Why Are Generative Adversarial Networks Vital for Deep Neural Networks? A Case Study on COVID-19 Chest X-Ray Images

M. Y. Shams, O. M. Elzeki, Mohamed Abd Elfattah, T. Medhat, and Aboul Ella Hassanien

Abstract  The need to generate large scale datasets from a limited number of determined data is highly required. Deep neural networks (DNN) is one of the most important and effective tools in machine learning (ML) that required large scale datasets. Recently, generative adversarial networks (GAN) is considered as the most potent and effective method for data augmentation. In this chapter, we investigated the importance of using GAN as a preprocessing stage to applied DNN for image data augmentation. Moreover, we present a case study of using GAN networks for a limited COVID-19 X-Ray Chest images. The results indicate that the proposed system based on using GAN-DNN is powerful with minimum loss function for detecting COVID-19 X-Ray Chest images. Stochastic gradient descent (SGD) and Improved Adam (IAdam) optimizers are used during the training process of the COVID-19 X-Ray images, and the evaluation results depend on loss function are determined to ensure the reliability of the proposed GAN architecture.

M. Y. Shams (✉)
Faculty of Artificial Intelligence, Kafrelsheikh University, Kafr Elsheikh 33511, Egypt
e-mail: mahmoud.yasin@ai.kfs.edu.eg

O. M. Elzeki
Faculty of Computers and Information, Mansoura University, Mansoura 35516, Egypt
e-mail: omar_m_elzeki@mans.eg

M. Abd Elfattah
Misr Higher Institute for Commerce and Computers, MET, Mansoura, Egypt
e-mail: mohabdelfatah8@gmail.com

A. E. Hassanien
Faculty of Computers and Artificial Intelligence, Cairo University, Giza, Egypt
e-mail: aboitcairo@fci-cu.edu.eg

T. Medhat
Department of Electrical Engineering, Faculty of Engineering, Kafrelsheikh University, Kafr Elsheikh 33516, Egypt
e-mail: tmedhatm@eng.kfs.edu.eg

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2020
A.-E. Hassanien et al. (eds.), Big Data Analytics and Artificial Intelligence Against COVID-19: Innovation Vision and Approach, Studies in Big Data 78, https://doi.org/10.1007/978-3-030-55258-9_9
Keywords Deep neural network · Generative adversarial network · Machine learning stochastic gradient descent (SGD) · Improved adam COVID-19 X-ray chest

1 Introduction

Deep Neural Networks (DNN) is inspired by the human biological brain consisting of neurons, synapses, and much more. Artificial neural networks (ANN) are key to building a DNN because they consist of multiple hidden layers stacked. DNN was formulated from hierarchical neural networks to improve the process of classifying supervised patterns [1]. In the process of training DNN, transfer learning is an effective and powerful tool to enable the training of large-scale datasets without over-fitting problem results from the target dataset that is much smaller than the basic dataset [2]. There are many attempts to formulate DNN, for example, multi-layer perceptron (MLP) as well as backpropagation that consists of feed-forward and feedback ANN [3]. DNN can be utilized as a feature extractor and classifier as well. However, to learn one layer of DNN feature vectors at a time, the multiple layers of feature vectors can be used as a starting point for a discriminative that is called the “fine-tuning” phase during which backpropagation through the DNN slightly adjusts the weights found in pre-training [4].

In this chapter, we utilized two different optimizers during the training step for generating COVID-19 X-Ray chest images based on GAN architecture. The first optimizer was the stochastic gradient descent (SGD), and the second was the improved Adam optimizer (IAdam) [5]. The loss function is determined for the two applied optimizers with minimum loss values. The main contribution of this chapter is to enlarge limited datasets to produce augmented COVID-19 X-Ray images with a minimum loss function. On the other hand, we prove the ability of GAN architecture using two both SGD and IAdam optimizers.

The rest of this chapter is organized as background in Sect. 2, which survey using DNN in COVID-19 X-Ray images detection and classifications. Section 3 demonstrates that the proposed methodologies include GAN architecture, SGD, and IAdam optimizers. Finally, the evaluation of experimental results is presented in Sect. 4.

2 Background

At present, researchers in the field of artificial intelligence (AI) are searching and investigating ways to treat this dangerous virus, which grows dramatically and changes a lot of data, which makes it more difficult for researchers to find the appropriate treatment for this virus [6]. The latest report of COVID-19 that updates daily.
AI and machine learning (ML) are considered as the most effective tools that are used to fight COVID-19 virus spread [6]. Rao [7] proposed a conceptual framework of data collection and possible COVID-19 identification. The collected data can be used to assist in the preliminary screening, and early identification of possible COVID-19 infected individuals since the sequence of events suggests that the coronavirus may have been transmitted by the asymptomatic carrier [8]. The incubation period for patient 1 was 19 days, which is long but within the reported range of 0 to 24 days [9].

A survey for Forecasting Coronavirus (COVID-19) Models is proposed by Shinde et al. [10]. This survey illustrates and categorizes several forecasting models available in the literature, challenges of these models, and recommendations for controlling this epidemic.

Pneumonia chest X-Ray detection based on generative adversarial networks (GAN) with a fine-tuned deep transfer learning for a limited dataset for COVID-19 is presented by Khalifa et al. [11]. They used Alexnet, Googlenet, Squeezenet, and Resnet18 were selected as deep transfer learning models with an accuracy reached to 99%. An automated infection detection system based on Computed-Tomography Scan Images of COVID-19 is proposed by Rajinikanth et al. [12] in order to compare the infection based on Infection/Lung pixel ratio.

In this chapter, we attempt to solve some critical issues that are faced by scientists in detecting and classifying COVID-19 X-Ray images. We proposed an algorithm based on Deep Convolutional Neural Networks (DCNN) that classifies the X-Ray image of infected COVID-19 persons. The proposed architecture not only classifies the Infected and uninfected people but also extracts the feature of the X-Ray images that may help extract details for the COVID-19 virus spread. Figure 1 shows the most common and recent types of DNN that is consists of unsupervised, convolutional, recurrent, and recursive neural network.

![Common types of Deep learning](image)

**Fig. 1** The most common types of DNN
On the other hand, there are further DNN architecture used, such as deep belief neural networks (DBNN) that are formulated from stacked Restricted Boltzmann machines (RBMs) [13]. Many applications utilized DNN, especially deep convolutional neural networks (DCNN) in medical, healthcare, and biomedical fields, as investigated in Fig. 2.

### 2.1 Small Data and Augmentation

Generally, some datasets are different in size and shape as well as ways to update and produce them. These datasets in the case they are small in size can be enlarged by using many methods [14]. These methods are called data augmentation. Many researchers in this field are suffering from the problem of searching for data, especially large data. There are methods, such as deep learning methods. In order for these methods to succeed, work efficiently, and to be highly efficient, it is necessary to give them a large database. Thus, people are directed to make augmentation of any data entering any system. Therefore, we will provide a method that benefits researchers to make augmentation of data so that they can deal with it later when it enters into deep learning and solves too many problems such as the problem of overfitting and memo, especially if these databases are imbalanced [15, 16].
2.2 Generative Adversarial Networks

GAN is the network that generally used for data augmentation to estimate generative models via an adversarial process. By which the trained models are simultaneously generated by capturing the data distribution, and the discriminative model for estimating the training data probability [17]. The framework of the GAN can be implemented and summarized, as shown in Fig. 3.

The Generative Adversarial Network (GAN) is a method for the training of generative models, which we briefly describe in the sense of image data. The framework pits two networks against each other: a generative model G that captures the distribution of data and a discriminating model D that distinguishes between samples taken from G and images taken from training data. Adversarial networks have opened up many new directions. Most prominent research in machine learning in the last several years, in the high-dimensional setting (like images), was focused on the discriminative side. At the same time, the need for vast amounts of data has increased as deep learning became common [18, 19].

Another type of data augmentation based on GAN is presented by Radford et al. [18] is called deep convolutional GAN (DCGAN) for unsupervised learning. They replace any pooling layers with stridden convolutions that are called discriminator and the fractional-stride convolutions that are called a generator. Moreover, Makhzani et al. [19] proposed adversarial auto-encoders (AAE) which as an extension to GAN that perform variational inference for the aggregated posterior of the hidden code vector of the autoencoder with an arbitrary prior distribution. They used AAE for dimensionality reduction of the input features. Information maximizing generative adversarial networks (InfoGAN) is an approach presented by Chen et al. [20]. This approach can learn disentangled representations in a completely unsupervised manner.

![Fig. 3 General framework of generative adversarial network](image-url)
3 Methodologies

A generated of augmented images are required to improve the architecture of deep neural networks, especially convolutional neural networks (CNN). Due to the limited dataset of the COVID-19 X-Ray images that are maybe imbalance and variable data, we present in this work GAN as an augmentation process. Figure 4 shows the proposed architecture of the generative stage of GAN by which we train CNN to produce augmented images. On the other hand, the discriminator of the GAN architecture takes the $64 \times 64$ X-Ray COVID-19 image results from the generator and the training sets. This process is typical and inspired by the decoder process in GAN is a discriminator, and the encoder is the generator in the communication channel.

During the augmentation process, two different optimization techniques have been performed, the first is the utilization of SGD optimizer as a preprocessing stage for the GAN entire network, while, the second is based on Adam optimizer. The two optimization techniques are applied to prove the ability of the proposed system to generate and manipulate the COVID-19 X-Ray images. In this section, a brief demonstration of the mentioned optimizers is presented.

3.1 Stochastic Gradient Decent (SGD) Optimizer

In SGD in order to minimize the computation time per iteration, an adaptive step size to estimate the most important details of COVID-19 X-Ray images is performed based on the measure of the similarity region in the image content and the transformation model as illustrated by Klein et al. [21]. In this case, the applied images learned by GANs in the training phase can produce a training loss with global minima based
on ReLU network. The initialization of SGD produces a sequence of iterations inside a small perturbation region that is around the initial centred weights as investigated by Zou et al. [22]. In this chapter, we present SGD for the trained X-Ray images in normal and COVID-19 cases. The weights are updated on each training image, and there is no need to perform the batch operation as a whole like in GD. The computational time of SGD is lower than traditional GD because the training process does not execute through the whole training set to update the weights. On the other hand, the SGD minimizes loss faster and produce noisy and variation accuracy and loss. In order to overcome the limitation of SGD, we utilized an adaptive momentum estimation (Adam) optimizer as investigated in the forthcoming section.

### 3.2 Improved Adam Optimizer

The non-convex nature of the optimization problem, as well as the need to design a powerful and reliable deep neural network, are a significant challenge, especially for the systems that require faster performance [23]. Bock et al. [24] proposed an improvement of Adam optimizer based on adaptive step size by changing the weights and momentum to attain the convergence of the applied neural networks. Moreover, Bock and WeiB [25] prove that Adam has a local convergence and posterior boundary for the hyper-parameters of the applied network. Not only, determining the local convergence in Adam helpful for optimization but also the improvement in the speed of convergence is required to achieve the minimum loss function. However, on the other hand, it will require more memory with a high complexity [26]. Therefore, in this work, we used the normalized preserving Adam algorithm presented by Zhang [5], IAdam. To maintain the gradient direction for each weight vector and produce more accurate weight decay with a minimum elapsed time.

### 3.3 Training and Generating the GAN

For building a GAN model to generate X-ray images, first, we need to build the structures of the network, including the layers of the generator unit and discriminator unit.
Next, we specify the different values of the hyper-parameters as training options including epochs, iterations, batch size and learning rates. Algorithm 1 represents the pseudocode algorithm for training and building the GAN model to generate X-Ray images based on understanding the given sample image dataset. The algorithm starts by setting up the environment options and training options. Next, the generator and discriminator models are initially constructed with initial weights randomly are setting up. The training process starts using sample images of the given dataset per epoch. During the same epoch, the training process repeats for every batch in the dataset doing the subsequent procedures. These procedures include loading the sample images, training the generator and optimize the learned generator using the Adam optimizer or the standard SGD. Also, these procedures include training the discriminator and optimize its behaviour using the same listed optimizers recently. Finally, the training procedures adjust the weights according to the hybrid activation function that is powered by sigmoid and cross-entropy function. The training process is repetitive for a specified number of iterations. The algorithm is customized and enhanced using the IAdam optimizer.

---

### Algorithm 1: Build a GAN model to generate X-Ray images

| Input | Dicremenaror_layers discLay, generator_layers genLay, hyper-parameters opts, sample_images imgSet |
|-------------------------------|------------------------------------------------------------------------------------------------|
| Output | GAN |

1. Begin
2. env ← setup(opts)
3. gen ← NNbuild(genLay)
4. disc ← NNbuild(discLay)
5. For e=1: env.Epoch
6. m ← rnd(num(imgSet))
7. For i=1: env.NBatch
8. imgs ← read(imgSet,m)
9. gen ← train(imgs,nosie)
10. // Optimize Generator using Adam optimizer instead of standard SGD
11. gen ← adamOptimizer(gen,disc,imgFakes)
12. disc ← train(imgs,nosie,imgFakes)
13. // Optimize Generator using IAdam optimizer instead of standard SGD
14. disc ← adamOptimizer(gen,disc,imgFakes)
15. // Calculate C-Loss & D-Loss
16. C_loss ← sigmoid_cross_entropy(expected,actual)
17. D_loss ← sigmoid_cross_entropy(expected,actual,c_loss)
18. End for
19. End for
20. GAN ← {gen,disc}
21. End
Why Are Generative Adversarial Networks Vital …

| Table 1 | Hyperparameter values of the proposed GAN architecture |
|---------|--------------------------------------------------------|
| Parameter          | Value               |
| Batch size        | 16                  |
| Iteration         | 100                 |
| Learning rate     | 0.001               |
| Optimizer name    | SGD and Adam        |

4 Experimental Results: Discussion and Analysis

Of course, for natural images, GANs do not currently generate very realistic images, but this work has sparked interest in generative models for natural images and has sort of created a new subarea in deep learning, which many research groups today are actively looking for it. So, algorithms that can make and generate data themselves can handle our real-world problems more efficiently.

The main objective of this section is to summarize and provide detailed results of the proposed architectures such that in order to enlarge the training data, we augmented the data using GANs by which the new examples learned from data is generated.

4.1 Dataset Description and Experiments Setup

In this chapter, we used COVID-19 X-ray image collections that were established in [27], which contain 50 images, 25 normal and the other 25 are COVID-19 states. Data sets are enlarged based on the GAN architecture. The images generated are 1000 images that result in a 20-fold GAN increase from 50 images. All experiments were performed using the software package (MATLAB 2020a), using a PC with the following features Core i7, 16-RAM. The hyper-parameters are shown in Table 1 during all experiments.

4.2 Experiments Scenarios

Two different experiments are carried out to compare the proposed approaches. The first experiment is powered by SGD optimizer, while the second experiment based on Improved Adam optimizer. In this chapter, the mean value of the trained images was subtracted from each image fed into GANs. The evaluation is based on two loss function, which is called C-Loss and D-Loss. The C-Loss is the cross-entropy loss and in some cases named as a logarithmic loss. In this type of loss, the predicted probability is compared to the output value (0 or 1), and the result is calculated based on the distance from the expected value. The Cross-entropy for the predicted class
prediction is calculated as the average cross-entropy across all examples. Furthermore, the second type of measurement in this chapter is the D-Loss which is the generator loss. Such that each generator tries to maximize this function and attempts to maximize the discriminator’s output for its instances.

4.2.1 First Experiment: Optimizing Using SGD Algorithm

Using the standard optimizer SGD generates the loss rates already visualized in Fig. 5 representing the C-loss and D-Loss. Based on the analytical investigation and visual inspection, we noticed the increment of C loss function and on the opposite side D loss decrease. The notice can be justified due to the cross-entropy of the input images which produce an enhancement at the same time in the D loss. The contradiction of loss rate in C loss and D loss leads to optimizer enhancement problem. The histogram of both C loss and D loss is shown in Figs. 6 and 7 respectively. In the histogram, the x-axis represents the loss value, and the y-axis represents the frequency value. Since the total number of iterations was 100 iterations, every frequency is less than

![Fig. 5 C-loss versus D-loss in SGD](image)

![Fig. 6 Histogram of C-loss value](image)
or equal to 100 per loss value. In Fig. 6, the error rates of C loss are binned into six bins each bin has 0.2 as the next step of the previous bin. The binning starts at 0.6 and ends at 1.8. So, the minimum error of C loss is 0.6, and the maximum error of C loss is 1.8. From Fig. 6 the C loss reaches up to 1.6, with a 35 as the frequency, which means C loss is higher using SGD optimizer. On the other hand, the error rates of D loss are binned into ten bins each bin has 0.05 as the next step of the previous bin. The binning starts at 0.2 and ends at 0.7. So, the minimum error of D loss is 0.2, and the maximum error of D loss is 0.7. From Fig. 7, the D loss scored loss rate in $[0.25, 0.3)$ with a 40 as the frequency which leads to the compatibility of the discriminator with SGD optimizer. The first experiment represents the research gap that motivates us to improve using Adam optimizer is helpful for us to determine which optimizer we need. In turn, the second experiment based on Adam optimizer to boost and ensure the accuracy of the proposed architecture. Figure 8 represents a montage of the generated samples for the proposed GAN architecture based on the SGD optimizer.

### 4.2.2 Second Experiment: Optimizing Using Adam Algorithm

As mentioned in Sect. 3.2, Adam optimizer is utilized for evaluating both C and D losses. Instead of evaluating the gradient of the current position as performed in the SGD, we used Adam optimizer to maintain the gradient direction for each weight vector. Figure 9 represents the comparative analysis of C loss verse the D loss of the proposed GAN using Adam optimizer. After 15 iterations, we noticed the harmony of loss rates where can be considered tightly reduced over iterations progressive. The Adam optimizer enhances the performance through auto-adjusting of the weights over the iteration progress in the early stage of training. The Adam optimizer tolerates the C loss rate almost in the third iteration to be correlated with D loss. In the same manner, histograms are shown in Figs. 10 and 11 for evaluating the error rates verse the frequencies.
Fig. 8  Sample of generated images using SGD optimizer in GAN architecture

Fig. 9  C-loss vs D-loss in Adam
Fig. 10  Histogram of C-loss value

Fig. 11  Histogram of D-loss value

In Fig. 10, the error rates of C loss are binned into five bins each bin has 2.0 as the next step of the previous bin. The binning starts at 0.0 and ends at 26.0. So, the minimum error of C loss is 0.0, and the maximum error of C loss is 26.0. From Fig. 10 the C loss reaches up to 2, with a 65 as the frequency and up to 4 with a 31 as the frequency, which means most of C loss is enclosed between [0,4) with 96 as frequent of total 100 using Adam optimizer. On the other hand, the error rates of D loss are binned into five bins each bin has <0.25 as the next step of the previous bin. The binning starts at 0.0 and ends at 4.5. So, the minimum error of D loss is 0.0, and the maximum error of D loss is 4.5.

From Fig. 11, the D loss scored a loss rate in [0.0, 0.80) with a 93 as the frequency, which leads to higher compatibility of the discriminator with Adam optimizer. We used the proposed GAN architecture with Adam optimizer to generate montage of the generated samples in Fig. 12. In turn, we can conclude the following points:

- Adam optimizer is better than the SGD optimizer, in enhancing the C-Loss and D-Loss together for X-ray images, especially for COVID-19 according to the case study.
The pros of Adam optimizer depend on considering the bias, learning rate, and heuristic of the weights verse the learning rate only as SGD behaves.

We believe that this approach can be generalized to solve and classify other medical classification applications for the improvement of diagnosis.

5 Conclusions

Generative adversarial network (GAN) is proved to be a very effective model for training Generative models. There are several reasons that GANs are essential for deep neural networks DNN: First, a generative model for natural images that evolve to generate more and more realistic looking data, because of the coupling with an adversarial network. Secondly, in principle, when you do not have enough data for understanding a problem, GANs can be used to generate more data rather than using tricks like data augmentation. A GAN architecture is proposed, using two different optimizers. The first is based on the SGD optimizer by which the trained images are evaluated by using both C loss and D loss. An increase of C loss occurred in the SGD optimizer. Therefore, Improved Adam optimizer is utilized as a second experiment used to enhance the results of SGD. The proposed architecture is promising and can
be utilized in other medical applications especially in the image processing field. In the future, we plan to use a Region of interest feature extractor instead of the whole image in order to extract more details for medical images.

References

1. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3642–3649. IEEE (2012)
2. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks?. In: Advances in Neural Information Processing Systems, pp. 3320–3328 (2014)
3. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.-R., Jaitly, N., Senior, A. et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Processing Magazine, Vol. 29, No. 6, pp. 82–97 (2012)
4. Hinton, G.E., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)
5. Zhang, Z.: Improved Adam optimizer for deep neural networks. In: 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), pp. 1–2. IEEE (2018)
6. Alimadadi, A., Aryal, S., Manandhar, I., Munroe, P.B., Joe, B., Cheng, X.: Artificial intelligence and machine learning to fight COVID-19, pp. 200–202 (2020)
7. Rao, A.S.S., Vazquez, J.A.: Identification of COVID-19 can be quicker through artificial intelligence framework using a mobile phone–based survey when cities and towns are under quarantine. Infection Control Hospital Epidemiology 41.7, 826–830 (2020)
8. Bai, Y., Yao, L., Wei, T., Tian, F., Jin, D.-Y., Chen, L., Wang, M.: Presumed asymptomatic carrier transmission of COVID-19. JAMA 323(14), 1406–1407 (2020)
9. Guan, W.-J., Ni, Z.-Y., Hu, Y., Liang, W.-H., Ou, C.-Q., He, J.-X., Liu, L. et al.: Clinical characteristics of 2019 novel coronavirus infection in China. MedRxiv (2020)
10. Mahalle, P., Kalamkar, A.B., Dey, N., Chaki, J., SHinde, G.R.: Forecasting models for coronavirus (covid-19): a survey of the state-of-the-art (2020)
11. Khalifa, N.E.M., Taha, M.H.N., Hassanien, A.E., Elghamrawy, S.: Detection of coronavirus (COVID-19) associated pneumonia based on generative adversarial networks and a fine-tuned deep transfer learning model using chest X-ray dataset. arXiv preprint: arXiv:2004.01184 (2020)
12. Rajinikanth, V., Dey, N., Raj, A.N.J., Hassanien, A.E., Santosh, K.C., Raja, N.: Harmony-search and otsu based system for coronavirus disease (COVID-19) detection using lung CT scan images. arXiv preprint: arXiv:2004.03431 (2020)
13. Shams, M.Y., Sarhan, S.H., Tolba, A.S.: Adaptive Deep Learning Vector Quantisation for Multimodal Authentication, Vol. 8, No. 3, pp. 702–722 (2017)
14. Mun, S., Park, S., Han, D.K., Ko, H.: Generative adversarial network based acoustic scene training set augmentation and selection using SVM hyper-plane. In: Proc. DCASE, pp. 93–97 (2017)
15. Salehinejad, H., Valaei, S., Dowdell, T., Colak, E., Barfett, J.: Generalization of deep neural networks for chest pathology classification in x-rays using generative adversarial networks. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 990–994. IEEE (2018)
16. Shamsolmoali, P., Zareapoor, M., Shen, L., Sadka, A.H., Yang, J.: Imbalanced Data Learning by Minority Class Augmentation using Capsule Adversarial Networks. arXiv preprint arXiv:2004.02182 (2020)
17. Mirza, M., Osindero, S.: Conditional Generative Adversarial Nets. arXiv preprint arXiv:1411.1784 (2014)
18. Radford, A., Metz, L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv preprint arXiv:1511.06434 (2015)
19. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)
20. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: interpretable representation learning by information maximizing generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2172–2180 (2016)
21. Klein, S., Pluim, J.P.W., Staring, M., Viergever, M.A.: Adaptive stochastic gradient descent optimization for image registration. Int. J. Comput. Vis. 81(3), 227 (2009)
22. Zou, D., Cao, Y., Zhou, D., Gu, Q.: Stochastic gradient descent optimizes over-parameterized deep relu networks. arXiv preprint arXiv:1811.08888 (2018)
23. Bello, I., Zoph, B., Vasudevan, V., Le, Q.V.: Neural optimizer search with reinforcement learning. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 459–468. JMLR.org (2017)
24. Bock, S., Goppold, J., Weiß, M.: An improvement of the convergence proof of the ADAM-optimizer. arXiv preprint arXiv:1804.10587 (2018)
25. Bock, S., Weiß, M.: A proof of local convergence for the Adam optimizer. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2019)
26. Tato, A., Nkambou, R.: Improving Adam Optimizer (2018)
27. https://github.com/smfai200/Detecting-COVID-19-in-X-ray-images/tree/master/dataset. Accessed in 21 May 2020