Optimizing Deep Network for Image Classification with Hyper Parameter Tuning

Munmi Gogoi, Shahin Ara Begum

Abstract**:** ***The deep network model comprises of several processing layers and deep learning techniques help us in representing data with diverse levels of abstraction. Based on the practical importance and the efficiency of machine learning, optimization of deep models are carried out relating to the objective functions and its parameters for a particular problem. The present work focuses on an empirical analysis of the performance of stochastic optimization methods with regard to hyperparameters for the deep Convolution Neural Network* (CNN) *and to understand the rate of convergence of the optimization methods in high dimensional parameter spaces. Experimentation has been carried out in deep* CNN *model with different optimization methods viz. SGD, AdaGard, AdaDelta and Adam. The empirical results are evaluated using benchmark* CIFAR10 *and* CIFAR100 *datasets. The optimal values of the hyperparameters obtained demonstrates that the optimizer Adam shows the best results compared to other methods viz. SGD, AdaGard, and AdaDelta over the considered datasets. Further, it is noted that classification accuracy can be increased by choosing the best optimization techniques with hyperparameter tuning to get the optimal configuration of the deep CNN model.***

Keywords**:** ***Optimization techniques,*** **CNN,** ***hyperparameter.***

I. INTRODUCTION

Image classification is one of the crucial jobs in the field of computer vision. The advancement of Deep Learning techniques in this area establishes superior performance than the preceding work and hence its objective is to move Machine Learning nearer to one of its novel aims: Artificial Intelligence. Deep learning refers to computational models containing numerous processing layers and can have compound levels of representation and abstraction that assist to make sense of data such as images, sound, and text. A deep network with multiple hidden layers is capable of recognizing more complex features. Subsequently, the combined nodes recombine features from the preceding layer [1]. Neural Network comprises of a layered set of neurons connected by links with some synaptic weights, through the activation functions, and the summation is processed. The concept of multiple hidden layers or deep network models aroused to solve problems of non-linearly separable domain and the complexity increases when problems related to arbitrary decision boundary and arbitrary accuracy with rational activation functions are encountered. For the differentiable objective functions, Gradient descent is the most common optimization method to solve the problem but most often objective functions are stochastic and stochastic gradient descent (SGD) is an effective optimization technique in many machine learning area. With the emerging trend of deep learning techniques and high dimensionality of data, this paper focus on optimizing stochastic objective functions w. r. t. its high dimensional parameter space. The main purpose of optimizing a network is to minimize the error by the common gradient descent algorithm *m* where we differentiate the error function to get the gradient of the error and update the weights to make the error smaller i.e. to minimize an objective function $J(\theta)$ parameterized by a model's parameters $\theta \in R^d$ by updating the parameters in the opposite direction of the gradient of the objective function $\nabla\theta J(\theta)$ w.r.t. to the parameters. The learning rate η determines the size of the steps we take to reach a global optimum. The complexity of the network increases as the number of layers increases and hence the necessity of optimization techniques. With this hypothesis, section 2 gives a brief description of work related to different optimization techniques such as SGD, Adam, AdaGard and AdaDelta. Section 3 describes the deep architecture of the CNN model and the hyperparameter optimization of the deep network. Section 4 presents the benchmark dataset from [2] used for the experimentation. Section 5 presents the experimental results of optimizing techniques and the comparison of performance of the optimizing techniques with regard to hyperparameters for the deep CNN. Section 6 concludes the paper.

II. RELATED WORK

The traditional machine learning techniques with hand-engineered feature design consumes much time than deep learning techniques in trend. Numerous machine learning problems deal with minimization or maximization of objective functions concerning some parameters i.e. optimization of objective functions. Stochastic gradient descent is the most general method of optimization in various machine learning task but with the rapid advances in deep learning many other efficient stochastic optimization techniques have been developed where stochastic objective functions with high dimensional parameter spaces are the matter of concern [3]-[4]-[5]-[6]-[7].

Revised Manuscript Received on December 30, 2019.
* Correspondence Author

Munmi Gogoi, Department of Computer Science, Assam University, Silchar,Assam,India.
Shahin Ara Begum*, Department of Computer Science, Assam University, Silchar, Assam, India.

Retrieval Number: B3515129219/2019©BEIESP
DOI: 10.35940/ijeat.B3515.129219
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

In 2011 Duchi [8] proposed another optimization technique named AdaGard

where optimization takes place with sparse gradients. It has the same updating rule with SGD but has a different learning rate for each parameter. The final update rule for AdaGard is given in (1) [8].

$$G^k = G^{k-1} + \nabla J(\theta^{k-1})^2$$

$$\theta^k = \theta^{(k-1)} - \frac{\alpha}{sqrt(G^{k-1})}.\nabla J(\theta^{k-1})$$

(1)

where, · and *sqrt* are element-wise operations. The historical gradient information is considered as *G* and it stores the sum of squares of its all historical gradients for each parameter and to scale up the learning rate, the calculated sum is used later. For each of the parameters, the AdaGrad learning rate is dissimilar from SGD. The learning rate is bigger for the parameters where the historical gradient is relatively small and the learning rate is smaller for greater historical gradients. AdaGard stores the sum of the square of its entire historical gradient (G) of each parameter which has been used to scale the learning rate later on. To overcome the weakness of AdaGard, Zeiler [9] introduced AdaDelta with respect to the learning rate converging to zero with augment of time. In contrast to AdaGard, AdaDelta uses only the current time window to scale the learning rate rather than consider the entire historical gradient like AdaGard. Adadelta combines two notions though - the first one uses only the recent time window gradient information rather considering the whole for scaling up the learning rate, and the second one comes up with the concept of acceleration term similar to momentum, and for that Adadelta use the component that serves an acceleration term. The update rules for Adadelta first computes the gradient g_t at current time *t*, then accumulates gradients as in (2) [9].

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho)g_t^2 \qquad (2)$$

After accumulating the gradients, (3) computes the Update, where, $E[g^2]_t$ and $E[\Delta x^2]_t$ is the accumulation variable and update variable respectively at time t.

$$\Delta x_t = -\frac{\sqrt{E[\Delta x^2]_{t-1} + \varepsilon}}{\sqrt{E[g^2]_t + \varepsilon}} g_t \qquad (3)$$

where, the parameter ρ is decay constant and ϵ (very small number) is considered for numerical stability. Another optimization algorithm Adam is one of the best choices for the neural network community. Adam [10] is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The Adam method has some advantages over other methods and it includes straightforward implementation, efficiency, and less memory requirement. Adam is the appropriate choice for bigger problems in terms of data and parameter, this method is also suitable for the problems with very noisy and sparse gradients and Adam method works well for non-stationary objectives as well. The update rule for Adam is determined based on the estimation of first (mean) and second raw moment of historical gradients. Adam update rule first computes the gradient g_t at current time *t* and then update biased first-moment estimate as in (4).

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t$$

(4)

where, m_t is the first-moment vector and β_1 is the decay rate for the moment estimates. Then, updates biased second raw moment estimate as in (5) where *v* and β_2 is the second-moment estimate and decay rate respectively, after updating biased for first and second raw moment, the (6) and (7) equation computes biased raw moment for first and second raw moment accordingly.

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2$$

(5)

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

(6)

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

(7)

Finally, the Adam optimizer updates the parameters as in (8), where θ is the parameter for stochastic objective function $f(\theta)$.

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}$$

(8)

III. CONVOLUTION NEURAL NETWORK MODEL

A. Convolution Neural Network (CNN) Architecture

The Convolution Neural Network (CNN) is a special kind of neural network which is composed of one or more convolution layers. A CNN convolves learned features with input data and uses a 2D convolution layer that makes this architecture well suited to processing 2D data, such as images. In the area of image classification and recognition, CNN has been used extensively and has proven very effective in this domain [4]. CNN model has a greater learning capacity compared to other feed-forward neural networks in the domain of image classification as the model has to go through millions of images to learn the latent pattern of the images for correct assumptions about the object images. The capability of a CNN model depends upon the parameters associated with it such as depth and breadth of the model. The training of a CNN model is much easier as compared to the similar-sized multilayer neural network as the CNN model has much fewer connections and parameters. An image is directly given as an input to the network and passes through various layers of convolution and pooling. Finally, the outcomes from these operations are given to one or more fully connected layers for the desired output often known as class label. Convolution layers detect local conjunctions from features and polling layers combine analogous features into one [11]. CNN uses convolutions instead of matrix multiplication in the convolution layers [12].

A typical CNN structure contains chains of convolutional layers, nonlinear activation layers, pooling layers and finally for output classification labels a fully connected layer is added in CNN architecture.

The deep model learns from low-level features to get more abstract features i.e. the feature detectors (filter) learn from minor regions of an image and sum up them to figure out more abstract features. Later on, these abstract features have been used by the fully connected layer for output the classification labels into different classes based on the training dataset. In the fully connected layer, every neuron of the layer is associated with every neuron of the previous layer. This layer computes class scores and that will decide the output of the network.

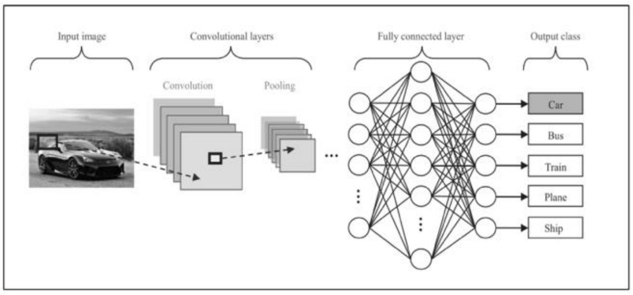


Fig. 1. Image Classification pipeline (CNN)[4]

B. Hyperparameter for CNN

Hyperparameter tuning is essentially used to make the network better and faster, and they deal with managing optimization functions and model selections during training with learning algorithms. It focuses on ensuring the two major problems of a deep network such as over-fitting and under-fitting of the training dataset while learning. Hyperparameter related to training algorithm includes learning rate, momentum, batch size, and number of epochs, etc. Learning rate refers to how fast a network is learning and it deals with the problem of slow convergence. A good choice of learning rate depends upon the optimizers used in the model such as SGD, Adam, AdaGard, and AdaDelta. The parameter "number of the epoch" defines one cycle through the training dataset and the number of epoch increasing depends upon the validation accuracy and training accuracy of the training data. Network parameter batch size is the number of samples given to the network after network parameter updates take place. CNN is sensitive to batch size; basically, minibatch size is preferable. To find out the method of hyperparameter selection [13] cast some light on hyperparameter optimization on large hierarchical models such as Deep belief network models. Bengio demonstrates empirically and theoretically that random search technique is more preferable over grid and manual search because not all the hyperparameters are mandatory to tune. In 2007, [14] experiments gird search and reports that grid search allocates too many trails to the investigation of dimensions and experiences poor convergence as compared to random search.

IV. EXPERIMENTAL SETUP AND DATASET

In this empirical analysis, the organizational design of the CNN model contains three alternating 5x5 convolution filters, 3x3 max pooling with stride 2 and a fully connected layer with 1000 rectified linear hidden units. The experimentation has been carried out on the HPC environment with Python language using the TensorFlow libraries. The description of hardware for this experimentation includes 16 Intel core processors, 16 GB RAM, with 2GB Nvidia Geforce GTX graphics card.

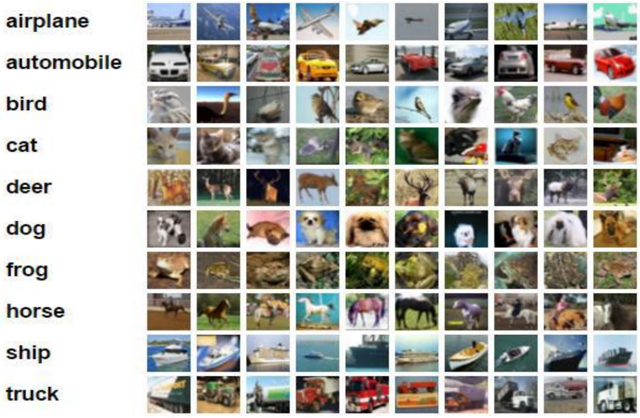


Fig. 2. CIFAR-10 dataset [2].

Experimentation has been conducted using CIFAR10 and CIFAR100 datasets. CIFAR stands for the Canadian Institute for Advanced Research [2]. The images of the dataset were composed by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. In the CIFAR10 dataset, "10 classes" are present and it comprises 50,000 training images and 10,000 test images. CIFAR-10 has 32 x 32 color real-world objects such as an airplane, automobile, cat, etc.

CIFAR100 datasets is similar to CIFAR10 dataset and has 100 class labels. Fig. 2 illustrates the diverse classes in the CIFAR10 dataset and ten random images from each one of the classes.

V. RESULTS AND ANALYSIS

To empirically evaluate the performance of different optimization techniques the experimentation has been conducted over the deep CNN model. Using a large model and dataset, the performance is measured in terms of accuracy and loss rate. The classification accuracy is presented in Table I and Table II over CIFAR 100 and CIFAR 10 dataset respectively. From the tables it is observed that the classification accuracy of Adam optimizer is higher than other stochastic optimization techniques *viz.* AdaGard, AdaDelta, SGD.

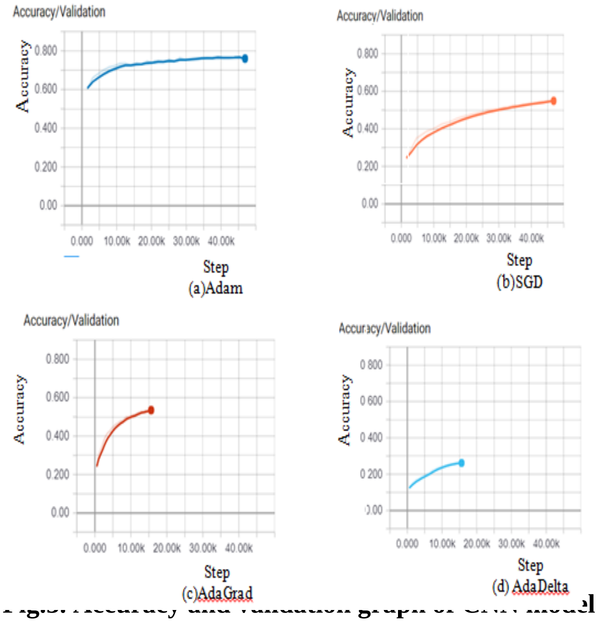
Table - I: Comparisons of optimization techniques over the CNN model with CIFAR 100

Classification Accuracy		
Optimizer	*Accuracy*	*Loss*
Adam	**90.55%**	9.5%
AdaGard	59.45%	40.55%
AdaDelta	24.19%	75.81%
SGD	59.91%	40.09%

Table - II: Comparisons of Optimization techniques over the CNN model with CIFAR 10

Classification Accuracy		
Optimizer	***Accuracy***	***Loss***
Adam	**90.55%**	9.5%
AdaGard	59.45%	40.55%
AdaDelta	24.19%	75.81%
SGD	59.91%	40.09%

The classification accuracy of different optimization methods after the hyperparameter tuning is shown in Table I and Table II respectively and from the tables it is seen that Adam shows best result in both the CIFAR 10 and CIFAR 100 datasets. Table III presents the performance of all the optimizers when the hyperparameters are set to an optimal value. Among all these hyperparameters, we considered the following three parameters during the experimentation, *viz.* epoch, learning rate, and batch size. The number of epoch considered is 15, 30 and 60 as the number of epochs is up surged till the gap between the test error and the training error is minimum. During the experimentations conducted a scale of 10 comparisons of learning rate has been carried out. However, in Table III only the optimal values have been shown for learning rate 0.001 and 0.002. For each learning rate of 0.001 and 0.002 the considered batch size is 32 and 64, the convnet is sensitive to batch size and in the learning process of convnet Mini-batch is generally desirable [13]. The range of 16 to 128 is a better choice to test with so the experimentation is carried out on that scale and the optimal results have been shown in Tale III with batch size 32 and 64. The experimentation is carried out on the CIFAR10 and CIFAR 100 dataset and from Table III, it is seen that the performance of Adam optimizer is best when the hyperparameters are set to an optimal value (highlighted in bold in Table III).



(a)Adam

(b)SGD

(c)AdaGrad

(d) AdaDelta

Fig.3. Accuracy and validation graph of CNN model using different optimization methods (a)Adam (b) SGD (c) AdaGrad (d) AdaDelta

The classification accuracy is increased when the value of hyperparameters is changed i.e. a good value of hyperparameter optimizes the network performance. In Table III. It is shown that the accuracy of Adam optimizer increases to 90.55 % at epoch 60, batch size 64 and learning rate 0.001. In Fig.3, the classification accuracy and validation of different optimizers are shown where Adam shows a better result as compared to other optimization methods *viz.* AdaGard, AdaDelta, SGD.

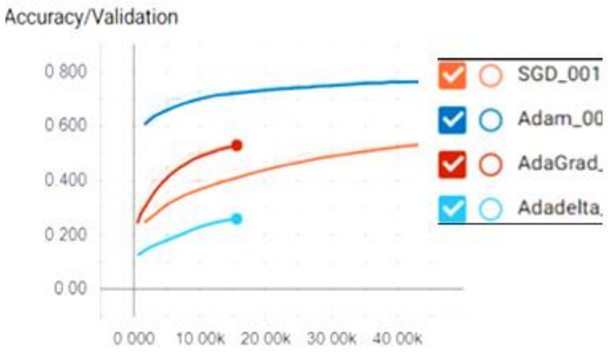


Fig.4.Comparison of Adam, SGD, AdaGrad, and Adadelta with respect to accuracy and validation over the CIFAR100 dataset.

In the above Fig.4 depicts the comparisons of all the optimizers *viz.* Adam, SGD, AdaGard, and AdaDelt over CIFAR 100 dataset where Adam shows better results compared to other optimization methods.

VI. CONCLUSION

This paper presents an empirical analysis of optimization techniques in a supervised architecture of the deep **CNN** model. Experimentation has been carried out to analyze the effect of different optimizers and the model has been tested over the CIFAR 10 and CIFAR 100 datasets. Along with the optimization methods, experimentations are conducted to identify optimal value for hyperparameters and to understand the rate of convergence of the optimization methods in high dimensional parameter spaces. From the experiments, it may be concluded that compared to other optimization techniques *viz.* AdaGard, AdaDelta and SGD - Adam shows better performance in nonconvex optimization problems. The experiments conducted confirm that classification accuracy can be increased by choosing the best optimization techniques with sophisticated hyperparameter tuning to get the optimal configuration of the deep CNN model. However, for general applicability further experimentation is required on different benchmark datasets.

Table - III: Comparison of Optimizers with hyperparameter tuning over CIFAR100 dataset

Table. 3 Comparison of Optimizers with hyperparameter tuning over CIFAR100 dataset

Optimizer	**Epoch**	**Learning Rate**	**Batch Size**	**Accuracy**
Adam	15	0.001	32	76.32
			64	77.31
		0.002	32	74.11
			64	74.12
	30	0.001	32	80.49
			64	80.60
		0.002	32	76.23
			64	76.50
	60	**0.001**	32	80.51
			64	**90.55**
		0.002	32	75.54
			64	75.60
AdaGrad	15	0.001	32	56.72
			64	56.80
		0.002	32	55.12
			64	55.23
	30	0.001	32	57.34
			64	58.45
		0.002	32	54.67
			64	56.89
	60	**0.001**	32	59.21
			64	**59.45**
		0.002	32	54.80
			64	55.76
SGD	15	0.001	32	53.12
			64	54.11
		0.002	32	53.15
			64	54.67
	30	0.001	32	58.45
			64	58.90
		0.002	32	57.45
			64	57.78
	60	**0.001**	32	59.90
			64	**59.91**
		0.002	32	57.12
			64	57.16
AdaDelta	15	0.001	32	22.13
			64	23.12
		0.002	32	22.11
			64	22.13
	30	0.001	32	24.15
			64	24.17
		0.002	32	22.11
			64	22.14
	60	**0.001**	32	24.18
			64	**24.19**
		0.002	32	23.89
			64	23.65

REFERENCES

1. Hinton, Geoffrey,. "Deep neural networks for acoustic modeling in speech recognition." IEEE Signal processing magazine 29 (2012).
2. Krizhevsky, Alex, and Geoffrey Hinton. Learning multiple layers of features from tiny images. Vol. 1. No. 4. Technical report, University of Toronto, 2009.
3. Deng, Li, et al. "Recent advances in deep learning for speech research at Microsoft." 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013.
4. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
5. Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." science 313.5786 (2006): 504-507.
6. Hinton, Geoffrey, et al. "Deep neural networks for acoustic modeling in speech recognition." IEEE Signal processing magazine 29 (2012).
7. Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013.
8. Duchi, John, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." Journal of Machine Learning Research 12.Jul (2011): 2121-2159.
9. Zeiler, Matthew D. "ADADELTA: an adaptive learning rate method." arXiv preprint arXiv:1212.5701 (2012).
10. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
11. LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436-444.
12. Goodfellow, I., Y. Bengio, and A. Courville. "Deep Learning (Book in preparation)." (2016).
13. Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." Journal of Machine Learning Research 13.Feb (2012): 281-305.
14. Larochelle, Hugo, et al. "An empirical evaluation of deep architectures on problems with many factors of variation." Proceedings of the 24th international conference on Machine learning. ACM, 2007.

AUTHORS PROFILE

Ms. Munmi Gogoi pursed Bachelor of Computer Application from Punjab Technical University in 2010 and Master of Computer Application from Dibrugarh University, Assam, India, in the year 2013. She is currently pursuing Ph. D. degree in Department of Computer Sciences, Assam University, Silchar, India, since 2015. Her research work focuses on Artificial Neural Network and Deep Learning.

Dr. Shahin Ara Begum pursed her Bachelor of Science from Bangalore University in 1994 and Master of Science from Jamia Millia Islamia University, India, in 1997. She has pursued her Ph. D. from Assam University, Silchar and currently working as Associate Professor in the Department of Computer Science, Assam University Silchar, India. She has published more than 30 research papers in reputed international journals including Thomson Reuters (Scopus & Web of Science) and conference papers of repute. Her research work focuses on Machine Learning and Soft Computing Techniques.