Optimizing Generalization in Deep Neural Networks through Adaptive Regularization

Ms.R.Jaseenash[1],Assistant Professor
Department of Computer Science and Engineering, Sri Shanmugha College of Engineering and Technology, Pullipalayam, Morur (Post), Sankari (Tk), Salem, Tamil Nadu, India
Jaseenash.r@shanmugha.edu.in
Mrs. Beaula Pinky B[2],Assistant Professor
Department of AI&DS, Sri Shanmugha College of Engineering and Technology, Pullipalayam, Morur (Post), Sankari (Tk), Salem, Tamil Nadu, India
beaulapinky@shanmugha.edu.in
Mrs.S.Gokulapriya[3],Assistant Professor
Department of Computer Science and Engineering, Sri Shanmugha College of Engineering and Technology, Pullipalayam, Morur (Post), Sankari (Tk), Salem, Tamil Nadu, India
gokulapriyas@shanmugha.edu.in
Mr.P.Tamilalagan[4],Assistant Professor
Department of Computer Science and Engineering, Sri Shanmugha College of Engineering and Technology, Pullipalayam, Morur (Post), Sankari (Tk), Salem, Tamil Nadu, India
tamilalagan@shanmugha.edu.in

Abstract- Recently, deep learning based techniques have garnered significant interest and popularity in a variety of fields of research due to their effectiveness in search for an optimal solution given a finite amount of data. However, the optimization of these networks has become more challenging as neural networks become deeper and datasets growing larger. The choice of the algorithm to optimize a neural network is one of the most important steps in model design and training in order to obtain a model that will generalize well on new, previously unseen data. In deep learning, adaptive gradient optimization methods are mostly preferred for supervised and unsupervised tasks. First, they accelerate the training of neural networks and since mini batches are selected randomly and are independent, an unbiased estimate of the expected gradient can be computed. This paper examined six state-of-the-art adaptive gradient optimization algorithms, namely, AdaMax, AdaGrad, AdaDelta, RMSProp, Nadam, and Adam on the generalization performance of convolutional neural networks (CNN) architecture that are extensively used in computer vision tasks.

Keywords – Adaptive gradient methods, optimization, deep learning, convolutional neural networks, image processing.

INTRODUCTION

The design goal of any machine learning classifier and any learning algorithm in general, based on a training set of a finite size, is to ultimately provide a good generalization performance (Theodoridis, 2020). The generalization performance is quantified by the difference between the training error and the test error where good machine learning algorithms are those where the test error and the training error have close values (Hu & Zheng, 2019)[1]. Recently, deep learning algorithms have shown to have better generalization performance than traditional machine learning

techniques in solving classification problems. However, explaining why these highly non-convex models trained by a specific optimization algorithm can generalize better has become a significant open question in deep learning (Wang, Meng, Chen, & Liu, 2021). In the literature, different approaches have been employed to improve on the generalization performance of neural networks[2]. Among these approaches, some explore the implicit regularization property of Stochastic Gradient Descent (SGD) (Hardt, Recht, & Singer, 2016; Zhang, Bengio, Hardt, Recht, & Vinyals, 2017); Neyshabur, Tomioka, Salakhutdinov, & Srebro, 2017). Another perspective relies on the geometry of loss function around a global minimum with the work of Keskar, Mudigere, Nocedal, Smelyanskiy, and Tang (2017) adopting this perspective to explain why small-batch SGD often converges to the solution generalizing better than large-batch SGD similar to the work of (Baldassi, et al., 2016) where discrete networks were considered. Optimization is a significant component in deep learning[3]. Optimization provides an approach to minimizing the loss function, often referred to as the objective function in stochastic nonconvex optimization (Kingma & Ba, 2015). Optimization considers different methods and algorithms used for learning the underlying mapping from input data to outputs by choosing the right set of parameters that will reduce the error during model training (Marin, Skelin, & Grujic, 2019). Through optimization, researchers seek to find a suitable model, which will generalize well on new, previously unseen data. Stochastic gradient descent (SGD) with mini-batches and its variant are undoubtedly the most prevalent methods for training deep neural networks, owing to its simplicity and greater performance than the alternatives (Wang & Srebro, 2019)[4]. For instance, the use of minibatch optimization algorithms makes the training process more stable as it reduces the variance of gradient estimate. These algorithms take smaller number of updates if a larger mini-batch size is used. Moreover, the backpropagation procedure on a larger mini-batch can utilize massive parallelization of linear algebra routines provided by advanced computational hardware such as graphical processing units and clusters (Hu & Zheng, 2019). Stochastic optimization methods are the dominant techniques in the training of deep neural networks. In this paper, we explore the commonly used adaptive optimization algorithms and conduct empirical analysis of their effect on the training process and the final generalization performance of deep learning models. In particular, convolutional neural networks (CNNs) are considered. Convolutional neural networks are one of the popular deep learning models that have a wide range of applications in the field of computer vision[5]. A comparative analysis on the validation accuracy, as well as the model loss (train and test) of each optimization algorithm in the generation of an optimization solution is done using three large image datasets, namely, Fashion MNIST, Kaggle Flowers and Scene classification.

LITERATURE REVIEW

The crux of machine learning algorithms is to develop an optimized model capable of learning the parameters in the objective function and the constraints placed from the given dataset. Several effective optimization methods have been put forward to stimulate development of machine learning, consequently improving their performance and efficiency (Shone, Ngoc, Phai, & Shi, 2018)[6]. According to Zhou (2018), majority of neural network applications are naturally formulated as non-convex optimization due to the complex mechanism of the underlying model. In addition, neural networks have many symmetric configurations such as exchanging intermediate neurons, hence non-convex. With the increasing interest in deep learning applications, researchers have deemed it necessary to deal with non-convex optimization progressively, more particularly because of the benefits hidden behind their complexity. By definition, a non-convex optimization

is any problem where the objective or any of the constraints are non-convex (Jain & Kar, 2017) predominantly because such algorithms operate in high-dimensional spaces[7]. The freedom to express the learning problem as a non-convex optimization problem gives immense modelling power to the algorithm designer (Mehdi, 2020). Training a deep neural network can be described as an optimization problem with non-convex objective function. The non-convex deep neural networks have been found to have large amount of global minima (Choromanska, Henaff, Mathieu, Arous, & LeCun, 2015) with only a few able to guarantee satisfactory generalization property (Brutzkus, Globerson, Malach, & Shalev-Shwartz, 2018)[8]. During the training process, model parameters are iteratively updated in order to reduce the cost on the training data. Optimization methods are used to optimize noisy functions, i.e., track key parameters of interest in data streams, which can have changing dynamics over time. Stochastic methods are mainly employed in non- convex problems where robust results have been demonstrated (Reddi, et al., 2016; Jain & Kar, 2017[9]; Mehdi, 2020). The adaptive nature of stochastic approximation methods such as stochastic gradient descent make them highly applicable in a range of applications, particularly in machine learning (Reddi, Zaheer, Sachan, Kale, & Kumar, 2018). Various optimization algorithms exist in the literature for training neural networks and they vary in the way they update network parameters[10].

OPTIMIZATION CHALLENGES IN DEEP LEARNING

Numerical optimization methods systematically vary the inputs to a function so as to realize the minimum or maximum value of the function. Normally, this requires numerous evaluations of the objective function (Fike, Jongsma, Alonso, & Weide, 2011). Optimization methods that utilize gradient information usually converge to the optimal solution in fewer iterations than methods that only rely on the function values and tends to drive parameters to certain kinds of global minima, which generalize well. However, optimization of deep neural networks comes with some challenges (Marin, Skelin, & Grujic, 2019). In the literature, the most vexing ones are local minima, saddle points, and vanishing gradients[11].

(i) Local Minima- The convergence of the optimization algorithm should be ensured to avoid falling into local optimum (Hu & Zheng, 2019). Researchers have empirically demonstrated that different local optima, attained from training deep neural networks do not generalize in a similar manner for the unseen data sets, albeit achieving similar training loss (Wang, Keskar, Xiong, & Socher, 2018). Figure 1 shows local minima in deep learning training[12].

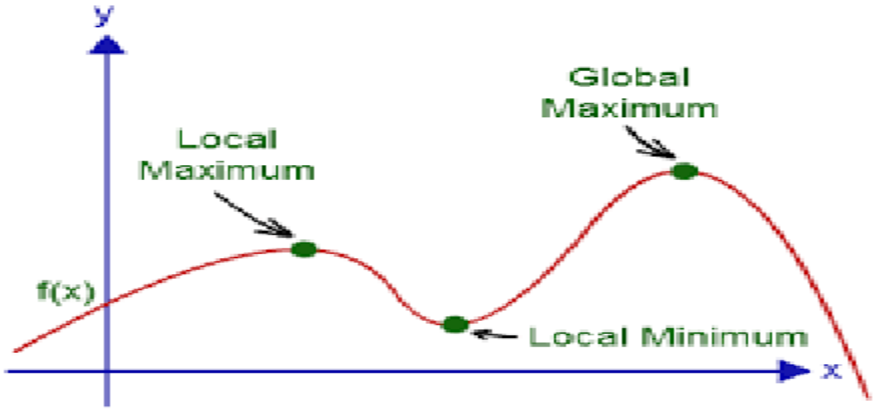

Figure 1- Local Minima in non-convex functions[8]

(ii) Saddle Points- A saddle point is any location where all gradients of a function vanish but which is neither a global nor a local minimum. Saddle points in higher dimensions problems are even more insidious since the likelihood that at least some the eigenvalues of the function's Hessian matrix are negative is quite high (Dauphin, et al., 2014). Figure 2 demonstrates sandle points in non-convex functions[10].

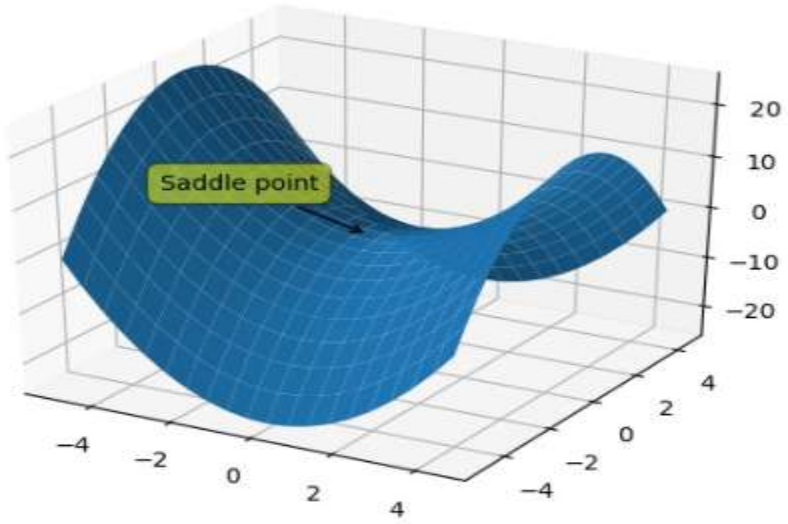

Figure 2- Saddle points in non-convex functions[5]

(iii) Vanishing Gradients- The vanishing gradient is the most insidious problem encountered in deep learning. The problem is mostly encountered when training deep learning models using gradient-based learning methods and backpropagation (Tan & Lim, 2019). As more layers using certain activation functions are added to a neural network, the gradients of the loss function approaches zero, making the network hard to train (Wang C.-F. , 2019). Figure 3 shows the vanishing gradient problem in training deep neural networks using the sigmoid activation function[13].

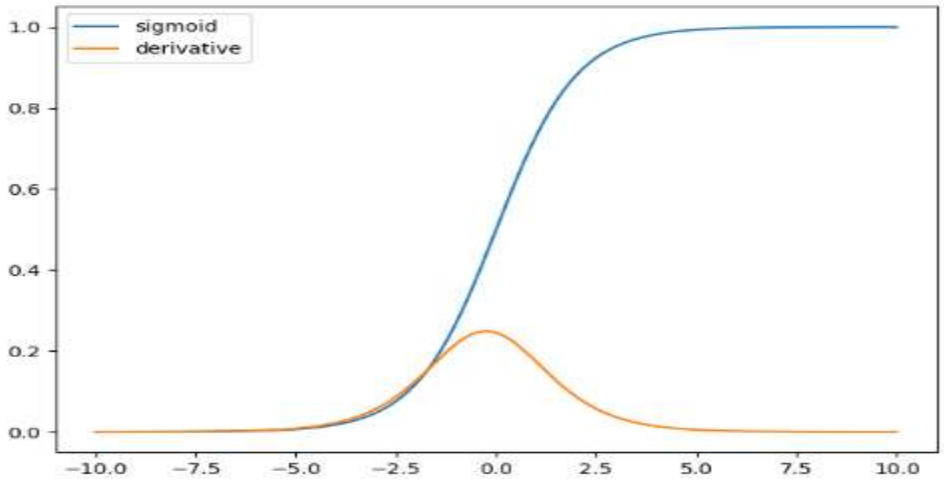

Figure 3- Vanishing gradient problem[3]

MATERIALS AND METHODS

the experimental design of the study. The setup and settings used in the experiments, the deep learning architecture used, the hyper-parameter setting, and the datasets used are described.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 224, 224, 32)	896
conv2d_2 (Conv2D)	(None, 224, 224, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 112, 112, 64)	0
dropout_1 (Dropout)	(None, 112, 112, 64)	0
conv2d_3 (Conv2D)	(None, 112, 112, 128)	73856
conv2d_4 (Conv2D)	(None, 110, 110, 256)	295168
max_pooling2d_2 (MaxPooling2	(None, 36, 36, 256)	0
dropout_2 (Dropout)	(None, 36, 36, 256)	0
flatten_1 (Flatten)	(None, 331776)	0
dense_1 (Dense)	(None, 256)	84934912
re_lu_1 (ReLU)	(None, 256)	0
dropout_3 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
re_lu_2 (ReLU)	(None, 256)	0
dense_3 (Dense)	(None, 5)	1285

Total params: 85,390,405
Trainable params: 85,390,405
Non-trainable params: 0

Table 1- Architecture of the fine-tuned VGG-16 CNN Model[11]

(i) Experiments- The aim of this experimental study was to compare the effect of adaptive optimizers on the training process and final generalization performance of classification models. Convolutional neural networks, deep learning architectures were train on three datasets. For implementation, a simulation experiment environment (Vieira, Koch, Sobral, Westphall, & de Souza Leao, 2019) was built on a 64-bit computer based on Win10 operating system platform with Intel® i7-9750 Hz 2.60 GHz CPU, 8 GB RAM, NVIDIA GeForce RTX 2060 6G GDDR6 GPU, and 10.2 CUDA. The experiments were written in Python language and the Scikit-learn and NumPy were used for data pre-processing. The neural networks was built based on the Tensorflow and Keras deep learning frameworks[14].

(ii) CNN Model Architectures- The network architecture plays a critical role in the performance of deep learning models. In this study, two well-known CNN model architectures were considered and further fine-tuned to ensure effective training process (Gong, Wang, Guo, & Lazebnik, 2014). The output layer for each model was fine-tuned to reflect the classification output expected from the chosen dataset. Model 1 architecture was based on the VGG-16 architecture developed by the Visual Geometry Group from the University of Oxford in 2014. VGG-16 (Simonyan & Zisserman, 2014) served as the mainstream CNN model for feature representation and classification and was used to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014[15]. The architecture consists of 16 layers, where convolutional layers (13) with 3x3 filters and 2x2 max pooling layers are stacked and dense layers incorporated before the output layer. The model contains a set of weights that are used to train the images. Table 1 shows the summary of the fine-tuned VGG-16 model that was used to train on the Kaggle Flowers recognition dataset.

CONCLUSION

In this paper, we explored the different adaptive optimization algorithms commonly used for training deep learning model architectures and compared their effect on the model's generalization performance on image classification problem. Experiments were conducted using two convolutional neural network models and the performance of each adaptive optimization method was compared while trained and tested on three large image datasets. The empirical evaluations demonstrate that the chosen optimization algorithm can positively affect model's generalization performance. The Adam, Adadelta and Nadam optimization algorithms were observed to be the best-performing algorithms on new data in our experiments. Adaptive optimization algorithms consider the cumulative changes of each parameter in their respective iterative optimization processes. They scale coordinates of the gradient by square roots of some form of averaging of the squared coordinates in the past gradients and automatically adjust the learning rate on a parameter basis. The theoretical background complemented with experimental results of the deep learning process is beneficial to anyone who seeks more in-depth insight into the fields of optimization of deep learning.

REFERENCES

[1]. Baldassi, C., Borgs, C., Chayes, J., Ingrosso, A., Lucibello, C., Saglietti, L., & Zecchina, R. (2022). Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes. Proceedings of the National Academy of Sciences, 113(48), pp. E7655–E7662.

[2]. Brownlee, J. (2023, January 8). A Gentle Introduction to the Rectified Linear Unit (ReLU). Retrieved July 16, 2021, from Machine Learning Mastery: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

[3]. Brutzkus, A., Globerson, A., Malach, E., & Shalev-Shwartz, S. (2022). Sgd learns over-parameterized networks that provably generalize on linearly separable data. International Conference on Learning Representations.

[4]. Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., . . . Zecchina, R. (2019, December 20). Entropy-SGD: biasing gradient descent into wide valleys. Journal of Statistical Mechanics: Theory and Experiment, 2023(124018).

[5]. Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B., & LeCun, Y. (2022). The loss surfaces of multilayer networks. In Artificial intelligence and statistics (pp. 192-204).

[6]. Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., & Bengio, Y. (2023, January). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. Advances in Neural Information Processing Systems, 4, 2933-2941.

[7]. Sanjay Kumar Suman, Dhananjay Kumar and L. Bhagyalakshmi, "Non Cooperative Power Control Game with New Pricing for Wireless Ad hoc Networks", International Review on Computers and Software, vol. 9, no. 1, pp. 18-28, 2014. ISSN: 1828-6003,

[8]. S. Porselvi, Sanjay Kumar Suman and L. Bhagyalakshmi, "Harvesting RF energy for mobile charging", Australian Journal of Basic and Applied Science, vol. 9, no. 20, pp. 454-465, June 2015.

[9]. K. Swapna, P. Rajalakshmi and Sanjay Kumar Suman, "Security Enhancement in MANET using Game Theory", Middle East Journal of Scientific Research, vol. 23, pp. 190-195, 2015.

[10]. VinaySrivatsan, Sanjay Kumar Suman, L. Bhagyalakshmi and S. Porselvi, "Non radiative wireless power transfer", Journal of Advances in Natural and Applied Sciences, vol. 10, no. 16, pp. 147-153, Nov. 2016.

[11]. Sujeetha Devi, Bhagyalakshmi L and Sanjay Kumar Suman, "Cluster based energy efficient joint routing algorithm for delay minimization in wireless sensor networks", International Journal of Pure and Applied Mathematics, vol. 119, no. 15, 307-313, 2018

[12]. Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2022). Understanding deep learning requires rethinking generalization. International Conference on Learning Representations (ICLR 2017). Toulon, France.

[13]. Zeiler, M. D. (2023). ADADELTA: An adaptive learning rate method. arXiv Preprints, arXiv:1212.5701. Retrieved from https://arxiv.org/abs/1212.5701

[14]. Zadeh, R. (2022, November 16). The hard thing about deep learning. Retrieved from https://www.oreilly.com: https://www.oreilly.com/radar/the-hard-thing-about-deep-learning/

[15]. Yi, D., Ahn, J., & Ji, S. (2022). An Effective Optimization Method for Machine Learning Based on ADAM. Applied Sciences, 10(1073), 20. doi:10.3390/app10031073