



# **Phase 4 Project**

**Author:** Freddy Abrahamson

Date created: 7-1-2022
Discipline: Data Science

## **Overview**

For this project, I will be comparing different neural network models. I will start with a ANN as a baseline model, and then create CNN(s), to see which is the most successful. We have a set of x-rays from both healthy patients, and ones with pneumonia. The goal for the model is to have the highest recall score. Effectively being able to correctly identify as many patients that have pneumonia as possible.

### **Business Problem**

Stakeholder: Board of directors of a national network of hospitals.

**Business Problem:** Covid has caused a surge in emergency room visits. The hospital is looking for a way to better prioritize paients by the severity of their ailments, particularly pulmonary diseases.

**Proposed Solution:** A machine learning model that could distinguish between the xray image of a healthy patient, and of one with pneumonia, thereby helping prioritize who the doctor will see first.

#### **Solution Benefits:**

- 1. Helps save lives, and protect from more severe damage caused by the disease.
- 2. non-invasive
- 3. cost-seffective
- 4. no medical background necesary to run the model

## **Data Understanding**

The data was taken from Kaggle.com. There are a total of 5856 images. This includes 1583 'normal' images, and 4273 'pneumonia' images. The ratio of 'pneumonia' images to 'normal' images is about 2.7: 1. I divided all these images between train, test, and val folders at a ratio of .8:.1:.1 respectively. I maintained the 2.7 to 1 ratio between the 'pneumonia' images and 'normal' images, for all the folders. Once how many of each image would go to each folder was established, all the 'normal', and 'pneumonia' images were chosen randomly. The primary concern with the dataset preparation would be to normalize the image values. All the values were scaled to a range between 0 and 1.

### Modeling

I used Keras and Tensorflow to create the models. Given that with the use of the filters, cnn(s) excel at detecting features of various sizes, I chose to use the less apt multi-layer perceptron as a baseline model. I then tried to overfit on purpose using a cnn. I began with a cnn model that has 4 activation layers for the feature extraction part, with the number of nodes for each each layer being 16,32,64, and 128 respectively. I used ReLu as my activation function for all feature detection, as well as for the classification layers. Given that this is a binary classification problem (0 for normal, and 1 for pneumonia), I used a sigmoid function for the output layer. From there, based on the results, I would either try to reduce the bias, by adding a layer, adding more nodes to existing layers, or both; or reduce the variance by increasing the filter size to improve generalizability, or add dropout layers.

### **Evaluation**

Given the importance of correctly identifying a patient with pneumonia, my primary goal was to find a model that produced the best recall scores. To this end, I was looking for a model that would produce the best bias/variance combination between the train and test data sets. I did this by creating a function best\_model(), which utilizes the auc() function from sklearn.metrics. The x-axis is represented by the absolute difference between the train and test scores, while the y-axis is represented by the test scores. The higher the test score, and the lower the train-test difference, the greater the area under the curve. The function returns a dataframe with the models, and their respective test scores, sorted by their auc. The model with the highest auc is the best. The secondary goal was a model that would have a good accuracy score, which the 'best' model in fact does, with a score over 90%.

## **Importing and Organizing Data**

I will create two new folders: 'all\_normal' and 'all\_pneumonia', and copy all the corresponding images from the 'chest\_xray' folder to these folders. I will then create a folder called 'train\_test\_val', with three folders inside of it:'train', 'test', and 'val'. Each of these three folders will have a 'normal', and a 'pneumonia' folder. I will randomly copy the images from the 'all\_normal', and 'all\_pneumonia' folders, to these three folders, with a split of 80%,10%,10% respectively, keeping the ratio between the number of 'normal' and 'pneumonia' images uniform across all three folders (stratified). The ratio of pneumonia images to normal images is 2.7:1.

The total number of validation images is: 586

The total number of normal validation images is: 158

The total number of pneumonia validation images is: 428

The total number of test images is: 586

The total number of normal test images is: 158

The total number of pneumonia test images is: 428

The total number of train images is: 4684

The total number of normal train images is: 1267

The total number of pneumonia train images is: 3417

# Viewing an image

View an example of an x-ray of a healthy patient, and one with pneumonia.

normal image



pneumonia image



## **Neural Network Models**

Six models were built all together. One ANN as a baseline model, and another five CNN(s). Below is a summary of the results. The datframe pictured is sorted by auc score. The model with the highest auc score is the model with the best bias-variance balance, and therefore the best model. In this case the best model is cnn\_model\_5, with a test score of 0.943038, and a train-test difference of 0.023813.

	model name	train score	test score	train-test diff	auc score
5	cnn_model_5	0.966851	0.943038	0.023813	0.959613
4	cnn_model_4	0.960537	0.930380	0.030157	0.950111
3	cnn_model_3	0.946330	0.917722	0.028608	0.944557
0	baseline_model	0.931334	0.905063	0.026271	0.939396
2	cnn_model_2	0.937648	0.898734	0.038914	0.929910
1	cnn_model_1	0.937648	0.879747	0.057901	0.910923

# **Best Model Classification Report and Confusion Matrix:**

## **Best Model Classification Report:**

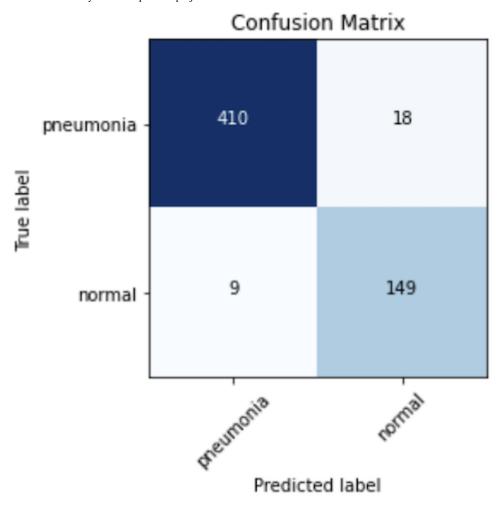
The table below is the classification report for the 'Best Model' (cnn\_model\_5), based on predictions of the test data set. Some take-aways from the report:

- 1. My primary concern was with the ability of the model to correctly identify patients with pneumonia, followed by the accuracy score. Row 2, column 2 of the report confirms my initial evaluation with a recall of 94%, while still maintaining an accuracy of 95%. Although 100% is always the goal, I would consider these both great scores.
- 2. In addition to this, we can see that both the precision, and f1-score metrics have very good scores of 89%, and 92% respectively.

	precision	recall	f1-score	support
0.0 1.0	0.98 0.89	0.96 0.94	0.97 0.92	428 158
1.0	0.09	0.94	0.92	130
accuracy			0.95	586
macro avg	0.94	0.95	0.94	586
weighted avg	0.96	0.95	0.95	586

### **Best Model Confusion Matrix:**

I created a confusion matrix plot for the presentation, as well as an easy way to calculate the accuracy score. With 559 Out of 586 images correctly classified, this model has an accuracy score of over 95%.



# **Project Conclusion: Possible Further Steps**

- 1. Request funding for a larger dataset to further calibrate the model
- 2. Once the model is ready, we can implement it in a subset of emergency rooms, use the feedback to make more changes if necessary, and then expand its use from there.

### For More Information

Please review my full analysis in my Jupyter Notebook or mypresentation.

For any additional questions, please contact Freddy Abrahamson at fred0421@hotmail.com,

## **Repository Structure**

├── README.md
├── student.ipynb
notebook
├── Phase\_4\_Project\_Presentation.pdf
└── images

- <- The top-level README for reviewers of this project
- <- Narrative documentation of analysis in Jupyter
- <- PDF version of project presentation
- <- Images used for this project

#### Releases

No releases published

Create a new release

#### **Packages**

No packages published

Publish your first package

#### Languages

Jupyter Notebook 100.0%