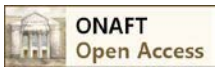
[15]. Bolger and Laurenceau (2013). Intensive Longitudinal Methods: An Introduction to Diary and Experience Sampling Research. New York: Guilford.
[16]. Ellis PD (2010) The Essential Guide to Effect Sizes: An Introduction to Statistical Power, MetaAnalysis and the Interpretation of Research Results. Cambridge, UK: Cambridge University Press.
[17]. Scherbaum CA and Ferreter JM (2009) Estimating statistical power and required sample sizes for organizational research using multilevel modeling. Organizational Research Methods 12(2): 347–367. https://doi.org/10.1177/1094428107308906
[18]. Breevaart K, Bakker AB, Demerouti E and Derks D (2016) Who takes the lead? A multi-source diary study on leadership, work engagement, and job performance. Journal of Organizational Behavior 37(3): 309–32. https://doi.org/10.1002/job.2041

УДК 004.032.26:004.896:616.314–073.7

THE USE OF CONTROL THEORY METHODS IN TRAINING NEURAL NETWORKS ON THE EXAMPLE OF TEETH RECOGNITION ON PANORAMIC X-RAY IMAGES

Smorodin A.

Odessa Polytechnic State University, Odessa, Ukraine
ORCID: 0000-0002-3370-3197
E-mail: andrey.v.smorodin@opu.ua



DOI:

Анотація. *У статті досліджено модифікацію стохастичного градієнтного спуску (SGD) на основі раніше розробленої теорії стабілізації дискретних циклів динамічної системи. Співвідношення між стабілізацією циклів у дискретних динамічних системах та знаходженням екстремальних точок дозволило застосувати нові методи управління для прискорення градієнтного спуску при наближенні до локальних мінімумів. Градієнтний спуск часто використовується для навчання глибоких нейронних мереж нарівні з іншими ітераційними методами. Експериментували з двома градієнтами SGD та Адам, було проведено порівняльні експерименти. Усі експерименти проводилися під час вирішення практичної задачі розпізнавання зубів на 2-D панорамних знімках. Мережеве навчання показало, що новий метод перевершує SGD за його можливостями, а що стосується обраних параметрів, то він наближається до можливостей Адам, що є “найсучаснішим” методом. Таким чином, показана практична корисність використання теорії управління у навчанні глибоких нейронних мереж та можливість розширення її придатності у процесі створення нових алгоритмів у цій важливій галузі.*

Abstract. *The article investigated a modification of stochastic gradient descent (SGD), based on the previously developed stabilization theory of discrete dynamical system cycles. Relation between stabilization of cycles in discrete dynamical systems and finding extremum points allowed us to apply new control methods to accelerate gradient descent when approaching local minima. Gradient descent is often used in training deep neural networks on a par with other iterative methods. Two gradient SGD and Adam were experimented, and we conducted comparative experiments. All experiments were conducted during solving a practical problem of teeth recognition on 2-D panoramic images. Network training showed that the new method outperforms the SGD in its capabilities and as for parameters chosen it approaches the capabilities of Adam, which is a “state of the art” method. Thus, practical utility of using control theory in the training of deep neural networks and possibility of expanding its applicability in the process of creating new algorithms in this important field are shown.*

Ключові слова: методи теорії управління, дискретна динамічна система, стабілізація циклів, нейронна мережа, навчання

Keywords: control theory methods, discrete dynamical systems, stabilization of cycles, neural networks, training

Introduction

Neural networks and their methods of their application in different areas of human activity are in the focus of many businesses and are included in the list of the most active areas of scientific research. The next round of activity in this area is mainly caused by the development of a new generation of graphics accelerators (GPU - Graphical Processing Unit). Modern versions of these devices have open interfaces, used not only for accelerating geometric transformations, but also for computations of any classes. These technologies gave an opportunity to accelerate all matrix operations thanks to works by Geoffrey Hinton and Yann LeCun as well. The first one of them was one of the creators of the modern algorithm *of back propagation algorithm* ([7]) making possible the computationally effective scheme for calculating partial derivatives of the neural network parameters from the loss function. While Yann LeCun, as a specialist in machine learning and computer vision, applied neural network technologies for solving problems of optical character recognition. As a result of his research, he created convolutional neural networks, which are the basis for all modern methods of video and image processing. In particular, in this work we used one of the varieties of convolutional networks.

Alongside with the task of training convolutional networks based on known examples and the method of back propagation, various methods of loss function minimization are used depending on network parameters. Among these methods, the central place belongs to methods derived from standard gradient descent which was modified to stochastic gradient descent during training neural networks. This modification takes into account the limited capabilities of modern GPUs in the amount of RAM, the slowness of information transfer from the main memory to the accelerator memory, and a large amount of training data required for training deep neural networks with dozens of millions of parameters. Thus, the problem concerning accelerated minimization of the stochastic gradient descent algorithm is important for both theoretical and practical purposes, since any acceleration of the process will save computational resources. It is known that training one network to achieve human-like results consumes electricity for dozens of millions of US dollars. In this article we apply a new scheme and show its advantages in solving the practical problem of teeth recognition on 3D panoramic images.

Analysis of Published Data and Problem Statement

Libraries for creating and training neural networks contain many different optimization algorithms based on various modifications of *gradient descent*. The main difference between the algorithms is in the method of using historical gradient values. The two algorithms SGD [10] and Adam [5] are considered as two extreme representatives in the spectrum of modifications. Adam considers historical values and tries to use this information to predict future gradient behavior, while SGD does not use only the gradient of a function at a point x_n to calculate the next value x_{n+1}.

The main problem in the process of applying neural networks in practical fields consists in their multidimensionality, which often exceeds dozens of millions of parameters. The process of learning these complex functions is time consuming. Practice shows that one training of this kind can take up to a week of processor time, and when obtaining results close to human capabilities, it is necessary to carry out the process of cross-validation and search for the best hyperparameters of the neural model. These technical difficulties pose a complex problem of selecting a fast and not resource-intensive optimization algorithm. Any improvement of even one percent would reduce the cost and use of the computer resources of huge computing clusters.

Purpose and Objectives

The objective of this study is to implement a new modification of the gradient descent algorithm. It is compared with advanced implementations of stochastic gradient descent (SGD) algorithms and Adam algorithm. All three algorithms were used during solving a practical medical problem of teeth recognition on panoramic images. To solve this problem a model of deep neural networks was involved and training of this network was carried out on the basis of all three optimization algorithms.

The main factor in evaluating algorithms is the training rate of a deep network. Training rate is the rate of approaching the target function minimum. This feature measures the network prediction proximity to the expected result on the test data.

When investigating, it is important to evaluate capabilities of the new algorithm with SGD capabilities. This is due to the proximity of these algorithms to adjusting the training coefficient and use of only historical gradient values. While predicting gradient behavior on future steps Adam gives an opportunity to get closer to the local extremum more quickly.

Methods and Materials of Research

Panoramic images used during the training of a neural article were divided by the standard method into three different, non-overlapping sets: training one, checking one and testing one. This technique is standard for any machine learning.

All calculations were performed on a personal computer with an Intel (R) Xeon (R) E-2286M @ 2.40GHz central processor, 64 GB RAM and an Nvidia Quadro RTX 3000 graphics accelerator. Neural networks are implemented in the PyTorch neural network development environment of Facebook Company.

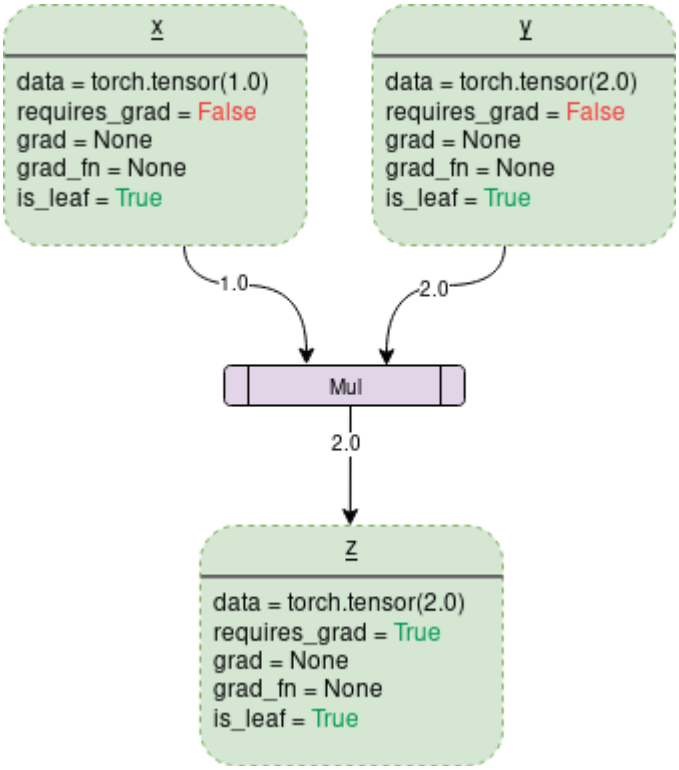
Algorithm

To analyze the work of the new gradient descent modification, we selected two tasks of neural network training, where we modified SGD. This algorithm is one of the most widely used algorithms and is a part of two main development environments for deep neural network architectures: TensorFlow (Google) and PyTorch (Facebook). This article views the PyTorch-based implementation.

Let's look at the iterative scheme

$$x_{n+1} = (1-\gamma)\left(\frac{2}{3}F(x_n)+\frac{1}{3}F(x_{n-1})\right)+\gamma\left(\frac{1}{2}x_n+\frac{1}{2}x_{n-1}\right),$$

where $F(x)=x-\lambda\nabla loss(x)$. In the neural network literature the multiplier λ is called the "learning rate" and it controls the length of the step opposite to the direction of the loss function gradient. In the standard algorithm iterations are limited with such a simple algorithm, and since the "loss' depends on the neural network model (with the loss function minimization performed according to parameters of this network) the gradient is calculated only once. The PyTorch library effectively computes the gradient based on Autograd algorithm. This algorithm is built on the basis of the calculation representation in the library based on an acyclic graph (its example is shown in the image)



Where two multidimensional tensors x and y are multiplied, and the result of their multiplication (tensor z) contains an indication of the operation it is obtained from. Saving the history of each calculation step gives an opportunity to calculate the gradient from any "sheet" of the graph from each of its "sources" by the opposite operation as for computation graph. Of course, this algorithm has limitations related to the fact that operations can only be functions with derivatives computable through analytical methods. But all known neural network architectures are built only of differentiable functions. Therefore, the value of the loss function is a "sheet" of the model calculation graph (neural network with weights) from the argument, which is a training sample. In order to understand whether the Autograd algorithm fits into this scheme it is sufficient to pay attention to the attribute that each **requires_grad** tensor has. When it is True it means that starting with it, PyTorch must monitor each operation and all new values (tensors) derived from this tensor. So, programmers may ask PyTorch to calculate the result derivative from the tensor the given graph or subgraph started with.

Now let's view some features or difficulties typical for implementation of our algorithm modification. In our scheme, we have not only to save the values of the weights of the neural network from the previous step, but also to calculate the gradient of the neural network as a part of calculation $F(x_n)$. It is this that was the main technical problem for implementation of this algorithm for complex and deep neural networks due to the peculiarities of Autograd algorithm which calculates gradients and stores them in each tensor during the reverse pass through the computation graph; and it is suboptimal to perform this operation for each level of the network, similarly to the implementation of Stochastic Gradient Decent algorithm from PyTorch. In order not to lose in rate and give Autograd an opportunity to calculate all derivatives at a new point $x_n = x_{n-1} - \lambda\nabla loss$ in one pass (backpropagation), we have chosen a different approach in implementation of the algorithm. It consists in loading all weights with the values x_{n-1} stored in the previous step and causing closure inside the algorithm between reloading the weights of all neurons and calculating the new step x_n. In this way we have got rid of all unnecessary

calculations and the performance indicators of the new algorithm are only a constant value different from the standard value and it uses additional memory amounting to 3 sizes of the set of neural network weights.

Teeth Segmentation

Since in case with the standard problem the new method was able to show better results, we decided to turn to the real problem of teeth detection and recognition in 2-D panoramic X-ray images of teeth. This problem has been actively considered in scientific studies of recent years since such images give doctors an opportunity to get an overview of the state of not only teeth, but also surrounding tissues including information about bone structure abnormalities of the jaw (see Figure 1).

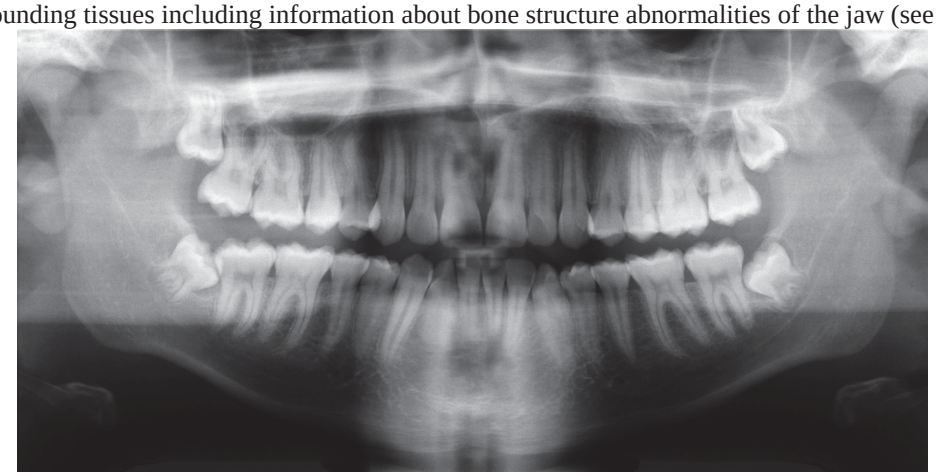


Fig 1. Source: https://en.wikipedia.org/wiki/Panoramic_radiograph

The latest results in the analysis of such images have been obtained thanks to using neural networks combined with image processing methods [1], but in this work we used a combination of U-Net networks [2] (decoder) and encoder of Resnet [4], and namely Resnet-18 from the standard library of PyTorch networks pretrained on ImageNET ([3]). Network data is usually trained with a help of Adam optimizer ([5]), but recent studies have shown that the benefits of Adam's fast convergence often lead to excessive network optimization for the training dataset, but to worse results on validation dataset in comparison with Stochastic Gradient Descent ([6]). Thus, improving the SGD algorithm and approaching its convergence rate to Adam's abilities give an opportunity to get the best of the two approaches (rate and generalization).

To begin with, we trained the network with Adam in order to get the best available result we can use to compare our algorithm and the standard SGD. All tests were run on the same number of images and training was performed during 60 epochs. The learning rate was taken equal to 0.0001 for Adam, which can change it in the process of training and at the same time it can adjust it for each hidden level of the neural network in an independent way (Figure 2).

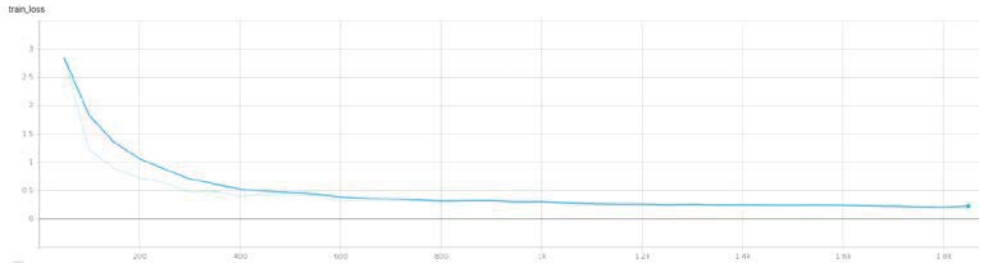
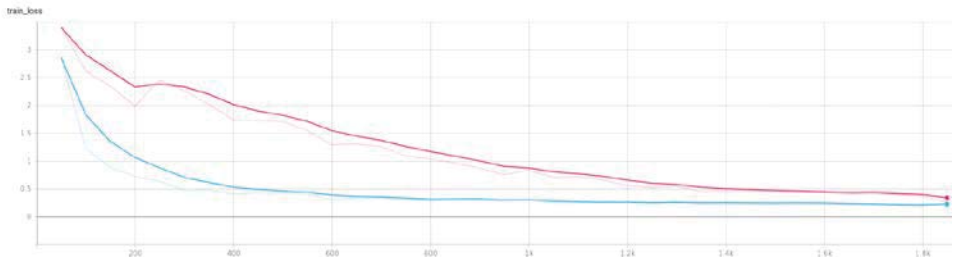


Fig. 2. Adam optimizer with lr = 0.0001

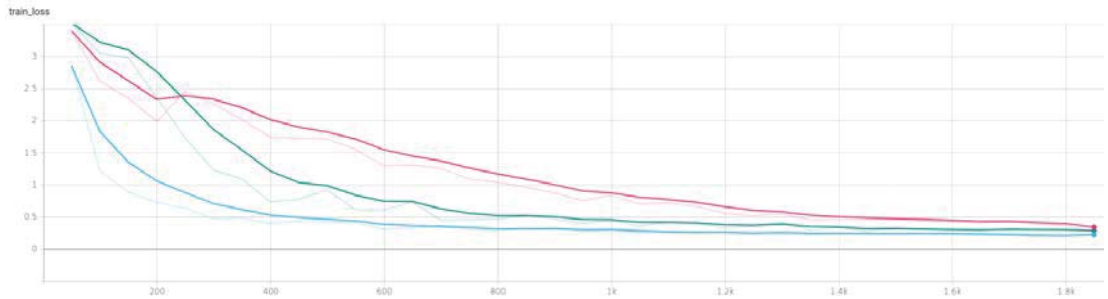
As expected, Adam quickly reached the plateau and the minimal for the given network architecture loss function and dataset plateau.

Our next step was to train the SGD network with an optimizer at lr=0.01, with the additionally enabled momentum equal to 0.99. This value was recommended by all standard machine learning libraries.



The graph shows that SGD converges to the optimal solution for the given problem much slower than Adam even with a greater learning rate and that it needed all 60 epochs to get closer to the best result, although it never to achieve it.

And finally, let's do the same experiment for our new modification with learning rate equal to 0.01, $\gamma = -0.5$, $\alpha_1 = 1.2, \alpha_2 = -0.2$.



As we can see, the new algorithm was inferior to the standard SGD (green graph) only at the beginning of training, but after 200 iterations of the algorithm it managed to obtain constantly better optimization or lower values of the loss function. After step 800, the improved algorithm was able to achieve the same result as that achieved by SGD after step 1800 and it was able to get closer to Adam's results.

Conclusions

The obtained modification of Stochastic Gradient Decent was able to improve the results of not only the SGD algorithm, but also those of its powerful modification with momentum. Neural network optimization gave an opportunity to accelerate the procedure of teeth recognition on 2-D panoramic X-ray images by 225%, and therefore it have become possible to reduce the time of training more than twice for obtaining the required accuracy. This advantage will save not only time of training, but also a large amount of computer resources, since serious research or obtaining better results in the industry close to human capabilities requires a selection of hyperparameters and a huge number of launches of neural network training, and such an advantage and acceleration of the process is a key factor in obtaining results.

Acknowledgments

This study was supported by Artifact.AI (Nantes, France), which gave an opportunity to use the neural network and provided access to the required set of panoramic X-ray images. The code that implements the network and the data remain private property of the company and will not be made available for free access.

References

[1] Mahdi, F.P., Motoki, K. & Kobashi, S. Optimization technique combined with deep learning method for teeth recognition in dental panoramic radiographs. Sci Rep 10, 19261 (2020). https://doi.org/10.1038/s41598-020-75887-9
[2] Ronneberger. O., Fischer. F., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2018. https://arxiv.org/abs/1505.04597
[3] ImageNet. http://www.image-net.org
[4] Kaiming H., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. pp. 770-778.
[5] Diederik P. Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization. 2014. https://arxiv.org/abs/1412.6980
[6] Keskar N. S., Socher R. Improving Generalization Performance by Switching from Adam to SGD. 2017. https://arxiv.org/abs/1712.07628
[7] Rumelhart D. E., Hinton G. E., Ronald W. J. Learning representations by back-propagating errors". Nature. 323 (6088). 1986. P: 533–536. Bibcode:1986Natur.323..533R. doi:10.1038/323533a0. S2CID 205001834
[8] Nesterov Y. A method of solving a convex program-ming problem with convergence rate O(1/sqr(k)). 1983. Soviet Mathematics Doklady, 27:372–376.
[9] Polyak, B.T. Some methods of speeding up the convergenceof iteration methods. USSR Computational Mathematics and Mathematical Physics. 1964. 4(5):1–17. https://doi.org/10.1016/0041-5553(64)90137-5
[10] Taddy M. Stochastic Gradient Descent // Business Data Science: Combining Machine Learning and Economics to Optimize, Automate, and Accelerate Business Decisions. - New York: McGraw-Hill, 2019. - ISBN 978-1-260-45277-8.