Covid-19 and Pneumonia Chest X-Ray Classification

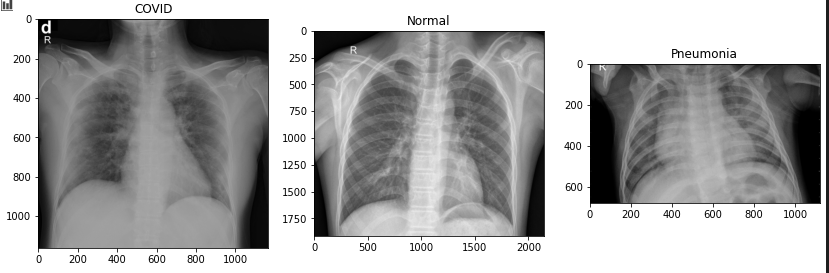
Daniel Chang

**Dataset and Data Preparation:**

Data came from Mendeley Data. The data consists of 603 chest x-rays in 3 folders labeled Normal, Covid-19 cases, and pneumonia. The chest X-rays were mostly taken from the front, but there were a few that were also from the side. Based on the naming of the files, I believe most of the X-rays that were taken from the side also had an image taken from the front, but there is no way to be sure. I left all X-rays taken from the side in the dataset.

In the sample images, we can see that pneumonia lungs are cloudier in the lung area than the normal X-ray. The Covid-19 X-ray appears to be slightly less clear when compared to the normal X-ray, but not as cloudy as the pneumonia X-ray.

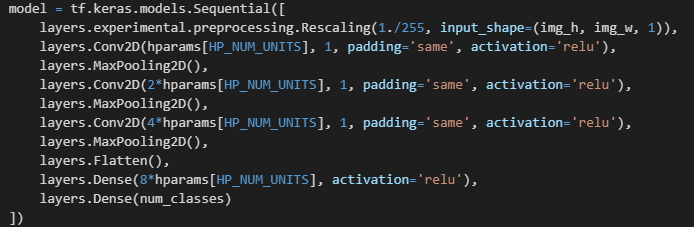
In the 3 folders, 221 are Covid-19, 234 are Normal, and 148 are Pneumonia. All files were imported using TensorFlow image\_dataset\_from\_directory function, and resized to 512x512 and used grayscale color space. I ended up using 512x512 resolution because I thought the default 256x256 image might not have enough detail to accurately predict. Some of the images were annotated, but I don’t think it would cause an issue since the annotations were on the edges of the images.



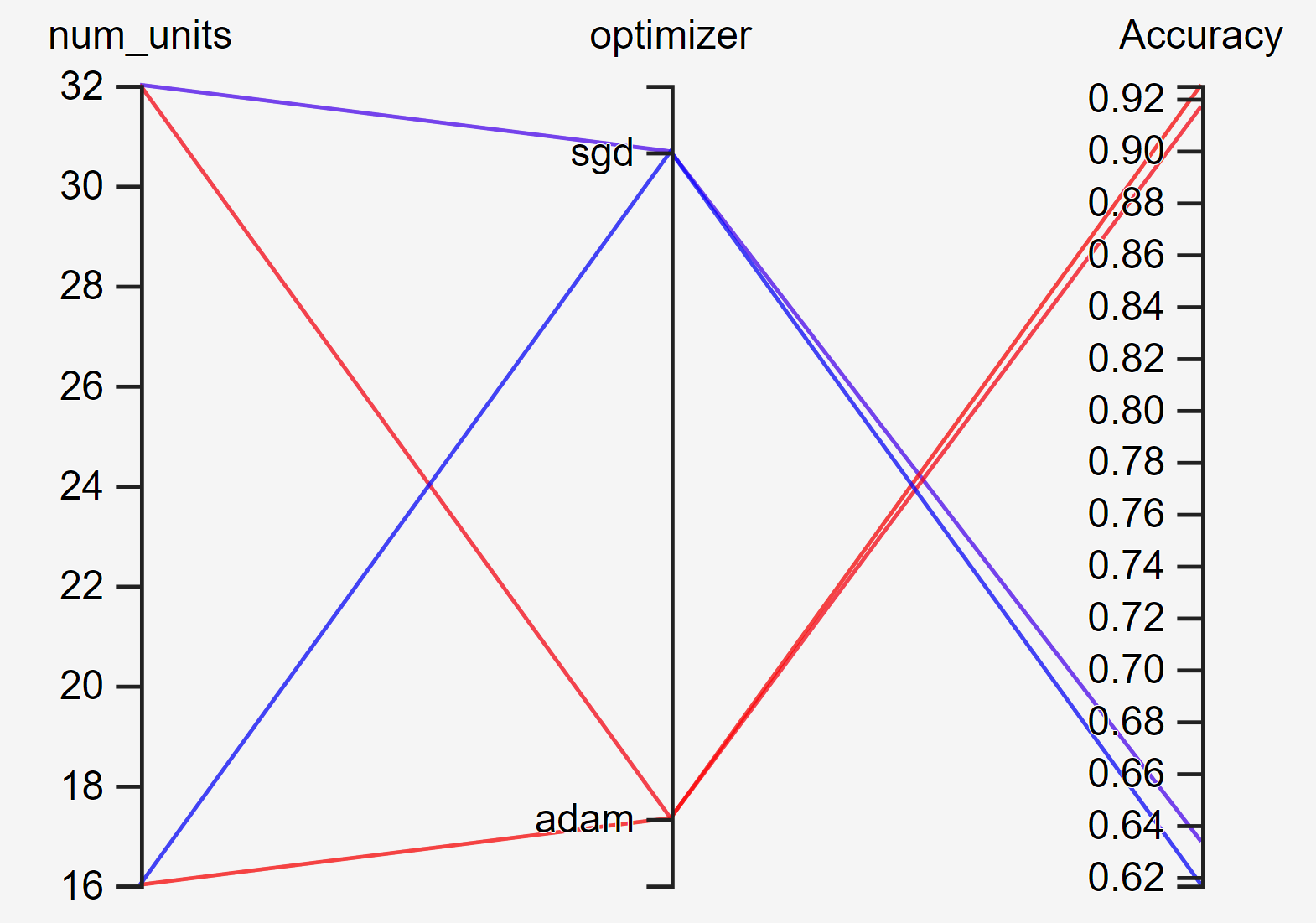
Data was split using standard split of 80-20 for training and validation.

**Data Analysis:**

To start the data analysis, I created a TensorFlow Sequence to tune the hyperparameters of number of filter layers for the 2D convolution, and the optimizer the TensorFlow model used. For the number of filter layers, I ran with 16 and 32 filter layers for each convolution layer. Each Convolution layer after the initial doubled the filter layers from the previous convolution layer. For the optimizer, I tested SGD and Adam.

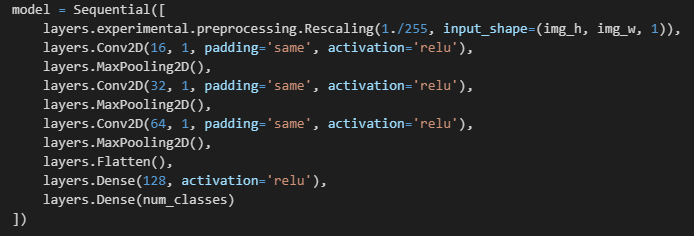


I scored the hyperparameter tuning based on the validation accuracy score. Each test case ran for 10 epochs.

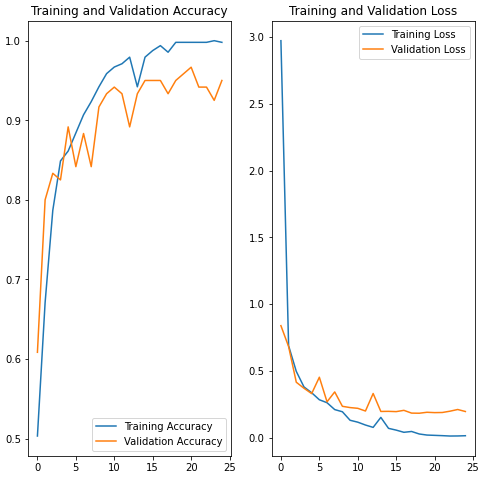


The Adam optimizer with 16 layers performed the best. Strangely enough 32 layers performed slightly worse, but it also doubled the training time. For the final model, I chose to use the 16 starting layers with Adam optimizer since it had faster training times.

From there, I rebuilt the model using the hyperparameters from above.



After 25 Epochs, the training was essentially perfect, but the validation accuracy was a little lower.



I was a little afraid that the model was overfitting, so I added a dropout layer to see if it improves the validation accuracy. However, after running another 25 epochs, it appears that a dropout layer didn’t improve the accuracy significantly.



Using either model with or without dropout, the model performed extremely well with training data being near perfect, and the validation accuracy being in the low to mid 90s.