Academic Journal of Nawroz University (AJNU), Vol.9, No.4, 2020

https://doi.org/10.25007/ajnu.v9n4a858

A New Optimizer for Image Classification using Wide ResNet (WRN)

[1] Arman I. Mohammed, [2] Ahmed AK. Tahir

[1,2] College of Science, Duhok University, Kurdistan Region, Iraq

ABSTRACT

A new optimization algorithm called Adam Meged with AMSgrad (AMAMSgrad) is modified and used for training a convolutional neural network type Wide Residual Neural Network, Wide ResNet (WRN), for image classification purpose. The modification includes the use of the second moment as in AMSgrad and the use of Adam updating rule but with $\epsilon = 10^{-1}$ and (2) as the power of the denominator. The main aim is to improve the performance of the AMAMSgrad optimizer by a proper selection of ϵ and the power of the denominator. The implementation of AMAMSgrad and the two known methods (Adam and AMSgrad) on the Wide ResNet using CIFAR-10 dataset for image classification reveals that WRN performs better with AMAMSgrad optimizer compared to its performance with Adam and AMSgrad optimizers. The accuracies of training, validation and testing are improved with AMAMSgrad over Adam and AMSgrad. AMAMSgrad needs less number of epochs to reach maximum performance compared to Adam and AMSgrad. With AMAMSgrad, the training accuracies are (90.45%, 97.79%, 99.98%, 99.99%) respectively at epoch (60, 120, 160, 200), while validation accuracy for the same epoch numbers are (84.89%, 91.53%, 95.05%, 95.23). For testing, the WRN with AMAMSgrad provided an overall accuracy of 94.8%. All these accuracies outrages those provided by WRN with Adam and AMSgrad. The classification metric measures indicate that the given architecture of WRN with the three optimizers performs significantly well and with high confidentiality, especially with AMAMSgrad optimizer.

KEYWORDS: Adam, AMSgrad, CNN, Deep learning algorithm, Deep neural networks, Image classification, Optimization algorithms, Wide ResNet.

1. Introduction

Image classification involves detection or/and identification of an object or attributes in a digital image[1]. Image classification has become an important tool in many applications that are based on computer vision and artificial intelligence such as medical imaging, security[2-5], authentication[6-8] and military surveillance[9-11].

In recent years, the approach of deep convolutional neural networks (CNN) of sequential type has dominated the field of image classification and becoming superior to the traditional approach of hand-crafted features[12-14]. In contrast to hand-crafted features, deep CNNs could learn rich highly abstract image features from the training dataset of large scale images to represent complex objects in an efficient way and can be faster. Many architectures of sequential CNN ranging from deep CNN to very deep CNN have already been developed and used for image classification [12, 15, 16].

Generally speaking, developing a CNN that performs well for image classification require a proper selection of CNN architecture, optimization algorithm, training parameters, training data etc.

More recently, a new type of CNN which is known as computational graph CNN was introduced. CNNs of this type are shown to be fast and more accurate for image classification than the sequential type of CNN. Examples of computational graph CNN are GoogleNet (Inception-v1) which was introduced by[17] and the more sophisticated one is the Wide Residual Network, Wide ResNet (WRN) which was introduced by[18].

In addition, many optimization algorithms that are based on gradient decent have been developed in order to enhance the performance of CNN for image classification. The most common of these algorithms that

have been used widely for image classification are: Stochastic Gradient Descent with Momentum (SGD with Momentum)[19, 20], Adaptive Subgradient Method (AdaGrad)[21], Adaptive Moment Estimation (Adam)[22], Adaptive Method Setup Gradient (AMSgrad)[23].

The objective of this paper is to develop a new optimizer called Adam Merged with AMSgrad (AMAMSgrad) for learning a WRN for the purpose of image classification using CIFAR-10 dataset. The major aim is to improve the performance of the classification system. In doing so, three models of classifications are implemented. One model is implemented with the new optimizer (AMAMSgrad) and the other two models are implemented with Adam and AMSgrad optimizers.

The remainder of this paper is organized as follows. In Section 2, a brief review of the related work is given. In Section 3, a description of the WRN architecture is given. In section 4, a detailed description of AMAMSgrad optimizer is presented. In section 5, training of Wide ResNet is given. In section 6, the results of applying three optimizers (AMAMSgrad, Adam and AMSgrad) on a Wide ResNet using CIFAR-10 dataset are presented and discussed. In section 7, the conclusions are given.

2. RELATED WORKS

In CNNs, Back-Propagation (BP) is used for training. The algorithm of BP performs two passes. In the first pass, errors are calculated from the output layers. In the second pass, these errors are used to calculate the gradient of the loss function then this gradient is propagated back to the optimizer which in turn uses it to update weights in such a way to minimize the loss function. Thus the main aim of the optimizer is to find the optimal minima of the gradient which indicates the convergence of the training process in CNNs and it has no tasks throughout the testing mode of CNNs. Therefore, the approach of optimization algorithms has brought the attraction of many researchers and much of research works have been accomplished to improve the performance of convolutional neural networks CNNs via a proper design of efficient optimizers. So far, many optimization algorithms have been developed and used successfully for implementing CNNs with various architectures. Examples of these optimization algorithms that have been used for image classification are: Stochastic Gradient Descent with Momentum (SGD with Momentum)[19,20], Adaptive Subgradient Method (AdaGrad)[21], Adaptive delta (Adadelta)[24], Root Mean Square Propagation Optimization (RMSProp)[25], Adaptive Moment Estimation (Adam)[22], Adaptive Method Setup Gradient (AMSgrad)[23], Adam with decoupled weight decay (AdamW)[26], Quasi-Hyperbolic Momentum (QHAdam)[27] and AdaptAhead[28]. However, the most commonly used of these algorithms are the SGD-Momentum, AdaGrad, RMSProp, Adam and AMSgrad, all of which are categorized as gradient descent methods. In 1986, Rumelhart and others[19] developed the SGD-Momentum algorithm which was well interpreted later by Qian in 1999[20]. The aim of SGD -Momentum was to dampen the high oscillations in the error function that occurs in the classical SGD in order to find the global minima in a faster time. It uses the gradient of the current iteration and the accumulated gradients of the previous iteration which acts as a momentum. This optimizer, SGD-Momentum, was used successfully to train different deep and very deep CNNs for image classification such as AlexNet[12] and VGG-19[17].

In 2011, Duchi and others[21] introduced a per-parameter learning rate optimizer called AdaGrad to be used effectively with sparse data. In this optimizer, the step size for each parameter is scaled according to the history of gradients for that parameter, which is done by

dividing the current gradient by the sum of previous gradients. However, AdaGrad suffers two main drawbacks. The first drawback is the decay of learning rate as the accumulated gradient becomes large. The second drawback is the need for a manually selected global learning rate. These two drawbacks were considered by[22] in modifying the Adadelta optimization algorithm. In Adadelta, the advantages of moment and AdaGrad by scaling the step size according to the last time historical gradient and using a component that acts as an accelerator (momentum). The problem of step size vanishing in AdaGrad was also considered by Tieleman and Hinton in 2012[25] in developing the RMSProp optimization algorithm. They tackled the problem of step size vanishing by using the decaying average of history gradients instead of the sum. For better tackling of the step size vanishing problem[22] introduced Adam optimizer by combining the advantages of AdaGrad method, which works well with sparse gradients, and RMSProp method, which works well in on-line and non-stationary settings. This method of optimization is based on computing the adaptive step for each parameter (weight) by using the estimations of first and second moments of gradient to adapt the learning rate for each weight of the neural network. The first moment uses decaying average of history gradients similar to AdaGrad, whereas the second moment emphasis uses the current gradient exactly like RMSProp. However, for some applications such as image classification and object recognition, it has been noticed by[29,30] that the adaptive learning rate optimizers such as Adam may fail to convergence and may not able to find the optimal minima. To overcome this problem, Reddi and others in 2018[23] modified AMSgrad algorithm from Adam algorithm by changing the rule of computing the second moment. The authors of AMSgrad pointed out that the reason of Adam failure is the improper use of the exponential moving average (second moment). They observed that during the training, some mini-batches provide larger and more informative gradient than others. They attributed this variability in the mini-batches gradients to the use of the second moment of the current iteration for adapting the learning rate for each weight update. In order to minimize the influence of the variability in the mini-batches gradients, they took the maximum of the previous and current second moments. By doing so, the authors of AMSgrad claimed that they achieved better performance by AMSgrad optimizer compared to Adam optimizer.

Finally, the field of optimization algorithm requires continuous investigation and more research work in order to enhance the performance of CNNs for various applications of computer vision and artificial intelligence, in specific for image classification.

In this paper, an attempt is made to modify a new optimization method from Adam and AMSgrad that is capable of improving the performance of CNNs for image classification.

3. WIDE RESNET WRN ARCHITECTURE

Basically, there are two types of CNN architectures which are sequential and computational graphs. The architecture that was introduced in chapter four is one type of the sequential architectures when the layers were stacked one above other. While in computational graphs each layer is a representative of mathematical operations, the advantages of these computational graphs are to compute more accurate gradient, it helps for parallel processing on GPUs much easier. The main difference between these two types is that in computational graphs the data from input may flow to be added to several next layers to preserve the resolution

i.e. gradient. The examples of some architectures of the computational graphs type are, GoogleNet (Inception-v1)[17], ResNet[31], Deep Recursive ResNet[32] and wide ResNet[18].

However in ResNets, accuracy improvement even by a fraction of a percent requires doubling the number of layers. Therefore, training very deep residual networks will face the problem of diminishing feature reuse, which makes these networks very slow to train. In order to solve this problem[18] proposed a novel architecture called the Wide Residual Networks (WRNs). Oppose to the ordinary ResNet which is characterized as being thin and very deep, in WRNs the depth of the network is decreased while the width is increased. In other words, the number of layers in WRNs is decreased and the number of filters is increased. Zagoruyko and Komodak have proved that this architecture will make the training process faster and can solve the problem of diminishing feature reuse. In addition, they made the structure such that the Batch Normalization layer and ReLU activation function precede each convolution layer. The architecture of the Wide ResNet as proposed by 18] is shown in figure (1).

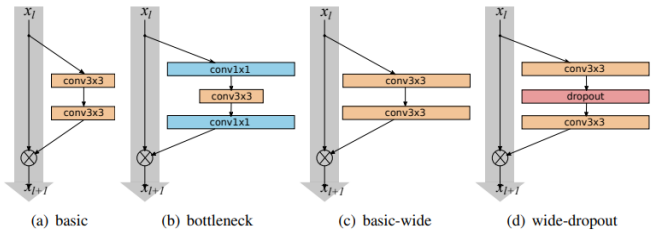

Fig. 1. Various blocks of Wide ResNet[18].

In this work, the basic-wide block is used within this Wide ResNet, the bias of convolution and dropout layer were not used. The network of 16x8 is used that means this net is 16 layer deep and has the widen factor of 8. The total number of parameters using equations (3.2) is (10, 968, 570), the trainable parameters are (10, 961, 370) and non-trainable parameters are (7,200) which are the batch normalization parameters. The blocks of different wide ResNet can be seen in figure (1). The supper convergence is also used without any modifications.

4. THE PROPOSED AMAMSGRAD OPTIMIZER

Despite AMSgrad was proposed to overcome the problem of finding the global minima in Adam, some recent works such as[33] proved that even after the modification, the performance of AMSgrad did not outrage Adam. In addition, making the step size larger as an aim to avoid learning rate decaying as in Adam and AMSgrad may lead to the problem of bypassing the global minima. Moreover, most of the modifications that have been done to the previous methods of optimization were restricted to the way of using the history gradient and the current gradients in order to scale the step size so that to prevent the decay of the learning rate, while the form of the denominator in the mathematical formula for calculating Δw_t, in specific the effect of the term (ϵ) and the power of the denominator on the step size have been ignored.

In this paper, an attempt is made to show graphically the impact of the denominator in each of Adam and AMSgrad on the process of updating the learning rate, in specific how the value of (ϵ) and the power of the denominator affect the weight change Δw_t. Then a new method of optimization called Adam Merged with AMSgrad (AMAMSgrad) will be modified with the aim to scale the step size in such a way to prevent learning rate decaying, while at the same time preventing the bypass problem of the global minima. The modifications are based on: First, merging the AMSgrad optimizer with the equation of calculating Δw_t of Adam optimizer. Second, changing the value of ϵ and the power of the denominator of the weight update formula of Adam. The value of ϵ is changed from 10-8 to 10-1 and the denominator of the weight update formula in Adam is

changed from $(\sqrt{\hat{v}_t}+\epsilon)$ to $(\sqrt{\hat{v}_t}+\epsilon)^2$, while the procedures of calculating the first and second moment are made as in AMSgrad.

The pseudo code of AMAMSgrad is given in table (1) below:

In order to describe how AMAMSgrad can perform better than AMSgrad and Adam, first the equations of weight update in both Adam and AMSgrad are reformulated to modulation functions without losing their mathematical meaning and second the effect of ϵ and the power of the denominator on the weight update are shown. The detailed descriptions are given in the following subsections.

TABLE 1

The pseudo Code of AMAMSgrad Optimizer with Criteria Definitions

	Pseudo Code		Parameter definition
1	**Input**: w, ϵ, η, $\beta1$, $\beta2$	w	weight
2	**Initialize**:$m_t = 0, v_t = 0, t = 0, \hat{v}_t = 0$	η	Learning rate
3	**While** w not converged **do**	ϵ	Small value = 10^{-8}
4	$t = t + 1$	t	Iteration number
5	$g_t = \nabla f_t(w_t)$	g_t	Gradient at iteration t
6	$m_t = \beta_1 . m_{t-1} + (1-\beta_1).g_t$	∇f_t	Computational gradient function
7	$v_t = \beta_2 . v_{t-1} + (1-\beta_2).g_t^2$	m_t	First moment
8	$\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$	$\beta1$	Hyperparameter (Decay rate=0.9)
9	$\Delta w_t = -\frac{\eta}{(\sqrt{\hat{v}_t}+\epsilon)^2}.m_t$	$\beta2$	Hyperparameter (Decay rate=0.99)
		v_t	Second moment
10	$w_t = w_{t-1} + \Delta w_t$	$\hat{v}_t$	Biased corrected second moment
		Δw_t	Weight change
11	**End while**		

4.1 Weight Update Formula

According to[22, 23] the two formulas for calculating Δw_t in Adam and AMSgrad are given below:

For Adam $$\Delta w_t = -\frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}.m_t \quad (1)$$

For AMSgrad $$\Delta w_t = -\frac{\eta}{\sqrt{\hat{v}_t}}.m_t \quad (2)$$

Where, Δw_t is the update in weight, η is the learning rate, m_t is the first moment, $\hat{v}_t$ is the second moment and ϵ is a small value to avoid the division by zero.

Equations (1 and 2) can be re-written as follow:

For Adam $$\Delta w_t = -\eta . M_1 . M_2 \quad (3)$$

For AMSgrad $$\Delta w_t = -\eta . M'_1 . M'_2 \quad (4)$$

Where, (M_1 and M'_1) are the first modulation functions in the two optimizers such that:

$$M_1 = \frac{1}{\sqrt{\hat{v}_t}+\epsilon} \quad (5)$$

$$M'_1 = \frac{1}{\sqrt{\hat{v}_t}} \quad (6)$$

The modulation functions (M_2 and M'_2) are less influential than (M_1 and M'_1) on weight update and will be discarded from the description.

The effect of ϵ and the power of the denominators on these modulation functions and how they affect the weight update will be described in the forthcoming subsections.

4.2 The Effect of ϵ

According to equations (5 and 6), the two modulation functions differ by the term ϵ. In the original algorithm of Adam, ϵ was taken very small (10-8) in order to avoid the division by zero[22]. Figure (2) shows graphically the difference between M_1 and M'_1 for two values of ϵ.

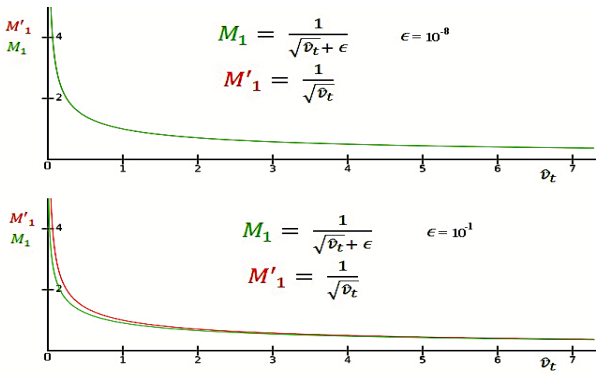

Fig. 2. Modulation functions of Adam and AMSgrad at different ϵ

In both parts of figure (2), the green color represents the modulation function of Adam and the red color represents the modulation function of AMSgrad.

According to the lower part of the figure it can be seen that adding a value of ϵ will shift the curve horizontally to the left and when ϵ =0 both Adam and AMSgrad coincide perfectly.

In the upper part, when $\epsilon = 10^{-8}$, both modulation functions have the same effect, the green and red colors are almost coincide perfectly.

In the lower part of figure (2), When $\epsilon = 10^{-1}$, the modulation function of AMSgrad is slightly higher than that of Adam at $\hat{v}_t > 2.5$ and becomes significantly higher than that of Adam at $0 < \hat{v}_t < 2.5$. For the same $\hat{v}_t$, the value of M'1 for AMSgrad is larger than that of M1 for Adam. This indicates that the changes on Δw_t that can be made by AMSgrad are higher than those made by Adam. Knowing that increasing Δw_t by large amount may lead to overpass the global minima, then it can be said that with AMSgrad optimizer there will be more chance to overpass the global minima than with Adam. In other words, increasing the value of ϵ will minimize the chance for the optimizer to overpass the global minima.

In the upper part, when $\epsilon = 10^{-8}$, both modulation functions have the same effect, the green and red colors are almost coincide perfectly.

4.3 The Effect of the Power of the Denominator

To show the effect of changing the power of the denominator for both modulation functions of Adam and AMSgrad, Three cases are considered with constant $\epsilon = 10^{-1}$. These are, the modulation functions of Adam, AMSgrad and AMAMSgrad (suggested optimizer) which is the same as Adam but the power of the denominator is raised to (2) as shown in the equation (7). The differences between these three functions are shown graphically in figure (3).

Modulation function for AMAMSgrad

$$MM_1 = \frac{1}{(\sqrt{\hat{v}_t} + \epsilon)^2} \qquad (7)$$

Figure 3 shows the values of modulation function versus $\hat{v}_t$ for the three optimizers.

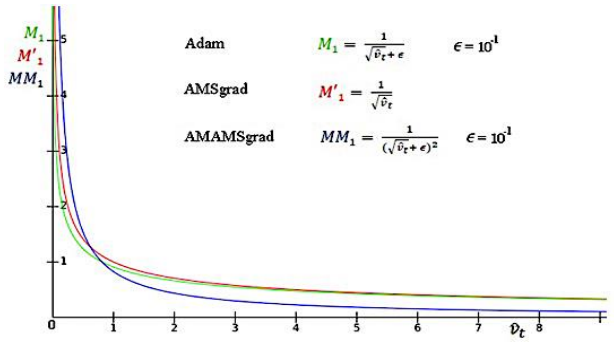

Fig 3. Modulation Functions of Adam and AMSgrad and AMAMSgrad.

In this figure, the modulation function of Adam and AMSgrad are shown in green and red color, while the modulation function of AMAMSgrad is shown in blue. The comparisons between the three modulation functions of Adam and AMSgrad and AMAMSgrad are shown below:

In this figure, the modulation function of Adam and AMSgrad are shown in green and red color, while the modulation function of AMAMSgrad is shown in blue. The comparisons between the three modulation functions of Adam and AMSgrad and AMAMSgrad are shown below:

i. At $\hat{v}_t > 1$ the modulation functions of Adam and AMSgrad have the same value which is greater than the value of the AMAMSgrad modulation function. The green and red curves for Adam and AMSgrad which are very close to each other at $\hat{v}_t > 1$. Accordingly, it can be said that the changes on Δw_t that is made by the modulation function of AMAMSgrad is less than those made by the modulation functions of Adam and AMSgrad. Keeping in mind that increasing Δw_t may lead to overpass the global minima, then it can be said that with AMAMSgrad optimizer there will be less chance to overpass the global minima compared to Adam and AMSgrad.

ii. In term of speed, two scenarios exist. In the first, the small changes in Δw_t that are made by AMAMSgrad modulation function may slow down the process of training. In the second scenario, the small changes in Δw_t (step size) that are made by AMAMSgrad will provide better chance to find the global minima in less number of epoch compared to Adam and AMSgrad. In practice, the impact of the second scenario was shown to have dominated the process of training with AMAMSgrad as it achieved the beset accuracy in less number of iterations compared to Adam and AMSgrad which were slower and achieved the best accuracy in more number of iterations.

iii. At $\hat{v}_t = 1$, figure (3) shows that the modulation functions of Adam and AMAMSgrad will have the same value. This can be shown mathematically as follows:

$$\frac{1}{\sqrt{\hat{v}_t} + \epsilon} = \frac{1}{(\sqrt{\hat{v}_t} + \epsilon)\text{^}2} \text{ only when } \sqrt{\hat{v}_t} = 1 - \epsilon \quad (8)$$

This means when $\epsilon = 0$ then the modulation functions of both Adam and AMAMSgrad will have the same value at $\hat{v}_t = 1$ and as $\epsilon > 0$ the value of Adam modulation function becomes higher than that of AMAMSgrad. In fact, the point at $\hat{v}_t = 1$ represents the turning point, at which the curve of the modulation function of AMAMSgrad in blue color looks as a clockwise-rotation of the modulation functions of Adam and AMSgrad in red and green color.

iv. At $0 < \hat{v}_t < 1$ the situation is reversed, the values of AMAMSgrad modulation function (blue color) is higher than those of Adam and AMSgrad. This means, at $0 < \hat{v}_t < 1$ the changes in Δw_t that are made by AMAMSgrad modulation function are higher than those made by the modulation functions of the Adam and AMSgrad. This will lead AMAMSgrad to help the process of training to continue without sticking in local minima.

5. TRAINING THE WIDE RESNET (WRN)

Training of the Wide ResNet is done using three models of image classification, one with AMAMSgrad optimizer and the other two with Adam and AMSgrad optimizers. The initial learning rate is set to (0.1) and it is scheduled for the rest of training as shown in table (2).

TABLE 2

Learning rate schedules.

Epochs	Learning Rate
1-60	0.1
61-120	0.02
121-160	0.004
161-200	0.0008

The value of ϵ for all models is taken as (0.1). All three models are trained on Google Colaboratory[34] on Tesla K80 GPU. The training time taken by each model for completing (200) epochs was around (11) hours. The mini-batch size is 128. The dataset CIFAR-10 is used, that contains 60000 images. This dataset is divided into two sets for training (50000 images) and for validation (10000 images). In addition, the data augmentation of type online augmentation (augmentation on the fly) is used. The images are flipped horizontally and shift in width and height shift within the range (0.125 of the input image size). Online augmentation does not change the batch-size. Thus the total number of weight updates (Iterations) is 50000/128 * 200 that is equal to (78, 125) updates. The weights are initialized with He-Normal[35], and L2 regularization of (0.0005) is used[36].

6. RESULTS AND DISCUSSIONS

The WRN is trained with CIFAR-10 dataset using three types of optimization, AMAMSgrad, Adam and AMSgrad. The evaluation metrics for the three models of classification are shown in the table (3). Also, graphs are

presented for the purpose of illustration. The detailed comparisons between the three models of classification are given in the following sections.

6.1 Training Accuracy

The results of training accuracy of the three models are given in figures (4 and 5) and table (3). According to these figures the following key points can be summarized and concluded:

i. According to table (3), the training accuracy of all the three models of classification at epoch (200) is 99.99. This is mainly due to: First, the use of Wide ResNet architecture which is very sophisticated network. Second, it is due to the use of schedule learning rate. Third, data augmentation is used.

ii. According to figures (4 and 5), the training accuracy of the model with AMAMSgrad optimizer starts to be higher than that of the other two models with Adam and MASgrad from epoch (5) and continues to be higher till the last epoch (200). Also, the two figures show that the accuracy curve of AMAMSgrad model is less oscillated compared to the models with Adam and AMSgrad. This indicates that the loss function values of AMAMSgrad are more stable than those of Adam and AMSgrad which can be attributed to step size of the modulation function of AMAMSgrad which is smaller than those of the modulation functions of Adam and AMSgrad as proven by figure (3).

iii. A comparison between the three models at different epoch number is shown in table (3). According to this table, the training accuracy of the model with AMAMSgrad optimizer at epochs (60, 120 and 160) are (90.45%, 97.79 and 99.98) respectively while for models with Adam and AMSgrad at the same epoch numbers are (87.52, 96.86 and 99.95) and (87.774%, 96.64 and 99.97) respectively. This result approves that AMAMSgrad achieves better accuracy than Adam and AMSgrad for the same epoch number, so it can be said that the model with AMAMSgrad reaches convergence faster than the other two models with Adam and AMSgrad. These comparisons can also be seen by inspecting figures (4 and 5). The accuracy of the model with AMAMSgrad starts to be higher than that of the other two models since epoch (5) till epoch (131).

TABLE 3
Training Evaluation Metrics for the three Models of classification

Method	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Time	Epoch No.
Adam	0.8752	0.9234	0.5774	2.8208	3.09	60
	0.9686	0.4176	0.8886	0.7118	6.18	120
	0.9995	0.1068	0.9428	0.3218	8.24	160
	0.9999	0.0820	0.9496	0.2706	10.3	200
AMSgrad	0.8774	0.9187	0.7895	1.2259	3.133	60
	0.9664	0.4249	0.9032	0.6519	6.266	120
	0.9997	0.1073	0.9480	0.3021	8.354	160
	0.9999	0.0831	0.9507	0.2640	10.443	200
AMAMSgrad	0.9045	0.7562	0.8489	0. 6764	3.133	60
	0.9779	0.3052	0.9153	0.5652	6.266	120
	0.9998	0.1169	0.9505	0.2599	8.354	160
	0.9999	0.0996	0.9523	0.2484	10.443	200

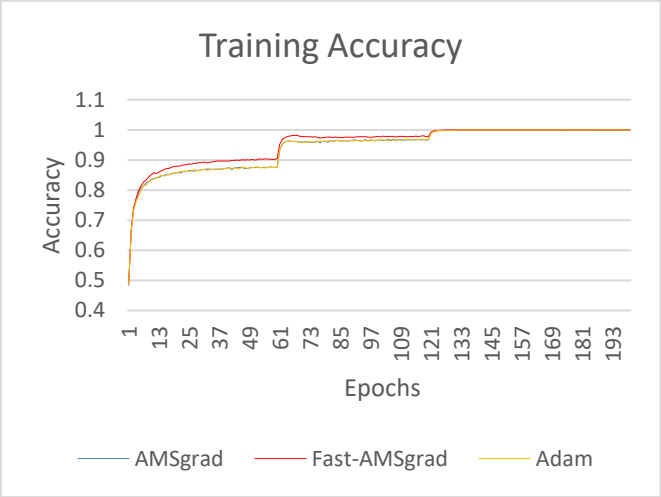

Fig. 4. Training accuracy of 200 epochs.

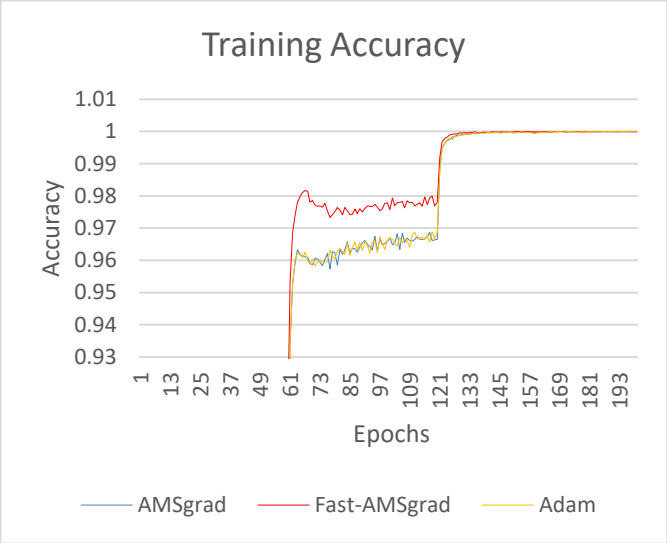

Fig. 5. A Magnified version of Fig. 4.

6.2 Training Loss

Figure (6) shows the loss values of the three models of classification. According to table (3) and figure (6), the increase in the loss value of the model with AMAMSgrad is slower than those of the other two models with Adam and AMSgrad for epoch (9) to epoch (160). This indicates that AMAMSgrad is handling the learning rate more adaptively than Adam and AMSgrad. This means that, if the schedule rate is not used, then the possibility of convergence failure could be more for Adam and AMSgrad than AMAMSgrad. Figure (6) also shows that the loss values of models with Adam and AMSgrad start to decrease more than that of the models with AMAMSgrad after epoch (160). However, this decrement did not lead to any improvement on the validation accuracy.

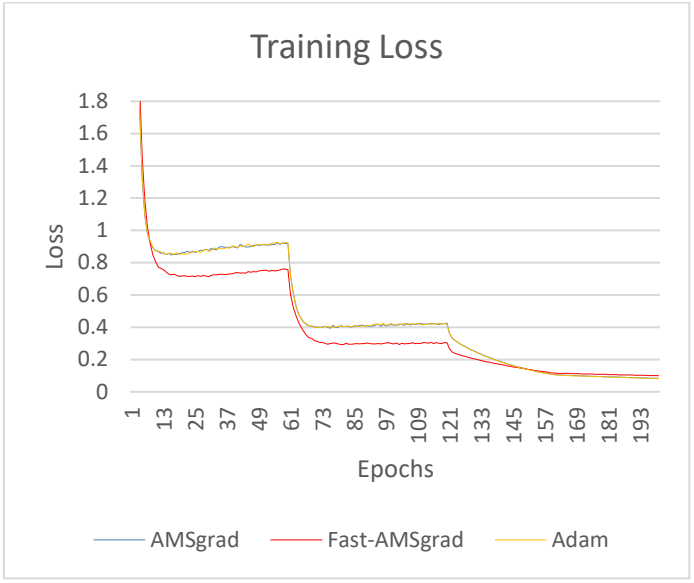

Fig. 6. Training loss of 200 epochs.

6.3 Validation Accuracy

The validation accuracy is an important metric that shows the level of performance for each model.

i. Figures (7 and 8) show the validation accuracies of the three models versus the epoch number. According to these figures and table (3), the validation accuracy curve of the model with AMAMSgrad shows more smoothness (less oscillation) than that of the other two models with Adam and AMSgrad. The oscillation of Adam and AMSgrad curve are severe between epochs (1-120).

ii. The validation accuracy of the model with AMAMSgrad at epoch number (61) is (92.89%) and continues to increase till 95.23% at epoch (200), while the accuracies of the models with Adam and AMSgrad are (91.02 and 91.9%) and continues to increase till (94.96 and 95.07%) at epoch (200), see table (3). That is, the validation accuracy is improved with the AMAMSgrad by 0.27 over Adam and by 0.16% over AMSgrad. While the improvement in AMSgrad model over Adam is 0.11%. These results indicate the efficiency of AMAMSgrad over the other two optimizers.

iii. As far as the training speed is concerned, figures (7 and 8) show how the validation accuracy is improved for each optimizer versus epoch numbers. According to these figures, AMAMSgrad can achieve around 95.00% of validation accuracy at epoch (134), while the same validation accuracy can be achieved by Adam and AMSgrad at epoch (162). This means that models with Adam and AMSgrad will require (20) more epochs of training in order to achieve the accuracy of AMAMSgrad model. But, this may cause high changes in the learning rate which may lead Adam and AMSgrad to stick in local minima. For example, at epoch number (160), AMAMSgrad optimizer has achieved (95.05%) of validation accuracy while AMSgrad has achieved this accuracy at epoch (200) and Adam did not reach this accuracy even at epoch (200). In term of time, the training time of (160) was (8.8) hours on GPU, while for (200) epoch was (11) hours. Thus, it can be concluded that AMAMSgrad has reduced the computations dramatically by saving (2.2) hours. In addition, the results revealed that as going from epoch to epoch, AMAMSgrad optimizer shows more accuracy stability than Adam and AMSgrad.

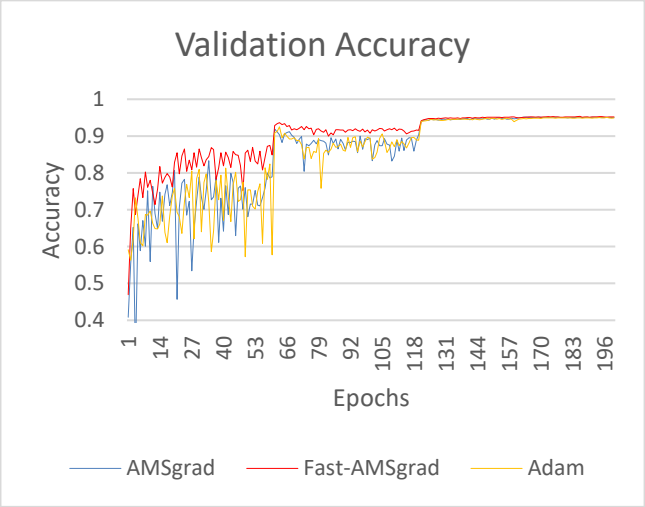

Fig. 7. Validation Accuracy of the AMAMSgrad and the MASgrad for 200 epochs.

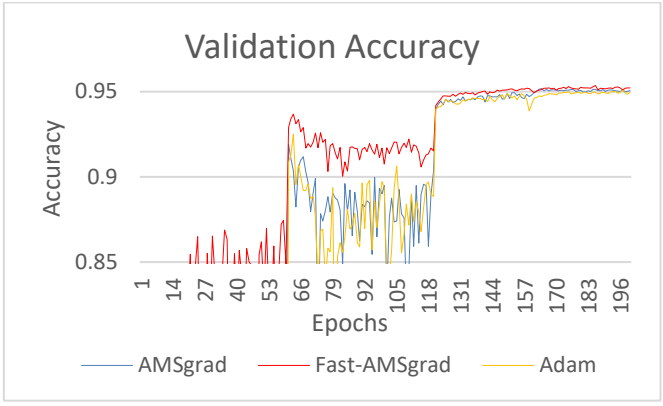

Fig. 8. A Magnified version of Fig. 7.

6.4 Validation Loss

The results of validation loss are presented in figure (9 and 10). It can be seen that the validation losses of models with Adam and AMSgrad are highly fluctuating from the beginning of training till epoch number (60) and these oscillations become smaller at later epochs but still much higher than the oscillations of the model with AMAMSgrad. Figure (10) shows that AMAMSgrad optimizer (Red color) has less loss than Adam and AMSgrad optimizer (Blue and Yellow colors) during all the stages of the training process. It can be seen that the validation loss of AMAMSgrad model is continuously decreasing. For instance, the validation loss of the model with AMAMSgrad at epoch (200) is (0.2484), while the validation loss of models with Adam and AMSgrad at epoch (200) are (0.2706 and 0.2640) respectively. This is a good indicator that the AMAMSgrad can find better global minima and converge much faster than AMSgrad optimizer.

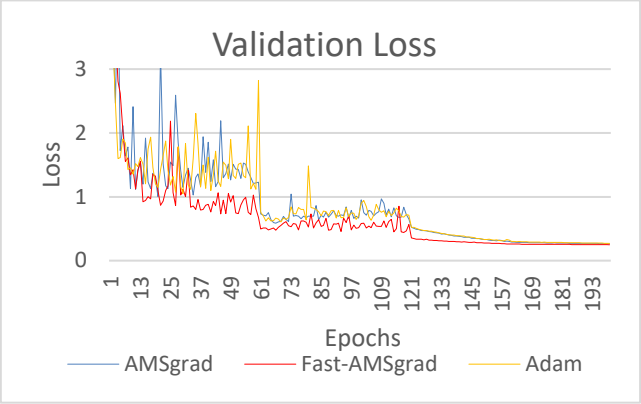

Fig. 9. Validation loss of 200 epochs.

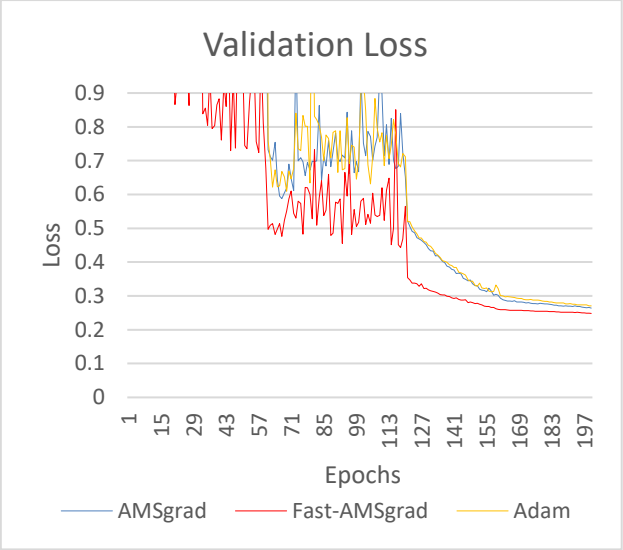

Fig. 10. Magnified Version of Fig. 9.

6.5 Testing Results

For testing mode, 2000 images from the validation data were randomly chosen and fed to the three models for classification. After predicting the labels of 2000 images these labels with their corresponding true labels are fed to confusion matrix to produce the matrix that was used to compute TP, TN, FP and FN. Thereafter, these variables are used to produce the performance measurements such as overall accuracy, error, precision and kappa.

Table (4) shows the true and false predictions for the three models. This table shows that the model with AMAMSgrad achieved the higher true prediction.

TABLE 4

True and False Acceptance of the samples computed from confusion matrix.

Method	TP	TN	FP	FN
Adam	188.5	1788.5	11.5	11.5
AMSgrad	188.8	1788.8	11.2	11.2
AMAMSgrad	**189.6**	**1789.6**	**10.4**	**10.4**

In order to evaluate and compare the performances of the three classification models, the classification metrics: overall accuracy, precision, error and Kappa coefficient are calculated for the three models[37] and given in table (5).

TABLE 5

Performance Evaluation Metrics of the Testing Samples for the Three Models of Classification

Method	Overall Accuracy %	Precision	Error	Kappa Coefficient
Adam	94.25	0.92.6	0.0574	0.9361
AMSgrad	94.40	0.9443	0.0560	0.9377
AMAMSgrad	**94.80**	**0.9486**	**0.0520**	**0.9422**

According to table (5), the performance of the model with AMAMSgrad is better than the performances of the two models with Adam and AMSgrad. The overall accuracy of the model with AMAMSgrad is improved by 0.55% over the model with Adam and by 0.4% over the model with AMSgrad. While the improvement achieved by the model with AMSgrad over that with Adam is only 0.15%. These results indicate that AMAMSgrad performs more efficiently than the other two optimizers. Also for the AMAMSgrad, the error is lower and the Kappa measure is higher which indicate the confidentiality of the results.

7. CONCLUSIONS

The results have shown that the value of ϵ and the power of the denominator of the updating rule equation have crucial effect on the performance of the optimizer during

the training mode of CNNs. This was evident from the results of the three optimizers, AMAMSgrad, Adam and AMSgrad. In AMAMSgrad, taking the value of $\epsilon = 10^{-1}$ and the power of (2) for the denominator have led the performance of AMAMSgrad to outrage those of Adam and AMSgrad. Increasing ϵ and the power of the denominator in AMAMSgrad make the weigh change (step size) smaller than those for Adam and MASgrad, thus reducing the possibility for bypassing the global minima. In addition, the training and validation accuracy as a function of epoch number that were achieved by AMAMSgrad were better than those of Adam and AMSgrad. Moreover, the classification metrics including overall accuracy, precision, error and Kapa coefficient of the testing mode for AMAMSgrad were improved over those of Adam and AMSgrad.

8. Acknowledgements

- This research work was implemented at the Computer Science Department / College of Science / University of Duhok as a part of the Master degree requirements, 2016-2019.
- The authors would like to express their sincere gratitude to the University of Duhok and the College of Science for their continuous support to make this work possible.
- The first author wants to use this opportunity to express his special thanks to the Duhok Polytechnic University for its invaluable support.

9. REFERENCES

1. Badrinarayanan, V., Kendall, A., and Cipolla, R., 2017, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation", IEEE Transactions on Pattern Analysis And Machine Intelligence, Vol. 39, No. 12, Pp. (2481-2495).
2. Borges, L. R., 2015, "Analysis of the Wisconsin Breast Cancer Dataset and Machine Learning for Breast Cancer Detection", Proceedings of XI Workshop de Visão Computacional - October 05th-07th, 2015, Pp. (15-19).
3. Khan, A. A., and Yong, S., 2016, "An Evaluation of Convolutional Neural Nets for Medical Image Anatomy Classification", Springer International Publishing Switzerland 2016, in P.J. Soh et al. (eds.), Advances in Machine Learning and Signal Processing, Lecture Notes in Electrical Engineering, DOI 10.1007/978-3-319-32213-1_26, Pp. (293-303).
4. Haryanto, T., Wasito1, I. and Suhartanto, H., 2017, "Convolutional Neural Network (CNN) for Gland Images Classification", International Conference on Information & Communication Technology and System (ICTS), Pp. (55-60).
5. Kapoor, I., and Mishra, A., 2018, "Automated Classification Method for Early Diagnosis of Alopecia Using Machine Learning", International Conference on Computational Intelligence and Data Science (ICCIDS 2018), ESEVIER, ScienceDirect, Pp. (437-443).
6. Antipov, G., Berrani, S.A., Ruchaud, N. and Dugelay, J.L., 2015, October. "Learned vs. hand-crafted features for pedestrian gender recognition". In Proceedings of the 23rd ACM international conference on Multimedia. pp. (1263-1266).
7. Akcay, S., Kundegorski, M. E., Willcocks, C. G., and Breckon, T. P., 2018, "Using Deep Convolutional Neural Network Architectures for Object Classification andDetection within cX-ray Baggage Security Imagery", IEEE Transactions on Information Forensics and Security,DOI 10.1109/TIFS.2018.2812196,Pp. (1-13).
8. Bian, P., Li, W., Jin, Y., and Zhi, R., 2018, "Ensemble feature learning for material recognition with convolutional neural networks", EURASIP Journal on Image and Video Processing, 2018:64, Pp. (1-11).
9. Zuo, J., Xu, G., Fu, K., Sun, X., and Sun, H., 2018, "Aircraft Type Recognition Based on Segmentation With Deep Convolutional Neural Networks", IEEE Geoscience And Remote Sensing Letters, Vol. 15, No. 2, Pp. (282-286).
10. Xu, H., Han, Z., Feng, S., Zhou, H., and Fang, Y., 2018 "Foreign object debris material recognition based on convolutional neural networks", EURASIP Journal on Image and Video Processing, https://doi.org/10.1186/s13640-018-0261-2, 2018:21, Pp. (1-10).
11. Wan, J., Chen, B., Xu, B., Liu, H., and Jin, L., 2019, "Convolutional neural networks for radar HRRP target recognition and rejection", EURASIP Journal on Advances in Signal Processing, https://doi.org/10.1186/s13634-019-0603-y, 2019:5, Pp. (1-17).
12. Krizhevsky A., Sutskever I. and Hinton G.E., 2012, "ImageNet classification with deep convolutional neural networks", Proceedings of the 25th International Conference on neural information processing systems (NIPS), Lake Tahoe, December, pp. (1097-1105).
13. Zhang P., Niu X., Dou Y., and Xia F., 2017, "Airport

Detection on Optical Satellite Images Using Deep Convolutional Neural Networks", IEEE Geoscience and Remote Sensing Letters, Vol. 14, No. 8, pp. (1183–1187).

14. Hoseini F., Shahbahrami A. and Bayat P., 2018, "An Efficient Implementation of Deep Convolutional Neural Networks for MRI Segmentation", Journal of Digital Imaging, Vol. 31, No. 5, pp. (738-747).
15. Zeiler M.D., Fergus R., 2014, "Visualizing and Understanding Convolutional Networks",in Fleet D., Pajdla T., Schiele B., Tuytelaars T., (eds) Computer Vision – European Conference on.
16. Simonyan, K. and Zisserman A., 2015, "Very deep convolutional networks for large-scale image recognition", International Conference on Learning Representations (ICLR), (pp. 1409.1556).
17. Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V. and Rabinovich, A., 2015, "Going deeper with convolutions", in Proceedings of the IEEE conference on computer vision and pattern recognition, (pp. 1-9).
18. Zagoruyko, S., and Komodakis, N., 2017, "Wide Residual Networks", rXiv:1605.07146v4, Pp. (1-15).
19. Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1986. "Learning representations by back-propagating errors". Cognitive modeling, Nature, LETERSTO NATURE, Vol. 323, PP (533-536).
20. Qian, N. 1999, "On the momentum term in gradient descent learning algorithms", Neural Networks, ELSEVIER, Vol. 12, Issue 1, (pp. 145–151).
21. Duchi J., Hazan E. and Singer Y., 2011, "Adaptive subgradient methods for online learning and stochastic optimization", Journal of Machine Learning Research, 12(Jul), pp. (2121-2159).
22. Kingma, D. P., and Ba, J. L., 2015, "Adam: A Method for Stochastic Optimization", in Proceedings of the International Conference on Learning Representations (ICLR), pp. (1-15).
23. Reddi S. J., Kale S. and Kumar S., 2018, "On the Convergence of Adam And Beyond", Proceedings of the International Conference on Learning Representations (ICLR), pp. (1-23).
24. Zeiler M. D., 2012, "Adadelta: An Adaptive Learning Rate Method", arXiv preprint arXiv, pp. (1212-5701).
25. Tieleman T. and Hinton G., 2012, "Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude", COURSERA: Neural Networks for Machine Learning, 4, pp. (26-31).
26. Loshchilov I. and Hutter F., 2019, "Decoupled Weight Decay Regularization", Proceedings of the International Conference on Learning Representations (ICLR), pp. (1-8).
27. Ma, J. and Yarats, D., 2019. "Quasi-hyperbolic momentum and Adam for deep learning". International Conference on Learning Representations (ICLR), Pp. (1-38).
28. Hoseini F., Shahbahrami A. and Bayat P., 2019, "AdaptAhead Optimization Algorithm for Learning Deep CNN Applied to MRI Segmentation", Journal of Digital Imaging, Society of imaging informatics in medicine, Springer, Vol. 32, issue 1, Pp. (105-115).
29. Huang G., Liu Z., Van Der Maaten L. and Weinberger K.Q., 2017, "Densely connected convolutional networks", in Proceedings of the IEEE conference on computer vision and pattern recognition. (pp. 4700-4708).
30. Johnson M., Schuster M., Le Q. V., Krikun M., Wu Y., Chen, Z., … Dean, J. (2017). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. Transactions of the Association for Computational Linguistics, Vol. 5, (pp. 339–351).
31. He K., Zhang X., Ren S. and Sun J., 2016, "Deep residual learning for image recognition", in Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). (pp. 770-778).
32. Tai, Y., Yang, J., and Liu, X., 2017, "Image Super-Resolution via Deep Recursive Residual Network", Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), DOI: 10.1109/CVPR.2017.1, Pp. (3147-3155).
33. Korzeniowski F., 2018, "Experiments with AMSGrad" Retrieved December 24, 2018, from https://fdlm.github.io/post/amsgrad/.
34. Carneiro, T., Da Nóbrega1, R. V., Nepomuceno, T., and others, 2018, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications", IEEE Access, DOI: 10.1109/ACCESS.2018.2874767, IEEE Access.
35. He, K., Zhang, X., Ren, S., and Sun, J., 2015, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", In Proceedings of the IEEE international conference on computer vision, Pp. (1026-1034).
36. Krogh, A. and Hertz, J.A., 1992. "A simple weight decay can improve generalization". In Advances in neural information processing systems (pp. 950-957).
37. Drăgulescu B., Bucos M., Vasiu R., 2015, "Predicting Assignment Submissions in a Multi-class Classification Problem", TEM Journal, Vol. 4, No. 3, Pp.(244-254).