Binary Classifier on Chest X-Ray Image to Detect Pneumonia Using Convolutional Neural Network

Jiwei Zhang
jiwei.zhang@ucdconnect.ie
17200334

Abstract

Deep learning is a sub-field of machine learning, using artificial neural networks as the architecture to learn the characteristics of data. This technique has been widely used in different research fields and computer vision is one of the most remarkable ones Bengio et al. (2012). Computer vision is to teach the machine how to "watch" and to do further image processing. One of its branches is to do image classification. Given a set of images that are each marked as only one category, then the machine is supposed to predict the category of a new set of test images and measure the accuracy of the prediction. In this project, I have developed a Convolutional Neural Network (CNN) model with adjusted parameters to predict whether there is pneumonia based on the chest X-ray image Mooney (2018). And I evaluate the model by its ROC curve and AUC value.

1 Introduction

Pneumonia refers to the symptoms of inflammation in the lungs, which are mainly affected by the alveolus. It usually improves gradually three days after the start of treatment; however, the patient may feel tired for more than one month in the future McLuckie (2009). Basically, the earlier pneumonia is detected, the easier for the patients to be cured. Approximately 450 million people (7% of the global population) suffer from pneumonia every year, and about 4 million people die from it every year Ruuskanen et al. (2011). Therefore the purpose of this project is to design an image classifier to distinguish whether a lung is normal (healthy) or pneumonia exists based on the chest x-ray image.

The current mainstream computer vision method based on deep learning is similar in principle to the working principle of the human brain. Construct a multi-layer neural network, the lower layer recognizes the primary image features, several bottom layer features form the higher layer features, and finally, through the combination of multiple levels, the classification is finally made on the top layer. CNN can be used to gradually extract higher and higher levels of image content representation results. Also, it does not derive features such as texture and shape by preprocessing the data, but only uses the original pixel data of the image as input, and then "learns" how to extract these features, and finally infer the objects composed of these features. Therefore, CNN is widely used for image classification tasks.

In order to quickly and efficiently detect whether there are symptoms of pneumonia, I proposed a CNN model using based on Tensorflow to learn the characteristics of lungs with pneumonia in order to determine whether a new image of the lung has pneumonia or not. In the related work section, I will introduce some similar approaches from other researchers and describe my own experiment in section 3. Then I demonstrate and discuss my experiment's result and evaluation based on the ROC curve. Last but not least, the conclusion is given and some future work about this project is discussed.

Preprint. Under review.

2 Related Work

2.1 Convolutional Neural Network and Image Classification

Convolutional neural networks is commonly used models in deep learning. An application of the CNN technology was designed to the problem of identifying particle interactions in sampling calorimeters used commonly in high energy physics and high energy neutrino physics in particular Aurisano et al. (2016). Among different type of models, convolutional neural networks has been demonstrated high performance on image classification Guo et al. (2017). Also it has been shown that there are advancements in CNN from LeNet-5 to SENet model Sultana et al. (2018) and a novel deep convolutional neural network has been introduced, which is deeper and wider than other existing deep networks for hyperspectral image classification Lee and Kwon (2017).

2.2 Multi-label Image Classification with CNN

Not only with great performance on single-label image classification, but CNNs could also be suitable for multi-label image classification. Combined with CNNs, the proposed CNN-RNN framework learns a joint image-label embedding to characterize the semantic label dependency as well as the image-label relevance, and it can be trained end-to-end from scratch to integrate both information in a unified framework Wang et al. (2016).

2.3 Image Classification on Medical Field

A customized Convolutional Neural Network with a shallow convolution layer to classify lung image patches with interstitial lung disease (ILD) has been designed to automatically and efficiently learn the intrinsic image features from lung image patches that are most suitable for the classification purpose Li et al. (2014). After few years, a diagnostic tool is built based on a deep-learning framework for the screening of patients with common treatable blinding retinal diseases Kermany et al. (2018).

3 Experimental Setup

3.1 Dataset

When I get the dataset from Kaggle Mooney (2018), I checked its structure (1) and the distribution (2) for each set first. Obviously, the dataset is imbalanced, positive samples (Pneumonia) is around three times of negative sample (Normal). And the dataset is too small for an accurate deep learning model as there are only about 5800 samples in total including testing. Also, the validation is far too small to have only 16 samples inside.

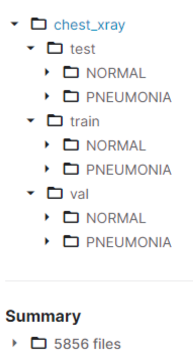

Figure 1: Dataset Structure

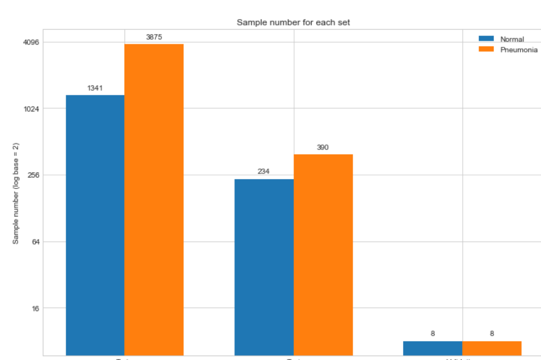

Figure 2: Dataset Distribution

3.2 Preprocessing

As there are some problems with these datasets, preprocessing the dataset is necessary for future learning and modeling. First of all, I have checked the size of each sample and it is too large, around

400KB for each. So I set the image size as 224×224, making the graph not too small to lose the characteristics but reducing the run-time. Then I decided to ignore the given validation set due to its size instead of using 20% of the training set for validation. I have tested several times before I made this decision with the original validation set, but it affected the performance of my model too much so I decided to ignore it. Moreover, I have made both the normalized version of data and the non-normalized version, testing whether the normalization is indispensable for training a CNN.

3.3 Model Building & Training

Because the task is to do a binary-class classifier, the best choice for the structure is CNN for sure as it performs the best to gather information with high precision. For the optimizer, I chose the "Adam" because Adam combines the best performance of the AdaGrad and RMSProp algorithms to provide an optimized algorithm that can deal with sparse gradients on noisy problems. And for the loss function, I used sparse categorical cross-entropy. Here is the summary of my model: The first layer

Model: "sequential"

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 224, 224, 16)	448
max_pooling2d (MaxPooling2D)	(None, 112, 112, 16)	0
conv2d_1 (Conv2D)	(None, 112, 112, 32)	4640
max_pooling2d_1 (MaxPooling2	(None, 56, 56, 32)	0
conv2d_2 (Conv2D)	(None, 56, 56, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 28, 28, 64)	0
flatten (Flatten)	(None, 50176)	0
dense (Dense)	(None, 128)	6422656
dense_1 (Dense)	(None, 2)	258

Total params: 6,446,498
Trainable params: 6,446,498
Non-trainable params: 0

Figure 3: Model Summary

is to normalize the data, which is removed for the non-normalized experiments. The rest remains the same for both experiments. There are three pairs of CNN 2D layers with max-pooling to gather information from the image. Then it ends with flatting the data and generating the final prediction by Dense layer.

4 Results

The model is evaluated by the AUC value on the test set. I have run the whole pipeline several times and chosen one of the most representative results to display here, including the analysis in each epoch and the ROC curve of the model on test set. Generally speaking, the AUC value for all experiments is in-between 0.85 ± 0.06, which is acceptable for the deep learning model with such a small-sized training set. Both experiments based on normalized data or non-normalized data has great performance. I used different epochs from [10, 15, 20] and I chose 10 as my final epoch number because the training data is not too large so a large epoch number will not make too much sense on boosting the performance.

Figure 4: Analysis by epoch (normalized)

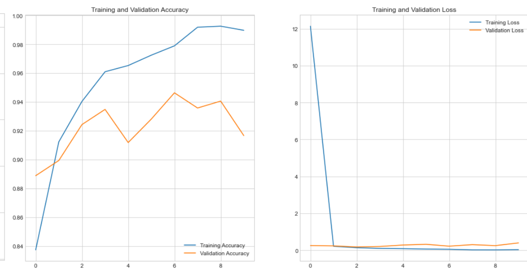

Figure 5: Analysis by epoch (non-normalized)

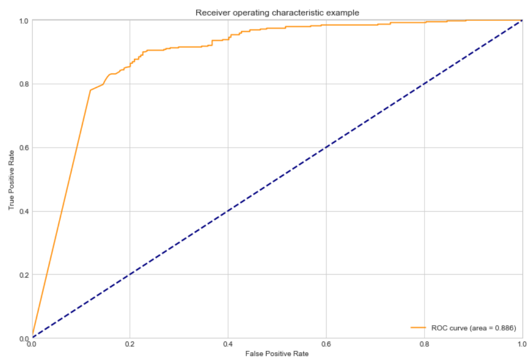

Figure 6: ROC (normalized)

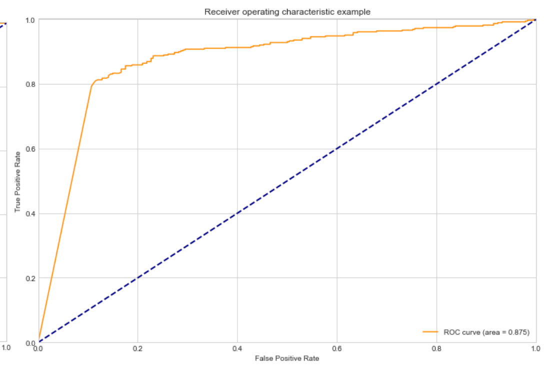

Figure 7: ROC (non-normalized)

From the result, it is obvious that the dataset is imbalanced from the rapid changes of the TP rate when the FP rate is low. And from the comparison between normalized training and non-normalized training, the former training process is much more smooth from epoch to epoch. In my opinion, the imbalance of the data had more effects on the non-normalized training. This is against the opinion that normalization is not indispensable for training a deep neural network Shao et al. (2020). At least when the dataset is size-limited, normalization seems to be able to help deal with the bias in the dataset.

5 Conclusion & Future Work

In this project, I have proposed a CNN model to predict whether a lung has pneumonia based on the chest X-Ray image. The model is trained on the dataset from Kaggle Mooney (2018) and performed an approximate 0.85 ± 0.06 AUC score on the test set. Some thoughts and opinions are obtained during my process to complete the task, such as whether the normalization is indispensable for CNNs and whether a lower epoch number will affect the result massively if the dataset is small enough. For this task, all samples are similar and there are only two categories in total, both of them make it easier for the model to do prediction. How to make the CNNs be used as a reliable and efficient diagnostic tool for all types of the common disease still needs to be emphasized. Kermany et al. (2018).

References

A Aurisano, A Radovic, D Rocco, A Himmel, MD Messier, E Niner, G Pawloski, F Psihas, Alexandre Sousa, and P Vahle. 2016. A convolutional neural network neutrino event classifier. *Journal of Instrumentation*, 11(09):P09001.

Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. 2012. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538.

Tianmei Guo, Jiwen Dong, Henjian Li, and Yunxing Gao. 2017. Simple convolutional neural network on image classification. In *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, pages 721–724.

Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. 2018. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131.

Hyungtae Lee and Heesung Kwon. 2017. Going deeper with contextual cnn for hyperspectral image classification. *IEEE Transactions on Image Processing*, 26(10):4843–4855.

Qing Li, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng, and Mei Chen. 2014. Medical image classification with convolutional neural network. In *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, pages 844–848.

Angela McLuckie. 2009. *Respiratory disease and its management*. Springer Science & Business Media.

Paul Mooney. 2018. Chest x-ray images (pneumonia). Data retrieved from Kaggle, `https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia`.

Olli Ruuskanen, Elina Lahti, Lance C Jennings, and David R Murdoch. 2011. Viral pneumonia. *The Lancet*, 377(9773):1264–1275.

Jie Shao, Kai Hu, Changhu Wang, Xiangyang Xue, and Bhiksha Raj. 2020. Is normalization indispensable for training deep neural network? *Advances in Neural Information Processing Systems*, 33.

Farhana Sultana, Abu Sufian, and Paramartha Dutta. 2018. Advancements in image classification using convolutional neural network. In *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pages 122–129.

Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. 2016. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294.