LINEAR REGRESSION MODELLING OF POPULATION AND GDP USING ADAPTIVE LEARNING RATE OPTIMIZATION ALGORITHMS

***Okonkwo, S.J., Adedoyin, E. D, Ayodele, O. U., and Abdulateef, S.**

Environmental Modelling and Biometrics Department
Forestry Research Institute of Nigeria
Forest Hill, Jericho Ibadan
*Email: sj.okonkwo@gmail.com

Abstract

Adaptive Learning Rate Optimization Algorithms (ALROA) are used in solving machine and deep learning problems. This study aims to adopt these algorithms to estimate the coefficients of traditional linear regression models between global population and GDP by countries for the Year 2019. Algorithms such as Stochastic Gradient Descent (SGD), Stochastic Gradient Descent with Momentum (SGD with Momentum), Nesterov Accelerated Gradient Descent (NAG), Modified Root Mean Squared Propagation (Modified RMSprop), and Modified Adaptive Moment Estimation (Modified Adam) were used in this study. A learning rate of 0.01 and exponential decay rates of 0.9 and 0.999 were used for the first and second momentum. Half Mean Square Error (HMSE) was used as the loss function while Root Mean Square Error (RMSE) was used to rank the algorithms in order of accuracy. The ranking showed that SGD with Momentum, Adam, and SGD could be modified and successfully adapted for estimating the coefficients of regression models with a high degree accuracy.

Keywords: Cost Function, Modified Adam, Gradient Descent, RMSE

INTRODUCTION

Estimating the coefficients of regression models can be formulated asan optimization problem(Muehlebach and Jordan, 2019).The key to the success of optimization algorithms, structural or stochastic,in minimizing convex functions is that these functions have local or global phenomena (Bubeck, 2015). Gradient Descent Optimization Algorithms (GDOA) provide an effective way to solve a good number of optimization problems with the aim of estimatingcoefficients of regression modelssuch that the Error Sum of Squares (ESS) between the predicted and the observed values of a variable of interest is minimized (Mayooran, 2018).Loss Functions have the same local and global minima and this behaviour makes faster convergence to be achieved using optimization algorithms (Bubeck, 2015). SGD and SGD with momentum are two of the most common algorithms for solving optimization problems involving large volumes of data (Loizou and Richtarik, 2018).Loizou and Richtarik (2018) proved the linear convergence of SGD and SGD with momentum algorithms.NAG descent converges to the global minimum faster than most extant optimization algorithms (Nesterov, 1983; Muehlebach and Jordan, 2019).This study provided answers to the following questions: Can deep learning and neural network algorithms be adopted estimating the coefficients of in linear regression models? What adjustments have to be made to these algorithms? And how well would they do compared to traditional regression methods? The study was limited to regression models using cross–sectional data.

LITERATURE REVIEW

Bubeck (2015) carried out an investigation into understanding what he termed the “oracle complexity of convex optimization” which is an understanding into how many iterations it will take for an algorithm to converge to the global minimum. Furthermore, Qianxiao *et al*. (2017) observed that the standard gradient descent (GD) attains convergence slower in comparison to SGD when the gradient evaluations progressively becomes large, although the high sensitivity of SGD with the learning rate makes its convergence rate equally slow. Consequently, a number of attempts have been employed. Notable amongst these are: to reduce the variance of the noise gradient using adaptive learning rate and momentum acceleration mechanisms (Zou and Shen, 2018). Zou and Shen (2018) further posited that adaptive learning rate and the use of momentum mechanisms showed more suitable solution to problems since no large memory was required for their computation. NAG is a momentum accelerated mechanism that has been theoretically and numerically investigated for both convex and non-convex optimization problems for global learning rate. Owing to conciseness and effectiveness of momentum accelerated mechanisms, the NAG momentum speeds up the training of deep networks. It is a product of "discretizing an ordinary differential equation with a semi–implicit Euler integration scheme" (Muehlebach and Jordan, 2019). Bubeck, *et al*. (2015) proposed a new method for unconstrained optimization of a smooth and strongly convex function which attains the optimal rate of convergence of NAG descent.

METHODOLOGY

Model Development

Let Y_{GDP} and $X_{Population}$ be independent variables denoting observations on GDP and Population, respectively. Also, let the relation between Y_{GDP} and $X_{Population}$ be given by:

$$Y_{GDP} = a_0 + a_1 X_{Population} + e_i \qquad (1)$$

where a_0 and a_1 are unknown parameters for the function and $E_{rror}$is a vector of measurement errors.

Log Transformation

The natural logarithm of Y_{GDP}, given as Y_{GDP_log}, in equation (1) gives a transformed model of form:

$$Y_{GDP\_log} = \theta_0 + \theta_1 X_{Population\_log} + E_{error} \qquad (2)$$

where $\theta_0 = log\, a_0$, $\theta_1 = log\, a_1$, $X_{Population\_log} = natural \log of\ X_{Population}$ and $E_{error} = natural \log of\ e_i$.

Ordinary Least Squares

Let $X_{n\times 2}$ be a matrix of predictor variables. The Ordinary Least Square (OLS) Estimator of (2) given by Mayooran (2018) is defined as:

$$\hat{\theta}_{2\times 1} = X^{+}_{n\times 2} Y_{GDP\_log} \qquad (3)$$

Where $X^{+}_{n\times 2} = (X^{T}_{n\times 2} X_{n\times 2})^{-1} X^{T}_{n\times 2}$

It should be noted that $X^{+}_{n\times 2}$ is the pseudoinverse of the matrix $X_{n\times 2}$ as described by Moore (1920) and Penrose (1955).

Loss Function

The loss function used is the Half Mean Squared Error (HMSE) and is defined as:

$$L(\theta_0, \theta_1) = \frac{1}{2n} \sum_{i=1}^{n} \left(Y_{predicted} - Y_{GDP\_log}\right)^2 \qquad (4)$$

Stochastic Gradient Descent

The objective of Stochastic Gradient Descent (SGD) is to minimise (5).The Stochastic gradient Descent Algorithm is defined by Ruder (2016) as:

$$\theta_j^{i+1} = \theta_j^{i} - \tau \frac{\partial}{\partial \theta_j} L(\theta_0, \theta_1) \qquad (5)$$

Where θ_j^irepresents the value of the j^{th} coefficient of thei^{th}iteration and τ is the learning rate. And $\frac{\partial}{\partial \theta_j} L(\theta_0, \theta_1)$ is a partial differentiation with respect to θ_j.

For $j = 0$, $\theta_0^{i+1} = \theta_0^{i} - \tau \frac{1}{n} \sum_{i=1}^{n} \left(Y_{predicted} - Y_{GDP\_log}\right)$ (6)

For $j = 1$,$\theta_1^{i+1} = \theta_1^{i} - \tau \frac{1}{n} \sum_{i=1}^{n} \left(Y_{predicted} - Y_{GDP\_log}\right) X_{Population\_log}$ (7)

Stochastic Gradient Descent with Momentum

The objective of Stochastic Gradient Descent with Momentum is to minimise (4): Stochastic Gradient Descent with Momentum as defined by Loizou and Richtarik (2018) is given as:

$$\theta_j^{i+1} = \theta_j^{i} - \tau m_t^{i} \qquad (8)$$

Where θ_j^irepresents the value of the j^{th} coefficient of thei^{th}iteration.m_t^i the momentum of thei^{th}iteration defined as:

$$m_t^{i+1} = \beta m_t^i + \frac{\partial}{\partial\theta_j} L(\theta_0, \theta_1) \qquad (9)$$

And β is the exponential decay rate of the momentum,m_t.

For $j = 0$,$m_t^{i+1} = \beta m_t^i + \frac{1}{n}\sum_{i=1}^{n}\left(Y_{predicted} - Y_{GDP\_log}\right)$ (10)

$$\theta_0^{i+1} = \theta_0^i - \tau m_t^{i+1} \qquad (11)$$

For $j = 1$,$m_t^{i+1} = \beta m_t^i + \frac{1}{n}\sum_{i=1}^{n}\left(Y_{predicted} - Y_{GDP\_log}\right)X_{Population\_log}$ (12)

$$\theta_1^{i+1} = \theta_1^i - \tau m_t^{i+1} \qquad (13)$$

Nesterov Accelerated Gradient Descent (NAG)

The objective is to minimize the modified cost function$L\ (\omega_0, \omega_1)$.

$$\omega_0 = \theta_0 - \beta m_t \qquad (14)$$

$$\omega_1 = \theta_1 - \beta m_t \qquad (15)$$

Where $m_t^{i+1} = \beta m_t^i + \frac{\partial}{\partial\theta_j} L\left(\left(\theta_0 - \beta m_t^i\right), \left(\theta_1 - \beta m_t^i\right)\right)$ is the momentum of the$(i+1)^{th}$iteration and β is the exponential decay rate of the Recast as: momentum (Ruder, 2018; Muehlebach and Jordan, 2019).NAG Algorithm is defined as:

$$\theta_j^{i+1} = \theta_j^i - \tau m_t^{i+1} \qquad (16)$$

Where θ_j^irepresents the value of the j^{th} coefficient of thei^{th}iterationand τ is the learning rate.

For $j = 0$,$m_t^{i+1} = \beta m_t^i + \frac{1}{n}\sum_{i=1}^{n}\left(Y_{predicted} - Y_{GDP\_log}\right)$ (17)

$$\theta_0^{i+1} = \theta_0^i - \tau m_t^{i+1} \qquad (18)$$

For $j = 1$, $m_t^{i+1} = \beta m_t^i + \frac{1}{n}\sum_{i=1}^{n}\left(Y_{predicted} - Y_{GDP\_log}\right)X_{Population\_log}$ (19)

$$\theta_1^{i+1} = \theta_1^i - \tau m_t^i \qquad (20)$$

Modified Root Mean Square Propagation (Modified RMSprop)

The objective of RMSprop is to minimize (4). RMSprop is defined by Bubeck (2015) as:

$$\theta_j^{i+1} = \theta_j^i - \frac{\tau}{\sqrt{m_t^{i+1} + 10e^{-8}}} \frac{\partial}{\partial \theta_j} L(\theta_0, \theta_1) \qquad (21)$$

Where θ_j^irepresents the value of the j^{th} coefficient of thei^{th}iteration, τ is the learning rate and $m_t^{i+1}$is defined as:

$$m_t^{i+1} = \beta m_t^i + (1 - \beta) \frac{\partial}{\partial \theta_j} L(\theta_0, \theta_1) \qquad (22)$$

β is the exponential decay rate of the momentum m_t.Using the RMSprop algorithm as defined in (21) for traditional regression problems resulted in complex number regression coefficients due to the possibility of negative momentum m_t in the algorithm. To solve this problem, we resorted to using Modified RMSprop which uses the absolute value of$|m_t^{i+1}|$.

$$\theta_j^{i+1} = \theta_j^i - \frac{\tau}{\sqrt{|m_t^{i+1}| + 10e^{-8}}} \frac{\partial}{\partial \theta_j} L(\theta_0, \theta_1) \qquad (23)$$

For $j = 0$,$m_t^{i+1} = \beta m_t^i + (1 - \beta) \frac{1}{n} \sum_{i=1}^{n} \left(Y_{predicted} - Y_{GDP\_log}\right) \qquad (24)$

$$\theta_0^{i+1} = \theta_0^i - \frac{\tau}{\sqrt{|m_t^{i+1}| + 10e^{-8}}} \frac{1}{n} \sum_{i=1}^{n} \left(Y_{predicted} - Y_{GDP\_log}\right) \qquad (25)$$

For$j = 1$,$m_t^{i+1} = \beta m_t^i + (1 - \beta) \frac{1}{n} \sum_{i=1}^{n} \left(Y_{predicted} - Y_{GDP\_log}\right) X_{Populatio\ \_log}$(26)

$$\theta_1^{i+1} = \theta_1^i - \frac{\tau}{\sqrt{|m_t| + 10e^{-8}}} \frac{1}{n} \sum_{i=1}^{n} \left(Y_{predicted} - Y_{GDP\_log}\right) X_{Population\_log} \qquad (27)$$

Modified Adaptive Moment Estimation (Modified Adam)

The objective of Adam is to minimize (4). Adam is defined by Kingma and Ba (2017):

$$\theta_j^{i+1} = \theta_j^i - \frac{\tau}{\sqrt{v_t^{i+1} + 10e^{-8}}} m_t^{i+1} \qquad (28)$$

Where θ_j^irepresents the value of the j^{th} coefficient of the i^{th} iteration,τ is the learning rate.m_t and v_t are the respective first and second momentum of the cost function of the algorithm respectively defined as:

$$m_t^{i+1} = \beta_1 m_t^i + (1 - \beta_1)\frac{\partial}{\partial\theta_j}L(\theta_0, \theta_1) \qquad (29)$$

$$v_t^{i+1} = \beta_2 v_t^i + (1 - \beta_2)\frac{\partial}{\partial\theta_j}L(\theta_0, \theta_1) \qquad (30)$$

β_1 and β_2 are the exponential decay rates for the first and second momentum respectively. Also, due to the possibility of negative momentum,v_t, using Adam as defined in (28) results in the coefficientsbeing complex numbers.However, using Modified Adam which uses the absolute value of$\left|v_t^{i+1}\right|$ results in (28) becoming:

$$\theta_j^{i+1} = \theta_j^i - \frac{\tau}{\sqrt{\left|v_t^{i+1}\right| + 10e^{-8}}} m_t^{i+1} \qquad (31)$$

Where m_t, v_t and $\frac{\partial}{\partial\theta_j}L(\theta_0, \theta_1)$ are as defined above.

For $j = 0$,$m_t^{i+1} = \beta_1 m_t^i + (1 - \beta_1)\frac{1}{n}\sum_{i=1}^{n}\left(Y_{predicted} - Y_{GDP\_log}\right)$ (32)

$$v_t^{i+1} = \beta_2 v_t^i + (1 - \beta_2)\frac{1}{n}\sum_{i=1}^{n}\left(Y_{predicted} - Y_{GDP\_log}\right) \qquad (33)$$

$$\theta_0^{i+1} = \theta_0^i - \frac{\tau}{\sqrt{\left|v_t^{i+1}\right| + 10e^{-8}}} m_t^{i+1} \qquad (34)$$

For$j = 1$,$m_t^{i+1} = \beta_1 m_t^i + (1 - \beta_1)\frac{1}{n}\sum_{i=1}^{n}\left(Y_{predicted} - Y_{GDP\_log}\right) X_{Population\_log}$

(35)

$$v_t^{i+1} = \beta_2 v_t^i + (1 - \beta_2)\frac{1}{n}\sum_{i=1}^{n}\left(Y_{predicted} - Y_{GDP\_log}\right) X_{Population\_log} \qquad (36)$$

$$\theta_1^{i+1} := \theta_1^i - \frac{\tau}{\sqrt{\left|v_t^{i+1}\right| + 10e^{-8}}} m_t^{i+1} \qquad (37)$$

DATA ANALYSIS

The cross–sectional data of total Population by countries and their respective GDP for 2019was sourced from World Population website. Data analysis was carried out using GNU's Not Unix! (GNU) Octave (Eaton *et al*., 2014). A histogram (Figure 1) for each of the two variables was found to be skewed to the right consequently suggesting that each variable in the dataset does not have a centred mean value. In other words, the plot suggests that the datasets are not normally distributed.

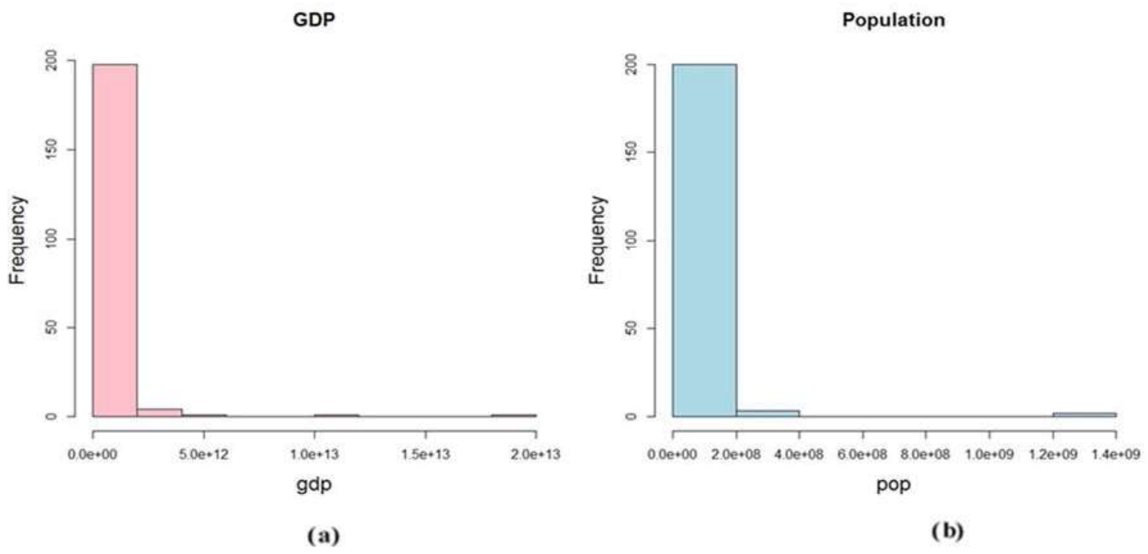

Figure 1: Histogram of GDP (a) and Population (b).

Heteroscedasticity in the original dataset was confirmed by Breusch–Pagan (BP) and Goldfeld–Quandt (GQ) Tests (Table 1).Consequently, a natural logarithm transformation was carried out on the data set. Assessment of the transformed data for heteroscedasticity using BP and GQ testsshowed that the transformed datasetwas homoscedastic (Table 1). As such, the OLS is expected to provide consistent estimators.

Table 1: Heteroscedasticity Test for Equation (1) and Equation (2)

Linear Model	Breusch–Pagan (BP) Test			Goldfeld–Quandt (GQ) Test			
	Chi-sq.	df	p-value	Chi-sq.	df1	df2	p-value
Original dataset	14.001	1	0.0002	955.51	101	100	<0.001
Log(original) dataset	3.0625	1	0.0801	1.1073	101	100	0.3053

The new data was randomly split into two parts; 80% (training), 20% (test). Adaptive learning rate algorithms were used on the training data witha learning rate of $\tau = 0.01$first with 100 then with 1000 iterations. Theta and momentum valueswere initialized to zero. The exponential decay rates for the first and second momentum were set to 0.9 and 0.999 respectively. The resultant parameters were then used to create linear regression models between the two variables, Population and GDP. The input variable from the test data was then used to predict GDP. The predicted data were then compared with the observed test data using Root Mean Square Error (RMSE) to rank the methods in terms of precision.

RESULTS AND DISCUSSION

The algorithms started from the same maximum loss of 290.28, but converged to different minimum losses.Table 2 contain the coefficients of the regression model based on the different algorithms under study after 100 iterations and those of the OLS method.

Table 2: Descriptive Statistics and Parameter Estimates of the Methods after 100 Iterations.

	Loss		Theta	
Algorithm	Minimum	Maximum	$\hat{\theta}_0$	$\hat{\theta}_1$
OLS	-	-	11.96333	0.79036
SGD	2.6099	290.28	0.12694	1.55085
SGD with Momentum	2.5536	290.28	0.34484	1.52887
NAG	290.28	Inf	-8.5509e+228	-1.3309e+230
RMS Propagation	249.56	290.28	0.10856	0.10856
Adam	2.2655	290.28	1.4622	1.4622

All algorithms showed steep convergence, with the exception of NAG which had no convergence point as it tends to infinity and RMSprop which hada shallow descent with a minimum loss of 249.56. Adam has a minimum loss of 2.2655 after 100 iterations which it maintained after 1000 iterations (Table 3).

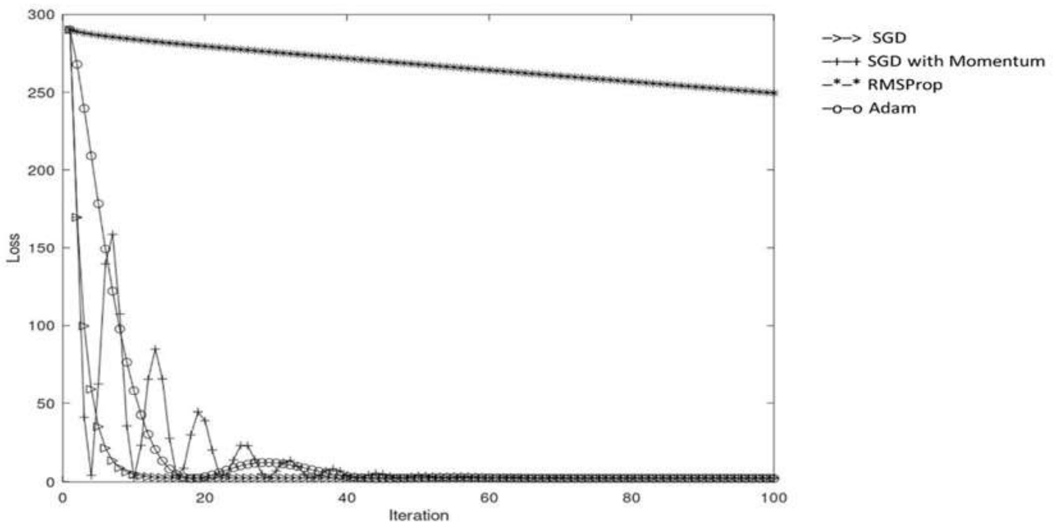

Figure 2: Loss of the Methods after Hundred iterations.

After 1000 iterations, RMSProp minimum loss dropped to 31.592but still had one of the highest minimum loss among the converging methods. A rank of the accuracy of the methods using the Root Mean Square Error (RMSE) showed that the OLS was the most accurate while RMSProp method was the least accurate (Table 4).

Table 3: Descriptive Statistics and Parameter Estimates of the Methods after 1000 Iterations.

	Loss		Theta	
Algorithm	Minimum	Maximum	$\hat{\theta}_0$	$\hat{\theta}_1$
SGD	2.5446	290.28	0.36887	1.53530
SGD with Momentum	2.0242	290.28	2.5173	1.3973
NAG	290.28	Inf	NaN	NaN
RMS Propagation	31.592	290.28	0.99899	0.99899
Adam	2.2655	290.28	1.4653	1.4653

Table 4: Rank of the methods using RMSE.

Algorithm	RMSE	Order of Precision
OLS	1.3375	1
SGD with Momentum	1.7351	2
Adam	1.8478	3
Stochastic Gradient Descent	1.9701	4
RMS Propagation	7.7081	5

CONCLUSION AND RECOMMENDATION

We investigated the possibility of adopting deep learning algorithms for estimating the parameters of regression models and found that with the exception of NAG which tended to infinity, SGD with Momentum, Modified Adam, and SGD can be successfully adopted for estimating the coefficients of regression models with a high degree accuracy. Modified RMSprop was as successful at estimating the coefficient of regression models.

REFERENCE

Bubeck, S. (2015). "Convex Optimization: Algorithms and Complexity". *Foundations and Trends in Machine Learning*, 8:231–357.

Bubeck, S., Lee, Y. T. and Singh, M. (2015). "A geometric alternative to Nesterov's accelerated gradient descent". arXiv:1506.08187 [math.OC]

Eaton, J. W., Bateman, D., Hauberg, S. and Wehbring, R. (2014). GNU Octave version 3.8.1 manual: A high–level interactive language for numerical computations. CreateSpace Independent Publishing Platform.ISBN 1441413006.

Kingma, D. P. and Ba, J. L. (2017). "Adam: A Method for Stochastic Optimization". arXiv: 1412.6980v9 [cs.LG].

Loizou, N. and Richtarik, P. (2018). "Momentum and Stochastic Momentum for Stochastic Gradient, Newton, Proximal Point and Subspace Descent Methods". arXiv:1712.09677v2 [math.OC].

Mayooran, T. (2018). "A Gradient–Based Optimization Algorithm for Ridge Regression by Using R". *International Journal of Research and Scientific Innovation (IJRSI)*, 5:38–44.

Muehlebach, M. and Jordan, M. I. (2019). "A Dynamical Systems Perspective on Nesterov Acceleration". arXiv:1905.07436v1 [math.OC].

Nesterov, Y. E. (1983). "A method of solving a convex programming problem with convergence rate O ($1/k^2$)". *Soviet Mathematics Doklady*, 27:372–376.

Penrose, R. (1955). A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51:406–413. doi:10.1017/S0305004100030401.

Qianxiao, L., Cheng, T. and Weinan, E. (2017). "Stochastic modified equations and adaptive stochastic gradient algorithms". In: Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia.

World Population Review. (2019). World Countries by GDP. Retrieved February 15, 2020 from http://worldpopulationreview.com/countries/countries-by-gdp/

Zou, F. and Shen, L. (2018). "On the convergence of AdaGrad with momentum for training deep neural networks". arXiv preprint arXiv.1810.03408v1.