CS230

Diagnosing Chest X-ray Diseases with Deep Learning

Jiawei Li
Energy Resources Engineering
Stanford University
jiaweili@stanford.edu

Zixi Xu
Mechanical Engineering
Stanford University
zixixu@stanford.edu

Yue Zhang
Mechanical Engineering
Stanford University
yzhang16@stanford.edu

Abstract

With the development of using deep learning techniques, automatic diagnosis of chest X-rays pictures can help thousands of patients to get a quick and accurate clinical assessment, especially for some places in lack of experienced doctors. In this project, we build three deep learning models (DenseNet-121, DenseNet-LSTM and DenseNet-GRU) to predict 14 thoracic diseases given a chest X-ray picture as the input. A modified loss function that better imitates the diagnostic process are used. The new loss function used on DenseNet-RNN models performs better than the simple multi-label cross-entropy loss. Both DenseNet-LSTM and DenseNet-GRU models help to increase the ROC-AUC scores of certain diseases. In particular, GRU as a decoder in DenseNet-RNN models performs better than LSTM in both loss functions.

1 Introduction

Detecting thoracic diseases in chest X-rays remains a challenging task that relies on the availability of expert radiologists. The diagnosis becomes even more complex when the patients develop complications (such as effusion and infiltration) or other pre-existing diseases. These situations are more common in young children, the elderly and those with pre-existing illnesses, such as diabetes [5].

Recently many exciting researches exhibit the ability of deep learning algorithms in helping diagnose the diseases, given a large number of high quality X-ray pictures. With the development of deep learning techniques, automated detection of diseases from chest X-rays would not only benefit clinical diagnosis, and also be invaluable in delivering health care to populations in lack of access to expert radiologists.

Inspired by the previous successes of CheXNet in pneumonia detection [6], we extend it to a CNN-RNN framework that inputs a chest X-ray image and outputs the diagnostic multi-labeling of all 14 different diseases. This idea is originated from a more generic framework for Multi-label Image Classification [8] that has been applied to web image database such as NUS-WIDE and Microsoft-COCO. Following this framework, we use the CNN as an encoder to process X-ray images and RNN as a decoder to detect all 14 diseases. Different from the previous research, our model takes the DenseNet-121 as the encoder which is connected by LSTM or GRU as the decoder.

2 Related work

A lot of investigations in automated diagnosis using deep learning techniques have been done in recent years. [9] released the ChestX-ray14 dataset which contains 112,120 frontal-view chest X-ray images individually labeled with up to 14 different thoracic diseases, and benchmarked different

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

CNN architectures that are pre-trained on ImageNet. Using this dataset, [10] exploited statistical dependencies between labels in order make more accurate predictions. However, the ROC-AUC scores of RNN model are lower than the pure CNN model given in [6], which applied the model DenseNet-121 to the ChestX-ray14 dataset to detect pneumonia using binary classification and transform it into 14 single-label classifications. The prediction exceeds average radiologist performance on the F1 metric. Similarly, [7] applied CNN-RNN models to OpenI, another open-source X-ray image dataset.

3 Dataset and Features

Our data set is from [9],there are 112,120 frontal-view chest X-ray PNG images in 1024 by 1024 resolution 1. We split our data set into 70% training set, 20% for dev set and 10% for the test set. We use the random flip to conduct data augmentation. We also transform our gray-scale image into 224 * 224 resolution images and replicate the resolution by 3 to get a RGB image as 224 by 224 by 3. This is the input data for our CNN model, denseNet121. We also read the meta information data for each image and generate a vector of dimension 14 to represent the exist of 14 disease as the training ground truth.

Figure 1: chest X-ray

4 Methods

Our base model is DenseNet-121, as devised by [3]. The main part of the model consists of 4 dense blocks. In each dense block, each layer l takes all the output features from every previous layer $1, \ldots, l-1$ as its input. This is called dense connection. The usage of dense connections encourages the reuse of features in the model. Also, due to shorter connections, layers can receive excessive supervision from the loss function. Similar to ResNet [2], DenseNet can also prevent the problem of gradient vanishing because of the addition of the shortcuts among layers. At the end of DenseNet-121, we add a fully-connected linear layer with output size 14 and a Sigmoid activation for each output.

To extend the base model, we treat DenseNet-121 as the feature encoder and add a bi-directional two-layer RNN model as a decoder to further exploit the correlations among the 14 diseases. We tried both Long-Short-Term-Memory (LSTM) and Gated Recurrent Units (GRU) as the RNN. The output size of the encoder is 50, and that of the decoder is 14. Different from [10] where a one-directional RNN is used, the bi-directional RNN decoder should capture the information of other diseases more accurately and completely.

Two loss functions are used separately throughout all three models:

$$L_1(X, y) = \sum_{i=1}^{14} \left(-y_i \log p(Y_i = 1|X) - (1 - y_i) \log p(Y_i = 0|X)\right)$$

$$L_2(X, y) = y_0 \sum_{i=1}^{14} (-y_i \log p(Y_i = 1|X) - (1 - y_i) \log p(Y_i = 0|X))$$
$$+ (-y_0 \log p(Y_0 = 1|X) - (1 - y_0) \log p(Y_0 = 0|X))$$

L_1 is a simple multi-class cross-entropy loss function. Y_i is defined as:

$$Y_i = \begin{cases} 0 & \text{if the i-th disease is not found} \\ 1 & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, 14$, and y_i is the corresponding model prediction. L_2 is a modified version of L_1 where Y_0 is defined as:

$$Y_0 = \begin{cases} 0 & \text{if no finding from the input figure} \\ 1 & \text{otherwise} \end{cases}$$

and y_0 is the corresponding model prediction. With the label Y_0, we expect the model to first reflect whether the figure has any disease or not. If there exist diseases, the model will further predict y_i. Otherwise, the loss function will not take Y_i into account. The reason of coming up with L_2 is because we want to make the model imitate the actual diagnostic process of radiologist in a better way. In the view of radiologist, if he or she decides that the given X-ray photo does not have any disease, the diagnosis simply comes to an end, just like the behavior of L_2.

We used the framework PyTorch to implement all the models.

5 Experiments/Results/Discussion

In this section, we analyze the results of the three models: DenseNet-121, DenseNet-LSTM and DenseNet-GRU, as discussed above. We use two loss functions, L_1 and L_2 for each model.

To tune the learning rate, we tested 10^{-3}, 5×10^{-4}, 10^{-4}, 5×10^{-5}, 10^{-5} and report the one giving the best dev loss and ROC-AUC value after 20 epochs. The batch size are all 32 to fit the GPU capacity. The weight decay is tuned by running on a subset of the total dataset for each case and 5×10^{-5} appears to give the best AUC-ROC value in all cases. We use Adam optimizer [4] for all the cases.

The first model is the pre-trained Densenet-121. The learning rate to achieve the best average ROC-AUC is 5×10^{-5}. It can be seen from Figure 2 that the training loss decreases almost exponentially with respect to epoch numbers. Also, the blue curve in Figure 2 shows that the AUC-ROC score increases as Epoch number increases. As seen in Table 1 column "L_1", the ROC-AUC score for each disease is close to the results in ([6]).

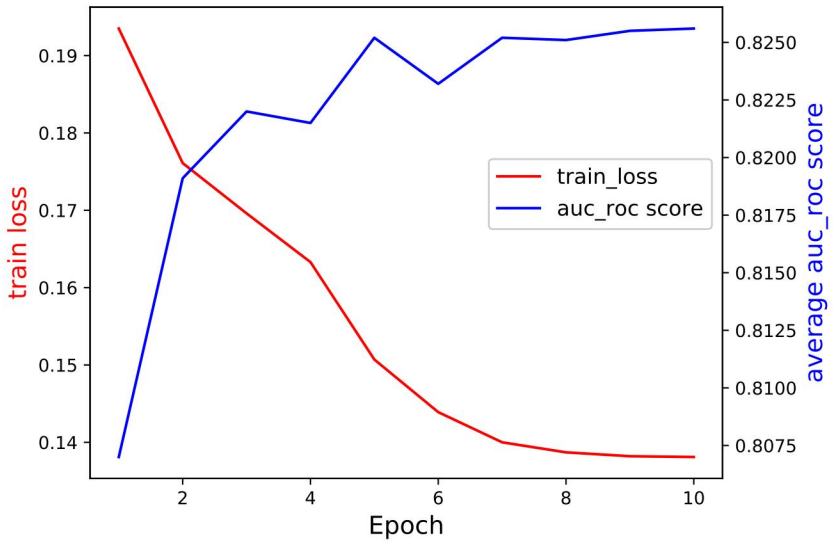

Figure 2: DenseNet-121 with $L_1(X, y)$

Table 1: DenseNet-121

Pathology	**Rajpurkar et al**	L_1	L_2
Atelectasis	0.809	**0.811**	0.781
Cardiomegaly	0.925	0.882	0.872
Effusion	0.863	**0.884**	**0.868**
Infiltration	0.734	0.714	0.647
Mass	0.867	0.846	0.836
Nodule	0.780	0.770	0.748
Pneumonia	0.768	0.745	0.727
Pneumothorax	0.888	**0.889**	0.873
Consolidation	0.790	**0.802**	**0.792**
Edema	0.887	**0.899**	**0.895**
Emphysema	0.937	0.915	0.912
Fibrosis	0.804	**0.812**	0.776
Pleural Thickening	0.806	**0.807**	0.792
Hernia	0.916	0.831	0.874
Average	0.841	0.829	0.814

Table 2: DenseNet-RNN

Pathology	**Yao et al.**	**LSTM-2**	**GRU-2**	**LSTM-1**	**GRU-1**
Atelectasis	0.772	0.768	0.770	0.769	0.776
Cardiomegaly	0.904	0.797	0.854	0.700	0.806
Effusion	0.859	**0.863**	**0.877**	0.857	0.860
Infiltration	0.695	0.557	0.617	0.552	0.691
Mass	0.792	**0.816**	**0.816**	0.633	0.878
Nodule	0.717	0.698	0.699	0.782	0.826
Pneumonia	0.713	0.640	0.667	0.717	0.751
Pneumothorax	0.841	**0.849**	**0.844**	0.689	0.727
Consolidation	0.788	0.777	0.785	0.796	0.860
Edema	0.882	0.861	0.878	0.730	0.780
Emphysema	0.829	**0.878**	**0.882**	0.700	0.840
Fibrosis	0.767	0.625	0.731	0.778	0.892
Pleural Thickening	0.765	0.693	0.741	0.699	0.763
Hernia	0.914	0.754	0.78	0.728	0.770
Average	0.798	0.760	0.782	0.723	0.801

Then we switch the loss function to L_2. The result is pretty good in comparison, though the average ROC-AUC is a bit lower. This is partly because we have no further information from the patients, such as their body temperatures and disease histories. We believe this loss function can work better when more information about patients is known.

Table 2 shows the results of DenseNet-LSTM and DenseNet-GRU. The learning rate is 10^{-4}. Both loss functions defined above are used. It can be observed that GRU performs better than LSTM in general. This is expected because there are less parameters in GRU which is less prone to the over-fitting. The results below are compared against [10]. Using L_2 works well in both cases, while GRU with L_1 gives a slightly better result. As discussed above, more information can help improve the performance of using L_2.

It can also be seen that bidirectional RNN models are able to capture the correlation between two diseases and help increase the AUC-ROC value. For example, of the four bold diseases in Table 2, the results of Mass and Emphysema have an obvious increase. In reality, lung Mass is often observed in patients who have Emphysema ([1]).

Additionally, ResNet-18 is also used as an alternative of DenseNet-121 under the encoder-decoder framework. We use ResNet-18 which has less parameters because the chest x-ray images are in gray-scale with a much simpler configuration. Our results show that the two models perform similarly, though Densenet gives a sightly higher ROC-AUC. The training time, however, can reduce by a factor of 3 when using Resnet18.

6 Conclusion/Future Work

In this project, we build three deep learning models: DenseNet-121, DenseNet-LSTM and DenseNet-GRU to predict 14 different diseases of a chest x-ray figure. Two loss functions, the simple multi-label cross-entropy and the other that better imitates the diagnostic process are used. The main conclusion is the following:

1. Using DenseNet-121, we get similar ROC-AUC results as [6].
2. The modified loss function L_2 examines the existence of disease before the disease classification, which better imitates the diagnostic process. In our DenseNet-RNN models, using L_2 does perform better than L_1. We expect it to perform even better when more information from patients like gender, age, body temperature, etc, are known.
3. Using bi-directional RNN helps to increase the ROC-AUC scores of Mass and Emphysema which have certain correlation that can be captured by sequence models.
4. In encoder-decoder framework, GRU as a decoder performs better than LSTM in both loss functions. GRU contains less parameters thus it is less prone to the over-fitting.

In the future, we will focus on extending the current RNN decoder to a more sophisticated attention model (that is add an one-directional RNN on top of the current bi-directional layers), so as to better capture the correlations among the diseases.

7 Contributions

All three members share the same amount of work. We all work in developing the code, iterating the models and writing reports. The codes can be found in this repo:

https://github.com/ZixiXu/CS230_CheXNet.

References

[1] H. Guenard. Lung density and lung mass in emphysema. *CHEST*, 102.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[3] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[5] NHS. NHS pneumonia. `https://www.nhs.uk/conditions/pneumonia/`.

[6] 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. *CoRR*, abs/1711.05225, 2017.

[7] H. Shin, K. Roberts, L. Lu, D. Demner-Fushman, J. Yao, and R. M. Summers. Learning to read chest x-rays: Recurrent neural cascade model for automated image annotation. *CoRR*, abs/1603.08486, 2016.

[8] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu. CNN-RNN: A unified framework for multi-label image classification. *CoRR*, abs/1604.04573, 2016.

[9] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *CoRR*, abs/1705.02315, 2017.

[10] L. Yao, E. Poblenz, D. Dagunts, B. Covington, D. Bernard, and K. Lyman. Learning to diagnose from scratch by exploiting dependencies among labels. *CoRR*, abs/1710.10501, 2017.