Assessment of Optimizers impact on Image Recognition with Convolutional Neural Network to Adversarial Datasets

Vidushi¹, Manisha Agarwal², Akash Rajak³, Ajay Kumar Shrivastava⁴

¹Research Scholar, Banasthali Vidyapith, Rajasthan- 304022, India, and Asst. Professor, KIET Group of Institutions, Delhi-NCR, Ghaziabad-201206, India.
²Banasthali Vidyapith, Rajasthan- 304022, India.
³KIET Group of Institutions, Delhi-NCR, Ghaziabad-201206, India.
⁴KIET Group of Institutions, Delhi-NCR, Ghaziabad-201206, India.

¹vidushi.mtech@gmail.com, ²manishaagarwal18@yahoo.co.in, ³akashrajak@gmail.com, ⁴ajay@kiet.edu.

Abstract. In Artificial Intelligence, the machine modeling technique means to behave in the manner of human reflects indistinguishable. To automatizes the development of rational model for data evaluation, machine learning mechanism of artificial intelligence, is used. Deep learning is the machine learning discipline, having objective to imbide the system and to discover pattern from input. In pattern recognition, deep learning has paramount importance and different advance powerful model’s architecture. The most effective, vital, and influential innovation in computer vision discipline is one of the architectures of deep learning called convolutional neural network. In this neural network various optimizers can be used for model molding into its appropriate form by weights futzing. Aiming to overcome the problem in getting the optimized result, the research used various algorithms of weight optimization. The article elaborates convolutional network concept as well as the idea behind the use of optimizers. Furthermore, the detailed study of optimizers is also presented in this paper. Along with it, experimental comparison and result of different learning paradigm optimizers is shown in the document. Considering the different image datasets, including MNIST, and CIFAR 10 dataset, the accuracies of convolutional model with different optimizers are verified.

Keywords- Machine learning, Deep Learning, Convolutional Neural Network, Optimization algorithm, Classification, Adadelta, Gradient Decent, Adam, RMSProp, Stochastic Gradient Decent.

1. Introduction

Deep learning [1] inspired from brain simulators, the most acknowledging, promising and advance machine learning methodology that brings the breakthrough achievement for interpreting and classifying the image data. As compared with diverse state-of-art models of machine learning, deep learning is achieving uppermost and leading success in heterogeneous fields. Deep learning consists of well-built and strong models claiming outstanding achievement in resolving issues related to pattern recognition. The successful extraction as well as selection of feature is the specialty of this network that makes deep learning more powerful. In order to learn accurately, the different architectures available in literature of deep learning model consist of numerous processing layers. Among other architectures, one powerful and successful one is convolutional neural network [2]. To learn accurately, along with the input-output and processing layers, convolutional, and pooling layers are also present. The operations of convolution as well as pooling processed the input layer. The operations of convolution include the function of rectified relu, slide, and shift. For converting the convolution neurons layers output to the identical kernel map, pooling layer is used [3]. The combination of convolution-pooling-fully connected layers can be repeated ‘n’ number of times, where ‘n’ is any positive integer. This complete working of all the layers makes the convolutional network more
efficient and powerful [3]. To optimize the result, the adjustment of weights iteratively is performed during backpropagation process [4]. There are many optimization algorithms present that adjust the weights efficiently [5]. Because of these optimization algorithms, the different neural network models are able to perform well with good accuracy [5].

This work contribution is towards the image recognition with convolutional neural network using different optimization algorithms. The fruitful and related past work done by researchers is present in section 2. However, section 3 describes convolutional neural network and also elaborates different optimization algorithms. Further, the experimental analysis with result discussion of diverse optimization algorithms with convolution network on adversarial image datasets is shown in section 4. Moreover, section 5 presents the conclusion and future work.

2. Background and Related Work

Yixiang et al. [5] analyzes the model sensitivity with not only the attack setups for white box and black box aspect but also with different datasets types aspects. The well known four optimizers called SGD, Adam, RMSProp, and Adadelta are judged and properly investigated using the unstructured as well as structured datasets. In case of structured datasets, Adam optimizer provides comparatively better quality of adversarial examples. Along with it, Adadelta optimizer with unstructured dataset provides comparatively better quality of adversarial examples. Additionally, adversarial examples transferability is not affected by optimizer choice. Jiandong et al. [6] proposed the GRU model and algorithm for GPS map-matching that meet the simultaneously required efficiency and accuracy. With the purpose of weight optimization, the paper used diverse optimization algorithm such like SGD, Adam, Adadelta, and RMSprop with the GRU model. The document verified the proposed method accuracies in different four scenarios that include workday, rainy, weekend, as well as accident.

The paper [7] researchers’ approach is towards learning the cost function that explains human motion. To perform it, the study gathers trajectories examples from participants pair that use capture motion to perform a task for collaborative assembly. Further, the research for learning cost function from the trajectories utilizes the inverse of optimal control. An adaptive localization is proposed in this study [8] by using RMSprop method for changing the gains from gradient decent that is adaptive between the distance of beacons and mobile node, thus improved and better convergence speed is given and stable localization is provided. The document [9] uses the CNN model on cats-dogs dataset in order to perform the classification task successfully. The paper uses different optimizers with CNN model and evaluate these optimizers performance by changing the number of epochs or iterations and corresponding learning rate. The paper tested using CPU and also the performance of optimizers is computed with respect to accuracy, along with error rate. Finally, the research result shows the performance of momentum optimizer with five epochs is much better than others.

The researchers [10] of this study compare the seven different well known optimizers by implementing them in CNN. To perform this experiment, the paper uses the dataset called Indian Pines and show the result graphically with Adamax 99.58 percentage. Arwa et al. [11] shows the experimental result with tensorflow CNN and on that basis shows the comparative result using Alzheimer dataset. In order to boost up the process data augmentation as well as multi optimizers contribution is used. While comparing the different optimizers, the paper experimental result shows Adam optimizer accuracy 95.8 percent and on the other side 100% accuracy achieved with RMSProp optimizer. The research [12] proposes PSO new variant called SALMPSO, which stands for Symbiosis-based Alternative Learning, inspired with mutual cooperation idea of symbiosis present for natural ecosystem. The study result shows the better performance of SALMPSO model with respect to convergence speed as well as optimal values.

3. Convolutional Neural Network (CNN), and Datasets

CNN is an algorithm with excellent performance in image processing [13]. It belongs to deep learning category [13]. Neurons with their connectivity and visual cortex present in human brain becomes inspiration for this model. Human brain part called cerebral cortex has a region named visual cortex. Similar to visual cortex, CNN also has multiple layers. CNN architecture consists of not only input-processing-output layers but also has convolution-pooling layers. Every layer is having neurons which are connected with other layer neurons [14],[15]. The responsibility of convolutional layer is to convolve the input and passing the generated output to other upcoming layers. High-level features extraction is the objective lies with this layer from image input [16]. Pooling layer in this architecture is responsible for dimension reduction and this
layer outcome goes to the next layer called fully connected layer. Because of all the layers that CNN has, it becomes the powerful one which can resolve hard problems. That’s why, it achieved remarkable progress and success. Visual imagery analysis is the main target of this architecture [17]. In this article, CNN model is implemented using two datasets named MNIST, and CIFAR-10.

Handwritten dataset having digits from 0 to 9 is MNIST [18]. This dataset has total 70 thousand images, in which 60,000 are present in training part and rest 10,000 present in testing part. In this, digits are represented in various ways [19]. Diversity in digits is present due to diversity in handwriting style [20]. Understanding different formats of digits is a typical task for a machine. Convolution model [21] is used with this dataset in this article.

This document also uses CIFAR 10 dataset [22],[23]. This dataset has ten variety of classes where each class is having 1000 images. All these 10 classes are completely mutually exclusive. In this, 50,000 images belong to the training part and rest 10,000 images belongs to testing class. Image recognition is the toughest job for a machine. This study also applies CNN model using this dataset. Table 1 shows the brief description of used dataset with Convolutional Network.

| Dataset Used | Epochs | Dataset Size | Training Sample | Testing Sample |
|--------------|--------|--------------|-----------------|----------------|
| CIFAR 10     | 100    | 60,000       | 50,000          | 10,000         |
| MNIST        | 12     | 70,000       | 60,000          | 10,000         |

### 4. Optimization Algorithms

The way to optimize the neural network model’s result is termed as optimization. To accomplish this task various optimizers or optimization algorithms are available. The methods whose purpose is to get more accurate result by adjusting again and again neural network parameters like learning rate, weights, etc. Aim of optimization is to achieve better result. This research is using CNN model with different optimizers to get better result in terms of accuracy. Brief detail of several optimizers is shown in Table 2.

| Optimizer Name            | Reference | Definition | Equation | Disadvantage                                                                 | Advantage                                      |
|---------------------------|-----------|------------|----------|------------------------------------------------------------------------------|-----------------------------------------------|
| Gradient Descent          | [24], [25]| Basic first-order algorithm of $W_{\text{new}} = W_{\text{old}} - \frac{-\partial L}{\partial W_{\text{old}}}$ | Local minima problem exists. It needs large memory for Computation, implementation, and understanding | Computation, implementatio n, and understanding |

Table 2. Comparative Study of CNN Optimizers
| Optimizer Type                  | Description                                                                 | Calculating the gradient on complete dataset. | Parameters are simple and easy. |
|--------------------------------|-----------------------------------------------------------------------------|-----------------------------------------------|---------------------------------|
| Stochastic Gradient Descent    | This optimizer is a gradient descent variant that updates the parameters frequently. | $W_{\text{new}} = W_{\text{old}} - \alpha \cdot \nabla J(W_{\text{old}}; x(i); y(i))$ | Parameters have high variance.  |
|                               |                                                                            |                                               | Parameters are updated frequently, so needs less convergence time. |
| Mini-Batch Gradient Descent    | This optimizer divides the dataset into batches and for each batch updates the parameters. | $W_{\text{new}} = W_{\text{old}} - \alpha \cdot \nabla J(W_{\text{old}}; B(i))$ | Learning rate selection to optimize the value is still a challenge. |
|                               |                                                                            |                                               | Parameters are updated frequently, so it faces less variance. |
| Momentum                      | This convergence is accelerated in the appropriate direction. Along with it, | $V(t) = \gamma V(t - 1) + \alpha \cdot \nabla J(W_{\text{old}})$ | It has extra hyper-parameter.   |
|                               |                                                                            |                                               | Parameters high variance is reduced by this optimizer. |
| Method           | Authors | Description                                                                 | Formula                                                                 | Parameter Selection Needs to Be Done Manually | Local Minima Are Not Missed by This Optimizer | Computational Rate Is Not Needed |
|------------------|---------|------------------------------------------------------------------------------|-------------------------------------------------------------------------|-----------------------------------------------|-----------------------------------------------|----------------------------------|
| Nesterov Accelerated Gradient | [31], [32] | It is also named look-ahead method.                                          | $V(t) = \gamma V(t - 1) + \alpha \nabla J(W_{old} - \gamma V(t - 1))$ | Hyper-parameter selection needs to be done manually. | Local minima are not missed by this optimizer. | Manual tuning is not needed. |
| AdaGrad          | [33], [34] | This optimizer is able to perform the changes in the learning rate.         | $W_{new} = \overbrace{W_{old} - \alpha \cdot g_{t,i}}^{\sqrt{G_{t,ii}} + \epsilon}$ | It is computationally expensive.              | Manual tuning is not needed.                  |                                  |
| AdaDelta         | [35], [36] | This optimizer is adagrad optimizer extension, which limits the accumulate past gradient windows to fixed size. | $E[g^2](t) = \gamma \cdot E[g^2](t - 1) + (1 - \gamma) \cdot g^2(t)$ | It has computation expansion.               | There is no decay of learning rate, training not stop. |                                  |
Adam [37], [38] It works with both first as well as second order momentum. $W_{\text{new}} = W_{\text{old}} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$ It has high computation cost. This method is fast. It also able to converges rapidly.

5. Experimental Analysis and Result Discussion
This research article uses MNIST, and CIFAR 10 datasets to perform experiments using CNN model with different optimizers. Different optimizers comparative study with their result in reference of accuracy and loss is done in this paper. Table 3. shows optimizer’s gained outcome in terms of accuracy and loss while implemented on CNN model.

Table 3. Optimizers Accuracy & Loss Comparison during Training on image datasets with CNN model

| Optimizer                          | MNIST         | CIFAR 10       |
|-----------------------------------|---------------|----------------|
|                                   | Accuracy      | Loss           | Accuracy      | Loss           |
| Stochastic Gradient Descent       | 0.9446        | 0.1824         | 0.835         | 0.4629         |
| Gradient Descent Momentum         | 0.9469        | 0.1797         | 0.853         | 0.4165         |
| Gradient Descent Nesterov         | 0.9688        | 0.1043         | 0.9646        | 0.1013         |
| accelerated gradient              |               |                |               |                |
| RMSProp                           | 0.9875        | 0.0421         | 0.7599        | 0.8199         |
| AdaDelta                          | 0.9932        | 0.0224         | 0.866         | 0.4475         |
| AdaGrad                           | 0.9962        | 0.0114         | 0.1001        | 14.5047        |

Table 3 shows training accuracy & loss comparison during training of optimizers. In Graphical form, optimizers comparison is represented in Figure 1.

Figure 1. Optimizers Training Accuracy and Loss Comparison on adversarial image datasets using CNN model
Figure 1 represents the graphical comparison of optimizers with their respective accuracy & loss during training. Similarly, Table 4 & Figure 2 shows the optimizers comparison during testing phase of CNN model in reference of accuracy and loss.

Table 4. Optimizers Validation/Testing Accuracy and Loss Comparison on adversarial image datasets using CNN model.

| Optimizer                        | MNIST       | CIFAR 10     |
|----------------------------------|-------------|--------------|
|                                  | Accuracy    | Loss         | Accuracy    | Loss         |
| Stochastic Gradient Descent      | 0.9718      | 0.0934       | 0.7636      | 0.6976       |
| Gradient Descent Momentum        | 0.9713      | 0.0937       | 0.7806      | 0.6551       |
| Gradient Descent Nesterov        | 0.9837      | 0.0539       | 0.8048      | 0.7776       |
| accelerated gradient             |             |              |             |              |
| RMSProp                          | 0.9898      | 0.0363       | 0.7         | 0.8627       |
| AdaDelta                         | 0.9931      | 0.0271       | 0.8234      | 0.5817       |
| AdaGrad                          | 0.9933      | 0.0256       | 0.1         | 14.5063      |

Table 4 shows optimizers comparison during validation phase using CNN model. Graphically this comparison is shown in Figure 2.

Figure 2. Optimizers Validation/Testing Accuracy and Loss Comparison on adversarial image datasets using CNN model.

The optimizers’ potential comparison in accuracy and loss terms during validation phase on CNN model is shown in Figure 2. It can be clearly observed in above comparative tables and figures that AdaDelta is performing good in both datasets used in this research. In MNIST dataset AdaDelta and AdaGrad optimizers performance is almost same but in other dataset called CIFAR 10, AdaDelta shows comparatively better performance. The difference between AdaDelta and Gradient descent based optimizers is the learning rate. In gradient based optimizers learning rate remain constant and in AdaDelta learning rate changes as per the requirement. And the difference between AdaDelta with RMSProp and AdaGrad is delta that refers to the difference between the current weight and the newly updated weight. Due to these mentioned reasons this optimizer shows good result comparatively to others.

6. Conclusion and Future Work

The paper presents the convolutional network and adversarial image datasets concept and detail study of different optimizers. Further, the comparative study and analysis of optimization algorithms on CNN model is shown through tables and figures in this article. It concluded that the AdaDelta which is an adaptive optimizer performed well in all the image datasets. This work can be utilized by the scientists.
who are working in the computer vision and particularly in image recognition field. The combination of convolutional network with AdaDelta optimizer results well is proved through the experiment analysis work. The output can be utilized in future to give more fruitful results.

References

[1] Rathore H, Al-Ali AK, Mohamed A, Du X, Guizani M. A novel deep learning strategy for classifying different attack patterns for deep brain implants. IEEE Access. 2019 Feb 20;7:24154-64.
[2] Xin R, Zhang J, Shao Y. Complex network classification with convolutional neural network. Tsinghua Science and technology. 2020 Jan 13;25(4):447-57.
[3] Liang F, Shen C, Wu F. An iterative BP-CNN architecture for channel decoding. IEEE Journal of Selected Topics in Signal Processing. 2018 Jan 15;12(1):144-59.
[4] Wang Y, Liu J, Mišić J, Mišić VB, Lv S, Chang X. Assessing optimizer impact on dnn model sensitivity to adversarial examples. IEEE Access. 2019 Oct 21;7:152766-76.
[5] Zhao J, Gao Y, Yang Z, Li J, Feng Y, Qin Z, Bai Z. Truck traffic speed prediction under non-recurrent congestion: Based on optimized deep learning algorithms and GPS data. IEEE Access. 2019 Jan 1;7:9116-27.
[6] Mainprice J, Hayne R, Berenson D. Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces. IEEE Transactions on Robotics. 2016 Jul 27;32(4):897-908.
[7] Park SH, Bae YB, Fidan B, Ahn HS. Distance-based Mobile Node Localization of Fixed Beacons Using RMS Prop. In2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE) 2019 Sep 10 (pp. 376-381). IEEE.
[8] Shaf A, Ali T, Farooq W, Javaid S, Draz U, Yasin S. Two classes classification using different optimizers in convolutional neural network. In2018 IEEE 21st International Multi-Topic Conference (INMIC) 2018 Nov 1 (pp. 1-6). IEEE.
[9] Lei X, Pan H, Huang X. A dilated CNN model for image classification. IEEE Access. 2019 Jul 8;7:124087-95.
[10] Chua LO, Roska T. The CNN paradigm. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications. 1993 Mar;40(3):147-56.
[11] Arena P, Fortuna L, Occhipinti L. A CNN algorithm for real time analysis of DNA microarrays. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications. 2002 Aug 7;49(3):335-40.
[12] Zarányi Á, Orzó L, Grawes E, Werblin F. CNN-based models for color vision and visual illusions. IEEE transactions on circuits and systems I: Fundamental theory and applications. 1999 Feb;46(2):229-38.
[13] Ahmed SB, Hameed IA, Naz S, Razzaq MI, Yusoof R. Evaluation of handwritten Urdu text by integration of MNIST dataset learning experience. IEEE access. 2019 Oct 9;7:153566-78.
[14] Mohapatra RK, Majhi B, Jena SK. Classification performance analysis of mnist dataset utilizing a multi-resolution technique. In2015 International Conference on Computing, Communication and
Security (ICCCS) 2015 Dec 4 (pp. 1-5). IEEE.

[20] Garg A, Gupta D, Saxena S, Sahadev PP. Validation of random dataset using an efficient CNN model trained on MNIST handwritten dataset. In 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN) 2019 Mar 7 (pp. 602-606). IEEE.

[21] Kyriakos A, Kitsakis V, Louropoulos A, Papatheofanous EA, Patronas I, Reisis D. High performance accelerator for cnn applications. In 2019 29th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS) 2019 Jul 1 (pp. 135-140). IEEE.

[22] Doon R, Rawat TK, Gautam S. Cifar-10 classification using deep convolutional neural network. In 2018 IEEE Punecon 2018 Nov (pp. 1-5). IEEE.

[23] Bankman D, Yang L, Moons B, Verhelst M, Murmann B. An Always-On 3.8J/μs J/86% CIFAR-10 Mixed-Signal Binary CNN Processor With All Memory on Chip in 28-nm CMOS. IEEE Journal of Solid-State Circuits. 2018 Oct 3;54(1):158-72.

[24] Wadayama T, Nakamura K, Yagita M, Funahashi Y, Usami S, Takumi I. Gradient descent bit flipping algorithms for decoding LDPC codes. In 2008 International Symposium on Information Theory and Its Applications Dec 7 (pp. 1-6). IEEE.

[25] Vidushi, Agarwal M, Khamparia A, Khatoon N. Wireless Environment Security: Challenges and Analysis Using Deep Learning. In Proceedings of International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT) 2019 Sep 27 (Vol. 1, pp. 1-4). IEEE.

[26] Li J, Li X, Zhao L. Hyperspectral unmixing via projected mini-batch gradient descent. In 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) 2017 Jul 23 (pp. 1133-1136). IEEE.

[27] Liu Y, Huangfu W, Zhang H, Long K. An efficient stochastic gradient descent algorithm to maximize the coverage of cellular networks. IEEE Transactions on Wireless Communications. 2018 May 7;18(7):3424-36.

[28] Liu Y, Huangfu W, Zhang H, Long K. An efficient stochastic gradient descent algorithm to maximize the coverage of cellular networks. IEEE Transactions on Wireless Communications. 2019 May 7;18(7):3424-36.

[29] Liu Y, Huangfu W, Zhang H, Long K. An efficient stochastic gradient descent algorithm to maximize the coverage of cellular networks. IEEE Transactions on Wireless Communications. 2019 May 7;18(7):3424-36.

[30] Liu Y, Huangfu W, Zhang H, Long K. An efficient stochastic gradient descent algorithm to maximize the coverage of cellular networks. IEEE Transactions on Wireless Communications. 2019 May 7;18(7):3424-36.

[31] Liu Y, Huangfu W, Zhang H, Long K. An efficient stochastic gradient descent algorithm to maximize the coverage of cellular networks. IEEE Transactions on Wireless Communications. 2019 May 7;18(7):3424-36.