Chest X-Ray (CXR) Disease Diagnosis with DenseNet

Doug Beatty, Filip Juristovski, Rushi Desai, Mohamed Abdelrazik
Georgia Institute of Technology, Atlanta, Georgia

Abstract

Chest X-ray imaging[1] is a crucial medical technology used by physicians to diagnose disease and monitor treatment outcomes. Training a human radiologist is a lengthy and costly process. Deep learning techniques combined with availability of larger data sets increases the feasibility of building automated models with performance approaching human radiologists.

We present a scalable deep learning model trained on the ChestXray14[7] data set of X-ray images to detect and correctly classify presence of pneumonia. Once a good baseline has been established, the model will be expanded to classify 14 different diseases.

Introduction

A chest radiograph[1], or a chest X-ray (CXR) is one of the oldest and most common forms of medical imaging. A human radiologist requires significant training time and cost to be able to perform a comprehensive chest X-ray analysis with minimal error. Several types of abnormalities can arise in a chest radiograph that helps lead to detection and diagnosis of a multitude of diseases. With the vast number of different abnormalities and the overlapping reasons that might cause them, human error becomes a major contribution to poor diagnosis.

The revolution of machine learning and deep learning techniques combined with the availability of larger data sets[2] and big data processing systems[3] makes the analysis of X-ray images increasingly more realistic and the creation of automated models more feasible. The objective of this project is to train an efficient and scalable deep learning model which can learn from a data set of X-ray images to detect and correctly classify 14 different diseases. Automating the X-ray analysis makes the overall diagnosing process faster and less error-prone which significantly improves a patients treatment procedure.

Approach

Our approach consists of 5 high-level activities:

1. Data acquisition
2. Image preprocessing - Apache Spark
3. Training DenseNet-121 deep learning model - Keras
4. Model validation and fine tuning
5. Model evaluation

The details of each of these activities are covered in subsequent sections.

Data acquisition

Two different datasets were acquired that contained chest radiographs. The first is ChestXray14 from the NIH, the current results of this paper utilize the ChestXray14 dataset. The second, is CheXpert, which provides a substantial improvement to ChestXray14 with more training images and labels, CheXpert will be used later to improve the model and results.

The full ChestXray14 dataset consists of 112,120 chest radiographs of 30,805 patients. We are using this dataset for our initial training and results.

The CheXpert dataset was acquired upon registration and acceptance of the Stanford University School of Medicine CheXpert Dataset Research Use Agreement terms and conditions.[2]

CheXpert consists of 224,316 chest radiographs of 65,240 patients. Each imaging study can pertain to one or more images, but most often are associated with two images: a frontal view and a lateral view. Images are provided with 14 labels derived from a natural language processing tool applied to the corresponding free-text radiology reports.

Dataset pre-processing

CheXpert[2] has high resolution images which are not suitable as input to the model. Using a high resolution image significantly increases the number of input feature vectors which would require an increase in the model complexity and training time. Also, our strategy is to use pre-trained DenseNet with ImageNet weights, so we need to use the same number of input features. Data set images were preprocessed before training using Apache Spark which is a scalable big data processing technology. Several down-sampling techniques were used to reduce image size.

A convolution neural network (CNN) is said to have an invariance property when it is capable to robustly classify objects even if its placed in different orientations. To enrich the input data set and increase the number of available training samples, we performed data augmentation by generating several images with different orientations from a subset of input images.

Each input image is down sampled by resizing to 224 by 224 pixels. An input image generates one or more augmented versions of itself (e.g. by horizontal flipping). Each output image is assigned an ID of type Long, and inherits all the labels from the original input image. The model will utilize transfer learning by having a base DenseNet model trained on ImageNet, because of this, the images are normalized against the mean and standard deviation of the ImageNet training set.

Pre-processed images are saved to HDFS, which is a highly distributed and scalable Big data storage system. Due to HDFS implementation and API limitations, storing several tiny image files is not an efficient operation.

We decided to change the output format to be textual. Each image can be represented with the unsigned values of its byte stream (a vector of length $224 \cdot 224 = 50,176$). Each Spark RDD Partition will save its images as a space separated file with this format:

[Image ID] [bytes values]

where Image ID is the newly assigned ID, and the bytes are space separated vector of row based pixel values.

A corresponding CSV file for labels is generated which starts with the Image ID along with all the inherited labels from the original image.

This new output storage format resulted in significant decrease of the pre-processing job run time, and was suitable as a direct input to the model.

Method

Residual Networks (ResNets) allow us to train much deeper networks than a conventional CNN architecture since they handle the vanishing/exploding gradient problem much more effectively by allowing early layers to be directly connected to later ones. Dense Convolution Networks (DenseNets) are a form of residual network. Theoretically, it is expected that performance of models should increase as architecture grows deeper, and we should get monotonically decreasing performance. But in reality we don't see that since as the layers get deeper, the optimizer finds it increasingly difficult to train the network due to the vanishing/exploding gradient problem. ResNet allow us to match the expected theoretical issue. Figure 1 shows depth vs performance.

ResNets have significantly more parameters than conventional CNN networks. DenseNet retains all features of ResNet and goes further by eliminating some pitfalls of ResNet. DenseNets have much less parameters to train compared to ResNets (typically up to 3x less parameters). The base model we are using is DenseNet-121 with pretrained weights from ImageNet. For feature extraction purposes, we load a network that doesn't include the classification layers at the top.

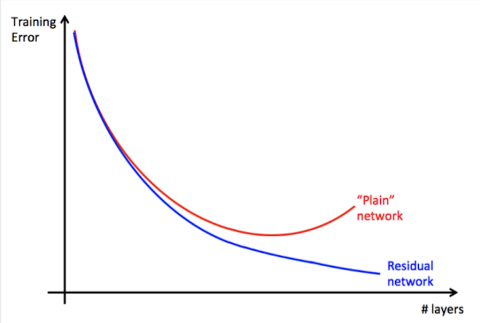


Figure 1: Model Accuracy

Pathology	Wang et al. (2017)	Yao et al. (2017)	CheXNet	Ours (SGD)	Ours (Adam)
Atelectasis	0.716	0.772	0.8094	0.8104	0.7985
Cardiomegaly	0.807	0.904	0.9248	0.8977	0.9055
Consolidation	0.708	0.788	0.7901	0.7961	0.7945
Edema	0.835	0.882	0.8878	0.8837	0.8849
Effusion	0.784	0.859	0.8638	0.8798	0.8792
Emphysema	0.815	0.829	0.9371	0.9143	0.8951
Fibrosis	0.769	0.767	0.8047	0.8284	0.8063
Hernia	0.767	0.914	0.9164	0.9097	0.8810
Infiltration	0.609	0.695	0.7345	0.6999	0.6979
Mass	0.706	0.792	0.8676	0.8214	0.8211
Nodule	0.671	0.717	0.7802	0.7506	0.7226
Pleural Thickening	0.708	0.765	0.8062	0.7713	0.7634
Pneumonia	0.633	0.713	0.7680	0.7678	0.7498
Pneumothorax	0.806	0.841	0.8887	0.8674	0.8533

Table 1: Our models perform near CheXNet and outperforms the previous best published results.

The DenseNet models trained on ImageNet have a depth of 121 (convolutional+pooling layer, 3 transition layers, and 1 classification layer and 4 dense block stages with 6, 12, 24, and 16 layers respectively.

Metrics and Experimental Results

Accuracy, loss, and AUC scores are the main metrics used to evaluate the performance of the model.

To guide our training and see if we are on track, we will use loss curves and model accuracy plots. This will help us diagnose if our model is over-fitting or under-fitting. From our initial training, we have the loss curve as shown in Figure 2.

The current loss curve indicates that we might have an issue with high variance. For this loss curve, we took a pre-trained DenseNet with ImageNet weights and added an additional 3 fully connected layers. We then froze all layers and trained only the additional layers. Once the additional layers were trained, we then unfroze a total of 9 layers including the newly added ones and went through training again. Layers near the end of the model become more specialized to their training set, while layers near the start of the model are more general. Due to this behavior, retraining a select few top layers allows them to learn characteristics of chest X-Rays, and will help improve overall model performance and accuracy.

The loss curve suggests trying approaches like procuring more data, training more layers in the architecture, and better data set analysis to check for class imbalances.

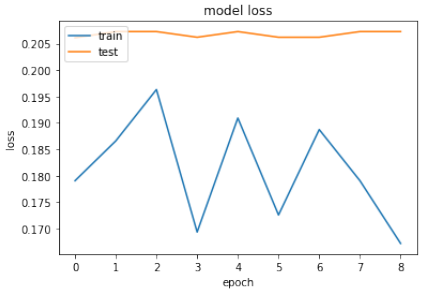


Figure 2: Model loss

Figure 3 shows the accuracy curve. At this point we aren't getting a monotonic trend in our accuracy curve so we will continue to investigate methods to improve the accuracy. We also see that like model loss, accuracy also is performing poorly on the test set as compared to the training set. So we will focus on getting both training loss lower and accuracy higher and bring train and test metrics as close as possible.

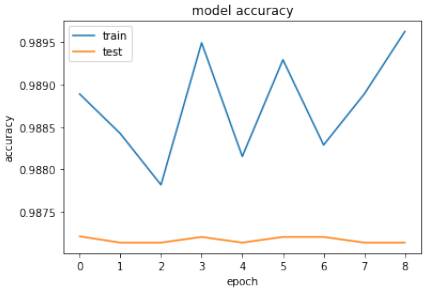


Figure 3: Model Accuracy

Discussion

Currently we are only focusing on improving performance on the pneumonia and infiltration disease classes and later we will expand to other diseases. This is to reduce complexity and to focus on producing a working model. Adding other diseases will mainly comprise of changing the classification layer and retraining on the dataset with all labels included. Currently we are using about 100K images for training. Using a sampled dataset allows us to iterate faster and focus on removing bugs in our model and training code.

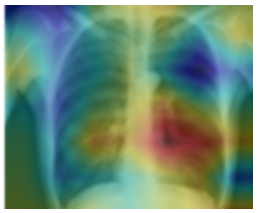
Our main theory for getting poor results is due to a class imbalance of cases of pneumonia (1430 cases out of a total of 112,120 images; only 1.2%). We hypothesize that re-balancing with a combination of data augmentation and sub-sampling the data set for non-pneumonia cases will improve results. Data augmentation methods include flipping images on horizontal and vertical axes, etc. We also used a phased approach to training where we kept most of the layers frozen during initial training and subsequently unfroze more and more layers. This helped in network troubleshooting during model development.

We have tried to resolve class imbalance issues by predicting for 'infiltration' which comprises for about 34 percent of dataset. We made the samples in both classes roughly 1:2 and tried training. Improvements in results are not satisfactory, but we believe with larger set we can get better balanced representation of the different disease classes.

Conclusion

Although the model currently has unsatisfactory performance characteristics, we feel we have a solid pipeline to start from and a strategy for improvement. Next we will re-balance the dataset and tune the model to get better performance. We also intend to create class activation maps (CAM) heatmaps for better visualization as seen in Figure 4.

(a) Source X-ray image



(b) Heatmap of pneumonia detection positive (85%)

Figure 4: CAM heatmap[5]

References

1. Siamak N. Nabili, M. (2019). Chest X-Ray Normal, Abnormal Views, and Interpretation. [online] eMedicine-Health.
2. CheXpert: A Large Dataset of Chest X-Rays and Competition for Automated Chest X-Ray Interpretation. [Internet]. Stanfordmlgroup.github.io. 2019.
3. FAN M, XU S. Massive medical image retrieval system based on Hadoop. Journal of Computer Applications. 2013;33(12):3345-3349.
4. Huang G, Liu Z, van der Maaten L, Weinberger K. Densely Connected Convolutional Networks [Internet]. arXiv.org. 2019.
5. Rajpurkar P, Irvin J, Zhu K, Yang B, Mehta H, Duan T et al. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning [Internet]. arXiv.org. 2019.
6. Liu H, Wang L, Nan Y, Jin F, Pu J. SDFN: Segmentation-based Deep Fusion Network for Thoracic Disease Classification in Chest X-ray Images [Internet]. arXiv.org. 2019.
7. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers R. ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. 2019.
8. Yao L. Weakly supervised medical diagnosis and localization from multiple resolutions [Internet]. Arxiv.org. 2019.
9. Guendel S, Grbic S, Georgescu B, Zhou K, Ritschl L, Meier A et al. Learning to recognize Abnormalities in Chest X-Rays with Location-Aware Dense Networks [Internet]. arXiv.org. 2019.
10. Raoof S, Feigin D, Sung A, Raoof S, Irugulpati L, Rosenow E. Interpretation of Plain Chest Roentgenogram. 2019.
11. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization [Internet]. Arxiv.org. 2019.
12. http://www.stat.harvard.edu/Faculty_Content/meng/JCGS01.pdf
13. Pryor TA, Gardner RM, Clayton RD, Warner HR. The HELP system. J Med Sys. 1983;7:87-101.
14. Gardner RM, Golubjatnikov OK, Laub RM, Jacobson JT, Evans RS. Computer-critiqued blood ordering using the HELP system. Comput Biomed Res 1990;23:514-28.