POE Práctica 5

Sonia Castro Paniello
Eduard Ramon Aliaga

June 2023

1 Exploratory Data Analysis, Error Analysis

When it comes to the data we utilize, there are several factors to consider. Firstly, there is a considerable variability in audio characteristics when recording respiration, coughs, or speech. As a result, this variability can contribute to higher error rates in the model's predictions.Additionally, although respiratory symptoms such as coughing and changes in speech patterns can be associated with COVID-19, it is important to note that these symptoms are not exclusive to the disease. Lastly, the limited training data available consists of only 2160 samples, which may affect the model's overall performance and generalizability.

In examining the input data, it comprises an index, features (consisting of 60 coefficients multiplied by 1000 samples), and a corresponding label. We have ensured that the classes are balanced, with 1080 samples for each class. Various statistical measures such as mean, maximum, minimum, standard deviation, and skewness have been computed for the different classes; however, they have not yielded significant insights. To visually explore the data, t-SNE dimensionality reduction technique has been applied to reduce the features to two dimensions, with color encoding representing each class. Unfortunately, this visualization does not exhibit clear differentiation among the classes, suggesting potential challenges in class separation for the model.

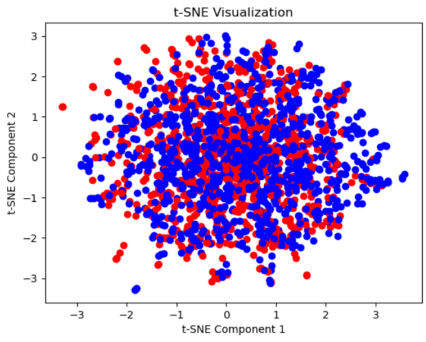

Figure 1: Tsne plot.

Additionally, in order to standardize the input sizes for both the VGG and Hubert models, certain preprocessing steps have been undertaken. For the VGG model, zero-padding and audio cropping techniques have been applied to achieve the desired input size. Consequently, this process introduces modified and "artificial" features, which may complicate

the learning process for the model. Similarly, for the Hubert model, resizing of inputs has been performed, accompanied by the removal of the mean (without normalization).

We also printed confussion matrix both for mel spectogram and for hubert.

```
[[7224 5736]
 [3027 9933]]
```

Figure 2: Mel confussion matrix.

```
[[11303  5977]
 [ 5352 11928]]
```

Figure 3: Hubert confussion matrix.

From the matrices, we can observe that both models struggle more with false negatives than false positives.

2 Hyperparameters and model variations

In this section, we have experimented with different hyperparameters for both models in order to observe how changes in these parameters affect performance and attempt to maximize the test score in both cases.

	Architecture	batch_size	test_bs	Epochs	lr	momentum	optimizer	w_si	w_str
Trial 1	VGG13	32	32	100	0,0001	0,9	Adam	0,04	0,02
Trial 2	VGG13	32	32	100	0,001	0,9	Adam	0,04	0,02
Trial 3	VGG13	32	32	100	0,00005	0,9	Adam	0,04	0,02
Trial 4	VGG13	32	32	100	0,005	0,9	Adam	0,04	0,02
Trial 5	VGG13	32	32	100	0,001	0,9	Adam	0,08	0,02
Trial 6	VGG13	32	32	100	0,001	0,9	Adam	0,1	0,02
Trial 7	VGG13	32	32	100	0,0001	0,9	Adam	0,04	0,01
Trial 8	VGG13	32	32	100	0,0001	0,9	Adam	0,04	0,03
Trial 9	VGG16	32	32	100	0,0001	0,9	Adam	0,04	0,02
Trial 10	VGG16	32	32	100	0,00005	0,9	Adam	0,04	0,02
Trial 11	VGG16	32	32	100	0,001	0,9	Adam	0,04	0,02
Trial 12	VGG16	32	32	100	0,0001	0,9	Adam	0,08	0,02
Trial 13	VGG16	32	32	100	0,0001	0,9	Adam	0,04	0,01
Trial 14	VGG16	32	32	100	0,0001	0,9	Adam	0,04	0,03
Trial 15	VGG16	32	32	100	0,0001	0,9	SGD	0,08	0,02
Trial 16	VGG16	32	32	100	0,001	0,9	SGD	0,04	0,02
Trial 17	VGG16	32	32	100	0,00005	0,09	SGD	0,04	0,02
Trial 18	VGG16	32	32	100	0,01	0,9	SGD	0,04	0,02
Trial 19	VGG19	32	32	100	0,0001	0,9	Adam	0,04	0,02
Trial 20	VGG19	32	32	100	0,00005	0,9	Adam	0,04	0,02
Trial 21	VGG19	32	32	100	0,0001	0,9	Adam	0,08	0,02
Trial 22	VGG19	32	32	100	0,0001	0,9	Adam	0,04	0,01
Trial 23	VGG19	32	32	100	0,0001	0,9	Adam	0,04	0,03
Trial 24	VGG19	32	32	100	0,0001	0,9	Adam	0,03	0,02
Trial 25	VGG19	32	32	100	0,001	0,9	SGD	0,04	0,02
Trial 26	VGG19	32	32	100	0,0001	0,9	SGD	0,04	0,02

Table 1: Hyperparameters in each trial of VGG models.

	Time	Stop Epoch	Validation	Test Score
Trial 1	964s	14	66,2	0,65856
Trial 2	657s	9	65,60	0,63698
Trial 3	958s	8	65,40	0,68552
Trial 4	442s	6	51	0,62437
Trial 5	554s	8	64,4	0,62345
Trial 6	508s	7	56,4	0,59232
Trial 7	1539s	12	65,8	0,66736
Trial 8	777s	14	65,6	0,69472
Trial 9	787s	10	65,6	0,65625
Trial 10	544s	7	65,3	0,66583
Trial 11	506s	7	65,10	0,64416
Trial 12	725s	10	66,2	0,66874
Trial 13	1837s	12	66,7	0,68736
Trial 14	537s	9	64,9	0,66483
Trial 15	3739s	55	59,9	0,666417
Trial 16	2222s	32	67	0,67204
Trial 17	7574s	100	59,9	0,66406
Trial 18	859s	12	65,4	0,66944
Trial 19	647s	9	65,5	0,66105
Trial 20	559s	14	66,3	0,67817
Trial 21	652s	8	65,3	0,64821
Trial 22	1385s	11	63,3	0,67944
Trial 23	566s	11	66,4	0,65619
Trial 24	607s	9	63,3	0,66192
Trial 25	2576s	30	66,4	0,64439
Trial 26	3577s	58	58,7	0,64109

Table 2: Resultsof VGG models

As we can see, we have performed several executions, tuning different hyperparameters that we believed would have a more significant impact and observing how they affect the model:

- Architecture: The initial tests for each architecture correspond to the baseline. We can see that they have similar scores, although VGG19 takes less time and achieves the best result. Generally, no architecture stands out significantly among the three; all provide similar results. However, on average, VGG19 obtains better accuracy in less time. Nevertheless, we can observe that the maximum precision is achieved with VGG13 (trial 8), whereas for the same experiment, it is lower for VGG19.
- Learning rate: In general, a lower learning rate for the Adam optimizer allows the model to converge in less time and with a better score than the baseline for all architectures. For most combinations, using a lower learning rate than the baseline is better. For SGD, the change in the learning rate has not resulted in a significant impact on accuracy, but the execution time has been considerably affected by the change. As expected, with a lower learning rate, the time increases considerably because, unlike the Adam optimizer, the learning rate is not learned but fixed, and

if it is smaller, it takes longer to reach the minimum (smaller steps).

- Optimizer: As mentioned, the most noticeable change is the execution time due to the non-learning nature of the learning rate in the SGD process. Regarding the results of the executions where only this parameter was changed, in general, Adam outperforms SGD, and considering efficiency, we deduce that Adam is the better optimizer.
- Window size: We have only tested window sizes larger than the baseline because we assumed that a larger window would make the effect of contours smoother and capture more context. However, we conducted a test for VGG19 (trial 24), slightly decreasing the size, and the result was practically the same. On the other hand, we can see that an increase, doubling the size, improves the model's accuracy in all cases except for 0.1. From this, we can deduce that beyond 0.1, the model is negatively affected by this change and begins to overfit.
- Window stride: We can observe that decreasing the window stride slightly improves the model, while increasing it proves to be detrimental. Clearly, this parameter significantly affects precision due to the subtle changes it produces, indicating its important impact. This effect was expected, as with a smaller stride, we were able to smooth out the edge effect more and thus reduce the variability introduced in the model.

In general, the performance of the model is not greatly affected by changes in the parameters. The precision remains around 0.65 (excluding isolated cases). So we can say the model is robust to changes in its parameters.

Changes could have been tried in other parameters, such as momentum in SGD, patience, number of epochs, etc. However, considering the limited execution time, the need to tune two different models, and the sometimes significant time constraints, we decided to prioritize the mentioned parameters.

For the Hubert model, we conducted the following trials and obtained the following results.

	batch_size	test_bs	Epochs	lr	momentum	optimizer
Trial 1	22	22	50	0,0002	0.9	Adam
Trial 2	22	22	50	0,002	0,9	Adam
Trial 3	22	22	50	0,00005	0,9	Adam
Trial 4	22	22	50	0,00001	0,9	Adam
Trial 5	22	22	50	0,00005	0,9	SGD
Trial 6	22	22	50	0,002	0,9	SGD
Trial 7	22	22	50	0,01	0,9	SGD
Trial 8	22	22	50	0,002	0,5	SGD
Trial 9	16	16	50	0,00005	0,9	Adam
Trial 10	8	8	50	0,00005	0,9	Adam
Trial 11	16	16	50	0,0001	0,9	Adam
Trial 12	16	16	50	0,002	0,9	SGD
Trial 13	16	16	50	0,00005	0,9	Adam
Trial 14	16	16	50	0,000	5 0,9	Adam

Table 3: Hyperparameters in each trial of Hubert models.

	Patience	Epochs	Test Score	Time	Validation
Trial 1	5	16	0,73481	1668.3s	72,20
Trial 2	5	18	0,70094	994s	72
Trial 3	5	39	0,73093	3891s	71,60
Trial 4	5	27	0,59018	2677s	65,10
Trial 5	5	9	0,4621	599s	49,60
Trial 6	5	17	0,60332	1686s	64,20
Trial 7	5	15	0,7	1490s	69,80
Trial 8	5	47	0,60609	4679s	63,80
Trial 9	5	31	0,73869	3125s	72,20
Trial 10	5	32	0,7384	3420s	72,10
Trial 11	5	24	0,74117	2424s	72,90
Trial 12	5	14	0,62061	1414s	65,10
Trial 13	10	50	0,73446	5025s	73
Trial 14	16	16	50	0,000	5 0,9

Table 4: Results of Hubert models.

As we can see, we have performed several executions, tuning different hyperparameters that we believed would have a more significant impact and observing how they affect the model:

- Batch size: Surprisingly, decreasing the batch size has improved the model compared to the baseline, achieving the best result of all experiments in Trial 11. Even reducing it to 8 maintains a similar performance with high precision. Therefore, we deduce that the impact is positive and quite significant. We couldn't perform trials by increasing this parameter due to memory issues; the model turned out to be too large.
- Learning rate: For the Adam optimizer, the learning rate does not seem to have a significant impact on the model. In the conducted trials, the results barely vary even with a large change in the parameter. On the other hand, SGD is considerably affected, and a significant decrease in the learning rate results in a fatal outcome (Trial 5).
- Optimizer: Once again, the difference between Adam and SGD is evident. In all cases, Adam achieves better results, and the difference is significant. SGD significantly reduces the model's performance, increases the execution time, and greatly decreases the accuracy. Moreover, it is highly affected by changes in the learning rate, making it relatively difficult to find an optimal value.
- Momentum: For the SGD executions, momentum does not affect the outcome as much as the execution time. Comparing Trials 6 and 8, we can see that reducing the momentum to almost half of the baseline results in nearly doubling the execution time. Therefore, we can deduce that a lower momentum makes the model much slower, which is expected since with a low learning rate and low momentum, the steps taken by SGD are very small and require more time to converge.

Based on the conducted trials, results, and analysis, it can be concluded that the Hubert model outperformed the Mel model. The precision achieved of the former remained higher

than the latter (although the execution times differ and in general is higher for Hubert). On the other hand, the Mel model showed less significant variations in performance but with lower precision scores. This indicates that the Mel model demonstrated better stability and robustness compared to Hubert but less accuracy.

3 Robust cross-validation procedures for small databases

In this section, in order to achieve robust cross-validation procedures for small databases, we applied dropout and layer normalization to the models.

For the VGG models, we only applied dropout since batch normalization was already implemented in the code, and we believed it was more suitable in this case than layer normalization. In the following table, we can see the results of applying dropout values of 0.5 and 0.6 to each VGG model, along with an additional experiment based on the results of these two cases. All experiments are conducted over the baseline.

	Architecture	Dropout	Time	Stop Epoch	Validation	Test Score
Trial 1	VGG13	0.5	662,90	8	65,30	0,67135
Trial 2	VGG13	0.6	765,7	10	65,00	0,67661
Trial 3	VGG13	0.65	6 749,8	10	64,20	0,66070
Trial 4	VGG16	0.6	1203,6	16	65,60	0,68402
Trial 5	VGG16	0.5	674,9	8	65,60	0,64271
Trial 6	VGG16	0.3	1038,3	14	66,40	0,69917
Trial 7	VGG19	0.6	838,7	12	66,50	0,65885
Trial 8	VGG19	0.5	892,6	12	68,20	0,68112
Trial 9	VGG19	0.55	677,7	9	66,90	0,63016

Table 5: Accuracy of VGG models with different regularization techniques.

As observed, in the case of VGG13, the best result was obtained with a dropout of 0.6, although a dropout of 0.5 already prevented overfitting. Improving the baseline (Table 1 Trial 1) from 0.65856 to 0.67135.

For VGG16, the best result was achieved with a dropout of 0.3, despite some overfitting, leading us to believe that the optimal value would be 0.6 again. Improving the baseline (able 1 Trial 9) from 0,65625 to 0.68402.

Lastly, for VGG19, the best result was obtained with a dropout of 0.5, and based on the validation and test scores, there is no indication of overfitting. Improving the baseline (Table 1 Trial 19)from 0,66105 to 0.68112.

In the case of Hubert, we experimented with various values of dropout, as well as adding a normalization layer to the adapter, the classifier, or both, and also varying the hidden size of the adapter. These experiments were conducted based on the baseline model.

	LayerNorm	Dropout	Hidden Adapt.	Time	Stop Epoch	Validation	Test Score
Trial 1	Adapt.+Classif.	0.1	64	1645,7	16	70,4	0,71878
Trial 2	Classif.	0.1	64	1650,1	16	72,5	0,74106
Trial 3	Adapt.	0.1	64	1551,1	15	71	0,73689
Trial 4	None	0.1	64	1668,3	16	72,2	0,73481
Trial 5	Adapt.+Classif.	0.2	64	1659,6	16	71	0,73163
Trial 6	Classif.	0.2	64	1662,8	16	72,3	0,71526
Trial 7	Adapt.	0.2	64	1457,1	14	71,2	0,73307
Trial 8	None	0.2	64	1664,7	16	70,6	0,71838
Trial 9	None	0.3	64	1842,3	18	70,5	0,72561
Trial 10	None	0.4	64	1854,6	18	72,4	0,72312
Trial 11	None	0.1	32	1551,8	15	71,7	0,73475
Trial 12	None	0.1	128	665,6	6	68,5	0,66765
Trial 13	None	0.1	256	1865,5	18	71,5	0,70750
Trial 14	Adapt.	0.1	128	1260,9	12	69,8	0,72793
Trial 15	Adapt.+Classif.	0.1	128	1561,5	15	69,5	0,73944
Trial 16	Adapt.+Classif.	0.2	128	1550	15	69,4	0,75251
Trial 17	Adapt.+Classif.	0.1	256	1565	15	71.8	0,72908
Trial 18	Adapt.+Classif.	0.2	256	1564,6	15	72,3	0,74117
Trial 19	Adapt.+Classif.	0.25	256	1563,7	15	71,7	0,73932

Table 6: Accuracy of Hubert models with different regularization techniques.

To begin with, we examined how the model's performance would improve by adding layer normalization layers without varying the dropout. We observed that the results improved when layer normalization was added either to the classifier or to the adapter, but deteriorated when it appeared in both.

Therefore, we decided to experiment by increasing the dropout to 0.2. In this case, notable improvements were observed only when the layer normalization layer was added to the adapter or to both the adapter and the classifier. Adding it solely to the classifier did not show any improvements.

However, simply adding dropout by itself did not lead to an improvement over the baseline (Trial 4). Increasing the dropout to 0.3 initially resulted in an improvement, but when it started to decrease at 0.4, we discontinued the experiments. From this initial phase, the best case obtained was 0.74106 (Trial 2).

In a second phase of the experiment, we modified the hidden layer size of the adapter. Starting from the baseline, we changed its value to 32, 128, and 256 (Trials 11-13). The best result was obtained with a hidden layer size of 32, but we suspected that this was due to overfitting in the other two cases. Therefore, we decided to conduct further experiments with these larger layer sizes, along with dropout and normalization, in order to achieve a maximum score.

In the case of the 128-hiddenlayer models, we first added a normalization layer to the adapter, then to the classifier, and finally increased the dropout to 0.2. It was at this point that we achieved the best result of 0.75251. Unfortunately, we did not have enough

time to conduct further experiments with higher dropout values, but it would have been interesting to explore. The next section of the paper discusses more experiments conducted with this particular case.

In the case of the 256-hiddenlayer models, we attempted to directly execute the configuration that had worked best for the 128-layer models. Since these later trials aimed to maximize the model's output rather than investigating the effects of parameters as in the initial experiments. However, we did not achieve as good results as with the 128-layer models.

We also decided to carry out the following experiments with Trial 11 from Table 4 (the best result from the previous section, 0.74117). We were able to achieve a slightly better result in Trial 2 of the following table by applying layer normalization in the Adapter.

	LayerNorm	Dropout	Hidden Adapt.	Time	Stop Epoch	Validation	Test Score
Trial 1	Classif.	0.1	64	1865,5	16	70,1	0,72532
Trial 2	Adapt.	0.1	64	1793,1	17	71,6	0,74146
Trial 3	None	0.1	64	2424	24	72,9	0,74117
Trial 4	None	0.2	64	1779,4	16	71,4	0,72052

Table 7: Accuracy of Table 4 Trial 11 with different regularization techniques.

4 Weighted average of layer's hidden states with trained parameters

In this last section, we have tried taking hidden states from layers prior to the last one or weighted averages between them. These experiments shown in the following table were conducted on the baseline.

	Layers	Time	Stop Epoch	Validation	Test Score
Trial 1	Penultimate	1405,2	14	71,5	0,75552
Trial 2	Antepenultimate	1557,7	16	70,3	0,70558
Trial 3	Mean Last 2	1847,8	18	72	0,73122
Trial 4	Mean Antepen. and Penult.	1988,1	20	70,5	0,73799
Trial 5	Mean Last 3	1956	19	71,7	0,70779

Table 8: Accuracy with different hiddenlayers taken from the model.

Surprisingly, the best result, both in validation and test, was obtained by taking the hidden states from the penultimate layer.

Based on the results from previous experiments and questions, we attempted to take the hidden states from the penultimate layer in several cases. First, we tried it with Trial 11 from Table 4 since it had been the best result so far, but the score decreased from 0.74117 to 0.73255. We tried again with Trial 2 from Table 7 in this case the score increased from 0.74146 to 0.74412.

Lastly, we attempted it with the best result from the previous section, Trial 17 from Table 6, but the score decreased from 0.75251 to 0.7189.