Spoken and Written Language Processing - POE

Assignment 5. COVID-19 detection from coughs (Audio Classification)

Gerard Martín Pey

Jose Àngel Mola Audí

16th June 2023

Assignment 3. Language identification (Sentence Classification)

In this assignment we aim at providing an accurate diagnosis of Covid-19 by using speech processing for a dataset of breathing,cough and voice recordings. To accomplish the task at hand, two different approaches are proposed: First, the unsupervised method of Mel filters; latter, the Hubert supervised model.

Exploratory Data Analysis and Error Analysis

To start with we are going to study and identify the main strengths and weaknesses of each of the proposed methods. Therefore, we aim at detecting in which cases would each method be more suitable and the reasons behind that choice.

On the one hand, the Mel algorithm is widely used in voice-based disease detection thanks to its effectiveness in capturing spectral features (global based approach). Moreover, voice recordings tend to have noise issues which are easily overcomed by this algorithm. It also handles the speaker variability properly.

Although it is computationally efficient, this fact is not quite relevant in our scenario, since the dataset available is small. This fact softens at the same time the issue with the lack of adaptability to variations in recording conditions, given that in a small dataset the variability between data will be lower. Another limitation would be the manual feature engineering (hyperparameter tuning, window for limited contextual information).

On the other hand, the Hubert model is capable of learning both low and high level features directly from the audio data (not need to preprocess), at the same time that they leverage temporal dependencies using an end-to-end learning. The data augmentation process followed to train it makes it more robust to scale (speed) variations in speech.

Nevertheless, the data requirements of deep learning models could pose an issue here because of the small dataset provided (not scalable to different speakers for instance). Moreover, another important drawback would be the lack of interpretability of deep learning models.

Hyperparameter tuning

In this section we are going to try different configurations of parameters for each method and observe how they are affected by the modifications.

First of all we are going to start with the Mel algorithm, which uses VGG models as classifiers. Therefore, we are going to start by studying how different versions of this network affect its performance regarding the prediction of the disease.

Models	# Train epochs	Train Loss	Val AUC	Val Loss	Elapsed time (sec)
Mel VGG11	10	0.5221	63.9%	1.0886	680 s
Mel VGG13	10	0.5125	65.5%	0.6801	696 s
Mel VGG16	10	0.5702	65.5%	0.7154	706 s
Mel VGG19	11	0.6244	66.3%	0.6629	792s

We can observe that it exists overfitting in most of the previous executions. This could potentially be attributed to the fact that our dataset is small. This fact leads to few data being able for model training, making it incapable of learning generalizable features and being prone to overfit.

However, the VGG19 classifier seems to deal properly with this issue, greatly reducing the difference between training and validation losses. Besides it is the one that provides the lower validation loss. Therefore we are going to use this configuration for the following experiments.

Next, we will experiment how changing the optimizer from *adam* to *SGD* affects the error of the model. In addition, we are going to combine it with different learning rate values, given that this parameter will no longer be modified during the execution, as we are removing the *adam* optimizer. We are going to also try different values for the window size:

Models	# Train epochs	Train Loss	Val AUC	Val Loss	Elapsed time (sec)
Mel VGG19	11	0.6244	66.3%	0.6629	792s
Mel VGG19, SGD	43	0.6927	60.8%	0.6929	3055s
Mel VGG19, lr=0.00001	22	0.0917	60.7%	1.3467	1567s
Mel, sgd, lr=0.00001	6	0.6933	42.5%	0.6932	438s
Mel VGG19, sgd, lr=0.001	25	0.6557	66.6%	0.6654	1743s
Mel VGG19, sgd, lr=0.01	7	0.6835	65.9%	0.6981	446s
Mel VGG19, sgd, lr=0.001 window_size = 0.02	2	0.6931	60.2%	0.6931	153s
Mel VGG19, sgd, lr=0.001 window_size = 0.06	27	0.6379	67.1%	0.6823	1984s
Mel VGG19, adam, window_size=0.02	10	0.6768	64.6%	0.6729	714s
Mel VGG19, adam, window_size=0.06	10	0.6652	65.8%	0.6721	680s

We observe that there are discrepancies between the validation AUC and the validation loss, given that they do not correlate. Therefore, we are going to rely on the loss function to make the decisions about which are the best models. By taking this into account, we observe that the configuration that achieves best results stills the baseline with the VGG19.

Next, we also experiment with the Hubert and Distil Hubert models. In this case, hyperparameter tuning is not going to aim only at improving the validation accuracy but at taking care of the overfit of the model, much likely in this scenario. Thus, some proposed modifications rely on solid procedures to overcome overfit such as experimenting with the dropout parameter and adding normalization layers. Besides, we are also going to experiment with the hidden size of the adapter in order to observe how changes in the bottleneck affect the performance of the model:

Assignment 3. Language identification (Sentence Classification)

Models	# Train epochs	Train Loss	Val AUC	Val Loss	Elapsed time (sec)	Best AUC
Hubert baseline	16	0.4968	72.1% (67.4% - 76.7%)	0.6539	1593s	72.4% (epoch 11)
Distil Hubert baseline	13	0.2432	68.4% (63.6%- 73.2%)	0.9477	828s	70.1% (epoch 1)
Hubert sgd	6	0.6926	48.5% (43.3% - 53.8%)	0.6934	599s	48.6% (epoch 1)
Distil Hubert sgd	50	0.6920	57.2% (52.0% - 62.4%)	0.6924	3167s	57.2% (epoch 50)
Hubert adam, dropout 0.2	16	0.4974	71.9% (67.2% - 76.5%)	0.6371	1598s	72.2% (epoch 11)
Distil Hubert adam, dropout 0.2	9	0.4299	68.8% (64.0% - 73.6%)	0.6994	575s	70.7% (epoch 4)
Hubert adam, dropout 0.3	23	0.4301	68.8% (64.0% - 73.6%)	0.6822	2302s	71.3% (epoch 18)
Distil Hubert adam, dropout 0.3	13	0.2411	67.2% (62.3% - 72.1%)	0.8982	827s	70.4% (epoch 8)
Distil Hubert sgd, dropout 0.2	50	0.6920	56.8% (51.5% - 62.0%)	0.6925	3252s	56.8% (epoch 50)
Distil Hubert sgd, dropout 0.3	50	0.6919	55.8% (50.6% - 61.1%)	0.6926	3165s	55.8% (epoch 50)
Hubert adam, drop 0.2, adapter h_s128	6	0.6096	68.0%(63.2- 72.9%)	0.6481	601s	71.4% (epoch 1)
Hubert adam, drop 0.2, adapter h_s 32	15	0.5435	71.8% (67.1% - 76.4%)	0.6434	1492s	72.8% (epoch 10)
Hubert adam, drop 0.2, adapter h_s 32, norm	21	0.5014	70.8% (66.1% - 75.5%)	0.7632	2085s	70-9% (epoch 16)

Hubert adam, drop 0.2, adapter h_s 128, norm	12	0.5293	67.6% (62.7% - 72.5%)	0.7536	1192s	69.6% (epoch 7)

Hubert layers combination

In this case, for the best model of the above combination, we have decided to calculate the final output as the average of the output of the last two layers of the model. In this case, our results have been:

Hubert adam, drop 0.2, adapter h_s 32 with mean of last two output layers	28	0.5294	70 % (65.2% - 74.7%)	0.6396	2795s	70.7% (epoch 23)

We can see how using the average of the two layers has lowered the error by a small magnitude but does not give a better AUC than the model without averaging.

Conclusions

In conclusion, this paper explored two different approaches for accurate diagnosis of Covid-19 using speech processing on a dataset of breath, cough and voice recordings. The first approach was based on the unsupervised Mel filter algorithm, which proved to be effective in voice-based disease detection due to its ability to capture spectral features and handle speaker variability. However, this approach requires manual feature engineering and can suffer from over-fitting on small datasets. On the other hand, the supervised Hubert model was used, which learns low- and high-level features directly from audio data and exploits temporal dependencies through end-to-end learning. Although this model can suffer from overfitting due to the deep learning data requirements, it was observed that making adjustments to the hyperparameters and incorporating normalisation and dropout layers can help mitigate this problem. Overall, both approaches offer their particular strengths and weaknesses, and the choice between them will depend on the size and characteristics of the dataset, as well as interpretability and computational efficiency requirements.