Benchmarking a suite for Stochastic Gradient Descent Optimization Algorithms for the image classification problems based on CIFAR10 dataset.

Rares FOLEA
Computer Science & Engineering Department
Faculty Of Automatic Control And Computers
University Politehnica Of Bucharest
Bucharest, Romania
rares.folea@stud.acs.upb.ro

Andrei VATAVU
Computer Science & Engineering Department
Faculty Of Automatic Control And Computers
University Politehnica Of Bucharest
Bucharest, Romania
andrei.vatavu@stud.acs.upb.ro

***Abstract*—This paper analyze the performance of various optimizers based on Stochastic Gradient Descent algorithms against the problem of image classification problems. We have benchmarked using a ResNet18 network trained and evaluated on CIFAR10 dataset.**

***Index Terms*—Stochastic Gradient Descent with Momentum, Adam, RAdam, AdaBound, Benchmark, Stochastic Gradient Descent, Optimization Algorithms, image classification problems, CIFAR10, ResNet, ResNet18, deep learning, optimizers, neural networks**

CONTENTS

I. INTRODUCTION

Image recognition is a common method for allowing computers to recognize items inside a graphical input. Image labeling, image search, medical condition recognition, and self-driving cars are just a few of the jobs that use it.[3]

A residual neural network (ResNet) is a type of neural network. Skip connections, or shortcuts, are used by residual neural networks to jump past some layers. The majority of ResNet models use double or triple layer skips with non-linearities (such as Rectified Linear Unit) and batch normalization in between.

The optimizers that we have used in this paper are the following:

1) **Stochastic Gradient Descent with Momentum**
2) **Adam**
3) **RAdam**
4) **AdaBound**

They will be presented in the followings section.

The iterative approach of stochastic gradient descent is used to optimize an objective function with sufficient smoothness criteria. Because it replaces the real gradient with an estimate, it can be considered a stochastic approximation of gradient descent optimization.

Throughout our research, we will benchmark all the optimizers against the CIFAR-10 dataset. This ensures an objective comparison between the models. The particularities of this dataset will be presented in the followings section.

We have trained for 100 epochs a ResNet18 using multiple optimizers: **sgd**, **adam**, **radam**, **adabound** and we have analyzed the training loss, toghether with the accuracies obtained for the training/testing dataset.

II. Dataset

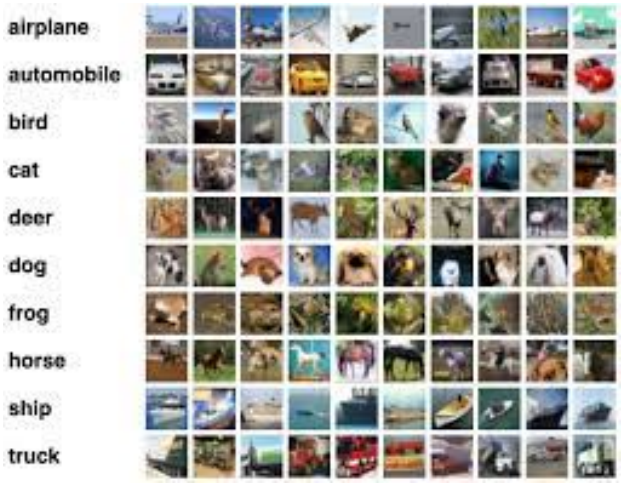

Figure 1: The classes in the CIFAR-10 dataset, as well as 10 random images from each class

CIFAR-10 is an established computer-vision dataset used for object recognition. The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. [4]

The 10 different classes represent **airplanes**, **cars**, **birds**, **cats**, **deer**, **dogs**, **frogs**, **horses**, **ships**, and **trucks**. There are **6 thousand** images of each class. All the classes in this dataset are mutually exclusive, with no overlap between the classes. The total size of the dataset is **163 MB**.

III. Neural Network Architecture

Residual Networks, or ResNets, instead of learning unreferenced functions, learn residual functions with reference to the layer inputs. Residual nets allow these layers to suit a residual mapping rather than expecting that each few stacked layers directly match a desired underlying mapping. They build networks by stacking residual blocks on top of each other: a ResNet-50, for example, has fifty layers made up of these pieces.

For classifying the CIFAR-10 dataset, we have used a **ResNet18** network.

The problem of image classification has been attempted multiple times in the literature, with such examples: [3], [1], [9].

IV. Optimizers

Optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses.

In this paper, we have aimed to benchmarking the following suite of algorithms for Stochastic Gradient Descent Optimization Algorithms for the image classification problems based on CIFAR10 dataset.

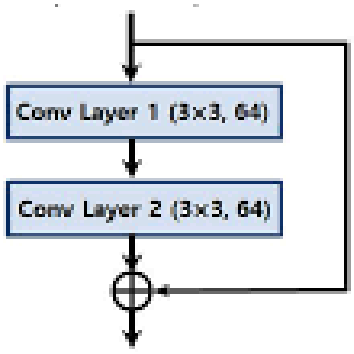

Figure 2: A ResNet18 "block", highlighting the "jump" past some layers.

1) **Stochastic Gradient Descent with Momentum** [6] (SGDM)
2) **Adam: A method for stochastic optimization** [2]
3) **RAdam: On the Variance of the Adaptive Learning Rate and Beyond** [5]
4) **AdaBound: Adaptive Gradient Methods with Dynamic Bound of Learning Rate** [7]

At each iteration, stochastic gradient descent samples a selection of summand functions in order to reduce the computational cost. In the case of large-scale machine learning problems, this is particularly effective.

Stochastic gradient descent with momentum remembers the update of deltas at each iteration, and determines the next update as a linear combination of the gradient and the previous update. Adam [2] is also an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments.

[5] states that the learning rate warm-up for **Adam** is a must-have trick for stable training in certain situations, while the underlying mechanism is largely unknown. In [5], it's shown that a fundamental cause is the large variance of the adaptive learning rates **Rectified Adam** does a analytical reduction of the large variance.

[8] presents **AdaBound** as an optimizer that behaves like Adam at the beginning of training, and gradually transforms to SGD at the end, therefore considered by the author an optimizer that trains as fast as Adam and as good as SGD.

V. Evaluation criteria

We have trained for 100 epochs a ResNet18 using multiple optimizers: **sgd**, **adam**, **radam**, **adabound** and we have analyzed the training loss. Also, we have analyzed the training/testing accuracy, and resource utilisation. For evaluating the loss function, we have used the Cross Entropy. This criterion computes the cross entropy loss between input and target and it is useful when training a classification problem.

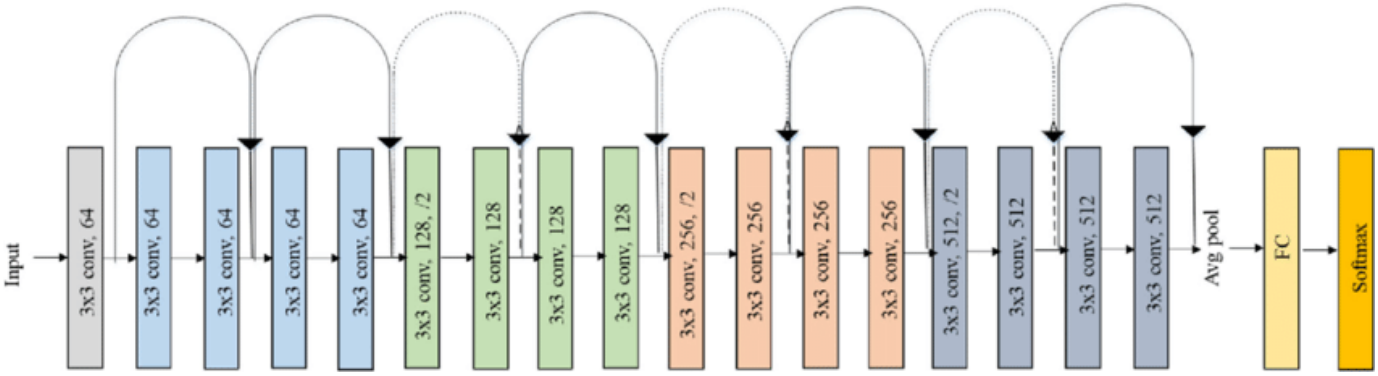

Figure 3: The architecture for the Deep Convolutional Network for ResNet18.

VI. Experiments

This section presents the experimental results for training 100 epochs on a ResNet18 using multiple optimizers: **sgd**, **adam**, **radam**, **adabound**.

A. Training loss

The goal of training a model is to find a set of weights and biases that have low loss, on average, across all examples. **adabound** reported the smaller training loss in the first 20 epochs, but then **radam** reduced the training loss to the minimum value between these optimizers. The losses after 100 epochs were: 2880.877 for **SGDM**, 1441.286 for **ADAM**, 881.258 for **AdaBound**, 583.347 **RADAM**

B. Training accuracy

Over 93% training accuracy was obtained using ResNet18 with RADAM, and 90.3% for AdaBound. 84% was the training accuracy for ADAM and just over 68% was achieved by sgd.

C. Test accuracy

An outstanding 90% accuracy on the training was achieved by on the the CIFAR-10 dataset using a ResNet18 with radam optimizer. AdaBound and Adam achieved similar accuracies, 87% and 85% respectively. The accuracy of the model using a stohastic gradient decent optimizer was only 64%.

VII. System stats

We have trained for 100 epochs a ResNet18 using multiple optimizers: **sgd**, **adam**, **radam**, **adabound** and we have analyzed the system utilization (CPU, RAM, GPU, etc.).

VIII. Code

The code that we have used is available open-source, at:

`https://github.com/raresraf/optimizer-benchmark`

IX. Acknowledgements

Credits for the framework developed around optimizer evaluation against multiple datasets and deep neural networks:

`https://github.com/ifeherva/optimizer-benchmark`

References

[1] Nie Jinliang. Cifar10 image classification based on resnet. 23(1), 2019.

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[3] Kamil Klosowski. Image recognition on cifar10 dataset using resnet18 and keras. 2018.

[4] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[5] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[6] Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum. *arXiv preprint arXiv:2007.07989*, 2020.

[7] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019.

[8] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, Louisiana, May 2019.

[9] Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

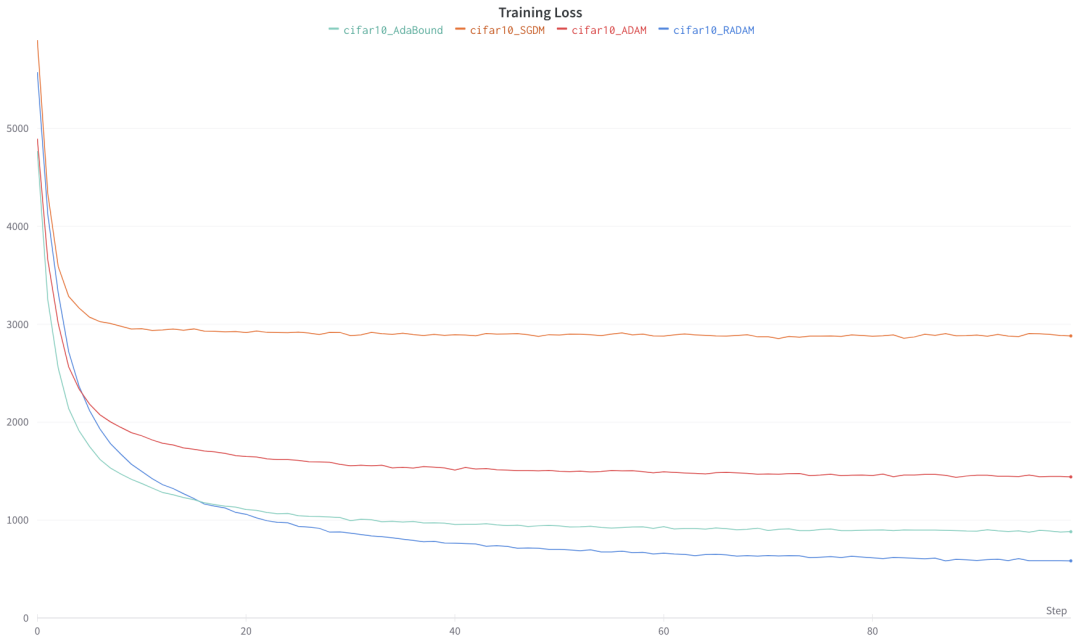

Figure 4: Training loss recorded by the ResNet18 network as a function of the number of trained epochs, using multiple optimizers: **sgd**, **adam**, **radam**, **adabound**. **adabound** reported the smaller training loss in the first 20 epochs, but then **radam** reduced the training loss to the minimum value between these optimizers. The losses after 100 epochs were: 2880.877 for **SGDM**, 1441.286 for **ADAM**, 881.258 for **AdaBound**, 583.347 **RADAM**.

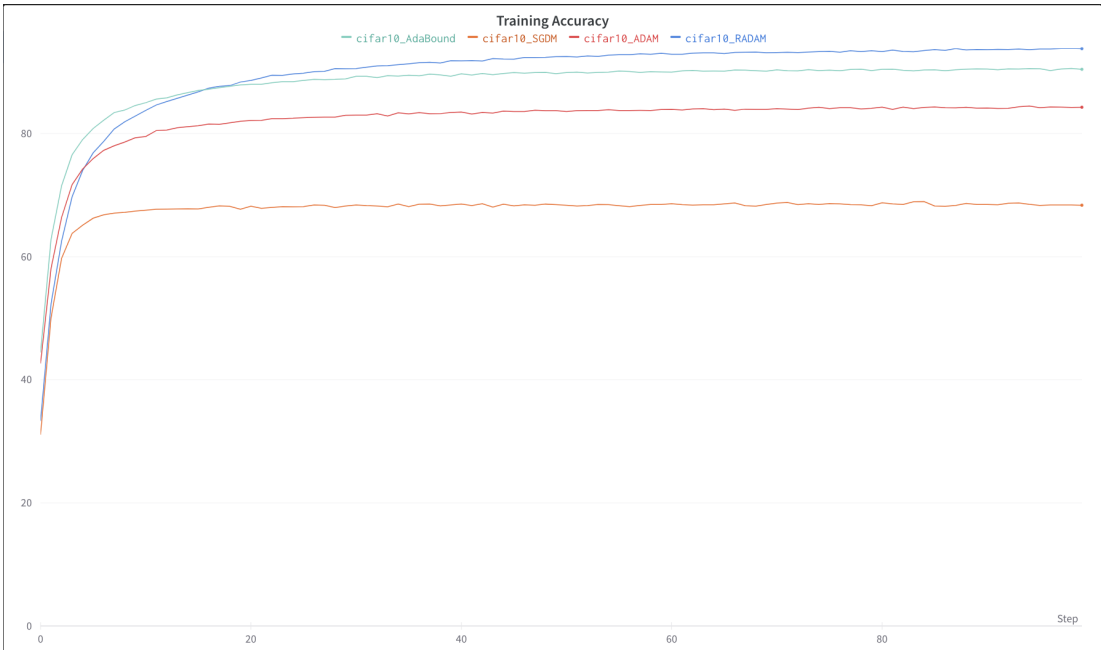

Figure 5: Training accuracy for ResNet18 as a function of the number of trained epochs, using multiple optimizers: **sgd**, **adam**, **radam**, **adabound**. Over 93% training accuracy was obtained using ResNet18 with RADAM, and 90.3% for AdaBound. 84% was the training accuracy for ADAM and just over 68% was achieved by sgd.

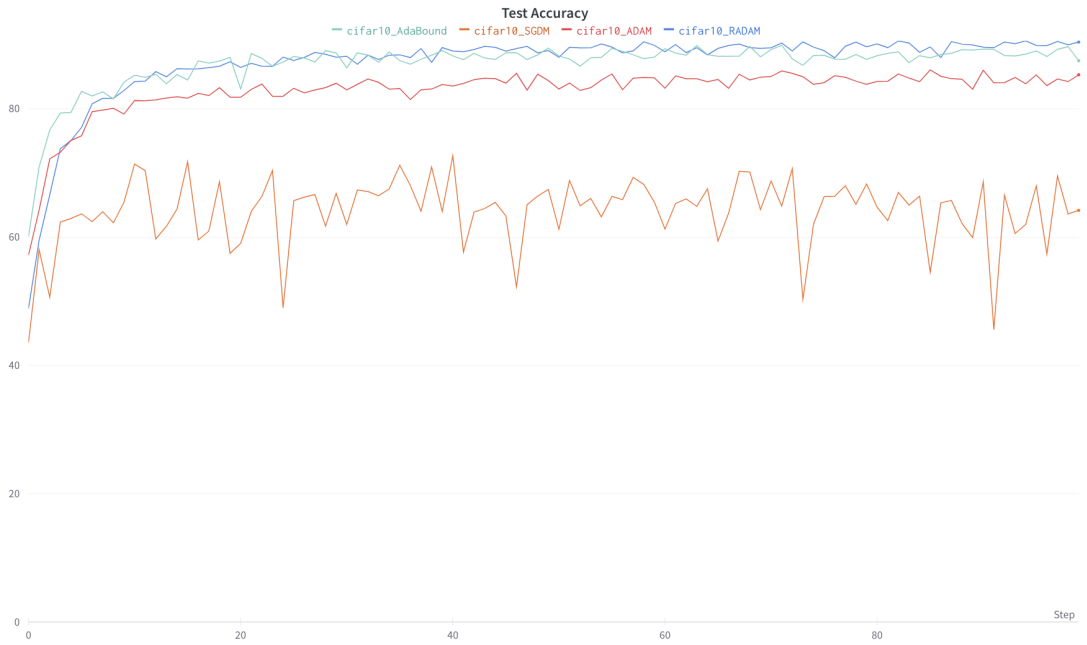

Figure 6: Test accuracy for ResNet18 as a function of the number of trained epochs, using multiple optimizers: **sgd**, **adam**, **radam**, **adabound**. An outstanding 90% accuracy on the training was achieved by on the the CIFAR-10 dataset using a ResNet18 with radam optimizer. AdaBound and Adam achieved similar accuracies, 87% and 85% respectively. The accuracy of the model using a stohastic gradient decent optimizer was only 64%.

Figure 7: GPU usage for ResNet18 as a function of the number of the training time, using multiple optimizers: **sgd**, **adam**, **radam**, **adabound**.

Figure 8: Process, RAM and CPU usage for ResNet18 as a function of the training time, using multiple optimizers: **sgd**, **adam**, **radam**, **adabound**.