Deep Learning

880008-M-6

Assignment

Using Deep Learning to Perform Multi-Class Classification on the Covid19 Chest X-ray Dataset

Report by:

Alessandro Fassina (2104671)

Group Number: 8

Group Members:

Andreea Cotfas
Alessandro Fassina
Antoine Guay-Molnar
Sebastian Vasquez

March 2023

1. Problem Definition

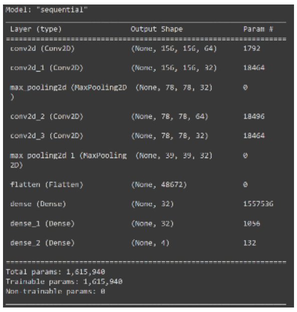
The problem at hand concerns the identification of Covid-19 and other forms of Pneumonia from lung scans. From a Machine Learning perspective, this task can be addressed as multiclass classification. The input is a single lung-scan image, while the output, or target variable, defines the state of health based on the image. The state of health is described by four alternative class labels, namely Bacterial Pneumonia, COVID-19, No Pneumonia (healthy), and Viral Pneumonia. The dataset used for this assignment is the COVID-19 X-ray dataset. The dataset contains 6,500 images of AP/PA chest x-rays.

2. Dataset Preprocessing

After preliminary operations on the input images, such as re-sizing to 156x156, output and input data are split into training, validation, and test sets in proportions of 60%, 20%, and 20% with respect to the whole dataset. Data preprocessing consists of data normalization and label encoding. As the minimum is 0 and the maximum is 255, normalization is simply performed by dividing by 255. Label encoding turns the labels of each sample into a binary vector with a 1 in correspondence of the represented category and zeros for the other categories.

3. Baseline Model

The baseline model is a feedforward neural network based on the Sequential Keras model. The model presents a first part where convolution takes place and a second part made up of two dense layers, and a final output layer. The output layer is made up of four units, since four are the categories of the dependent variable. The activation function for every layer is Rectified Linear Units (ReLU), except for the last layer where the activation function is softmax. The baseline model's architecture contains also the specification of an optimizer, a loss function, and metrics of interest. The optimizer is set to Adam. As this is a case of multiclass classification, the type of loss function selected is categorical cross entropy, and the metric of interest is accuracy. An overview of the model's architecture is provided by Figure 1. Then, the baseline model is fitted to the training set, and evaluated on the validation set. Also, the batch size is set to 32 within the fit function, implying that a stochastic method called mini batch is used to train the model. Finally, the number of epochs is set to ten. Since the dataset suffers from data imbalance with regards to Covid-19 class, the F1 score holds significant importance for model selection. F1 score for the baseline is 73.86%. The accuracy score against test data amounts to 73.38%. The two following graphs compare validation and training loss, and validation and accuracy, respectively. At the fourth epoch we can notice that the validation loss starts to increase while the training loss keeps decreasing. In this scenario, the model is said to overfit the training data. At the same time, training accuracy keeps increasing, while validation accuracy starts decreasing, another signal of overfitting (see Figures 2-3).



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 156, 156, 64)      1792

 conv2d_1 (Conv2D)           (None, 156, 156, 32)      18464

 max_pooling2d (MaxPooling2D  (None, 78, 78, 32)       0
 )

 conv2d_2 (Conv2D)           (None, 78, 78, 64)        18496

 conv2d_3 (Conv2D)           (None, 78, 78, 32)        18464

 max pooling2d 1 (MaxPooling  (None, 39, 39, 32)       0
 2D)

 flatten (Flatten)           (None, 48672)             0

 dense (Dense)               (None, 32)                1557536

 dense_1 (Dense)             (None, 32)                1056

 dense_2 (Dense)             (None, 4)                 132

=================================================================
Total params: 1,615,940
Trainable params: 1,615,940
Non-trainable params: 0
_________________________________________________________________
```

Figure 1.

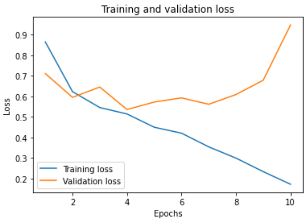


Figure 2.

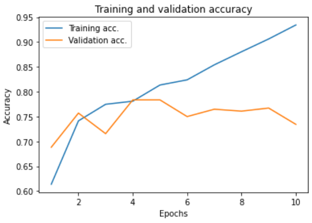


Figure 3.

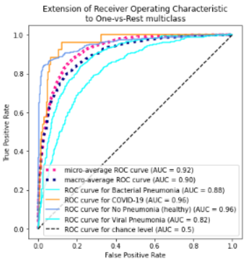


Figure 4.

The Receiver Operating Characteristic variant for multiclass classification is plotted in Figure 4. The ROC curve in multiclass classification tracks the trade-off between True Positive Rate and False Positive Rate in a "One vs. Rest" approach (cite). In addition, the micro and macro averages are plotted. Lastly, the confusion matrix provides information on misclassification, where rows correspond to the true class and the columns to the predicted class (see Figure 5).

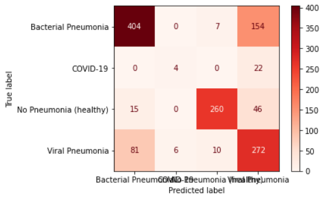
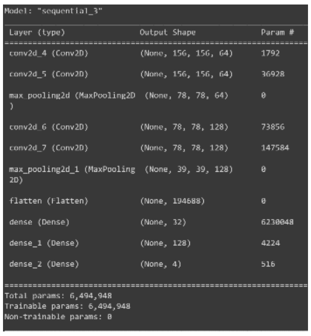


Figure 5.

4. Improved (Fine-tuned) Model

In this phase, the baseline model is subject to hyperparameter tuning. Tuning is carried out by means of firstly naïve trials, and secondly by Talos tuner. Naïve trials consist of manual incremental changes in order to detect eventual increases in the accuracy score compared to the one of the baseline. These manual changes include the addition of 2D-convolutional layers, the reduction of the learning rate, and the addition of dropout layers. In general, the addition of layers to neural network architecture increases its complexity, thus either returning better performance, or triggering overfitting. In fact, this experiment resulted in worse performance than the one of the baseline. The reduction of the learning rate does not provoke a gain in accuracy. Lastly, dropout layers act as a regularization solution. Nevertheless, the addition of dropout layers does not return the expected improvement in performance. Overall, none of the tuned models resulting from initial trials performs better than the baseline. Tuning is then performed through Talos Tuner (Zamzmi et al., 2021). The parameters to be tested are the optimizer, the regularizer, the number of neurons per layer, and the batch size. The loss function and the number of epochs is limited to categorical cross entropy, and ten, respectively. The possible optimizers are Nadam and Adam, while the possible regularizers include L2, elastic net, and none. The number of neurons per layer takes on values of either 16, 32, 64, or 128. Lastly, the batch size is set to either 16 or 32. Activation functions remain untouched. The outcome from



```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 156, 156, 64)      1792

 conv2d_5 (Conv2D)           (None, 156, 156, 64)      36928

 max_pooling2d (MaxPooling2D  (None, 78, 78, 64)       0
 )

 conv2d_6 (Conv2D)           (None, 78, 78, 128)       73856

 conv2d_7 (Conv2D)           (None, 78, 78, 128)       147584

 max_pooling2d_1 (MaxPooling  (None, 39, 39, 128)      0
 2D)

 flatten (Flatten)           (None, 194688)            0

 dense (Dense)               (None, 32)                6230048

 dense_1 (Dense)             (None, 128)               4224

 dense_2 (Dense)             (None, 4)                 516

=================================================================
Total params: 6,494,948
Trainable params: 6,494,948
Non-trainable params: 0
_________________________________________________________________
```

Figure 6.

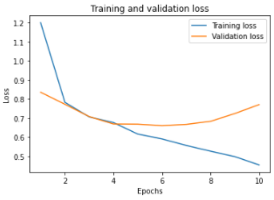


Figure 7.

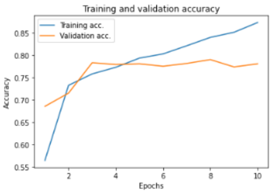


Figure 8.

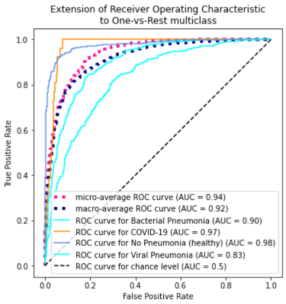


Figure 9.

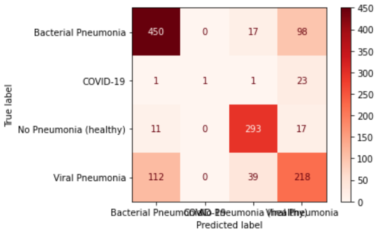
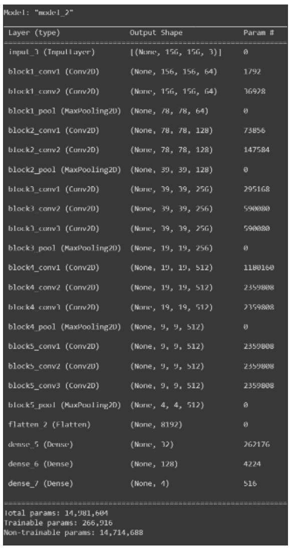


Figure 10.

Talos Tuner is a model that possesses the architecture shown in Figure 6, and its training and validation loss, as well as its training and validation accuracy are reported in Figures 7-8. The model's F1 score amounts to to 74.20%, which is higher than the one for the baseline. Figure 7-8 display the training and validation loss and accuracies of tuned model. Compared to the baseline, also the tuned model starts to overfit after the fourth epoch. However, the increase in validation loss is less steep through the epochs and validation accuracy remains stable instead of dropping. Figures 9-10 report the class-specific ROC curves and the confusion matrix. The choice for the possible alternatives of the optimizer Adam and Nadam is driven by the content of both Chapter 8 of the course book, where Adam optimizer is described not only to be based on an adaptive learning rate algorithm, but to also take momentum into account to reach local minima faster. Nadam is then considered as a possible alternative due to satisfactory results in terms of accuracy in similar studies (Silva et al., 2020).

5. Transfer Learning Model

As for transfer learning, resnet50 and vgg16 are employed in parallel (Tammina, 2019). The two models build from the tuned model produced by Talos and are expected to improve its performance based on training and hyperparameter tuning on a large amount of data. The two resulting models from transfer learning are compared and vgg16 performs better overall. Therefore, the vgg16 model is selected to be compared against the baseline and the tuned model. The architecture of the vgg16 model, its training and validation losses, and its training and validation accuracy are reported in Figure 11, 12, ad 13 respectively. Additionally, vgg16 reaches a F1 score of 77.32%. The class-specific ROC curves and the confusion matrix for vgg16 are reported in Figures 14-15.



```
Model: "model_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_3 (InputLayer)        [(None, 156, 156, 3)]     0

 block1_conv1 (Conv2D)       (None, 156, 156, 64)      1792

 block1_conv2 (Conv2D)       (None, 156, 156, 64)      36928

 block1_pool (MaxPooling2D)  (None, 78, 78, 64)        0

 block2_conv1 (Conv2D)       (None, 78, 78, 128)       73856

 block2_conv2 (Conv2D)       (None, 78, 78, 128)       147584

 block2_pool (MaxPooling2D)  (None, 39, 39, 128)       0

 block3_conv1 (Conv2D)       (None, 39, 39, 256)       295168

 block3_conv2 (Conv2D)       (None, 39, 39, 256)       590080

 block3_conv3 (Conv2D)       (None, 39, 39, 256)       590080

 block3_pool (MaxPooling2D)  (None, 19, 19, 256)       0

 block4_conv1 (Conv2D)       (None, 19, 19, 512)       1180160

 block4_conv2 (Conv2D)       (None, 19, 19, 512)       2359808

 block4_conv3 (Conv2D)       (None, 19, 19, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 9, 9, 512)         0

 block5_conv1 (Conv2D)       (None, 9, 9, 512)         2359808

 block5_conv2 (Conv2D)       (None, 9, 9, 512)         2359808

 block5_conv3 (Conv2D)       (None, 9, 9, 512)         2359808

 block5_pool (MaxPooling2D)  (None, 4, 4, 512)         0

 flatten_2 (Flatten)         (None, 8192)              0

 dense_5 (Dense)             (None, 32)                262176

 dense_6 (Dense)             (None, 128)               4224

 dense_7 (Dense)             (None, 4)                 516

=================================================================
Total params: 14,981,604
Trainable params: 266,916
Non-trainable params: 14,714,688
```

Figure 11.

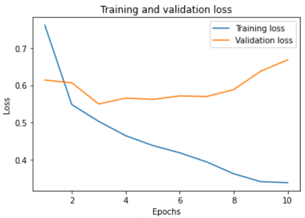


Figure 12.

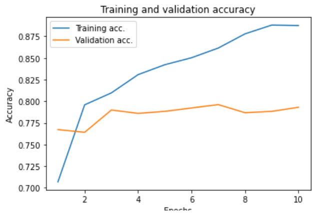


Figure 13.

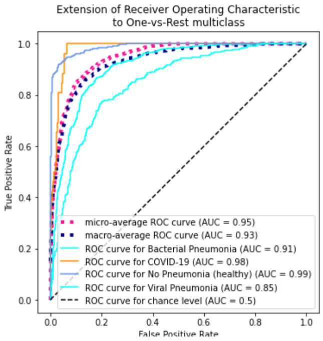


Figure 14.

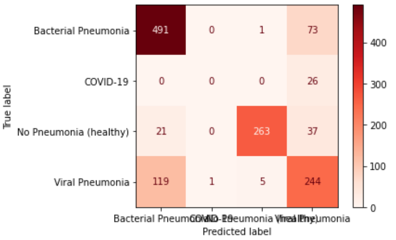


Figure 15.

6. Discussion

In the case of multiclass classification, the One vs. Rest approach allows to display a class-specific ROC curve. For instance, Covid-19 has its True Positive Rate's denominator consisting of the sum of all cases where Covid-19 is correctly detected (True Positives) and the cases where it is misclassified (False Negative). Whereas, the False Positive Rate's denominator includes the sum of all cases where the prediction is Covid-19, but the actual class label is different (False Positive), and all cases where Covid-19 is correctly not detected (True Negatives). That said, due to class imbalance, Covid-19 always has a significant amount of True Negatives, and with the improvement of model prediction in the other classes, the ROC curve for Covid-19 is expected to have a high AUC, comparable to the one of the other classes, even though the True Positives (correct predictions) count is smaller. As expected, in the tuned model the AUC for Covid-19 increases due to the improvement in recognizing the other classes (True Negatives). As for hyperparameter tuning through Talos, the best model among 30 is selected based on both validation accuracy and F1 score. The selected model (27) scores best both in terms of validation accuracy (78.28%) and F1 score (78.19%). When comparing the baseline model, the tuned model, and the vgg16 model, there is a significant difference in the values that the validation loss assumes in the graphs. While the baseline's validation loss ranges from 0.7 reaches 0.9 in ten epochs, the one for the tuned model presents a narrower range, and the one for vgg16 consistently hoovers around 0.6. This observation supports the conclusion that vgg16 is better at keeping validation error under control. Further, while validation accuracy in the baseline model eventually drops, it remains consistent through the epochs in vgg16, and at a slightly higher level than the tuned model. Moreover, by looking at the AUC scores of each model, we can observe that vgg16 outperforms both the tuned model and the baseline. The confusion matrices also report different behavior from the models. In particular, while the tuned model becomes better than the baseline at predicting Bacterial Pneumonia and No Pneumonia, vgg16 is more able to recognize Bacterial Pneumonia and No Pneumonia (healthy), at the expense of Viral Pneumonia and Covid-19 True Positive counts. In addition, vgg16 achieves the highest F1 score. Provided the importance of F1 score in this scenario and the observations so far, vgg16 is considered to be the most suitable model to address the task.

7. References

Silva, A. B., Santos, D. F., Tosta, T. A., Martins, A. S., Neves, L. A., Travenclo, B. A., Faria, P. R., & Nascimento, M. Z. (2020). Segmentation of oral epithelial dysplasias employing mask R-CNN and color normalization. *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. https://doi.org/10.1109/bibm49941.2020.9313101

Tammina, S. (2019). Transfer learning using VGG-16 with deep Convolutional neural network for classifying images. *International Journal of Scientific and Research Publications (IJSRP)*, *9*(10), p9420. https://doi.org/10.29322/ijsrp.9.10.2019.p9420

Zamzmi, G., Rajaraman, S., & Antani, S. (2021). UMS-rep: Unified modality-specific representation for efficient medical image analysis. *Informatics in Medicine Unlocked*, *24*, 100571. https://doi.org/10.1016/j.imu.2021.100571