Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 5
AIDML 2024

The influence of network structure on different gradient descent optimization algorithms

Lingyun Yan

Engineering Faculty The University of New South Wales (UNSW) Sydney, Australia

lingyun.yan1@student.unsw.edu.au

Abstract. The gradient optimization algorithms and network architectures play pivotal roles in the field of artificial intelligence. However, there is limited research comparing multiple optimization algorithms across different network structures. In this paper, the effectiveness of several optimization techniques in image processing tasks is investigated, along with an investigation of their effects on different neural network architectures. LeNet, AlexNet, and the backpropagation neural network are the three popular neural network architectures included in the selection, along with three different optimization algorithms. An extensive assessment of training loss, test accuracy, convergence time, and other metrics was carried out to determine how well these algorithms worked in various network designs through rigorous experimentation on image datasets. The results show complex differences in the impact of optimization techniques across various neural network configurations, providing essential information for the choice of the best algorithms and network structures.

Keywords: optimization algorithm; network structure; model performance.

1. Introduction

Over the past two decades, machine learning has come a long way from being a research curiosity to a practical technology with wide-ranging commercial applications. Regarding speech recognition, computer vision, and other applications, machine learning has become the go-to technique within artificial intelligence (AI) [1].

In the realm of artificial intelligence, the architectural design of neural network models alongside the refinement of gradient descent algorithms stands as pivotal elements within the landscape of machine learning. Various neural network structures and constantly updated gradient descent algorithms have continuously improved the accuracy of machine learning models, allowing machine learning to be applied to aspect of life such as medical diagnosis, financial analysis, and data statistics, providing convenience for social development.

In the past development, artificial neural networks and convolutional neural networks were important technologies for solving practical problems, and the gradient descent algorithm was widely used in the training and optimization process of the model. Among them, back-propagation (BP) is one of the most widely used parameter search techniques in ANN [2]. Convolutional neural networks (CNNs) demonstrated impressive performance on a variety of image categorization tasks [3].

Among the many gradient descent algorithms, stochastic gradient descent (SGD) is the most basic method and the workhorse algorithm of deep learning technology [4]. And researchers continue to propose and improve various gradient descent optimization algorithms. To expedite model convergence and enhance performance, such as AdaGrad, Adam [5], etc.

Although the optimization algorithm of gradient descent has been widely researched and applied in theory and practice, the network structure is a core component of the deep learning model, and the performance of the optimization algorithm may be different under different neural network structures. Factors such as the complexity of the structure, parameters count, and data characteristics may affect the convergence speed and performance of the optimization algorithm.

The objective of this study is to examine the influence of various network structures on several prevalent gradient descent optimization algorithms and validate their efficacy across diverse scenarios via empirical analyses. By conducting this research, we try to understand the interplay between network architecture and optimization algorithms. Additionally, we seek to offer theoretical insights and practical recommendations for constructing more efficient and robust deep learning models.

This article will comprise the following components. The initial section will elucidate the experimental methodologies employed. Subsequently, the experimental data will be presented in the second part. Following that, the third part will undertake the analysis and comparison of the experimental data. Finally, the fourth part will encapsulate the summary of experimental results and the formulation of conclusions.

2. Methods

This section will provide a concise exposition of the principles underlying the optimization algorithms and network architectures involved in the experiments. It will delineate the underlying mathematical logic and gradient descent formulations beneath optimization algorithms, as well as the structure of neural networks, and discuss their distinctions and similarities.

2.1. Introduction to Optimization Algorithms

This study examines the performance of three optimization algorithms—SGD, Adaptive Moment Estimation (Adam) and Adaptive Gradient Algorithm (Adagrad) to compare the performance on different network structures. And then, this article will introduce the mathematical logic of three optimization algorithms [6].

SGD stands up as a simple and incredibly successful approach in the context of machine learning models. Large amounts of effort have been focused on improving optimization methods and solving optimization problems in recent years. The general form of SGD for a given loss function $f(x)$ is:

$$x_{k+1} = x_k - \alpha \nabla f_i(x_k) \tag{1}$$

where α represents the learning rate, $\nabla f_i(x^{(k)})$ represents the gradient and i denotes the index of a randomly selected subset of samples from the dataset.

Adam combines adaptive learning rate with momentum techniques. The equation for each update is as follows:

$$x_{k+1} = \alpha x_k + (1-\alpha) g_{k+1} \tag{2}$$

$$y_{k+1} = \beta y_k + (1-\beta) g_{k+1}^2 \tag{3}$$

x_{k+1} and y_{k+1} are moment of the gradients.

$$\widehat{x_k} = \frac{x_k}{1-\alpha^k} \qquad \widehat{y_k} = \frac{y_k}{1-\beta^k} \tag{4}$$

$$z_{k+1} = z_k - \frac{\alpha}{\sqrt{\widehat{y_k}}+\varepsilon} \widehat{x_k} \tag{5}$$

where $0<\alpha<1$ and $0<\beta<1$.

AdaGrad adjusts the learning rate dynamically according to the gradients obtained in prior epochs.

The equation for each update is as follows:

$$x_{k+1} = x_k - \frac{\alpha}{\sqrt{G_k+\varepsilon}} g_{k+1} \quad (6)$$

where α is defaulted as 0.001, G_k is a diagonal matrix where each element is the sum of the squares of the past gradients and g_{k+1} represents the gradient.

Among these three algorithms, SGD computes gradients using a randomly selected batch of data samples, with the learning rate pre-declared and kept constant throughout the computation process.

However, for both Adam and AdaGrad algorithms, a learning rate needs to be manually specified before the code execution. During the algorithm runtime, the learning rate dynamically adjusts with the variation of gradients to discover the optimal learning rate, facilitating faster model convergence. Notably, Adam algorithm introduces a momentum parameter, which remains unexplored in this paper, thus default parameters of Adam algorithm will be employed.

2.2. Introduction to Network Structure

This study employs three classical neural network architectures: BPNN, LeNet and AlexNet, as the experimental subjects to explore the impact of diverse network structures on the effectiveness of optimization algorithms.

BPNN: The backpropagation neural network (BPNN) is characterized by its non-linear mapping capability and flexible network architecture [7], comprising multiple neurons distributed across input, hidden, and output layers. The topological structure of the BPNN is illustrated as figure 1.

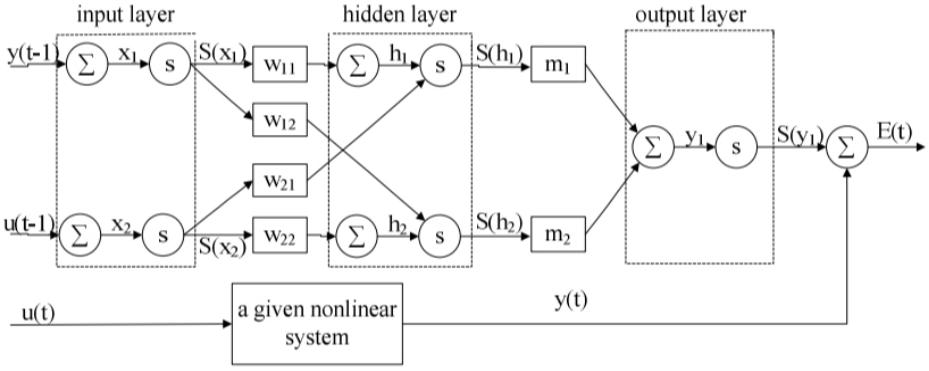

Figure 1. The structure of BPNN[7]

LeNet is a convolutional neural network model [8]. It is able to proficiently perform image processing tasks. The topological structure of the LeNet is illustrated as figure 2: AlexNet features a novel deep CNN architecture [9]. The topological structure of the LeNet is illustrated as figure 3:

Among these three network architectures, BPNN is the fundamental neural network structure, while LeNet and AlexNet are both examples of deep neural networks. They are all capable of handling image recognition tasks, but their performance may vary. In the subsequent sections of this paper, we will discuss the impact of network architecture and optimization algorithms on model performance.

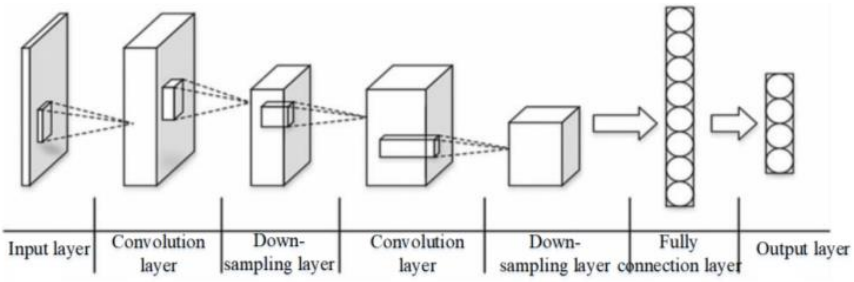

Figure 2. The structure of LeNet [8]

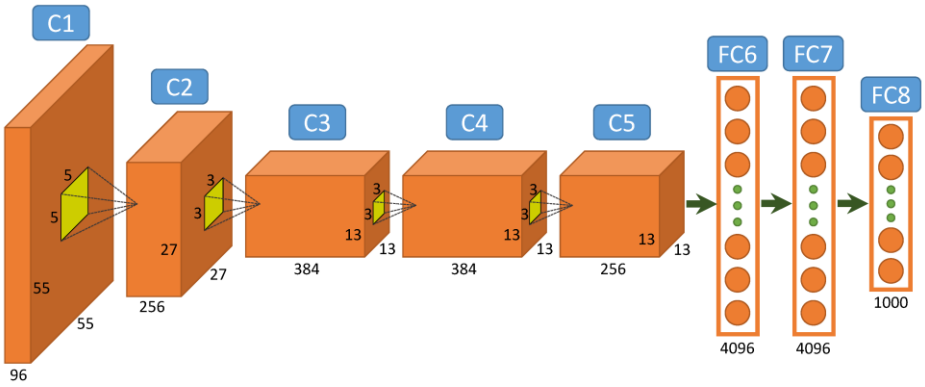

Figure 3. The structure of AlexNet [9]

3. Results

This study focuses on the evaluation of three distinct optimization algorithms across three varied network nodes. By employing different optimization algorithms across diverse network structures and evaluating various performance metrics such as training loss, test accuracy, and running time, this research aims to elucidate the influence of network architecture on optimization algorithms.

3.1. Experimental Design

For each combination of network structure and optimization algorithms to maintain consistency, identical learning rates, communication epochs, and batch sizes are employed across the three optimization algorithms. The training procedures are executed under uniform hyperparameters.

1) Dataset: The standardized dataset utilized in this study is Fashion-MNIST, which comprises grayscale images of diverse clothing items presented in a 28x28 format [10].

2) The Conceptual Framework of the Code: This article combines three types of network architectures with three different optimization algorithms, for example, BPNN and SGD, BPNN and Adam, BPNN and AdaGrad, LeNet and SGD, LeNet and Adam, LeNet and AdaGrad, AlexNet and SGD, AlexNet and Adam, AlexNet and AdaGrad.

This article sets the same hyperparameters, learning-rate = 0.001, batch-size = 32, num-epochs = 50. This article will use the training set for model training, record the data during the training process, and output the training loss, training accuracy, testing accuracy, runtime, as well as the graphical representation depicting the relationship between loss and epochs, testing accuracy and epochs, and runtime and epochs, are presented in this paper. Furthermore, the graphs depicting the performance of various optimization algorithms within identical network architectures are consolidated into a singular visual representation.

3) Evaluation Criteria: After the model training is completed, we will assess its performance using the test set and document the test accuracy.

This article will select the number of epochs of convergence points of different network structures and optimization algorithm combinations, and compare the training accuracy, test accuracy and time required for model convergence between different network structures and optimization algorithm combinations under this epoch time. And the curves for each algorithm under every network structure are compared to discern the differences in performance among the three algorithms.

3.2. Experimental Results and Analysis

This section showcases the findings from the experiments, focusing primarily on comparative visualizations of the three different optimization algorithms under each network architecture. Discussions and comparisons will be conducted regarding the performance of each algorithm within

each network structure. Finally, a comparative analysis will be conducted among the three network architectures to identify the optimal combination of network structure and algorithm performance on the dataset employed in this study.

1) The Experimental Results for BPNN: The following figure 4 illustrates the relationship between training loss and epochs, testing accuracy and epochs, and runtime and epochs for three different optimization algorithms under the BPNN structure.

From the figure 4, it is evident that in the BPNN structure, due to the inability of the SGD algorithm to dynamically adjust the learning rate like Adam and AdaGrad algorithms, the training loss values of SGD algorithm are larger than those of Adam and AdaGrad algorithms within a finite number of epochs. Consequently, the test accuracy of the SGD algorithm is lower. However, as Adam and AdaGrad algorithms require dynamic updates of the learning rate hyperparameter and involve momentum calculations in the case of Adam algorithm, the runtime of Adam and AdaGrad algorithms is longer compared to that of the SGD algorithm.

2) The Experimental Results for LeNet : The following figure 5 illustrates the relationship between training loss and epochs, testing accuracy and epochs, and runtime and epochs for three different optimization algorithms under the LeNet structure.

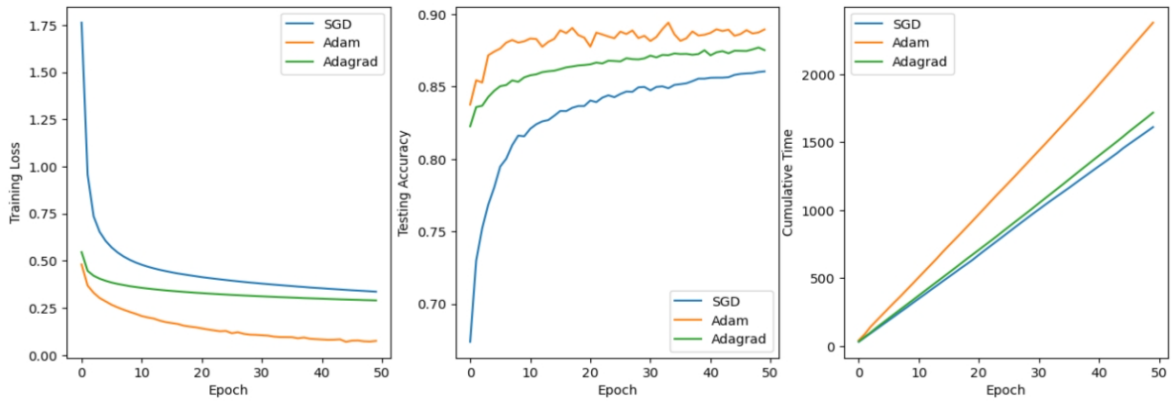

Figure 4. The experimental results plot for BPNN (Photo/Picture credit :Original)

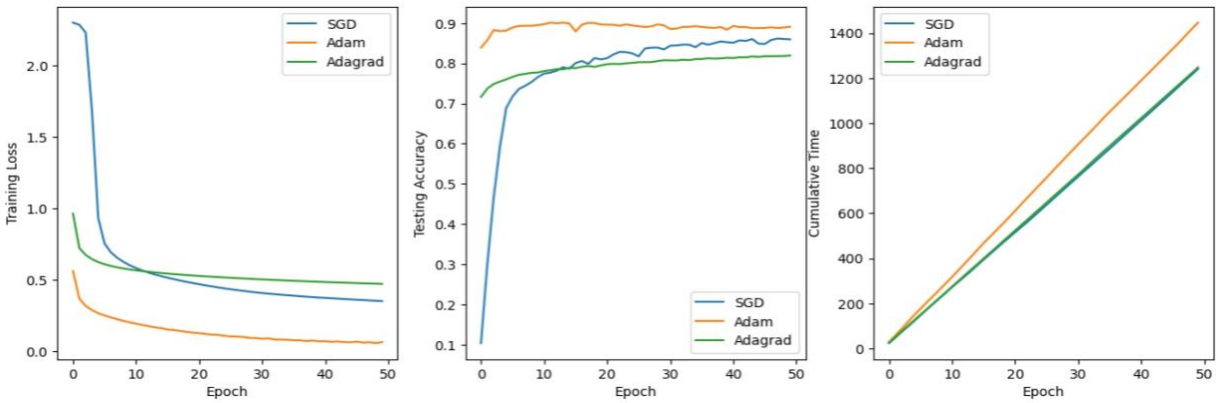

Figure 5. The experimental results plot for LeNet (Photo/Picture credit :Original)

From the figure 5, it is evident that in the LeNet structure, unlike the BPNN structure, the SGD algorithm exhibits a smaller loss value compared to the AdaGrad algorithm. Although the AdaGrad algorithm achieves convergence at a faster rate, its test accuracy is lower than that of the SGD algorithm, while the runtime of both algorithms is almost identical. Among these three algorithms, however, Adam exhibits superior performance in terms of both training loss and test accuracy, despite requiring the longest runtime.

3) The Experimental Results for AlexNet: The following figure 6 illustrates the relationship between training loss and epochs, testing accuracy and epochs, and runtime and epochs for three different optimization algorithms under the AlexNet structure.

From the figure 6, it is evident that in the AlexNet structure, the results indicate a close similarity between the SGD and AdaGrad algorithms, with nearly identical training loss values and test accuracies by the 50th epoch, alongside a relatively small difference in runtime. Among these three algorithms, however, Adam exhibits superior performance in terms of both training loss and test accuracy, despite requiring the longest runtime.

From Figures 4, 5, and 6, it can be observed that the performance of the three optimization algorithms varies across different network architectures. Specifically, the SGD algorithm requires more epochs to converge, while the Adam and AdaGrad algorithms converge in fewer epochs. Therefore, if a limited number of epochs is available, opting for the Adam and AdaGrad algorithms is preferable.

4) Comparative Analysis

The following table1 presents the training loss values, testing accuracies, and total runtime for each optimization algorithm at the 50th epoch under various network architectures. From Table I, it's evident that the three optimization algorithms perform optimally when paired with the AlexNet architecture on the dataset used in this study. The effectiveness arises from how AlexNet combines depth and width in its network architecture, allowing it to capture more intricate features and thereby enhance performance. Specifically, the Adam algorithm exhibits the best performance. However, due to the intricacies of the network architecture and the design of the optimization algorithms, the AlexNet architecture requires the longest runtime, with the Adam algorithm also having a longer runtime compared to the other algorithms. Consequently, the combination of the Adam algorithm and the AlexNet structure demonstrates the best performance, albeit with a lengthy runtime. To improve runtime while maintaining performance, an alternative option would be to choose the combination of the Adam algorithm with the LeNet structure.

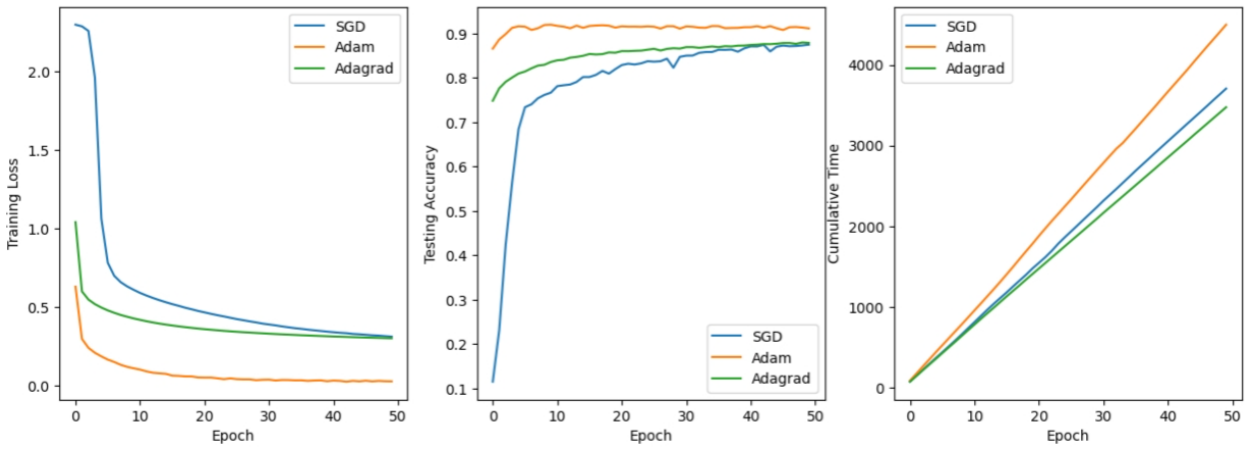

Figure 6. The experimental results plot for AlexNet (Photo/Picture credit :Original)

Table 1. The experimental results at the 50th epoch

Neural Network / Optimization Algorithms		BPNN	LeNet	AlexNet
SGD	Loss	0.3362	0.3529	0.311
	Accuracy	86.05%	85.97%	87.52%
	Time	1611.15s	1240.41s	3705.12s
Adam	Loss	0.075	0.0662	0.0258
	Accuracy	88.95%	89.1%	91.18%
	Time	2380.05s	1445.98s	4497.58s
AdaGrad	Loss	0.2892	0.4733	0.3005
	Accuracy	87.52%	81.95%	87.91%
	Time	1717.54s	1247.18s	3474s

4. Conclusion

This study's main goal is to investigate how gradient descent optimization algorithms are applied inside neural network topologies, with an emphasis on how well they operate with the Fashion-MNIST dataset. We can have a more thorough grasp of the benefits and drawbacks of different network architectures and optimization techniques in real-world applications by contrasting their performances.

Experimental results reveal that the Adam optimization algorithm exhibits the highest test accuracy, particularly within the complex AlexNet architecture. However, it is also observed that compared to simpler network structures, AlexNet incurs longer execution times, highlighting the crucial importance of considering both performance and efficiency when selecting network structures. Regarding the three optimization algorithms examined, it is found that due to the incorporation of momentum computation, the Adam algorithm requires longer execution times compared to the other two algorithms. Nevertheless, the Adam algorithm demonstrates the best performance in this experiment.

As a result, it's critical to take performance and efficiency into careful consideration while choosing optimization techniques. These experimental results offer important insights for the future development of optimization algorithms and network structures, in addition to suggestions for choosing the best algorithms and network architectures for certain tasks. To sum up, this research has important applications for improving neural networks' effectiveness and performance in image processing applications.

References

[1] M. Jordan, T. Mitchell, Machine learning: Trends, perspectives, and prospects. Science 349,255-260, 2015.

[2] Y. Xue Y, Y. Wang, J. Liang, A self-adaptive gradient descent search algorithm for fully-connected neural networks, Neurocomputing, Volume 478, 2022, Pages 70-80.

[3] Y. Sun, B. Xue, M. Zhang, et al., "Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification," IEEE Transactions on Cybernetics, vol. 50, no. 9, pp. 3840-3854, Sept. 2020.

[4] F. Mignacco, P. Urbani, The effective noise of stochastic gradient descent. J. Stat. Mech: Theory Exp. 2022, 083405, 2022.

[5] S. Haji, A. Abdulazeez, Comparison of Optimization Techniques Based on Gradient Descent Algorithm: A Review. PalArch's J. Archaeol. Egypt/Egyptol, 18(4), 2715-2743, 2021.

[6] Y. Tian, Y. Zhang, H. Zhang, Recent Advances in Stochastic Gradient Descent in Deep Learning. Mathematics 2023, 11, 682.

[7] K. Chen, S. Yang, C. Batur, "Effect of multi-hidden-layer structure on performance of BP neural network: Probe," ICNC 2012, China, 2012, pp. 1-5.

[8] J. Zhang, X. Yu, X. Lei, et al. A novel deep LeNet-5 convolutional neural network model for image recognition. Comput. Sci. Inf. Syst. 2022, 19, 1463–1480.

[9] F. Hu, G. Xia, J. Hu, et al. Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. Remote Sens. 2015, 7, 14680-1470.

[10] O. Nocentini, J. Kim, M. Bashir, et al. Image Classification Using Multiple Convolutional Neural Networks on the Fashion-MNIST Dataset. Sensors 2022, 22, 9544.