Machine Learning Engineer Nanodegree Capstone Proposal

Jingde Guo

March 19, 2020

Domain background

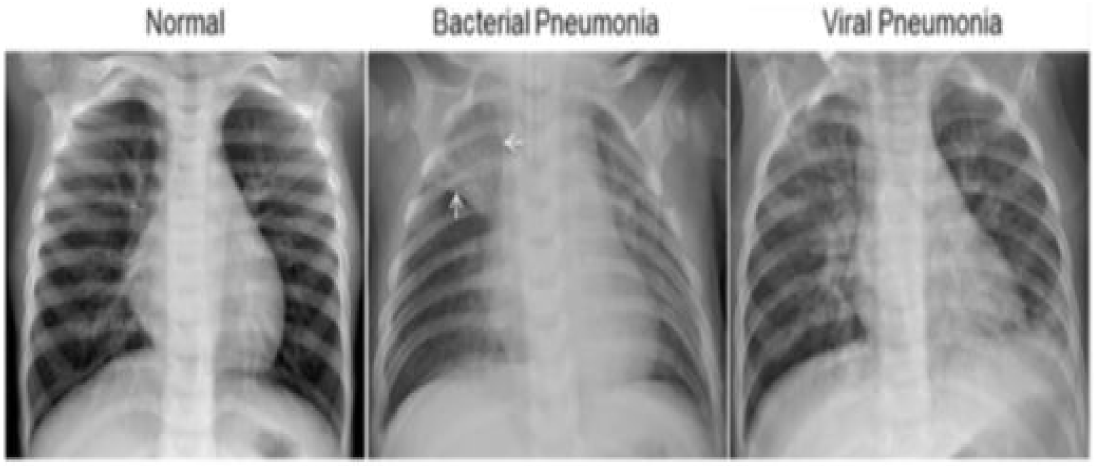
Deep learning is one of the family members of the machine learning based on artificial neural networks. With the neural networks, images can be classified well. In medical image science deep learning can be used to classify the X-ray images. In this project Chest X-rays are used to classify whether Pneumonia exists. With the model that can classify whether the patient has Pneumonia, doctors can reduce time to read X-ray images. The model will give the possibility of the existence of Pneumonia, which can be used as a auxiliary diagnosis. The reason I choose this topic is due to the outbreak of the coronavirus, such classification model can be used to speed up the diagnostic process. Furthermore, it is also a good challenge for me to put everything I learned in practice.

Problem statement

The problem is to create a model that can classify the chest x-ray image in terms of whether Pneumonia exists. A deep learning method will be used, and Amazon SageMaker will be utilized. A high level method for SageMaker training will be deployed in building the model.

Datasets and inputs

The dataset is retrieved from Kaggle:https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.



The sample of the images as above. The size of images varies around 300kb.

The dataset is organized into 3 folders (train, test, val) and contains subfolders for each image category (0/1). 0 represents Normal and 1 represents Pneumonia. There are 5,863 X-Ray images (JPEG). Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients' routine clinical care. For the analysis of chest x-ray images, all chest radio graphs were initially screened for quality control by removing all low quality or unreadable scans. The diagnoses for the images were then graded by two expert physicians before being cleared for training the AI system. In order to account for any grading errors, the evaluation set was also checked by a third expert.

Solution statement

Since this is the binary classification problem, deep learning model will be used. The built-in algorithms in Sagemaker will be helpful for solving the problem. The Amazon SageMaker image classification algorithm is a supervised learning algorithm. It uses a convolutional neural network (ResNet) that can be trained from scratch or trained using transfer learning when a large number of training images are not available.

The libraries will be used in this project are mainly sagemaker, pandas, and matplotlib.

First the exploration of the dataset will be done, for example, the number of normal cases or the number of pneumonia can be visualized. Also the sample images of the dataset can be shown. Then the labels of each image will be created in order to fit the requirement of the algorithm.

The input images for the Amazon SageMaker image classification algorithm can either be RecordIO format or Image format. In my case, the Image format is chosen to be the input format. Therefore, '.lst' files should be created. A .lst file is a tab-separated file with three columns that contains a list of image files. The first column specifies the image index, the second column specifies the class label index for the image, and the third column specifies the relative path of the image file. The data channel should be established afterwards. In the end, we can upload our processed dataset to S3.

Then the parameters and hyper-parameters of the model can be tuned. In our case, "num_classes" is 2, "augmentation_type" can be "crop_color_transform". Also other hyper-parameters can be tuned so that the model can perform better. More information of the hyper-parameters can be found at :
https://docs.aws.amazon.com/sagemaker/latest/dg/IC-Hyperparameter.html.

Then we can train our model and deploy it as an endpoint.

Finally after training job is done, the cutoff value will be analysed in order to get the higher recall ratio. The back-up plan is that the transfer learning with existed model VGG16 might be used.

After everything has been done, the endpoint should be deleted such that no extra costs incurred.

Benchmark model

The benchmark model can be found in Kaggle. The best model in Kaggle Kernels right now has recall ratio 0.98 and precision ratio 0.79. The model he used is based on partial transfer learning and rest of the model will be trained from scratch. My goal is to get the similar result based on the same test dataset.

The link is: https://www.kaggle.com/aakashnain/beating-everything-with-depthwise-convolutionModel

Evaluation metrics

Accuracy ratio, recall ratio and precision ratio can be used to evaluate the model performance. In this case recall ratio should be focused on since the number of False Negatives should be as low as possible.
The formulas can be written as:

$$\text{Precision } = \frac{tp}{tp+fp}$$

$$\text{Recall } = \frac{tp}{tp+fn}$$

$$\text{Accuracy } = \frac{tp+tn}{tp+tn+fp+fn}$$

where tp denotes true positive, fp denotes false positive, tn denotes true negative, and fn denotes false negative.

Project design

First, the dataset will be explored. Then the pictures will be transformed to the format that fit the requirement of the algorithm. Then model parameters and hyper-parameters will be set. Finally model will be tested and improved in order to get the higher recall ratio.