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 14 thoracic pathologies.

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 pathologies. 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 was initially planned on being used to further improve performance, but due to rising costs of training the model it was not pursued.

The full ChestXray14 dataset consists of 112,120 chest radiographs of 30,805 patients. The current model is trained off ChestXray14 and in some cases outperforms comparing models in performance for select pathologies.

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 increasing overall model complexity and training time. Another restriction to input image size is due to using a pretrained DenseNet model which was trained on ImageNet. Data set images were preprocessed before training using Apache Spark which is a scalable big data processing technology. The dataset was stored in Google Cloud Storage to provide a scale-able mechanism for handling the large data set. 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, horizontal flipping was applied randomly to images. This follows the DenseNet paper which found performance increases by adding horizontal flipping to the dataset.

Each input image is down sampled by resizing to 224x224 pixels. An input image generates one or more augmented versions of itself (e.g. by horizontal flipping). The DenseNet model utilizes transfer learning, and was originally trained on the ImageNet dataset, because of this the training data was normalized by the mean and standard deviation of the ImageNet dataset to normalize the distribution.

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.

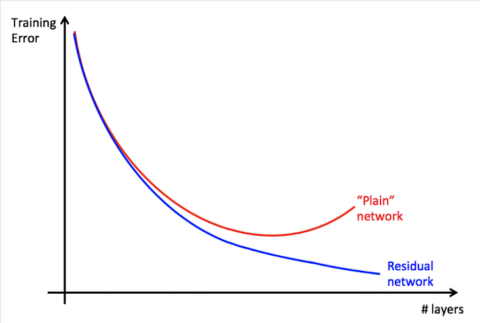


Figure 1: Model Accuracy

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 BC with pretrained weights from ImageNet. For feature extraction purposes, we load a network that doesn't include the classification layers at the top.

The DenseNet models trained on ImageNet have a depth of 121. The architecture consists of a convolutional+pooling

layer followed by 4 dense block stages. The dense block stages contain 6, 12, 24, and 16 units respectively, and each unit has 2 layers (composite layer with bottleneck layer). There is a single transition layer in between each dense block stage (for a total of 3 transition layers). Finally, there is 1 classification layer. 1 + 2(6) + 1 + 2(12) + 1 + 2(24) + 1 + 2(16) + 1 = 121.

The model utilized transfer learning due to a training size of only 112,120, compared to the millions which modern day models use. The base layers of a model are very generic to the data set, while later layers get more specific and tailored to their data set. This means that the last few layers may be retrained by unfreezing their weights and training on the ChestXray data set. These layers then learn representations of thoracic diseases in these top layers while retaining general information in the earlier layers.

Metrics and Experimental Results

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

Training and validation loss curves were used to provide insight into the model performance. An initial loss curve may be seen in in Figure 2. It may be seen that the model started over-fitting near the second epoch, because of this the training was stopped at only 8 epochs since further training would lead to no benefits. Based on the loss curves, adding more layers and increasing complexity would most likely lead to more over-fitting. Further investigation into better data pre-processing may help alleviate over-fitting, such as further data augmentation to increase sample size, better handling of class imbalances, or using more thorough data sets such as CheXpert. For class imbalances specifically, some of the classes such as pneumonia had extreme imbalances, 1430 cases out of a total of 112,120 images; only 1.2% of the overall data set.

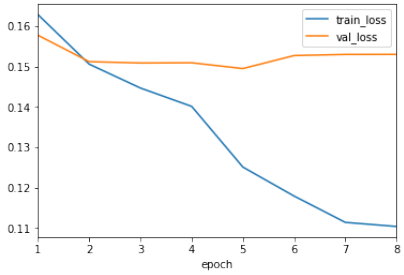
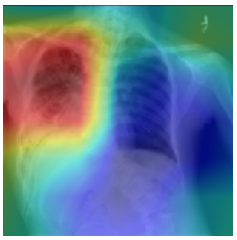


Figure 2: Model loss

The overall AUC score of the model compared to previous attempts may be viewed in Figure . Our variations of the model were able to get better performance in Atelectasis, Consolidation, Effusion, Fibrosis, and Pneumonia. Using stochastic gradient descent with momentum helped the model generalize better on a few of the diseases. Adam was able to provide good performance for most diseases as well, and further investigation into hyper parameter tuning may have provided better results.

(a) Original Patient X-Ray



(b) Infiltration of right lung highlighted

Figure 3: Patient X-Ray CAM Heatmap

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

Discussion

The initial goal of this paper was to reproduce the competitive results from the ChexNet paper, and then improve upon the performance by using a more substantial data set. Due to increasing costs of training the model on Amazon Sagemaker, the decision was made to not use CheXpert as originally planned.

Using only ChestXray14, better performance was achieved by using stochastic gradient descent (SGD) with momentum instead of the Adam optimizer. This was found to generalize better[15] and provide better results on the test set. Other improvements to the performance would be utilizing deeper versions of the DenseNet model such as DenseNet-169, and DenseNet-201. Utilizing these models in an ensemble pattern would also provide benefits to the overall performance.

Further plans to improve performance of the model were to integrate and train on the CheXpert dataset, this dataset has roughly twice the amount of images and also provides lateral chest X-rays, which have been found to account for 15% accuracy in diagnosis of select thoracic diseases[2]. The pre-processing spark code was developed to be agnostic to datasets and easily provide the necessary pre-processing on the data set.

Class Activation Maps (CAMS) were generated to visualize where the model was focusing to make its classification, an example may be seen in 3. In the specific example generated by our model, the model correctly predicted Infiltration as the diagnosis and highlighted the right lung region which lead to the diagnosis. This is a useful tool for verifying correct and incorrect model predictions and help further fine-tune the model.

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 3.

Contributions

We had full participation and collaboration from all group members. We met frequently on Google Hangouts to discuss strategies and progress, about a dozen meetings in all. We also collaborated via Slack (over 1000 messages).

Filip - modeling in Keras, prototype in Colaboratory, GitHub setup, Google Drive setup

Rushi - modeling in PyTorch and SageMaker, Class Activation Modeling (CAM), billing monitoring

Mohamed (Fawy) - modeling in PyTorch and SageMaker, Class Activation Modeling (CAM), pre-processing in Spark

Doug - EMR prototype, SageMaker Keras prototype, LaTeX formatting, presentation slides

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.
15. Wilson A, Roelofs R, Stern M, Srebro N, Recht B. The Marginal Value of Adaptive Gradient Methods in Machine Learning [Internet]. Arxiv.org. 2017.