Deep Learning Final Project Status

BMEN 4000

Alex Kim & Kevin Mao

Determining Goals

For this project, we initially planned to use accuracy as the preferred performance metric to gauge the level of desired performance. However, this has its shortcomings in medical applications, especially in a potential diagnostic tool for pneumonia. It is worse to classify affected lungs as normal (false negatives) compared to if the model classifies normal lungs as affected (false positives). In order to combat potentially dangerous misdiagnoses, precision and recall performance metrics can be used to give false negatives higher costs than false positives.

Precision measures the fraction of true diagnoses amongst positive classifications, while recall measures the fraction of true diagnoses amongst people with pneumonia. Both scores will be plotted on a PR curve. We plan on reporting the total area underneath the PR curve or calculating a comprehensive F-score that accounts for both precision and recall metrics.

Working End to End Pipeline for Baseline Model

A convolutional neural network with rectified linear units was used for analyzing images. An Adam optimizer was used with a learning rate of 0.01. Two layers of 1024, 512 units and a softmax pool layer of 3 units (classes) are added to the top of a Xception classifier pre-trained on Google's ImageNet. More learning rates and different numbers of hidden layers (4, 8) will be tested for improved performance.

The inputs are x-ray grayscale (r=1) images approximately 2000 x 2000 pixels large. The Kaggle files provided split the data into three separate folders—test, train, and validation. In each folder were three folders named corresponding to one of normal, bacterial pneumonia, and viral pneumonia.

Label analysis showed that the number of images in each folder were inherently skewed. There were 0 viral pneumonia images and only 17 images total in the validation data set, while the testing data set had 234, 242, and 148 images and the training data had 1342, 2530, 1345 images for normal, bacterial, and viral cases, respectively. In order to normalize the number of images, two measures were taken. First, validation and test data were combined, randomized, and split evenly between validation and test data sets. Secondly, a random selection of the minimum number of images for each label type was selected. The table below shows the data before and after normalization.

	Before normalizing	After normalizing

	Normal	*Bacterial*	*Viral*	*Normal*	*Bacterial*	*Viral*
Train	1342	2530	1345	1342	1342	1342
Validate	9	8	0	74	74	74
Test	234	242	148	74	74	74

Table 1: Number of images for train, validate, and test sets per label, before and after pre-processing.

Proposed Incremental Approach

Future steps involve iteratively tuning hyperparameters (training step, number of hidden layers) to maximize our F-score and generating a supporting confusion matrix. The first iteration of the model prioritized accuracy, which is not ideal for diagnostical applications. Even when we prioritized accuracy, we only obtained an accuracy of about 30% on the test data which suggests that the problem was due to underfitting/overfitting or a problem with the training data. Taking a look at specific images that were negatively classified may provide some insight into how we can normalize the images even further. We will also explore the benefit of adding a dropout and/or batch normalization layer.