Pneumonia Classification Through X-Ray images

Introduction

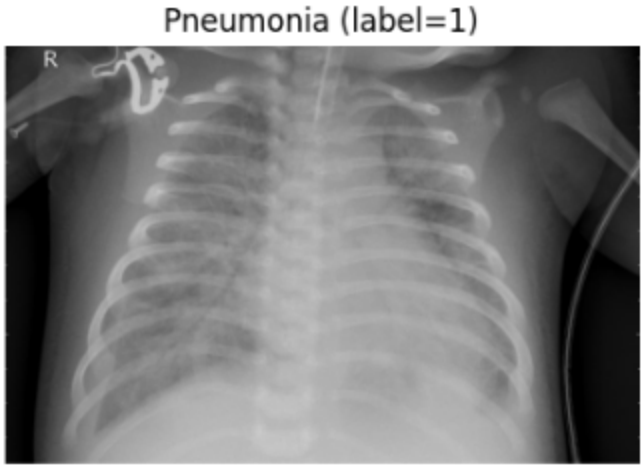
Machine learning in healthcare is a very important growing field that requires incredibly accurate models. An inaccurate model can have detrimental effects on people's health. Machine learning has been used often to classify different results from X-Rays. The task I decided to attempt to complete is classifying pneumonia from X-Ray images.

The first models I made were random forests with varying amounts of trees. I selected 100 trees, then 500 trees, and finally 1000 trees. I tested which model would be best and was able to get accuracies in the 90% range.

The other model I chose was the model I was most interested in seeing results. I created a convolutional neural network and was able to get an accuracy of 86%. Convolutional neural networks have been used very often in the past for this task with ResNet and DenseNet being top choices with fantastic results used in the actual medical field. The model I created was still leaning heavily towards predicting pneumonia and would be dangerous to actually push to production in its current state so nobody gets prescribed anything without the actual illness.

X-Ray Image Dataset

The dataset I used for this project was a dataset containing images of X-Rays and a binary label attached to label each image if pneumonia is present (1) or not (0). There were around 6000 rows of data. Looking at the images below to the untrained eye it is almost impossible to tell them apart. My goal was not only to be able to create a model capable of yielding results similar to any normal person but to try to get results similar or better than radiologists



The data was around 73% pneumonia and 17% normal so balancing the dataset was very important before creating my models. Stratifying on the binary label for balance between testing and training sets and then setting class weights to avoid my model always predicting the dominant class. The data was split in a 75/25 train test split.

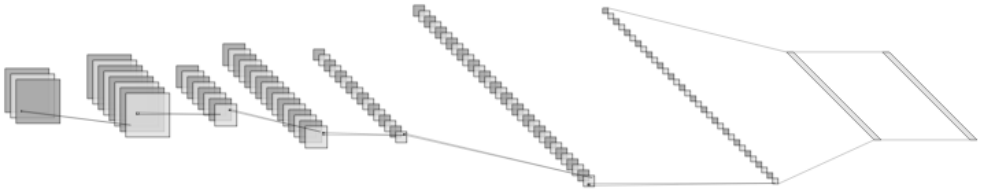
This dataset was sourced from hugging face. It was a pre-split dataset so in order to complete the project from start to finish I concatenated the pre-split sets and converted them to one pandas dataframe. Once the data was in one dataframe I was able to continue the project from start to finish and generate my own splits for each model created.

Random Forest

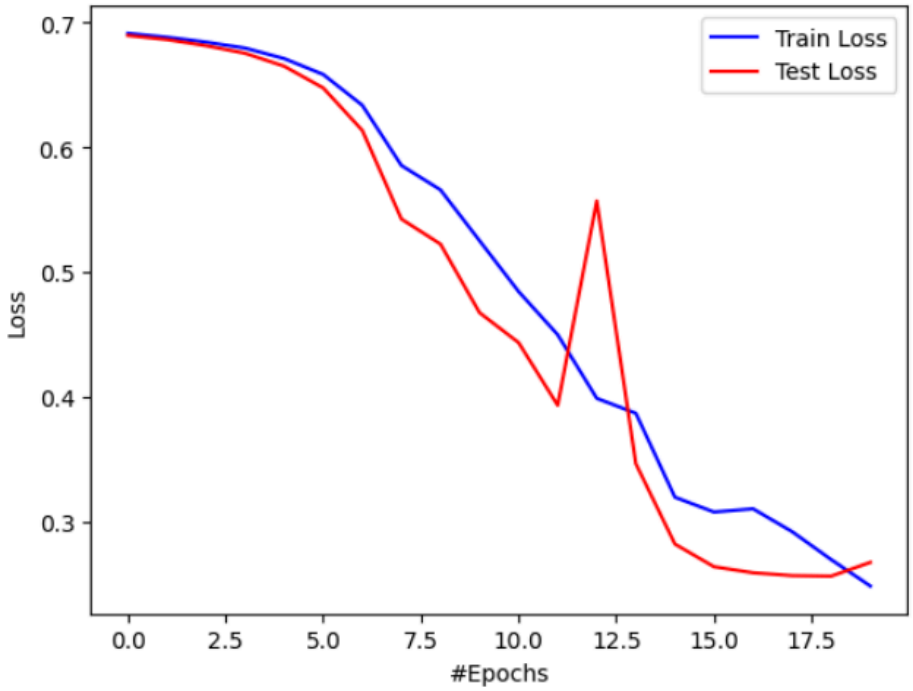
I began by creating a few random forests to quickly get a few accuracies on models that can be easily trained and quickly created. I used the train test split function and separated the data 75% into the testing set and 25% into the training set. I stratified on the binary label to balance the ratio of pneumonia to normal between each set. Then while making my models I set the class weight to balanced to even out the ratio of pneumonia to normal inside each set. I made three models, first with 100 trees, second with 500 trees, and finally with 1000 trees. I found that the model with the best results was surprisingly the 100 tree model. Even with these models being very simple to implement they were able to return very impressive results that I did not expect.

Convolutional Neural Network

The next method I used was a convolutional neural network. The network consisted of three convolutional layers followed by ReLu activation functions and max pooling. These 3 convolutional blocks are then followed by an output layer that predicts either pneumonia or normal. The network diagram is shown below.



I trained the network for 20 epochs using stochastic gradient descent with .01 for the learning rate. I also adjusted class weights to avoid always predicting the dominant class. The loss graph after training is displayed below.



Results

To record my results I looked at accuracy, precision, recall, and F1 score. For the random forests I used the classification report function to generate a report of results and then for the neural network I generated a confusion matrix and computed the values myself.

Random Forest

The first model I created was the 100 tree random forest. It ended up being a very accurate model. I was able to achieve an accuracy of 93%. Macro averages were .93 for precision, .89 for recall, .91 for F1 score. Displayed below is the full classification report for the 100 tree model. Surprisingly this was the best model that was created to solve this problem.

	precision	recall	f1-score	support
0	0.93	0.80	0.86	396
1	0.93	0.98	0.95	1068
accuracy			0.93	1464
macro avg	0.93	0.89	0.91	1464
weighted avg	0.93	0.93	0.93	1464

The second model I moved onto was a 500 tree model, I was testing a few different tree amounts to see how they would impact the results. This model achieved an accuracy of 92%. Macro averages were .93 for precision, .88 for recall, and .90 for F1 score. Dispalyed below is the full classification report for the 500 tree model. This was a slight dip in performance compared to the previous model.

	precision	recall	f1-score	support
0	0.93	0.78	0.85	396
1	0.92	0.98	0.95	1068
accuracy			0.92	1464
macro avg	0.93	0.88	0.90	1464
weighted avg	0.93	0.92	0.92	1464

The third model implemented was a 1000 tree model. This model achieved results exactly the same as the 500 tree model. Once again this was a slight dip in performance compared to the 100 tree model. Results showed an accuracy of 92%. Macro averages were .93 for precision, .88 for recall, and .90 for F1 score. Full results are shown below.

	precision	recall	f1-score	support
0	0.93	0.78	0.85	396
1	0.92	0.98	0.95	1068
accuracy			0.92	1464
macro avg	0.93	0.88	0.90	1464
weighted avg	0.92	0.92	0.92	1464

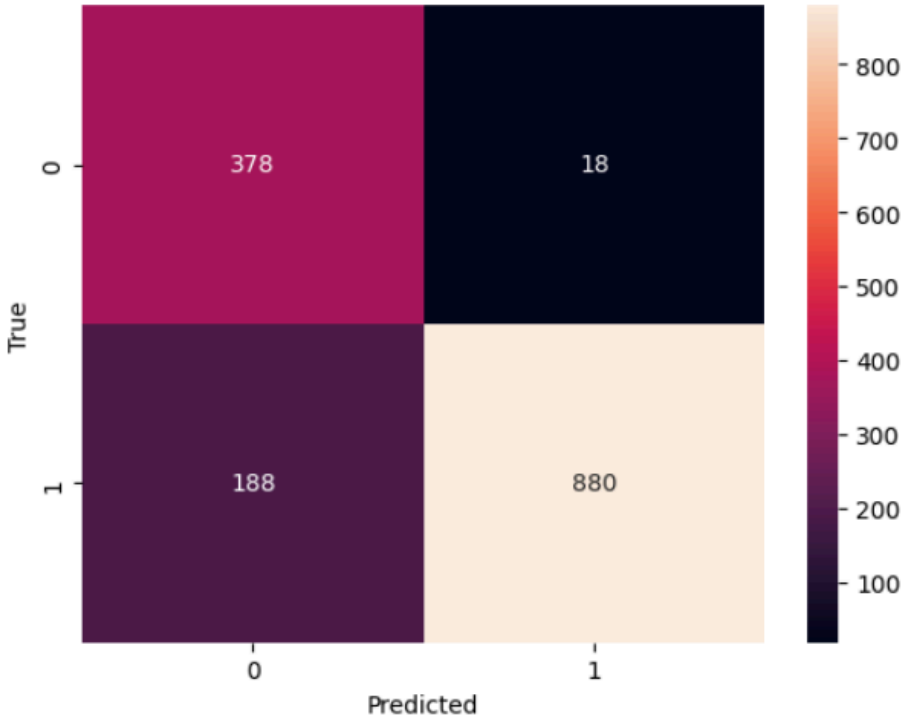
CNN

The last model I implemented was a convolutional neural network. As stated above the network consisted of three convolutional layers followed by ReLu activation functions and max pooling. This model initially had 2 layers trained for 5 epochs and had an accuracy of 73% only due to it only predicting pneumonia. I then added in the 3rd layer and increased the epochs to 20. This immediatly improved the accuracy to a number that seemed more correct. The accuracy was raised to around 86%.

```
Train Set: Accuracy: 3770/4392 (85.8%)

Test Set: Accuracy: 1258/1464 (85.9%)
```

I then created a confusion matrix and calculated the precision, recall, and F1 score. This model had .98 for precision, .82 for recall, and .89 for F1 score. The confusion matrix is shown below.



Discussion

In this project I used a few random forests and a convolutional neural network to predict if a patient had pneumonia from X-Ray images. The random forests achieved higher accuracy than the convolutional neural network which in the future I would like to continue to tune the network to push its accuracy above the random forests. I believe the main reason the network had lower results than the random forests was mainly due to a time constraint while testing as the runtimes were around an hour in its final form. I wouldnt be confident using the convolutional neural network in its current state in a real scenario due to its tendency to choose pneumonia as people may get perscribed unnecessary medications due to false classification. The random forests on the other hand may be able to yield successful results in a real scenario. I believe with extra tuning the neural network could become a fantastic choice and achieve better results than the random forest and become a great choice to implement for use on this task as it is already commonly used in healthcare for this task and other similar problems as well.