ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

Training Convolutional Networks with Weight–wise Adaptive Learning Rates

Alan Mosca and George D. Magoulas

Department of Computer Science and Information Systems
Birkbeck College, University of London
Malet Street, London, United Kingdom

Abstract. Current state–of–the–art Deep Learning classification with Convolutional Neural Networks achieves very impressive results, which are, in some cases, close to human level performance. However, training these methods to their optimal performance requires very long training periods, usually by applying the Stochastic Gradient Descent method. We show that by applying more modern methods, which involve adapting a different learning rate for each weight rather than using a single, global, learning rate for the entire network, we are able to reach close to state–of–the–art performance on the same architectures, and improve the training time and accuracy.

1 Introduction

In the field of supervised classification with Deep Learning methods, specifically Convolutional Networks [1], it is typical to train a model on a large dataset for a very long number of epochs, which equates to long training periods [2]. One of the common patterns of training these models is the use of Stochastic Gradient Descent (SGD) method with Momentum. In the field of traditional ANNs, improved gradient-based *update rules* have been developed, which greatly improve training speed and classification performance [3, 4, 5, 6, 7]. Some of these rules have been already used in conjunction with Convolutional Neural Networks [8], and new ones have been developed specifically with Deep Learning in mind [9, 10]. We conducted a study to show whether it is possible to train Convolutional Neural Networks efficiently in relatively short times, and concluded that there are still some improvements that can be made, in the form of weight-wise learning rates. We formulated a new gradient-based update rule, called *Weight–wise Adaptive learning rates with Moving average Estimator*– WAME, which draws on some of the strengths of existing algorithms and adds a so–called *per weight acceleration factor*. We then conducted an experimental comparison between WAME and other update rules to show that it reaches better results in less training time, on some well–known benchmark datasets in computer vision.

2 Background

In this section, we provide a brief overview of previous work relevant to this paper.

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

2.1 The Resilient propagation method– Rprop

Rprop [3] is a weight update algorithm which does not depend on the magnitude of the partial derivative of the error with respect to the weights $\frac{\partial E(t)}{\partial w_{ij}}$. Instead, it makes use of the sign of the product of the current and previous gradients $\frac{\partial E(t)}{\partial w_{ij}} \cdot \frac{\partial E(t-1)}{\partial w_{ij}}$, to determine whether an adaptive step size Δ_{ij} should be increased or decreased multiplicatively. Some variants based on this basic principle have been proposed in the literature [4, 11]. With small adaptations [8], this approach has been shown to be usable with modern Deep Neural Networks–DNNs and Convolutional Neural Networks–CNNs equipped with Dropout [12].

In Deep Learning, being able to train in *mini–batches* is often cited as an important feature of the learning algorithm, because doing so provides speed and accuracy improvements. However, this is not possible with Rprop because in SGD the gradient values follow a stochastic process and in order to maintain convergence, an update rule has to maintain consistency across multiple mini-batches, e.g. if the sum of several successive partial derivatives for a weight is zero, then the sum of the updates for those weights must be zero.

2.2 The Root mean square propagation method– RMSprop

The RMSprop method, introduced by Hinton et al. during lectures delivered on Coursera [9], has not been studied extensively in the literature, as far as we are aware. The development of the method was motivated by the need to train using mini–batches and although it was inspired by Rprop, it does not use the sign of the derivatives as Rprop does. RMSprop attempts to alleviate the dependence on the size of the partial derivatives of the error with respect to the weight that causes phenomena like the *vanishing gradients problem* or getting stuck in *saddle points* when training with SGD.

This is achieved with the introduction of a divisor $\theta_{ij}(t)$ for each gradient value, which is then multiplied by the learning rate $\lambda(t)$ at time t. The $\theta_{ij}(t)$ is then updated as an exponentially decaying mean of the square of the gradient, with a decaying rate *alpha*, as in Eq. 1, where $\alpha = 0.9$ is suggested.

$$\theta_{ij}(t) = \alpha\theta_{ij}(t-1) + (1-\alpha)\left(\frac{\partial E(t)}{\partial w_{ij}}\right)^2 \tag{1}$$

2.3 Adaptive Moment Estimation– Adam

Adam [10] is closely related to RMSprop, especially when RMSprop is used with momentum. The main difference between the two methods is that while the momentum component of RMSprop is calculated on the rescaled gradient $\Delta w_{ij}(t-1)$, Adam utilises running means of the first and second moments of the gradient. Adam also includes an initial *bias correction factor*, because the running means are initialised at 0.

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

3 Weight–wise adaptive learning rates with moving average estimator– WAME

In this section we propose a new algorithm for mini–batches and online training which employs a *per–weight acceleration factor* ζ_{ij}, used multiplicatively in the weight update. This leads to adaptively tuning the learning rate of each weight, effectively producing weight–wise adaptive learning rates. The acceleration factor is evolved following the same criterion as Δ_{ij} in Rprop: the sign of the product of the current and previous gradients. We clip these values between $[\zeta_{min}, \zeta_{max}]$ to avoid runaway effects. If used directly, this factor would have the same non–linear effect as Δ_{ij} in Rprop, so in order to use it in mini–batches and online, we have to apply smoothing that will guarantee an asymptotic agreement between different batch sizes. To achieve this, we divide by an exponentially–weighted moving average (EWMA) of ζ_{ij}, with exponential decay α. The full update rule is presented in Algorithm 1. Empirically, we have established that good values for the hyperparameters are $\alpha = 0.9$, $\eta_+ = 1.2$, $\eta_- = 0.1$, $\zeta_{min} = 0.01$, $\zeta_{max} = 100$– these are used in all experiments reported in Section 4.

Algorithm 1 WAME

pick α, η_+, η_-, ζ_{min}, ζ_{max}
$\theta_{ij}(0) = 0, Z_{ij}(0) = 0, \zeta_{ij} = 1 | \forall i, j$
for all $t \in [1..T]$ **do**
  if $\frac{\partial E(t)}{\partial w_{ij}} \cdot \frac{\partial E(t-1)}{\partial w_{ij}} > 0$ **then**
    $\zeta_{ij}(t) = \min\{\zeta_{ij}(t-1) * \eta_+, \zeta_{max}\}$
  else if $\frac{\partial E(t)}{\partial w_{ij}} \cdot \frac{\partial E(t-1)}{\partial w_{ij}} < 0$ **then**
    $\zeta_{ij}(t) = \max\{\zeta_{ij}(t-1) * \eta_-, \zeta_{min}\}$
  end if
  $Z_{ij}(t) = \alpha Z_{ij}(t-1) + (1-\alpha)\zeta_{ij}(t)$
  $\theta_{ij}(t) = \alpha \theta_{ij}(t-1) + (1-\alpha)(\frac{\partial E(t)}{\partial w_{ij}}(t))^2$
  $\Delta w_{ij}(t) = -\lambda Z_{ij} \frac{\partial E(t)}{\partial w_{ij}} \frac{1}{\theta_{ij}(t)}$
  $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$
end for

4 Experimental results

We tested WAME on three commonly used benchmark datasets in Deep Learning : MNIST, CIFAR–10 and CIFAR–100. Details about these datasets are given below and the architectures used are presented in Table 1. We followed common practice [13, 14]: MNIST uses training, validation and test sets, while CIFAR–10 and CIFAR–100 have no validation sets with the test set used for validation purposes. No augmentations were applied to the training data. We compared WAME to SGD with momentum, RMSprop and Adam, using commonly used

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

64 conv, 5×5
64 conv, 1×1
2×2 max-pooling
128 conv, 5×5
128 conv, 1×1
2×2 max-pooling
Dense, 1024 nodes
50% dropout

(a) MNIST

2×96 conv, 3×3
96 conv, 3×3, 2×2 strides
2×192 conv, 3×3
192 conv, 3×3, 2×2 strides
192 conv, 3×3
192 conv, 1×1
10 conv, 1×1
global average pooling

(b) CIFAR–10, CIFAR–100

Table 1: Network structures used in the experiments.

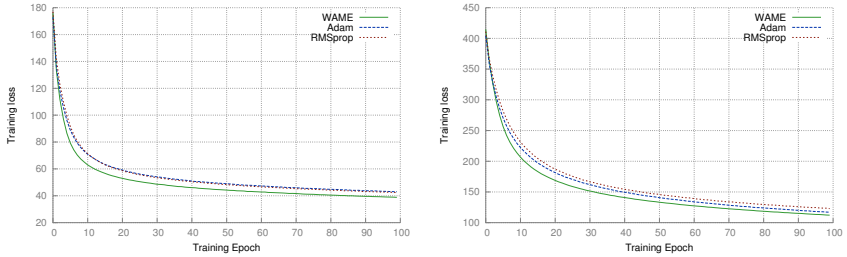
mini–batch sizes. We did not include Rprop in our evaluation because it is already known to not work well on mini–batches [9]. We used the same architecture for each algorithm. We performed our comparison by generating a random set of initial weights and then using them as the starting point for each algorithm, so as to avoid any possible distortion of the results by the random initialisation process. We repeated each experiment 20 times, each time with a different random seed, and took the means of each value. All experiments were run on the Toupee Deep Learning experimentation library, which is based on Keras. It is available at `http://github.com/nitbix/toupee`. Results are reported in Table 2, showing the number of epochs required to train the CNNs to reach the best validation accuracy, and the test accuracy recorded at that epoch. Figure 1 shows how WAME is more effective during training, by having an average loss that is consistently below that of Adam and RMSprop. A Friedman aligned ranks tests shows that the accuracy improvement on Adam and RMSprop is not significant (p–value=0.0974), while the speed improvement is significant at the 5% level both w.r.t. epoch counts and (if we exclude SGD which has lower accuracy), real time.

MNIST [13] is a dataset for labelling pre–processed images of hand–written digits. We used the network in Table 1a, trained for 100 epochs. The difference in test accuracy between Adam, RMSprop and WAME does not appear to be significant, but the improvement in speed is noticeable. MNIST is a dataset that is easy to overfit, and the accuracy levels obtained are close to the state–of–the–art without data augmentation, so the lack of significant decay of learning ability is positive.

CIFAR–10 [14] contains 60000 images of 10 categories of objects. We used the network in Table 1b, trained for 100 epochs. 50% dropout was applied after each convolution with stride 2×2. WAME achieves better maximum and average accuracy than the other methods in less epochs, and is close to state–of–the–art without data augmentation, despite being a small network.

CIFAR–100 [14] contains 60000 images of 100 categories of objects. The same CNN and training schedule as CIFAR–10 was used. WAME requires less epochs than other methods to reach better performance, producing accuracy that is close to state–of–the–art without data augmentation. Also, the minimum best

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.



(a) CIFAR-10 (b) CIFAR-100

Fig. 1: Average training loss

accuracy on all experiments with WAME is higher than the maximum of all other update rules, suggesting that there is a significant generalisation improvement.

5 Conclusions

The paper introduced WAME, a new training algorithm based on Stochastic Gradient Descent that is equipped with a self–tuning weight–wise factor and learning rate adaptation schedule for each weight. WAME was compared with some of the most popular and well–performing update rules in Deep Learning, by training CNNs on commonly used benchmark datasets in computer vision. The experiments showed that WAME offers increased learning speed without compromising generalisation performance. Further experimentation with more architectures and fine-tuned heuristics is needed in order to draw solid conclusions w.r.t. the accuracy improvements.

References

[1] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361:310, 1995.

[2] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[3] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *proceeding of the IEEE International Conference on Neural Networks*, pages 586–591. IEEE, 1993.

[4] Aristoklis D Anastasiadis, George D Magoulas, and Michael N Vrahatis. An efficient improvement of the rprop algorithm. In *Proceedings of the First International Workshop on Artificial Neural Networks in Pattern Recognition (IAPR 2003), University of Florence, Italy*, page 197, 2003.

[5] Aristoklis D Anastasiadis, George D Magoulas, and Michael N Vrahatis. New globally convergent training scheme based on the resilient propagation algorithm. *Neurocomputing*, 64:253–270, 2005.

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

	test accuracy (%)				best validation epoch				time (min)
	μ	σ	min	max	μ	σ	min	max	μ
MNIST									
SGD	99.29	0.04	99.20	99.36	46.30	11.08	11	92	23
RMSprop	99.53	0.03	99.47	99.60	44.80	19.56	12	98	47
Adam	99.53	0.04	99.45	99.60	49.30	18.78	14	90	49
WAME	99.52	0.03	99.47	99.58	33.90	11.08	13	94	28
CIFAR-10									
SGD	79.83	0.34	79.36	80.49	99.35	0.79	98	100	392
RMSprop	89.87	0.26	89.31	90.58	90.85	6.44	71	98	456
Adam	90.26	0.26	89.66	90.60	91.55	7.02	76	100	478
WAME	90.80	0.17	90.41	91.11	86.20	9.38	62	100	421
CIFAR-100									
SGD	41.52	0.26	41.23	41.92	99.33	0.82	98	100	419
RMSprop	63.97	0.56	62.95	64.80	93.33	5.58	84	100	471
Adam	64.83	0.42	64.24	65.45	95.78	3.82	87	100	490
WAME	67.25	0.32	66.75	67.87	93.05	6.72	77	100	468

Table 2: Test data accuracy for the trained CNN, number of training epochs required to reach best accuracy on the validation set and average real time for a single run if early stopping is used

[6] M.N. Vrahatis, G.D. Magoulas, and V.P. Plagianakos. From linear to nonlinear iterative methods. *Applied Numerical Mathematics*, 45(1):59 – 77, 2003. 5th IMACS Conference on Iterative Methods in Scientific Computing 28-31 May, 2001, Heraklion, Crete (Greece).

[7] Aristoklis D Anastasiadis, George D Magoulas, and Michael N Vrahatis. Improved sign-based learning algorithm derived by the composite nonlinear jacobi process. *Journal of computational and applied mathematics*, 191(2):166–178, 2006.

[8] Alan Mosca and George D Magoulas. Adapting resilient propagation for deep learning. *UK Workshop on Computational Intelligence*, 2015.

[9] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

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

[11] Christian Igel and Michael Hüsken. Improving the Rprop learning algorithm. In *Proceedings of the second international ICSC symposium on neural computation (NC 2000)*, volume 2000, pages 115–121. Citeseer, 2000.

[12] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint*, 2012.

[13] Yann Lecun and Corinna Cortes. The MNIST database of handwritten digits.

[14] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images, 2009.