A Novel Adaptive Learning Rate Algorithm for Convolutional Neural Network Training

S.V. Georgakopoulos(✉) and V.P. Plagianakos

Department of Computer Science and Biomedical Informatics, University of Thessaly,
Papassiopoulou 2–4, 35100 Lamia, Greece
{spirosgeorg,vpp}@uth.gr

Abstract. In this work an adaptive learning rate algorithm for Convolutional Neural Networks is presented. Harvesting already computed first order information of the gradient vectors of three consecutive iterations during the training phase, an adaptive learning rate is calculated. The learning rate is increasing proportionally to the similarity of the direction of the gradients in an attempt to accelerate the convergence and locate a good solution. The proposed algorithm is suitable for the time-consuming training of the Convolutional Neural Networks, alleviating the exhaustive and critical for the performance of trained network heuristic search for a suitable learning rate. The experimental results indicate that the proposed algorithm produces networks having good classification accuracy, regardless the initial learning rate value. Moreover, the training procedure is similar or better to the gradient descent algorithm with fixed heuristically chosen learning rate.

Keywords: Convolutional Neural Networks · Adaptive learning rate

1 Introduction

Convolutional Neural Networks (CNNs) are state-of-the-art classification algorithms with many recent successes, mainly in computer vision problems. CNNs have been suggested for pattern recognition [4], object localization [15], object classification in large scale database of real world image [10], abnormalities recognition on medical images [8], etc. Despite the increase of CNNs' usage in the recent years, the CNN model is rooted back to a model of 1980 [7], which had adopted layers of learnable filters and a supervised classifier for the output layer. This model is modified by simplifying the architecture and introducing the backpropagation algorithm to train it [11]. However, the high number of trainable weights that this type of networks require was an inhibitor for their wide use for many years. This problem was tackled with the advent of parallel architecture and the utilization of Graphic Processing Units (GPUs).

Because of the high volume of training data that the CNNs need for training, the on-line Stochastic Gradient Descent (SGD) learning algorithm is usually utilized. A crucial parameter for the SGD algorithm's convergence is the user

G. Boracchi et al. (Eds.): EANN 2017, CCIS 744, pp. 327–336, 2017.
DOI: 10.1007/978-3-319-65172-9_28

chosen learning rate. The heuristic search of the proper learning rate value for each problem or for each specific dataset can be difficult and time consuming, since, the typical training time for a CNN varies from hours to days even when high-end GPUs are used. Additionally, it has been observed that a learning rate decay after a number of model training iterations tends to increase their performance. Thus, the human effort needed to find suitable parameter values, leads to the adoption of adaptive learning rate methods.

In this study, we present an adaptive learning rate algorithm for CNN training, which relies on the SGD algorithm and uses the last three gradients to adapt the learning rate in every training iteration.

The rest of the paper is structured as follows. In Sect. 2 related work of adaptive learning rate algorithms and preliminary materials are presented. In Sect. 3, we introduce the proposed algorithm for the adaptive learning rate and in Sect. 4 we present the experimental results and analysis. Finally, we conclude with pointers for future work.

2 Background Methods

In this section, we present related adaptive learning methods and review the basic tools used in the proposed methodology. In particular, we briefly present the CNN classification algorithm and the popular Stochastic Gradient Descent optimization method.

2.1 Convolutional Neural Networks

Convolutional Neural Networks are multistage trainable architectures used for classification tasks. Each of these stages consist of three types of layers [12]:

1. *Convolutional Layers*, which are the major components of the CNNs. A convolutional layer consists of a number of kernel matrices that perform convolution on their input and produce an output matrix (feature image) where a bias value is added. The learning procedures aim to train the kernel weights and biases as shared neuron connection weights.
2. *Pooling Layer*, which are also integral components of the CNNs. The purpose of a pooling layer is to perform dimensionality reduction of the input feature images. Pooling layers make a subsampling to the output of the convolutional layer matrices combing neighboring elements. The most common pooling function is the max-pooling function, which takes the maximum value of the local neighborhoods.
3. *Fully-Connected Layer*, is a classic Feed-Forward Neural Network (FNN) hidden layer. It can be interpreted as a special case of the convolutional layer with kernel size 1×1. This type of layer belongs to the class of trainable layer weights and it is used in the final stages of CNNs.

The training of CNN is relies on the BackPropagation (BP) training algorithm [12]. The requirements of the BP algorithm is a vector with input patterns x and a vector with targets y, respectively. The input x_i is associated with the output o_i. Each output is compared to its corresponding desirable target and their difference gives the training error. Our goal is to find weights that minimize the cost function

$$E(w) = \frac{1}{n} \sum_{p=1}^{P} \sum_{j=1}^{N_L} (o_{j,p}^L - y_{j,p})^2, \tag{1}$$

where P the number of patterns, $o_{j,p}^L$ the output of j neuron that belongs to L layer, N_L the number of neurons in output layer, $y_{j,p}$ the desirable target of j neuron of pattern p. To minimize the cost function $E(w)$, a pseudo-stochastic version of SGD algorithm, also called mini-batch Stochastic Gradient Descent (mSGD), is usually utilized [3].

2.2 Stochastic Gradient Descent

Stochastic Gradient Descent is an optimization algorithm with many successes in the Machine Learning field. Given a cost function $E(w)$ it aims to find the minimizer $w^* = (w_1^*, w_2^*, \cdots, w_v^*) \in \mathbb{R}^v$, such that:

$$w^* = \min_{w \in \mathbb{R}^v} E(w). \tag{2}$$

The SGD algorithm iteratively minimizes Eq. 1, updating the vector w after the presentation of each training example. To speedup the training process, the pseudo-stochastic variant of SGD uses a small number of training examples (mini batches) to update the weights. The mini-batches SGD has been proven very successful in CNN training [10].

To minimize the aforementioned function, the SGD calculates the gradient vector at every step and updates the network weights using a heuristically defined learning rate (n_0). Because of the CNN multi-layer architecture the gradient calculation is performed at each layer output with the chain rule. However, the learning rate parameter is very critical for the convergence of the SGD; usually large values accelerate convergence to local minima, while smaller values, result in larger training time, may locate "better" local minima.

2.3 Related Work

The problem of selecting a proper learning rate is crucial for the convergence of the SGD and in general to train high classification accuracy CNNs. Many adaptive schemes have proposed in the recent literature to deal with this problem. The AdaDelta [19] algorithm dynamically adapts the learning rate using information only from the gradient vector. The method does not require manual learning rate tuning, while it is robust to noisy gradient information. Another popular method AdaGrad [6] incorporates the knowledge of the geometry that the observed data provide on previous iterations to perform more informative gradient-based learning.

The most recently proposed adaptation method, which outperforms the previous ones, is the ADAM [2] algorithm. It is an optimization algorithm that utilizes the gradient information of the stochastic objective function, based on adaptive estimates of lower order moments. More specifically, the method calculates adaptive learning rates from estimates of first and second moments of the gradients. However, the selection of the initial learning rate is critical for the success of the adaptation.

3 The Proposed Method

The proposed algorithm is inspired by the methods presented in [1], where information of the gradient direction of the previous and the current step is used to modify the global learning rate of the SGD algorithm for FNNs. More specifically, when the gradient vector directions of two consecutive steps tend to be similar, convergence is accelerated by increasing the learning rate. On the contrary, the learning rate is decreased when the directions of the gradient vectors tend to significantly differ. This can be roughly approximated with the inner product ($\langle \cdot, \cdot \rangle$) of the normalized gradient vectors. Note that the inner product of orthogonal gradient vectors is zero, while it is equal to one when the vectors are parallel. The update rule is following:

$$n_i = n_{i-1} + \gamma \langle \nabla E_{i-1}(w_{i-1}), \nabla E_i(w_i) \rangle, \tag{3}$$

where γ is a control parameter of the learning rate adaptation.

This work is based on an adaptive learning algorithm designed for FNN training [13,16]. The suggested Adaptive Learning Rate (AdLR) algorithm relies on the latest three gradient vector directions, exploiting already computed information. The learning rate is adjusted taking into consideration the current and the previous inner product of the gradient vectors. The update rule is the following:

$$\begin{aligned} n_i = n_{i-1} &+ \gamma_1 \langle \nabla E_{i-1}(w_{i-1}), \nabla E_i(w_i) \rangle + \\ &+ \gamma_2 \langle \nabla E_{i-2}(w_{i-2}), \nabla E_{i-1}(w_{i-1}) \rangle, \end{aligned} \tag{4}$$

where the γ_1 and γ_2 are control parameter called meta-learning rates. We suggest that $\gamma_1 > \gamma_2$, since the direction of the latest gradient vectors should contribute more to the adaptation of the learning rate. In addition, to prevent the divergence of the algorithm when the learning rate takes negative values, we apply the following reset rule:

$$n_i = \begin{cases} n_0 \cdot \gamma_3, \text{ if } n_i < 0 \\ \quad n_i, \text{ if } n_i \geq 0 \end{cases} \tag{5}$$

where the n_0 is the initial learning rate value and γ_3 is a decrease factor of the original learning rate. The use of the decrease factor helps the stable convergence of the algorithm, especially in the case where the reset rule is activated after many epochs and the initial learning rate n_0 is rather large, the algorithm could overshoot the reached minimum and/or need many steps to properly readapt the learning rate.

The requirements of the proposed algorithm in computational resources is comparable to those of the SGD algorithm, since only the previous gradient vectors need to be stored and the inner products can be rapidly calculated. The AdLR[1] algorithm is implemented in C++ and CUDA using the CAFFE framework [9]. The pseudo-code of the proposed algorithm is presented below.

```
Input: $w_0$, $n_0$, $\gamma_1$, $\gamma_2$, $\gamma_3$, $iter$ ;
$i = 0$;
while $i < iter$ do
    $i = i + 1$;
    Calculate $E(w_i)$ and $\nabla E_i(w_i)$;
    Update the weights
    $w_{i+1} = w_i - n_i \nabla E_i(w_i)$;
    Update learning rate
    $n_i = n_{i-1} + \gamma_1 \langle \nabla E_{i-1}(w_{i-1}), \nabla E_i(w_i) \rangle +$
              $+ \gamma_2 \langle \nabla E_{i-2}(w_{i-2}), \nabla E_{i-1}(w_{i-1}) \rangle$
    if $n_i < 0$ then
        $n_i = n_0 \cdot \gamma_3$
    end
end
Output: $w_{i+1}$
```

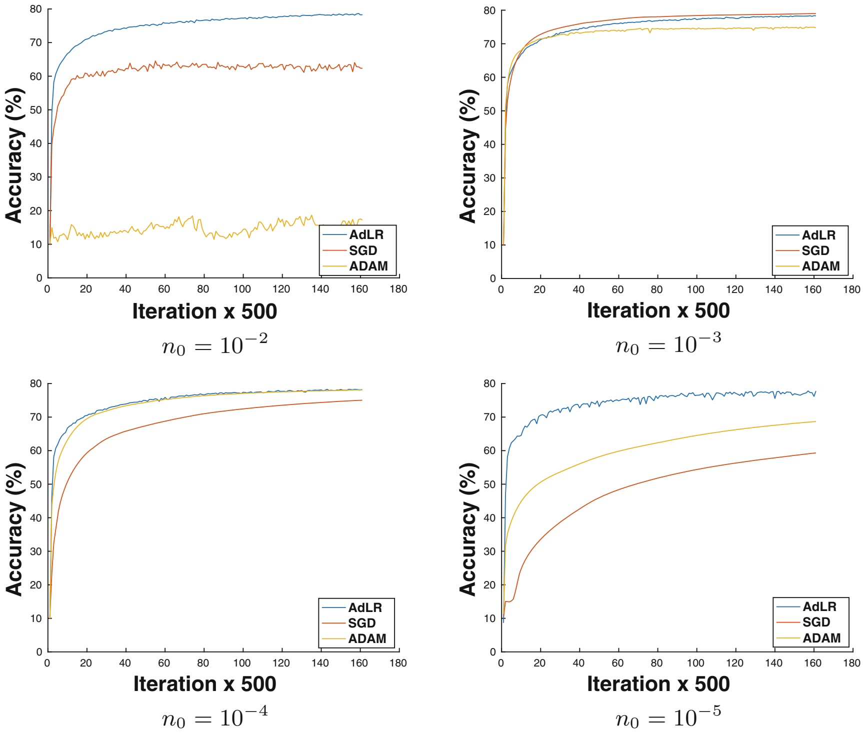
Algorithm 1. The Stochastic Gradient Descent with Adaptive Learning Rate Algorithm (AdLR)

4 Experimental Results

To evaluate the proposed algorithm the CIFAR-10 object recognition dataset [17], was used. The dataset consists of 60,000 tiny color images with size 32 × 32 pixels, which contain ten classes with 6,000 images each. The dataset was split to $50,000$ images for training, while the rest were kept for testing. The CNN architecture model for this problem is the one proposed in the CAFFE framework. More specifically, the CNN consists of two convolutional layers of 32 feature maps with 5 × 5 convolutional kernels, each followed by one 3 × 3 max pooling layer. Consecutively, another convolutional layer of 64 feature maps of 5 × 5 convolutional kernels followed by a 3 × 3 max pooling layer is utilized. Finally, a fully-connected layer with 10 neurons and a softmax logistic regression layer is integrated to the previous layers. After each convolutional layer a ReLu activation function [14] is applied, while after the first two pooling layer a local region normalization of their output is performed.

The proposed AdLR algorithm is compared to the SGD and the ADAM algorithms, using different initial learning rate values n_0, i.e. 10^{-2}, 10^{-3}, 10^{-4}

[1] The code is available on https://github.com/Georgakopoulos-Sp/.



$n_0 = 10^{-2}$

$n_0 = 10^{-3}$

$n_0 = 10^{-4}$

$n_0 = 10^{-5}$

Fig. 1. The accuracy of the AdLR, the SGD and the ADAM algorithms with different initial learning rate values.

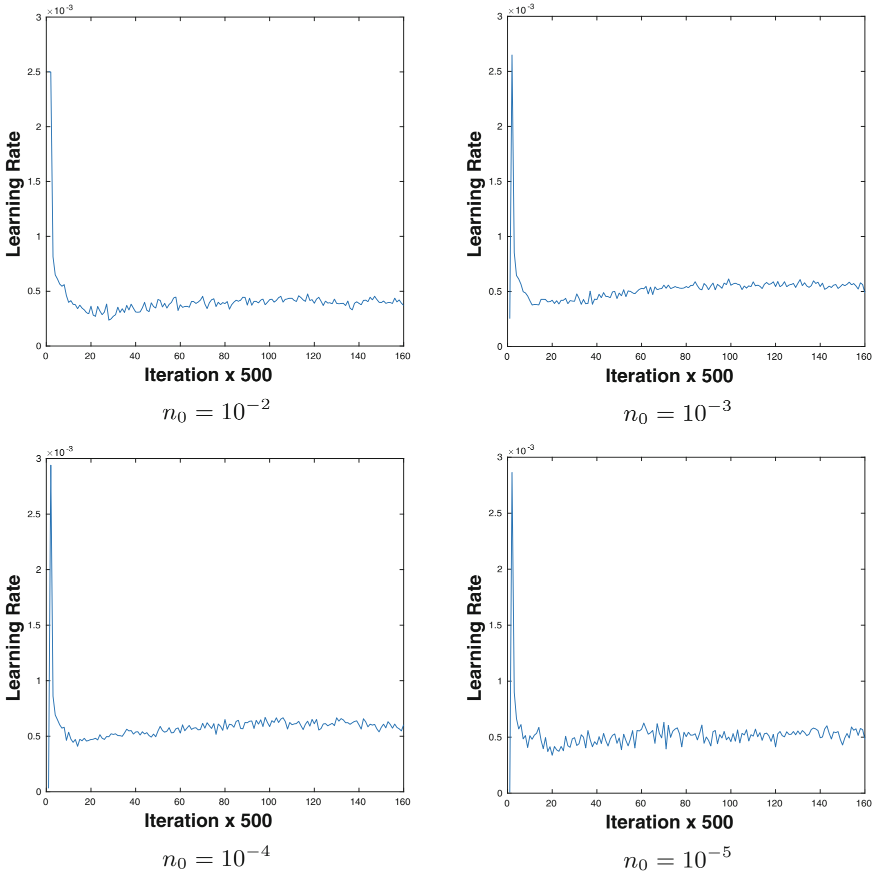
and 10^{-5}. The maximum number of iterations was set to $80,000$, the mini-batches of all algorithms contained 100 samples and the momentum term was set to 0.9. The parameters of the ADAM algorithm were set as proposed by its authors [2]. For the proposed AdLR algorithm the meta-learning parameters γ_1 and γ_2 had fixed values of 10^{-2} and 10^{-3}, respectively, while $\gamma_3 = 10^{-2}$. To validate the accuracy of the results, 100 independent executions for each initial learning rate n_0 for each algorithm were performed.

In Fig. 1 the mean accuracy is depicted, while Table 1 presents the average performance of the algorithms for 100 independent executions. The mean accuracy of the proposed algorithm is consistent and similar or better to the performance of the SGD and the ADAM algorithms, regardless the initial learning rate. As expected, the SGD algorithm is found to be very sensitive to the selection of the initial learning rate. On the other hand, although the ADAM algorithm adapts the direction of search, the user selected learning rate can be critical, as well. The proposed AdLR algorithm is suitable for CNN training, alleviating the exhaustive heuristic search for a suitable learning rate.

To investigate the results of the proposed AdLR algorithm against the other algorithms, a two-sided Wilcoxon non-parametric significant test [18] at the 5% significance level is conducted for every initial learning rate.

Table 1. The mean accuracy and standard deviation (Std) of the proposed Algorithm (AdLR), the SGD and the ADAM algorithm on CIFAR-10 dataset for different learning rate values.

	AdLR		SGD		ADAM	
	Mean (%)	Std	Mean (%)	Std	Mean (%)	Std
$n_0 = 10^{-2}$	78.4 (+/+)	0.7	62.3	1.9	17.2	8.3
$n_0 = 10^{-3}$	78.4 (=/+)	0.5	78.9	0.6	74.7	1.0
$n_0 = 10^{-4}$	78.3 (+/=)	0.7	75.0	0.3	78.1	0.5
$n_0 = 10^{-5}$	78.0 (+/+)	0.5	59.3	0.4	68.7	0.5



$n_0 = 10^{-2}$

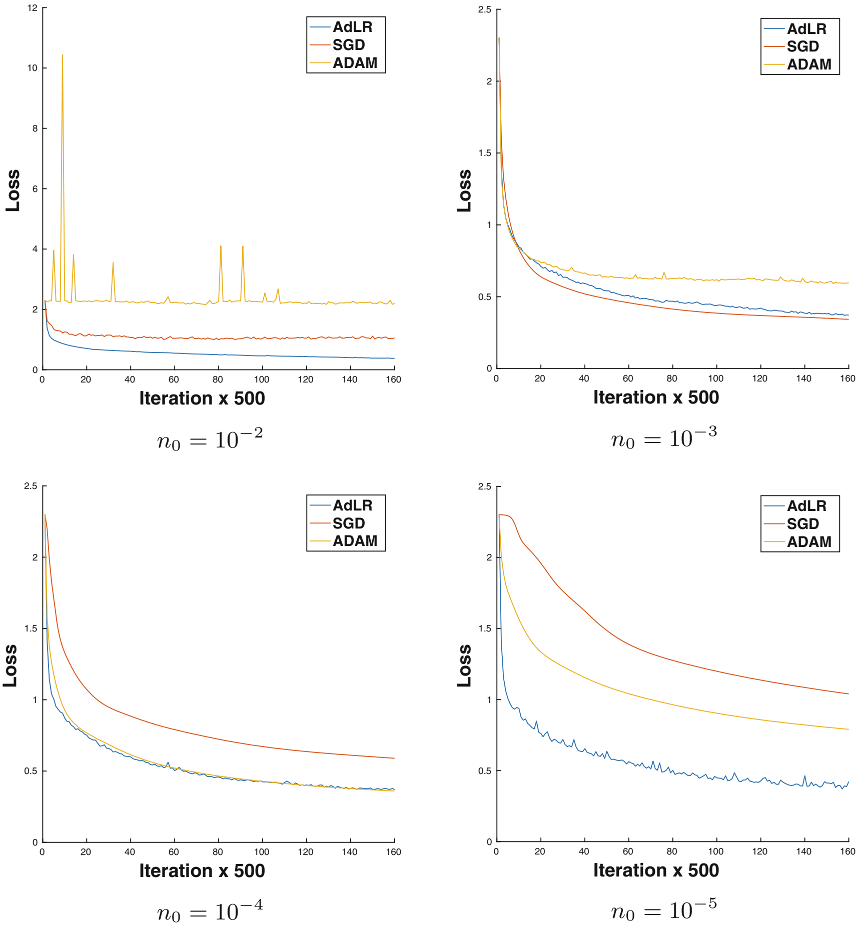
$n_0 = 10^{-3}$

$n_0 = 10^{-4}$

$n_0 = 10^{-5}$

Fig. 2. The adaptation of learning rate of the proposed AdLR algorithm with different initial learning rate values.

The null hypothesis is that the samples of each comparison are independent and derived by identical continuous distributions with equal medians. In Table 1, we mark with the "+" sign the cases when the null hypothesis is rejected at the 5% significance level and the proposed algorithm exhibits superior performance, with the "−" sign when the null hypothesis is rejected at the same level of significance and the proposed algorithm exhibits inferior performance and with "=" when the performance difference is not statistically significant. The usage of the notation (·/·) for the AdLR algorithm indicate the result of Wilcoxon test against the SGD and the ADAM algorithm, respectively.



$n_0 = 10^{-2}$

$n_0 = 10^{-3}$

$n_0 = 10^{-4}$

$n_0 = 10^{-5}$

Fig. 3. The training loss of the proposed AdLR, the SGD and the ADAM algorithm with different initial learning rate values.

The proposed algorithm adjusts the learning rate on each iteration and the Fig. 2 presents the mean value of learning rate adaptations during the training for different initial values. Independently to the initial learning rate the learning rate adapts to similar values. In addition, in Fig. 3 the mean values of the loss function during training samples are depicted.

5 Conclusions

The heuristic search for the initial learning rate value for Convolutional Neural Networks training can be difficult and time consuming, since the training time is high even when powerful GPUs are used. Usually, a trial-and-error procedure is performed. Thus, the human effort needed to find suitable parameter values, leads to the adoption of adaptive learning rate methods.

In this work we present an adaptive learning rate algorithm for CNN training. Leveraging first order gradient information of three consecutive iterations during the training phase, the learning rate is adapted. The proposed AdLR algorithm is evaluated against the popular SGD optimization algorithm and the recently proposed ADAM adaptive algorithm on the CIFAR-10 dataset. Different initial learning rate values were used and the performance of the AdLR algorithm was found to be very promising.

In future work, we intent to establish a theoretical bound of the meta-learning values of the algorithm convergence and examine the accuracy of the algorithm with different meta-learning values. Additionally, we indent to evaluate the proposed algorithm on other datasets, such as the well known ImageNet [5] and perform a comparative analysis including more state-of-the-art adaptive learning rate algorithms.

Acknowledgments. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. This work was partially supported by a Hellenic Artificial Intelligence Society (EETN) scholarship.

References

1. Almeidaa, L.B., Langloisa, T., Amaral, J.D., Plakhov, A.: Parameter adaptation in stochastic optimization. In: On-line Learning in Neural Networks, pp. 111–134. Cambridge University Press (1998)
2. Ba, J., Kingma, D.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (2015)
3. Bottou, L.: On-line learning and stochastic approximations. In: On-line Learning in Neural Networks, pp. 9–42. Cambridge University Press (1998)
4. Delibasis, K.K., Georgakopoulos, S.V., Kottari, K., Plagianakos, V.P., Maglogiannis, I.: Geodesically-corrected zernike descriptors for pose recognition in omni-directional images. Integr. Comput.-Aided Eng. **23**(2), 185–199 (2016)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR 2009 (2009)
6. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
7. Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. **36**(4), 193–202 (1980)

8. Georgakopoulos, S.V., Iakovidis, D.K., Vasilakakis, M., Plagianakos, V.P., Koulaouzidis, A.: Weakly-supervised convolutional learning for detection of inflammatory gastrointestinal lesions. In: 2016 IEEE International Conference on Imaging Systems and Techniques (IST), pp. 510–514, October 2016
9. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105. Curran Associates, Inc. (2012)
11. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)
12. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
13. Magoulas, G., Plagianakos, V., Vrahatis, M.: Adaptive stepsize algorithms for online training of neural networks. Nonlinear Anal.: Theory Methods Appl. **47**(5), 3425–3430 (2001)
14. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Fnkranz, J., Joachims, T. (eds.) Proceedings of 27th International Conference on Machine Learning (ICML-2010), pp. 807–814. Omnipress (2010)
15. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free? - weakly-supervised learning with convolutional neural networks. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 685–694, June 2015
16. Plagianakos, V.P., Magoulas, G.D., Vrahatis, M.N.: Global learning rate adaptation in on-line neural network training. In: Proceedings of 2nd International ICSC Symposium on Neural Computation (NC 2000), Berlin, Germany (2000)
17. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: a large data set for nonparametric object and scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. **30**(11), 1958–1970 (2008). http://dx.doi.org/10.1109/TPAMI.2008.128
18. Wilcoxon, F.: Individual comparisons by ranking methods. Biom. Bull. **1**(6), 80–83 (1945)
19. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. CoRR abs/1212.5701 (2012)