can grow very high, which is seen as inefficient because there is redundancy in such high dimensions. Another disadvantage is that it disregards spatial information as it takes flattened vectors as inputs.

6.6 Mel Frequency Cepstral Coefficients

Mel Frequency Cepstral Coefficients (MFCC) is the default choice for most speech processing tasks including speech emotion recognition. However, MFCC is not the optimal one as it lacks prosody information, long-term information. That is why MFCC is often augmented with pitch (to be more specific log F0) and shifted delta coefficients. This additional information help to boost the emotion recognition performance. MFCC lacks phase information but the role of phase in emotion recognition performance is not much investigated. The parameters for MFCC computation such as the number of filters, the frequency scale are chosen experimentally and they are dependent on the dataset and the backend classifier.

7 Future work

For future research purposes, one should consider applying attention to the LSTM model for achieving better results. One could also change the feature from Mel frequency cepstral coefficients (MFCC) to linear prediction cepstral coefficients (LPCC). More advanced models to try out on the problem includes recognition models are vector quantization (VQ), dynamic time warping (DTW), and different artificial neural networks (ANN). Regarding training data, one could extend the RAVDESS dataset with the SAVEE dataset for potentially better results. The reason why SAVEE was not used for training in this work is that it has a different naming schema which requires a different function for parsing the files. Lastly, it remains to be seen how the models perform when tested on datasets other than RAVDESS. This is important if the models is to be useful in real world applications.

7.1 Summary

It is clear that CNN is the preferred model amongst the ones we have implemented in this work. This is because CNN can account for local connectivity while the weights are smaller and shared which is less wasteful, it is also easier to train than MLP. CNN is more effective while it also goes deeper. Layers are sparsely connected rather than fully connected. It takes matrices as well as vectors as inputs. Every node does not connect to every other node.

8 How to run

To run the project, we make use of Google colab. The notebook is configured to pull the dataset from a public repository on GitHub. The only thing required by the user is to login to their Google drive so that the notebook can be attached. After that, click "Run all cells".

9 Conclusion

This work has implemented a Speech Emotion Recognition (SER) system with different machine learning models. The results shows that our best model (CNN) managed to predict the emotion of spoken sentences with an accuracy of 88%. Suggestions for future work include applying attention to LSTM, and implementing more advanced models that may suite the problem better. Using a larger dataset for training may also be advantageous.