CS230

A Screening Tool for Abnormalities in The Chest and Thoracic Region Using Chest X-Ray Images and Convolutional Neural Networks

Jad Faraj
faraj2@stanford.edu

Rishabh Sirdesai
rishabhs@stanford.edu

Abstract

In large developing countries such as India and China, there exists an acute problem stemming from the vast disparity between the population and the number of radiologists which puts a strain on radiologists and potentially decreases human performance. For this project, we wanted to develop a chest x-ray abnormality classifier using a variety of Convolutional Neural Network models. Network depth and connectivity were varied and performance was evaluated using F1 score. The best model was found to be the ResNet. It seems that deep neural networks are more desirable for solving this multi-object classification problem.

1 Introduction

Excessive number of patients and doctor burn out have been shown to increase medical error in hospital environments, and so excessive workload may also lead to faulty diagnosis and false negative results (i.e. giving a patient an all clear when there exists a pathophysiological condition or traces of a disease in the patient) which can prove to be fatal to the patient.
The purpose of this project is to alleviate pressure off of physicians, by building a deep learning model which can take as an input, frontal and lateral x-rays of a human chest, and output the presence of abnormalities to a reasonable degree of error. One thing to be noted is that the Convolutional Neural Network will function exclusively as a screening tool and not as a diagnosis tool. Specifically, the role of the CNN will be to screen the x-ray images and identify the images that are free of any abnormalities or pathophysiological conditions. The images which do contain abnormalities or even contain uncertainties about abnormalities are subsequently sent to a radiologist or team of radiologists for further consultation.

2 Related work

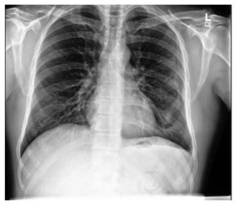
ChexNet(1) is a 121-layer convolutional neural network developed to predict presence of pneumonia to a high degree of accuracy. One of the problems that was faced when developing the model for this project was choosing a reasonable image size, where the model could be trained relatively quickly, but features could still be visually discerned. ChexNet used 224 X 224 pixel images and given that the ChexNet performed better than physicians when evaluating presence of pneumonia, we also decided to resize our images to 224 X 224 pixels.

3 Dataset and Features

The dataset used is called CheXpert (2) and is a large dataset of chest X-rays which features uncertainty labels and radiologist-labeled reference standard evaluation sets. CheXpert is a large public dataset for chest radiograph interpretation, consisting of 224,316 chest radiographs of 65,240 patients. The images were retrospectively collected from chest radiograph examinations at Stanford Hospital, performed between October 2002 and July 2017 in both inpatient and outpatient centers, along with their associated radiology reports. This dataset contains 4 different types of labels for 13 documented chest abnormalities. The labels are as follows:

- Blank for unmentioned
- 0 for Negative
- 1 for positive
- -1 for Uncertain

For our purposes, the uncertain labels were modified to be positive labels, because false positives are preferable to false negatives, and further consultation will confirm or deny the presence of any abnormalities. Also, the blank labels were presumed to be negative, and so all Nan labels were modified to be 0.
The data set is divided into Train and Valid segments with the train set consisting of 220,000 images and the valid set consisting of 350 images. Multiple images may be present for the same patient, since there are two types of images, frontal views and lateral views, shown in Figure 1 below.


(a) Frontal Chest X-Ray view

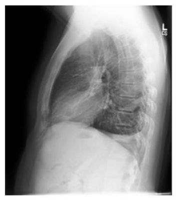

(b) Lateral Chest X-Ray view

Figure 1: Examples of input image data

Each image in the data set is cropped to some resolution, so all the images needed to be re-sized before training. All images were cropped to a size of 224 by 224 pixels, and the data set was re-segmented into a training set of size **216,000** and a dev set of size **4000**.

4 Methods

Initially, a baseline model was implemented, and performance was evaluated. From now it will be referred to as CustomNet, and the model was implemented in PyTorch(3) as a 4-layer convolutional neural network, with each hidden layer consisting of a Convolutional step, a ReLU step, and a Max pooling step. The input was a $(224 * 224)$ matrix with one color channel, and $5x5$ filters were used at each Convolutional layer. The first, second, and third convolutional steps used 32, 64, and 128 filters, respectively. SAME padding was used to ensure there was no visual information lost in the process, and the max pooling layers had a stride and padding of 2, effectively halving the size of input matrices. The loss function that is being minimized is shown in Equation 1 below.

$$L\left(w\right) = -\frac{1}{N}\sum\left[y_n log(\hat{y_n}) + (1-y_n)log(1-y_n)\right] \tag{1}$$

After observing that the learning rate was too high, it was decided that a DenseNet(4) model should be implemented. A DenseNet is a network where each layer is connected to each other layer, thus making connections between distant layers shorter. A visualization is shown in Figure 2. The model

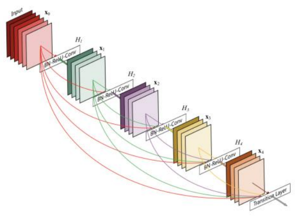
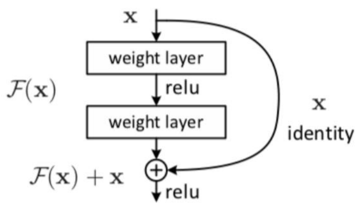


Figure 2: 5-layer dense block (4)

was acquired from GitHub and results can be seen in Section 5. To further compare performance results, a ResNet(5) model was also implemented and trained on the input data. ResNet models use residual blocks, shown in Figure 3 below, to speed up learning when multiple passes are taken through very deep neural networks.


x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu
Figure 3: Residual layer block (5)

5 Experiments/Results/Discussion

For the project milestone, we submitted a Tensorflow model which was a Convolutional Neural Network consisting of a convolutional Layer followed by a ReLU and a Maxpool and another convolutional Layer followed by a ReLU and Maxpool and finally a fully connected layer with 14 elements for the the 14 conditions to be classified. One of the primary findings for the pre-baseline model was the fluctuation in the cost function with every epoch as shown in Figure 4.

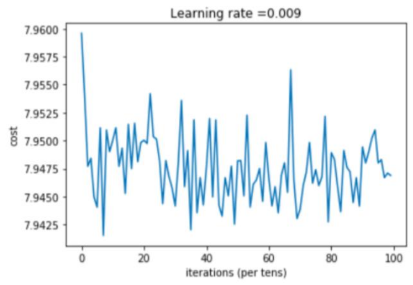


Figure 4: Cost function plotted over 100 iterations

The first step to fix this fluctuation was to decrease the learning rate from 0.009 to 9e-06. This made an immediate change in the cost function over all the iterations and we saw a decrease in the cost over the whole training set. Figure 5 is the plot of the cost function over 100 iterations with learning rate as 9e-06.

The cost function for the baseline model exhibited predicted behaviour and the cost decreased of the whole training set as seen in figure 6.

Similarly, for the DenseNet Model, Cost function also exhibited predicted behavior and showed slight fluctuation within mini-batches but decreased overall over epochs as can be seen in figure 6.

The Cost function of the ResNet Model however fluctuated over the entire training set and the suspicion is that the learning rate could be reduced further to get expected results as shown in 7

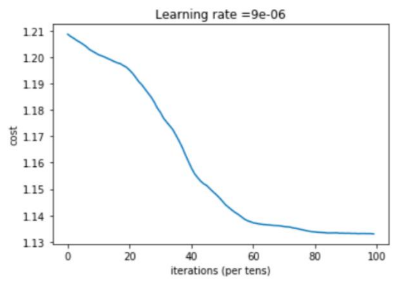
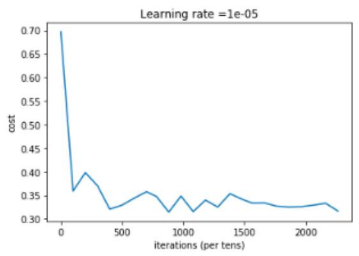
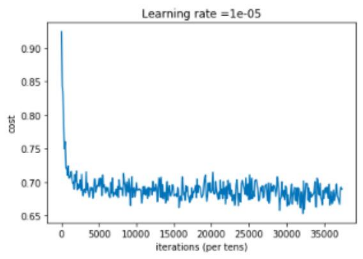


Figure 5: Cost Function after reduced learning rate



(a) CustomNet Cost Function Plot



(b) Cost Function Plot for DenseNet model

Figure 6: Cost Function plots for CustomNet and DenseNet Models

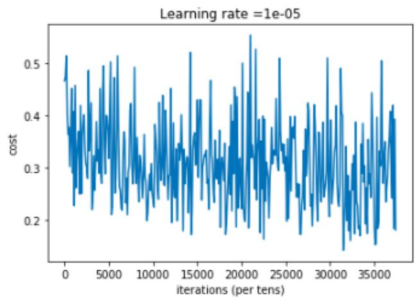


Figure 7: Cost function fluctuated in the ResNet Model

The Metric chosen to evaluate these models was F1 score and the ResNet model exhibited the best performance over the training set. The baseline CustomNet model exhibited a better F1 Score on the test set as compared to the training set. Refer to table below for the F1 scores on the train and test set.

Model	Train	Test
CustomNet	0.3541	0.3745
DenseNet	0.2679	0.2521
ResNet	0.4836	0.4690

6 Conclusion/Future Work

From the three models that were implemented, the ResNet exhibited the best F1 score which was the metric chosen to evaluate the models. Immediate next steps for this model would be to address the issue of the fluctuating cost function by trying out a lower learning rate. Apart from this, future work would be to conduct error analysis on the data set to see which examples were classified incorrectly by the models and also to implement a heat map function such that the area of disease can be localised on the image. Another aspect that can be explored is by training the model on exclusively frontal X-Ray images and testing it on lateral images to identify similar condition but from a different angle.

References

[1] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng, “Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning,” 2017.

[2] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng, “Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison,” vol. 1, 2019.

[3] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.

[4] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” 2016.

[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.