Deep Learning Approach For Prediction Of Pneumonia

Kalyani Kadam, Dr.Swati Ahirrao, Harbir Kaur, Dr. Shraddha Phansalkar, Dr. Ambika Pawar

Abstract: Pneumonia is most important cause of death worldwide even though it is a vaccine preventable disease. It can be detected by analyzing chest x-rays. Analyzing chest x-rays is a difficult task and requires precision. A better and advanced artificial intelligence system for pneumonia detection can go a long way in decreasing the mortality rate and increasing life expectancy. The proposed paper presents a deep neural network based on convolutional neural networks and residual network along with techniques of identifying optimum differential rates using cosine annealing and stochastic gradient with restarts to achieve an efficient and highly accurate network which will help detect and predict the presence of pneumonia using chest-x-rays.

Index Terms— convolutional neural network, cosine annealing, residual networks, stochastic gradient with restarts etc.

1 INTRODUCTION

Pneumonia in India accounts for 20 percent of the death worldwide caused by pneumonia. It is an acute respiratory infection which affects the lungs which can be detected by analyzing chest x-rays. This can be credited to the way that x-rays are very efficient investigative tools in revealing the pathological changes, in addition to its bloodless characteristics and financial considerations [1]. Analyzing and classifying chest x-rays can be very tedious for radiologists since x-rays are often affected by noise due to sensors, electronic devices, and implantation. Utilization of ML (machine learning) and AI (man-made consciousness) in medicinal services field is ascending because of its ability to deal with enormous datasets which is past the extent of human potential, and after that dependably convert examination of that information into clinical bits of knowledge that help doctors in getting ready and giving assistance in consideration, inevitably giving better results, less expenses of consideration and improved patient fulfillment.

Numerous algorithms have been proposed by researchers to effectively analyze x-rays for disease detection [7, 9, and 11]. However, these algorithms could not achieve significant level of accuracy and decision making and hence could not be deployed in medical applications. A number of researches have been conducted in the field of diagnosing diseases using chest x-rays using artificial intelligence. A paper [2] proposed used convolutional neural networks to diagnose disease using x-rays. For comparative analysis, BPNNs (back propagation neural networks) with supervised learning, competitive neural networks (CpNNs) with unsupervised learning were also developed. However, the network did not provide satisfactory results. In [12], the performance measure of various CNN architectures to identify pneumonia from radio graphs have been discussed. All the CNN architectures discussed in the paper did not have satisfactory accuracy value. The success in the field of deep learning motivated researchers to further improve these networks for medical image analysis for disease detection. CNN is widely used in image classification due to its capability to handle spatial characteristics from images [3, 6, and 8].

This paper focuses on developing a deep neural network which will help predict the presence of pneumonia using chest x-rays. In order to achieve this, CNN along with residual networks have been employed to increase efficiency and accuracy. To further add to the performance, optimum differential learning rates have been selected using the techniques of cosine annealing and stochastic gradient with restarts. The detailed explanation is discussed in the further sections.

2 METHODOLOGY

The proposed model aims to achieve maximum accuracy for pneumonia detection using techniques of data augmentation, residual networks, and stochastic gradient with restarts, cosine annealing and differential learning rates. Fig 1 shows the process flow of model development. Further sub-sections give detailed information about the process of the proposed work.
Fig. 1 Process flow of model development

A. Dataset
The Kaggle dataset is used consists of total, 5863 X-Ray images (JPEG). The dataset structured into 3 parts such as train, test, and validation, contains sub folders for each image category or class such as Pneumonia or Normal. Chest X-ray images (anterior-posterior) were looked over review accomplishes of pediatric patients of 1 to 5 years from Guangzhou Women and Children’s Medical Center, Guangzhou. All chest X-ray imaging was gathered and executed as a major aspect of patients' normal clinical consideration. [4].

B. Data Augmentation
In AI when model is prepared, it tunes its parameters with the end goal that it can outline specific info (an image) to some yield (a class label). As number of parameters increase, a proportional amount of examples need to be shown to the model for achieving good performance. A poorly trained neural network may not be able to predict satisfactory results if the images (x-rays) in the target application change color, brightness etc. As the dataset is quite less, and our target application may exist in a variety of conditions, data augmentation is used to create new data with different orientations. Data augmentation was done in order to generate more data and avoid the problem of overfitting. In order to achieve this, random lighting transformation capabilities were used. Using a balance of 0 and contrast of 10 data were augmented. The result of augmentation of some images is shown in Fig. 2.

C. Analysis using neural networks
CNNs (Convolutional Neural Networks) are instrumental for performance in image analysis due to convolutional units and, it’s ability to include various hidden layers handling convolution and subsampling in order to pull out low to high levels of structures of the input data and hence were used for building the model architecture. A dropout of 0.6 to reduce overfitting. The model was first trained with smaller 64 images using CNN and then steadily increase in image size is done for better CNN and then gradually an increase in the image size were done for better efficiency. Only the last fully connected layer added on top of ResNet34 is trained first.

D. Learning rate with cosine annealing and stochastic gradient with restarts (SGDR)
In typical Gradient Descent improvement, like Batch Gradient Descent, the batch is taken to be the entire dataset. Despite the fact that, utilizing the entire dataset is extremely helpful for getting to the minima in a less uproarious or less arbitrary way, however the issue emerges when the datasets get extremely tremendous. In case of Gradient Descent the entire set is required for completing one iteration and it has to be done for each iteration until minima is reached. Due to this, it becomes computationally very expensive. In order to solve this problem Stochastic Gradient Descent is used. In this, it uses a single sample rather than the whole dataset for each iteration. The sample is randomly shuffled and chosen for doing the iteration.

For choosing idle learning rate, the graph of loss versus learning was plotted. The estimation of the learning rate was picked dependent on where it is most noteworthy and the misfortune is at yet sliding. As shown in Fig. 2 the optimum learning rate is 0.01. As the network gets closer to a global minimum value of loss with each batch of stochastic gradient descent, the learning rate ought to likewise decrease so the calculation does not overshoot. Cosine tempering [5], a method of decreasing functions admirably with the learning rate, coming about extraordinary outcomes in a computationally proficient way.

During training the model might get stuck at local minima instead of approaching to the global minima. To avoid this, stochastic gradient descent with restarts (SGDR) [5] was incorporated into the model. This adds to the performance of the model in a manner that by rising the learning rate all of sudden, gradient descent may “hop” out of the local minima and find its way near the global minimum. To handle the
problem of the model getting stuck at local minima, the learning rate is retune at the start of each epoch to the original value entered as a parameter, and then reduce again over the epoch as described cosine annealing. Further to achieve better accuracy, differential learning rates were introduced. As per the paper [10], an ideal learning rate can be evaluated via preparing the model at first with a low learning rate and increase its value at each step. When learning rate is too small, loss does not change much, but as learning rate goes higher, loss should decreases faster and faster until a point where it does not decrease anymore and eventually starts increasing.

As per this, the model was trained on three different learning rates - [1e-4, 1e-3, 1e-2] for 500 epochs, with a cycle length of 1 and cycle multiple of 2. The model was trained for image size 64x64 initially followed by gradual increase in image size (128x128, 256x256, 512x512) to increase accuracy and tackle overfitting. An overall decrease in training loss was observed which is nearly similar to validation loss suggesting that the model is a good fit to the problem.

<table>
<thead>
<tr>
<th>epoch</th>
<th>trn_loss</th>
<th>val_loss</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.182979</td>
<td>0.483302</td>
<td>0.8125</td>
</tr>
<tr>
<td>1</td>
<td>0.142599</td>
<td>0.418486</td>
<td>0.8125</td>
</tr>
<tr>
<td>2</td>
<td>0.122832</td>
<td>0.421858</td>
<td>0.8125</td>
</tr>
<tr>
<td>3</td>
<td>0.128605</td>
<td>0.491195</td>
<td>0.8125</td>
</tr>
<tr>
<td>4</td>
<td>0.109111</td>
<td>0.178241</td>
<td>0.975</td>
</tr>
<tr>
<td>5</td>
<td>0.094232</td>
<td>0.264976</td>
<td>0.875</td>
</tr>
<tr>
<td>6</td>
<td>0.100595</td>
<td>0.183232</td>
<td>0.875</td>
</tr>
<tr>
<td>7</td>
<td>0.097753</td>
<td>0.293054</td>
<td>0.875</td>
</tr>
<tr>
<td>8</td>
<td>0.086892</td>
<td>0.068743</td>
<td>1.0</td>
</tr>
<tr>
<td>9</td>
<td>0.087249</td>
<td>0.144018</td>
<td>0.9375</td>
</tr>
<tr>
<td>10</td>
<td>0.080406</td>
<td>0.031748</td>
<td>1.0</td>
</tr>
<tr>
<td>11</td>
<td>0.066073</td>
<td>0.017114</td>
<td>1.0</td>
</tr>
<tr>
<td>12</td>
<td>0.068921</td>
<td>0.015575</td>
<td>1.0</td>
</tr>
<tr>
<td>13</td>
<td>0.068817</td>
<td>0.015818</td>
<td>1.0</td>
</tr>
<tr>
<td>14</td>
<td>0.064297</td>
<td>0.013403</td>
<td>1.0</td>
</tr>
</tbody>
</table>

[0.013403348624706268, 1.0]

64 x64
512x512

Fig. 3. Results of epochs with training and validation losses

3 Results

After training the model for 500 epochs, test time augmentation was done and the model was tested for performance on the test data. The model accomplished an accuracy of 92.9%, is considerably superior than the baseline accuracy of 63%. The precision and recall values were approximately around 0.9088 and 0.9927 respectively. These values prove that the model is well-fit. After training and building the model, the weights are saved into a pytorch file for prediction of new single image. The pre-trained model is then loaded and evaluated based on the architecture of the trained model. The new data image for classification is transformed as per the trained model and converted to tensor for prediction. Prediction function of the model is used which classifies the image as “pneumonia” or “normal” based on probabilistic predicted value. Multi-level thresholding is applied to the Chest X-rays for segmentation. This is achieved using Otsu thresholding technique which keeps clusters as tight as possible and at the same time maximizes the separation between two clusters (to minimize overlap).[10] Using thresholding the X-ray image is converted into an image with 4 discrete levels. This is achieved by quantising the image with the threshold values obtained via Otsu thresholding.

As per Fig.5 and Fig .6, the collection of mass (colored light-grey) in the chest cavity of a pneumonia affected person can be seen clearly. On the other hand, a clear dark-grey chest cavity is observed for a person not affected with pneumonia.
4 LIMITATIONS
Following limitations were observed during the development of the model:
1. Processing and building the model requires fast and efficient processors which is time and cost consuming.
2. 100% accuracy is not achievable.
3. The challenges of large variation in sensing modality which is complicated by human anatomy are faced in medical image analysis.
4. Parameters affecting medical images fluctuate from organ to organ.
5. Medical Images are affected by noise due to sensors, device implantation, electronics leading to inefficiency while detection.

4 CONCLUSION AND FUTURE SCOPE
The proposed work will help doctors better predict pneumonia in minimal time with high efficiency. The aggregation of this will contribute to the health care system for better patient satisfaction and care. This work is in its early stages and can be improved by adding more images to the dataset, incorporating better architectures, training the model based on more transformations and orientations.

REFERENCES