

Ignacio Rojas  
Olga Valenzuela  
Fernando Rojas Ruiz  
Luis Javier Herrera  
Francisco Ortuño (Eds.)

LNBI 13920

# Bioinformatics and Biomedical Engineering

10th International Work-Conference, IWBBIO 2023  
Meloneras, Gran Canaria, Spain, July 12–14, 2023  
Proceedings, Part II

2 Part II

 Springer

Lecture Notes in Computer Science

**Lecture Notes in Bioinformatics**

**13920**

Series Editors

Sorin Istrail, *Brown University, Providence, RI, USA*

Pavel Pevzner, *University of California, San Diego, CA, USA*

Michael Waterman, *University of Southern California, Los Angeles, CA, USA*

Editorial Board Members

Søren Brunak, *Technical University of Denmark, Kongens Lyngby, Denmark*

Mikhail S. Gelfand, *IITP, Research and Training Center on Bioinformatics, Moscow, Russia*

Thomas Lengauer, *Max Planck Institute for Informatics, Saarbrücken, Germany*

Satoru Miyano, *University of Tokyo, Tokyo, Japan*

Eugene Myers, *Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany*

Marie-France Sagot, *Université Lyon 1, Villeurbanne, France*

David Sankoff, *University of Ottawa, Ottawa, Canada*

Ron Shamir, *Tel Aviv University, Ramat Aviv, Tel Aviv, Israel*

Terry Speed, *Walter and Eliza Hall Institute of Medical Research, Melbourne, VIC, Australia*

Martin Vingron, *Max Planck Institute for Molecular Genetics, Berlin, Germany*

W. Eric Wong, *University of Texas at Dallas, Richardson, TX, USA*

The series Lecture Notes in Bioinformatics (LNBI) was established in 2003 as a topical subseries of LNCS devoted to bioinformatics and computational biology.

The series publishes state-of-the-art research results at a high level. As with the LNCS mother series, the mission of the series is to serve the international R & D community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings.

Ignacio Rojas · Olga Valenzuela ·  
Fernando Rojas Ruiz · Luis Javier Herrera ·  
Francisco Ortuño  
Editors

# Bioinformatics and Biomedical Engineering

10th International Work-Conference, IWBBIO 2023  
Meloneras, Gran Canaria, Spain, July 12–14, 2023  
Proceedings, Part II

*Editors*

Ignacio Rojas   
University of Granada  
Granada, Spain

Olga Valenzuela   
University of Granada  
Granada, Spain

Fernando Rojas Ruiz   
University of Granada  
Granada, Spain

Luis Javier Herrera   
University of Granada  
Granada, Spain

Francisco Ortuño   
University of Granada  
Granada, Spain

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Bioinformatics

ISBN 978-3-031-34959-1

ISBN 978-3-031-34960-7 (eBook)

<https://doi.org/10.1007/978-3-031-34960-7>

LNCS Sublibrary: SL8 – Bioinformatics

© The Editor(s) (if applicable) and The Author(s), under exclusive license  
to Springer Nature Switzerland AG 2023

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland



# Cyclical Learning Rates (CLR'S) for Improving Training Accuracies and Lowering Computational Cost

Rushikesh Chopade<sup>1</sup>, Aditya Stanam<sup>2</sup>, Anand Narayanan<sup>3</sup>, and Shrikant Pawar<sup>3,4</sup>(✉)

<sup>1</sup> Department of Geology and Geophysics, Indian Institute of Technology, Kharagpur, India

<sup>2</sup> Department of Toxicology, University of Iowa, Iowa City, IA 52242-5000, USA  
aditya-stanam@uiowa.edu

<sup>3</sup> Yale Center for Genomic Analysis, Yale School of Medicine, Yale University, New Haven, CT 30303, USA

{anand.narayanan, shrikant.pawar}@yale.edu

<sup>4</sup> Department of Computer Science & Biology, Claflin University, Orangeburg, SC 29115, USA

**Abstract.** Prediction of different lung pathologies using chest X-ray images is a challenging task requiring robust training and testing accuracies. In this article, one-class classifier (OCC) and binary classification algorithms have been tested to classify 14 different diseases (atelectasis, cardiomegaly, consolidation, effusion, edema, emphysema, fibrosis, hernia, infiltration, mass, nodule, pneumonia, pneumothorax and pleural-thickening). We have utilized 3 different neural network architectures (MobileNetV1, Alexnet, and DenseNet-121) with four different optimizers (SGD, Adam, and RMSProp) for comparing best possible accuracies. Cyclical learning rate (CLR), a tuning hyperparameters technique was found to have a faster convergence of the cost towards the minima of cost function. Here, we present a unique approach of utilizing previously trained binary classification models with a learning rate decay technique for re-training models using CLR's. Doing so, we found significant improvement in training accuracies for each of the selected conditions. Thus, utilizing CLR's in callback functions seems a promising strategy for image classification problems.

**Keywords:** One-Class Classifier · Optimizer · Cyclical Learning Rates

## 1 Introduction

Speech recognition, computer vision and text analysis are major fields in which deep learning is prominently used for image classification [1–3]. Cyclical learning rates (CLR's) allow the learning rates to vary between a range of boundary values. Selecting learning rate manually is a time consuming and computationally costly task [4]. Optimal learning rate is important as the model can converge slowly if the learning rate is too slow or the model can diverge from the minima of the cost function if the learning rate is too high [5]. Even if an optimal learning rate for the model is achieved, the model can take many epochs to reach the minima of the loss function. The model doesn't have a

regular cost function, moreover, the gradient of the cost function is different in different parts of the cost function curve [6]. To overcome this issue, instead of using constant single learning rate, a learning rate decay policy can be used to obtain better results. However, the learning rate decay also has several drawbacks including getting stuck in a local minimum or plateau of cost function due to very small learning rates in later epochs [7]. CLR's can be an effective technique to make the model converge faster in minimal number of epochs and to decrease the efforts of finding optimal learning rates.

## 2 Experimental Results and Analysis

### 2.1 Data Collection, Preprocessing, Model Architecture, and Learning Rates

#### 2.1.1 Data Collection

The data used for binary and one-class classification has been made available by National Institutes of Health (NIH), USA [8]. This dataset consists of 112,120 chest X-ray images, each with a 1024 \* 1024-pixel resolution. Images belong to 15 classes, 14 classes of diseased individuals and 1 class of healthy individuals ('No Finding'). The disease classes contain 'Atelectasis', 'Cardiomegaly', 'Consolidation', 'Effusion'; 'Emphysema', 'Edema', 'Fibrosis', 'Infiltration', 'Mass', 'Nodule', 'Pneumonia', 'Pneumothorax', 'Pleural Thickening' and 'Hernia'. A metadata associated with the image dataset consists of patient's age, gender, unique patient id, and the view position (anterior-posterior and posterior-anterior) of the X-ray image.

#### 2.1.2 Exploratory Data Analysis

From the total set, 60,361 images have the label 'No Finding' (healthy), while others have multiple labels with combinations of 14 classes. Overall, the unique constitutes to around 836 labels. Unique can be any of the 14 primary classes ('No Finding' label excluded) or any combination of these 14 primary classes. Figure 1 depicts the distribution of these 15 unique labels.

A one-hot encoding was applied to convert 836 unique labels to 15 primary class labels [9]. Comparison of the number of images in 15 primary classes before and after performing one-hot encoding is shown in Table 1. A plot for the number of images after performing one-hot encoding is shown in Fig. 2.

Binary classifiers have been developed on each disease and the 'No Finding' class. The 'No Finding' class has approximately 3 times more images than the 'Infiltration' class, this type of unbalanced dataset can raise a state where the algorithm will overfit the class having more images. To avoid this, the number of images in the 'No Finding' class has been taken approximately the same as the number of images in the class for which the binary classifier was developed.

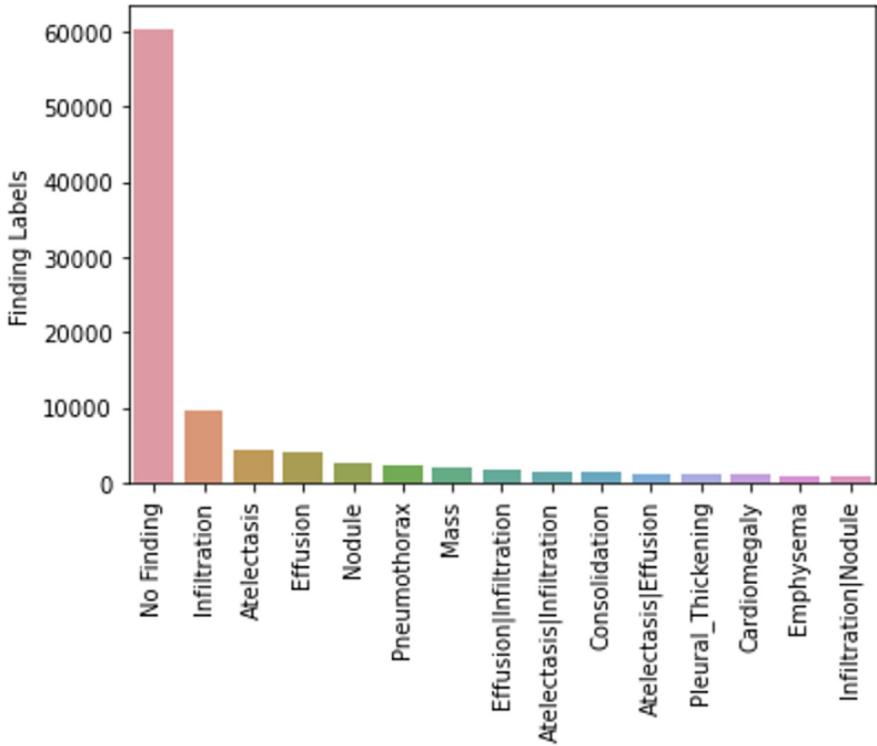


Fig. 1. Number of top 15 unique labels.

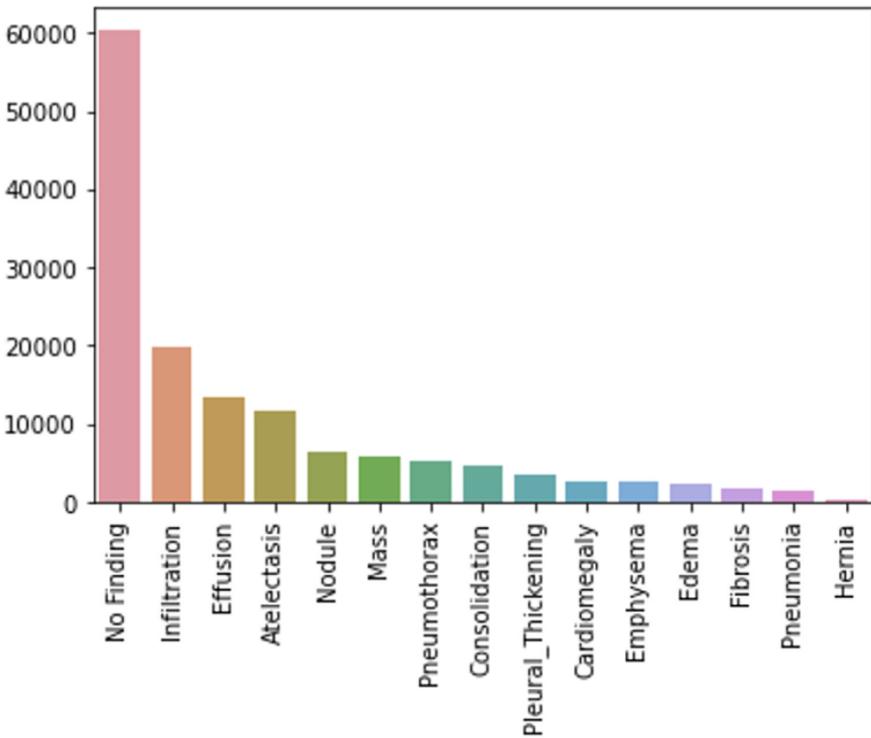
Table 1. Counts per class for primary labels before and after one-hot encoding.

| Image Label          | No. of Images before One Hot Encoding | No. of Image before One Hot Encoding |
|----------------------|---------------------------------------|--------------------------------------|
| <b>No Finding</b>    | 60361                                 | 60361                                |
| <b>Atelectasis</b>   | 4215                                  | 11559                                |
| <b>Cardiomegaly</b>  | 1093                                  | 2776                                 |
| <b>Consolidation</b> | 1310                                  | 4667                                 |
| <b>Edema</b>         | 628                                   | 2303                                 |
| <b>Emphysema</b>     | 892                                   | 2516                                 |
| <b>Effusion</b>      | 3955                                  | 13317                                |
| <b>Fibrosis</b>      | 727                                   | 1686                                 |
| <b>Infiltration</b>  | 9547                                  | 19894                                |
| <b>Mass</b>          | 2139                                  | 5782                                 |

(continued)

**Table 1.** (continued)

| Image Label               | No. of Images before One Hot Encoding | No. of Image before One Hot Encoding |
|---------------------------|---------------------------------------|--------------------------------------|
| <b>Nodule</b>             | 2705                                  | 6331                                 |
| <b>Pneumothorax</b>       | 2194                                  | 5302                                 |
| <b>Pneumonia</b>          | 322                                   | 1431                                 |
| <b>Pleural Thickening</b> | 1126                                  | 3385                                 |
| <b>Hernia</b>             | 110                                   | 227                                  |



**Fig. 2.** Counts per class for primary labels after one-hot encoding.

### 2.1.3 Pre-processing of Data

*Binary Classifier* A 1:4-fold split of test to training set was performed for 14 binary classifiers (Table 2). To save overhead memory and making model more robust, we passed all the images through a ImageDataGenerator class of Keras [10] (shear range of 0.05, zoom range of 0.1, rotation range of 7°, width, and height shift range of 0.1, brightness range of 0.4 to 1.5 with a horizontal flip), while subsequently applying image

augmentation technique. These techniques helped the model to generalize and reduce the overfitting state.

**Table 2.** List of binary classifiers and the number of images in their training and test sets.

| Binary Classifier         | No. of images containing respective disease label | No. of Images with ‘No Finding’ Label | Total Images | No. of training images (80% of total images) | No. of test images (20% of total images) |
|---------------------------|---|---------------------------------------|--------------|--|--|
| <b>Atelectasis</b>        | 11559   | 12000                                 | 23599        | 18847  | 4712                                     |
| <b>Cardiomegaly</b>       | 2776  | 2800                                  | 5576         | 4460   | 1116                                     |
| <b>Consolidation</b>      | 4667  | 4700                                  | 9367         | 7493   | 1874                                     |
| <b>Edema</b>              | 2303  | 2300                                  | 4603         | 3682   | 921                                      |
| <b>Emphysema</b>          | 2516  | 2600                                  | 5116         | 4092   | 1024                                     |
| <b>Effusion</b>           | 13317   | 13500                                 | 26817        | 21453  | 5364                                     |
| <b>Fibrosis</b>           | 1686  | 1700                                  | 3386         | 2708   | 678                                      |
| <b>Infiltration</b>       | 19894   | 20000                                 | 39894        | 31915  | 7979                                     |
| <b>Mass</b>               | 5782  | 6000                                  | 11782        | 9425   | 2357                                     |
| <b>Nodule</b>             | 6331  | 6500                                  | 12831        | 10264  | 2567                                     |
| <b>Pneumothorax</b>       | 5302  | 5500                                  | 10802        | 8641   | 2161                                     |
| <b>Pneumonia</b>          | 1431  | 1500                                  | 2931         | 2344   | 587                                      |
| <b>Pleural Thickening</b> | 3385  | 3500                                  | 6885         | 5508   | 1377                                     |
| <b>Hernia</b>             | 227   | 250                                   | 447          | 381  | 96                                       |

A dynamic batch training was utilized to decrease computational time and memory. Based on optimal performance, an iterative loop of 32 images/batch was used for training till all the images in batch were exhausted. Apart from utilizing less memory, this method helps to save fewer errors in the memory for updating hyperparameters through backpropagation which increases the training speed drastically. The high-resolution X-ray images for training have higher fractional improvements in area under curve (AUC) [11], and also can help localize a disease pattern (Table 3).

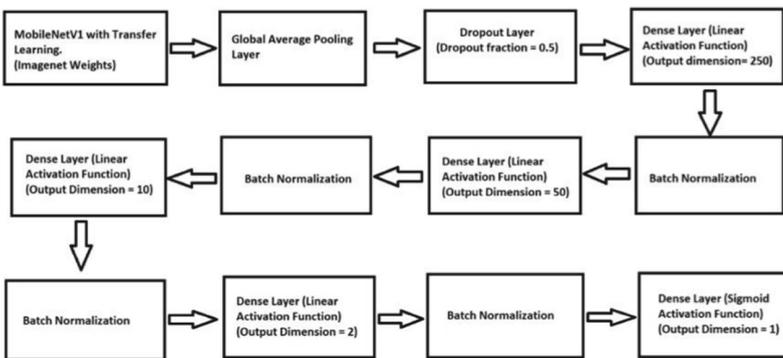
*One Class Classifier.* With the idea of choosing a balanced data, the dataset for one-class classifier contains 2,800 images of “No Finding” class and 200 images from each disease class. We again choose 1:4-fold split of test to training set to be consistent with binary classifiers. Further, the preprocessing through ImageDataGenerator class with same parameters as binary classifiers was performed for this split. Dynamic training with an optimal batch of 16 images/batch was performed.

**Table 3.** Number of training and testing batches with respective batch sizes for all the binary classifiers.

| Binary Classifier         | Total Images | Batch Size | No. of training batches | No. of test batches |
|---------------------------|--------------|------------|-------------------------|---------------------|
| <b>Atelectasis</b>        | 23599        | 16         | 1178                    | 295                 |
| <b>Cardiomegaly</b>       | 5576         | 32         | 140                     | 35                  |
| <b>Consolidation</b>      | 9367         | 32         | 235                     | 59                  |
| <b>Edema</b>              | 4603         | 32         | 116                     | 29                  |
| <b>Emphysema</b>          | 5116         | 32         | 128                     | 32                  |
| <b>Effusion</b>           | 26817        | 32         | 671                     | 168                 |
| <b>Fibrosis</b>           | 3386         | 32         | 85                      | 22                  |
| <b>Infiltration</b>       | 39894        | 16         | 1995                    | 499                 |
| <b>Mass</b>               | 11782        | 32         | 295                     | 74                  |
| <b>Nodule</b>             | 12831        | 32         | 321                     | 81                  |
| <b>Pneumothorax</b>       | 10802        | 32         | 271                     | 68                  |
| <b>Pneumonia</b>          | 2931         | 16         | 147                     | 37                  |
| <b>Pleural Thickening</b> | 6885         | 32         | 173                     | 44                  |
| <b>Hernia</b>             | 447          | 4          | 96                      | 24                  |

**2.1.4 Model Architectures for Binary & One-Class Classifiers**

*Binary Classifier:* A 2D convolutional neural network is applied using an MobileNetV1 network architecture [12]. The model parameters of MobileNet previously trained on ImageNet have been utilized using transfer learning (Fig. 3).



**Fig. 3.** Model architecture used for all the binary classifiers.

For MobileNetV1 previously trained ImageNet weights are passed through a global average pooling layer considering averages of each feature map instead of adding fully

connected layers. This technique helps to easily interpret feature maps as categories confidence maps, to reduce overfitting, and is more robust to spatial translations of the input as it sums out the spatial information [13]. To further reduce overfitting, a dropout regularization layer to drop ~50% of the input units for variance reduction has been applied after the global average pooling layer. The model is then passed through 4 dense layers of output nodes 250, 50, 10, and 2 with linear activation functions in them. In each dense layer, L1 and/or L2 regularization is applied to the layer's kernel, bias, and activity. Kernel regularizer with both L1 and L2 penalties of 0.001 and 0.01 respectively are applied on the kernel's layer. A bias regularizer with an L2 penalty of 0.01 is applied on the layer's bias. Activity regularizer with an L2 penalty of 0.001 is applied on the layer's output. After each dense layer, batch normalization is used to stabilize the learning process and dramatically reduce the number of training epochs required to train a deep neural network. Finally, the model architecture is complete with application of a dense layer comprising of sigmoid activation function and 1 output node. The stochastic gradient descent (SGD) optimizer with learning rate decay has been used to train the model as it gave a superior performance compared to RMSProp and adam optimizer for all the classifiers except "Hernia". Adam optimizer with a learning rate of 0.01 has been found to perform better in case of "Hernia". A momentum parameter has been used to help accelerate gradient vectors in right directions (Table 4).

**Table 4.** Chart showing optimizer, its momentum, learning rates, and the decay constants used with SGD optimizer for all binary classifiers (Except Hernia which has Adam optimizer).

| Binary Classifier         | Optimizer Used | Learning Rate | Decay constant | Momentum |
|---------------------------|----------------|---------------|----------------|----------|
| <b>Atelectasis</b>        | SGD            | 0.01          | 0.001          | 0.9      |
| <b>Cardiomegaly</b>       | SGD            | 0.1           | 0.0005         | 0.9      |
| <b>Consolidation</b>      | SGD            | 0.05          | 0.0005         | 0.9      |
| <b>Edema</b>              | SGD            | 0.01          | 0.0005         | 0.9      |
| <b>Emphysema</b>          | SGD            | 0.01          | 0.0005         | 0.9      |
| <b>Effusion</b>           | SGD            | 0.01          | 0.001          | 0.9      |
| <b>Fibrosis</b>           | SGD            | 0.001         | 0.00005        | 0.9      |
| <b>Infiltration</b>       | SGD            | 0.01          | 0.001          | 0.9      |
| <b>Mass</b>               | SGD            | 0.01          | 0.001          | 0.9      |
| <b>Nodule</b>             | SGD            | 0.01          | 0.001          | 0.9      |
| <b>Pneumothorax</b>       | SGD            | 0.01          | 0.0005         | 0.9      |
| <b>Pneumonia</b>          | SGD            | 0.01          | 0.0001         | 0.9      |
| <b>Pleural Thickening</b> | SGD            | 0.05          | 0.0005         | 0.9      |
| <b>Hernia</b>             | Adam           | 0.01          | 0.0001         | 0.9      |

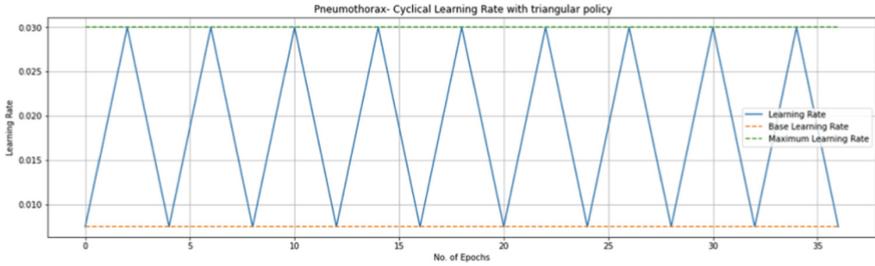
*One Class Classifier:* A false-positive predictions arise when the algorithm is unable to identify the "No Finding" class, a problem falling under the category of "Anomaly

Detection”. One-class classifier is an unsupervised learning algorithm focusing on the problem of anomaly detection [14]. The model contains a negative class (inlier or normal class) and a positive class (outlier or anomaly class). In our case, the normal class or inlier class is the “No Finding” class. The anomaly class is formed by combining 200 images of each disease class. The benefit of this approach is that if the prediction/test image fed to the algorithm is not from any of the 14 disease classes, it will still categorize it as an “Anomaly” simply because the algorithm could not classify it as an image with “No Finding” class. If the algorithm classifies the image with a disease other than these 14 diseases as a “No Finding” class, it will give rise to a problem of false negative prediction. One-class classifier serves the purpose of solving the problem of both false positives and false negative predictions. The model architecture for one-class classifier is same as the binary classifier.

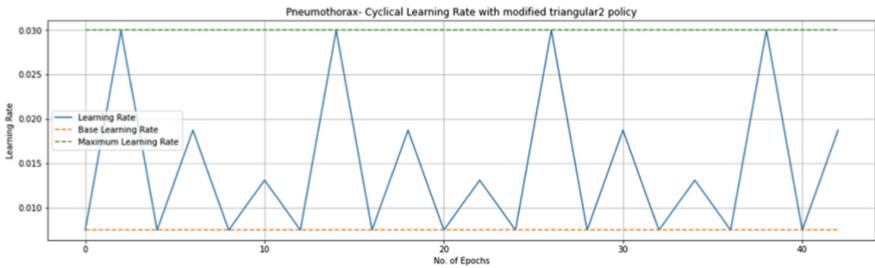
### 2.1.5 Cyclical Learning Rates

The first step in applying CLR’s is to define a maximum learning rate and a base learning rate [4]. The learning rate can then be allowed to vary between maximum learning rate and base learning rate. We have utilized learning rate finder technique (described in section A6) to decide maximum learning and the base learning rates. For one condition, “Pneumothorax” binary classifier, maximum and the base learning rates of 0.03 and 0.0075 respectively were obtained using learning rate finder. A step size is an important parameter which simply is the number of batches in which the learning rate will become equal to the maximum learning rate starting from the base learning rate or vice-versa. It is the number of training batches to reach half cycle. Typically, the step size of 2–8 times the number of training batches in 1 epoch is ideal [4]. For “Pneumothorax”, the total number of training batches in 1 epoch is equal to 541. Therefore, a step size of 1082 was used for learning rate finder. Finally, a mode policy needs to be defined for calculating learning rates. Mode is the pattern in which the learning rate will vary within the bounds of maximum and minimum learning rates. The “triangular” policy for “Pneumothorax” binary classifier is shown in Fig. 4. The learning rate monotonically increases to maximum learning rate from base learning rate in two epochs and decreases back to base learning rate in the next two epochs. Since the “Pneumothorax” model with CLR technique and “triangular” policy is trained for 36 epochs, a total of 9 full cycles can be observed in Fig. 4.

We also have parallelly utilized a more complex policy called a “modified triangular2” policy. In this policy, the maximum learning rate is not taken to be the average of previous maximum learning rate unlike “triangular2” policy. After 3 complete cycles of the “triangular2” policy, the training is continued with “triangular2 policy” with original maximum learning rate obtained from the learning rate finder technique. This process is carried out until whole training is exhausted. In the “Pneumothorax” binary classifier, the maximum learning rate in the first cycle is 0.03 from the first learning rate finder cycle, followed by second cycle with maximum learning rate of 0.01875, followed by third cycle with maximum learning rate of 0.013125 (Fig. 5), etc.



**Fig. 4.** Plot showing the “Triangular” policy for “Pneumothorax” binary classifier trained for 36 epochs.



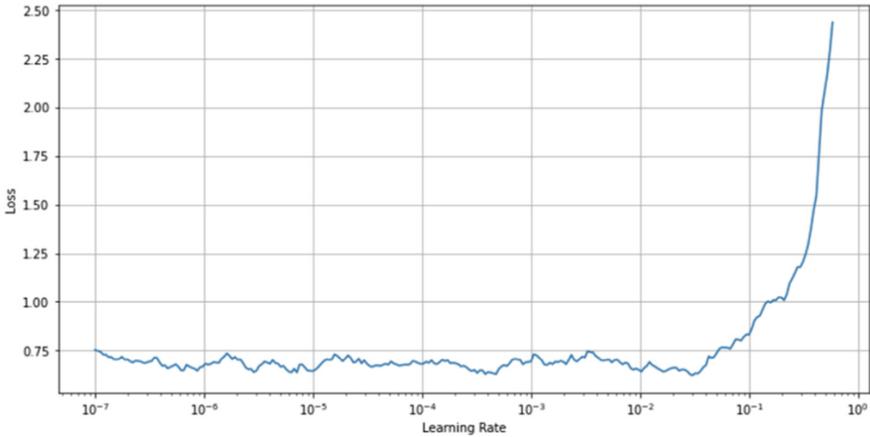
**Fig. 5.** Plot for the “modified triangular2” policy of “Pneumothorax” binary classifier trained for 42 epochs.

### 2.1.6 Learning Rate Finder

The upper and lower bounds of the CLR have been determined by learning rate finder technique where the cost function is minimum. Training the model with a learning rate finder as a callback for 1-5 epochs was enough to get the learning rate with minimum cost function. In case of the “Pneumonia” binary classifier, the minimum and maximum values for the learning rates were  $1e-7$  as minimum and 1 as maximum (Fig. 6). The training increases exponentially after each batch on minimum learning rate. The “Pneumothorax” model loss vs. learning rate curve trained for 10 epochs is found to have a learning rate of  $3e-2$  with minimum loss (Fig. 6). This loss increased as the learning rate approached to 1. The base learning rate for CLR can be accounted to one-fourth of the maximum learning rate [4].

### 2.1.7 With Binary Classifiers CLR’S Out-Perform Normal Training with a Learning Rate Decay Policy

We have run 3 model architectures (MobileNetV1, AlexNet, and DenseNet121) for comparing the performance (computational cost & accuracy) of classifiers [15, 16]. MobileNetV1 with an SGD optimizer was found to be most efficient, while DenseNet121 had good accuracy but significantly more computational cost, AlexNet had significantly lower accuracies when trained for the same number of epochs (Table 5).



**Fig. 6.** Loss vs. learning rate plot for “Pneumothorax” binary classifier trained for 10 epochs.

**Table 5.** Accuracies of all the binary classifiers after training for given number of epochs.

| Binary Classifier           | No. of Epochs | Accuracy (in %) |
|-----------------------------|---------------|-----------------|
| <b>Atelectasis</b>          | 10            | 75.10           |
| <b>Cardiomegaly</b>         | 12            | 75.78           |
| <b>Consolidation</b>        | 10            | 73.32           |
| <b>Edema</b>                | 12            | 93.37           |
| <b>Emphysema</b>            | 10            | 85.60           |
| <b>Effusion</b>             | 10            | 86.53           |
| <b>Fibrosis</b>             | 10            | 66.58           |
| <b>Infiltration</b>         | 10            | 64.60           |
| <b>Mass</b>                 | 10            | 70.11           |
| <b>Nodule</b>               | 10            | 68.23           |
| <b>Pneumothorax</b>         | 10            | 70.12           |
| <b>Pneumonia (with CLR)</b> | 30            | 88.43           |
| <b>Pleural Thickening</b>   | 10            | 71.67           |
| <b>Hernia</b>               | 30            | 90.81           |

The problem of false-positive predictions was addressed using one-class classifiers. For the models of “Infiltration”, “Atelectasis”, “Fibrosis” & “Pneumothorax” the accuracies have been consistently low after training for the selected number of epochs. So, we chose these conditions to test CLR’s on (Table 6).

The problem of false-positive and false-negative predictions was resolved with one class classifiers. After which, a selected model trained for 32 epochs using CLR’s with

**Table 6.** Comparison of the network architectures for “Atelectasis” binary classifier.

| Name of Model Architecture | Approx. training time per epoch in hours | Training Epochs | Accuracy (in %) |
|----------------------------|--|-----------------|-----------------|
| <b>MobileNetV1</b>         | 2.5                                      | 10              | 75.10           |
| <b>DenseNet121</b>         | 5  | 10              | 78.07           |
| <b>AlexNet</b>             | 1.5                                      | 30              | 75.50           |

a maximum learning rate of 0.1, a base learning rate of 0.025, a step size of 2, and with a “triangular” policy provided a final training accuracy of 83.01%. CLR’s showed improved accuracy and a lower computational cost compared to training a network with constant learning rates (Tables 7 and 8).

**Table 7.** Classifier accuracies after application of CLR’s.

| Binary Classifier   | Accuracy before CLR application (in %) | Epochs taken to achieve the accuracy before CLR application | Accuracy after CLR application (in %) | Epochs taken to achieve the accuracy after CLR application | Policy Used          |
|---------------------|--|---|---------------------------------------|--|----------------------|
| <b>Atelectasis</b>  | 75.10                                  | 10  | 79.59                                 | 32   | Triangular           |
| <b>Infiltration</b> | 64.6                                   | 10  | 76.15                                 | 10   | Modified Triangular2 |
| <b>Fibrosis</b>     | 66.58                                  | 10  | 88.96                                 | 32   | Modified Triangular2 |
| <b>Pneumothorax</b> | 70.12                                  | 10  | 79.83                                 | 36   | Triangular           |
| <b>Pneumonia</b>    | –                                      | –   | 88.43                                 | 30   | Triangular           |

**Table 8.** Parameters and specifications of the CLR’s.

| Binary Classifier   | Policy Used          | Step Size | Epochs | Maximum Learning Rate | Base Learning Rate |
|---------------------|----------------------|-----------|--------|-----------------------|--------------------|
| <b>Atelectasis</b>  | Triangular           | 2         | 32     | 0.1                   | 0.025              |
| <b>Infiltration</b> | Modified Triangular2 | 2         | 10     | 0.02                  | 0.005              |
| <b>Fibrosis</b>     | Modified Triangular2 | 2         | 32     | 0.002                 | 0.0005             |
| <b>Pneumothorax</b> | Triangular           | 2         | 36     | 0.03                  | 0.0075             |
| <b>Pneumonia</b>    | Triangular           | 2         | 30     | 0.01                  | 0.0001             |

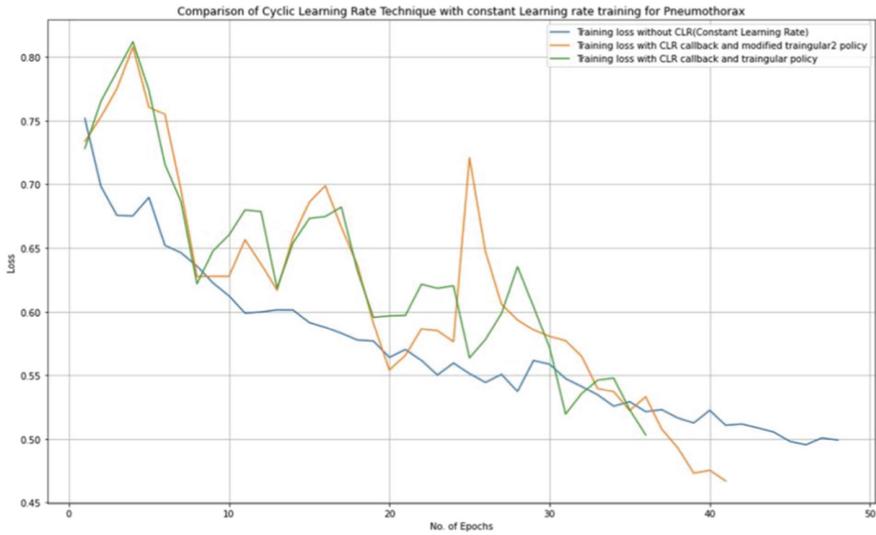
The “Pneumothorax” model is found to perform best when the CLR’s is used with a “triangular” policy. As shown in Fig. 7, it took 47 epochs for the model with a constant learning rate to reach an accuracy of 79.26%. With CLR using “modified triangular2” policy crossed the accuracy level of 79.26% at 38 epoch and reached the accuracy of 80.92% in 41 epochs. While, the “Pneumothorax” model with CLR using a “triangular” policy crossed the accuracy level of 79.26% in just 36 epochs to achieve final accuracy of 79.83%.



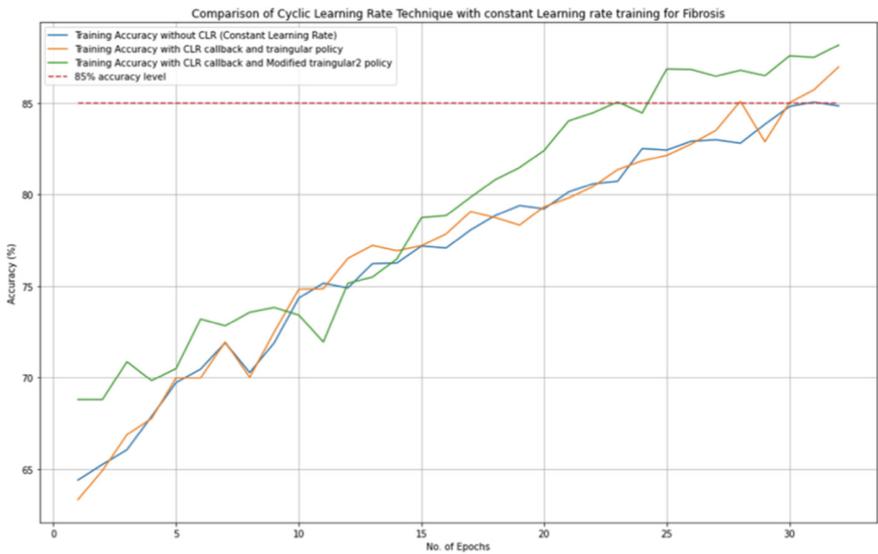
**Fig. 7.** Accuracy plot for “Pneumothorax” binary classifier with constant learning rate, CLR with “triangular” and CLR with “modified triangular2” policies.

The loss compared to the number of epochs was seen to be decreased with CLR’s in both “triangular” and “modified triangular2” policies (Fig. 8). The loss of the “Pneumothorax” model with CLR reduced quicker than the “Pneumothorax” model with a constant learning rate.

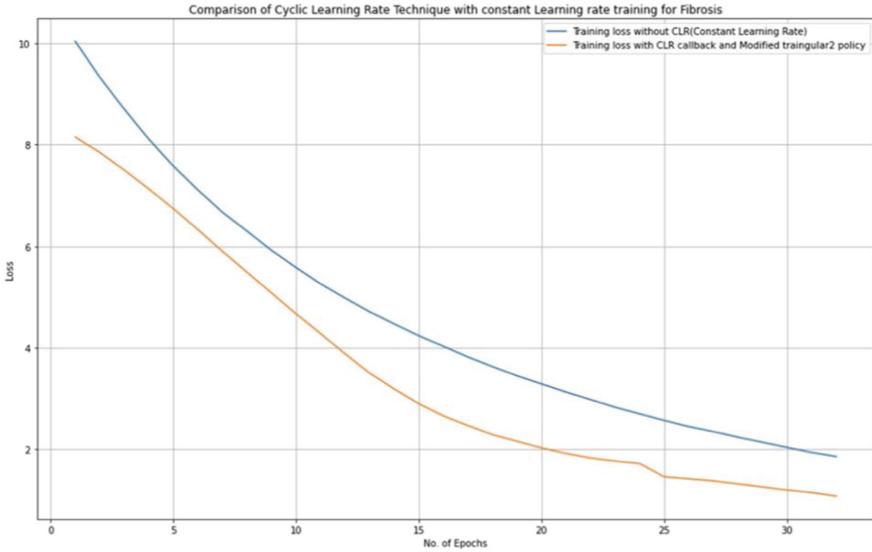
The “Fibrosis” model was found to give better results in the case of the CLR technique with a “modified triangular2” policy. A comparison of “fibrosis” model trained for 32 epochs is shown in Fig. 9. The model reached an accuracy of 85.04% in 32 epochs when trained with a constant learning rate policy. The model reached an accuracy of 86.96% in 32 epochs when trained with CLR using a “triangular” policy. It crossed the 85% accuracy level in 30 epochs. The model reached an accuracy of 88.15% in 32 epochs when trained with CLR using a “modified triangular2” policy. It crossed the accuracy level of 85% in just 25 epochs. The loss was observed to be always less in CLR’s with a “modified triangular2” (Fig. 10).



**Fig. 8.** Loss for “Pneumothorax” model with constant learning rate, CLR with “triangular” policy and CLR with “modified triangular2” policy.



**Fig. 9.** Accuracy plot comparing “Fibrosis” binary classifier with constant learning rate, CLR with “triangular” policy and CLR with “modified triangular2” policy.



**Fig. 10.** Loss for “Fibrosis” binary classifier with constant learning rate and CLR with a “modified Triangular2” policy.

### 3 Discussion and Future Scope

Depthwise separable convolutions like MobileNets have been gradually pruned for improving the speed of dense network [17]. MobileNetV1 Imagenet weights with SGD optimizer is found to outperform other optimizers and architectures in terms of training time taken and accuracy attained. Achieving a high test accuracy is directly depended on learning rate hyper-parameter for training neural networks [18–21]. Three forms of triangle CLR’s have been stated to accelerate neural network training [18, 19]. Further, tuning the batch size hyper-parameter for adjusting learning rates have also been shown to improve learning accuracy [22]. Some hyperparameter tools like Hyperopt, SMAC, and Optuna, using grid search, random search and Bayesian optimization have been seen efficient in tuning batch sizes [23, 24]. To the best of our knowledge, our work is the first to present a comprehensive characterization of CLR function on training and testing accuracy of dense network models. In general, training any model with a CLR technique is found to perform better than training with a constant learning rate. For the “Pneumothorax” binary classifier, the CLR technique with the “triangular” policy is found to outperform both CLR with the “modified triangular2” policy and constant learning rate training. For the “Fibrosis” binary classifier, the CLR with the “modified triangular2” policy was found to give better results than the rest two policies. Primarily, we found that there are two main advantages of training with CLR’s over constant learning rates, with decay learning rates the model can get stuck into the saddle points or local minima due to low learning rates, and secondly CLR’s reduces the effort of choosing an optimal learning rate by hit and trial method. Poor choice of initial learning rate can make the model circle infinitely. In setting a learning rate, there is a trade-off between the rate of convergence and overshooting, a high learning rate will make the learning jump over

minima but a too low learning rate will either take too long to converge or get stuck in an undesirable local minimum [25]. The CLR's cyclically provided higher learning rates too, which helped the model to jump out of the local minima of the cost function. With these findings, implementing CLR's for improving prediction accuracies seems a promising strategy for object detection and machine translation.

**Author Contributions.** SP, AS, AN, and RC conceived the concepts, planned, and designed the article. SP, AS, AN, and RC primarily wrote and edited the manuscript.

**Funding.** This work was primarily supported by the National Science Foundation EPSCoR Program under NSF Award # OIA-2242812.

**Data Availability Statement.** The datasets generated during and/or analyzed during the current study are available in the repository: [https://bitbucket.org/chestai/chestai\\_rushikes\\_code/src/master/](https://bitbucket.org/chestai/chestai_rushikes_code/src/master/) <https://www.kaggle.com/nih-chest-xrays/data>.

**Conflict of Interest for All Authors.** None.

## References

1. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580–587 (2014)
2. Graves, A., Jaitly, N.: Towards end-to-end speech recognition with recurrent neural networks. In Proceedings of the 31st International Conference on Machine Learning (ICML14), pp. 1764–1772 (2014)
3. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1701–1708. IEEE (2014)
4. Smith, L.N.: Cyclical Learning Rates for Training Neural Networks (2017). [arXiv:1506.01186v6](https://arxiv.org/abs/1506.01186v6)
5. Wilson, R.C., Shenhav, A., Straccia, M., et al.: The Eighty Five Percent Rule for optimal learning. *Nat. Commun.* **10**, 4646 (2019). <https://doi.org/10.1038/s41467-019-12552-4>
6. Pattanayak, S.: A mathematical approach to advanced artificial intelligence in Python. In: *Pro Deep Learning with TensorFlow* (2017). <https://doi.org/10.1007/978-1-4842-3096-1>
7. Bukhari, S.T., Mohy-Ud-Din, H.: A systematic evaluation of learning rate policies in training CNNs for brain tumor segmentation. *Phys. Med. Biol.* **66**(10) (2021). <https://doi.org/10.1088/1361-6560/abe3d3>
8. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. *IEEE CVPR* (2017)
9. Zhang, S.W., Zhang, X.X., Fan, X.N., Li, W.N.: LPI-CNNCP: prediction of lncRNA-protein interactions by using convolutional neural network with the copy-padding trick. *Anal. Biochem.* **601**, 113767 (2020). <https://doi.org/10.1016/j.ab.2020.113767>
10. Chollet, F., et al.: Keras. GitHub (2015). <https://github.com/fchollet/keras>
11. Sabottke, C.F., Spieler, B.M.: The effect of image resolution on deep learning in radiography. *Radiology Artif. Intell.* **2**(1), e190015 (2020). <https://doi.org/10.1148/ryai.2019190015>

12. Pang, S., Wang, S., Rodríguez-Patón, A., Li, P., Wang, X.: An artificial intelligent diagnostic system on mobile Android terminals for cholelithiasis by lightweight convolutional neural network. *PLoS ONE* **14**(9), e0221720 (2019). <https://doi.org/10.1371/journal.pone.0221720>
13. Lin, M., Chen, Q., Yan, S.: Network in Network (2014). <https://arxiv.org/pdf/1312.4400v3.pdf>
14. Dai, H., Cao, J., Wang, T., Deng, M., Yang, Z.: Multilayer one-class extreme learning machine. *Neural Netw.* **115**, 11–22 (2019). <https://doi.org/10.1016/j.neunet.2019.03.004>
15. Chen, J., et al.: Medical image segmentation and reconstruction of prostate tumor based on 3D AlexNet. *Comput. Methods Programs Biomed.* **200**, 105878 (2021). <https://doi.org/10.1016/j.cmpb.2020.105878>
16. Urinbayev, K., Orazbek, Y., Nurambek, Y., Mirzakhmetov, A., Varol, H.A.: End-to-end deep diagnosis of X-ray images. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference, pp. 2182–2185 (2020). <https://doi.org/10.1109/EMBC44109.2020.9175208>
17. Tu, C.-H., Chan, Y.-M., Lee, J.-H., Chen, C.-S.: Pruning depthwise separable convolutions for MobileNet compression. In: IEEE WCCL. <https://doi.org/10.1109/IJCNN48605.2020.9207259>
18. Smith, L.N., Topin, N.: Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates (2017). arXiv e-prints: [arXiv:1708.07120](https://arxiv.org/abs/1708.07120)
19. Smith, L.N.: Cyclical Learning Rates for Training Neural Networks (2015). arXiv e-prints: [arXiv:1506.01186](https://arxiv.org/abs/1506.01186)
20. Goyal, P., et al.: Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR* (2017). [http://arxiv.org/abs/1706.02677](https://arxiv.org/abs/1706.02677)
21. Zulkifli, H.: Understanding learning rates and how it improves performance in deep learning (2018). <https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10>. Accessed 23 Sep 2018
22. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Coello, C.A.C. (ed.) *Learning and Intelligent Optimization*, pp. 507–523. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-25566-3\\_40](https://doi.org/10.1007/978-3-642-25566-3_40)
23. Hyperopt Developers. hyperopt – distributed asynchronous hyperparameter optimization in python (2019). <http://hyperopt.github.io/hyperopt/>. Accessed 13 Aug 2019
24. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next generation hyperparameter optimization framework. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD 2019, pp. 2623–2631 (2019). ACM, New York, NY, USA
25. Buduma, N., Locascio, N.: *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*, p. 21 (2017). O’Reilly. ISBN: 978-1-4919-2558-4