Studying exponential moving averages of model weights

Julien Sven Adda* David Mizrahi* Oğuz Kaan Yüksel*
Swiss Federal Institute of Technology Lausanne (EPFL)
{julien.adda,david.mizrahi,oguz.yuksel}@epfl.ch

Abstract

We study the performance of deep neural networks composed of an exponential moving average of SGD iterates under different learning rate schedules and decay rates. We show that, on CIFAR-10, an averaged network almost always matches or outperforms its non-averaged counterpart and performs best when using a constant schedule.

1. Introduction

Recently, a variant of Polyak-Ruppert averaging, consisting of an exponential moving average of network weights, has been seeing more and more applications in deep learning across different domains. This form of weight averaging has become a standard practice to improve the performance of a model with little cost or effort, and is now used in most SOTA architectures for image classification [14, 3]. In addition, it is an essential element of many recent self-supervised learning techniques, such as mean teachers and momentum encoders [15, 5, 4, 1]. It has also been shown to help with GAN training [16].

However, despite its popularity, there has been a lack of studies on the properties of this averaged network.

In this report, we study the performance of this averaged network for different decay rates and learning rate schedules to better understand when and how it should be used. In a simple image classification setting, we observe that an averaged network almost always matches or outperforms its non-averaged counterpart and performs best when using a high learning rate with a constant schedule, even though the non-averaged network performs poorly in this setting.

2. Exponential moving average (EMA)

The idea of weight averaging goes back to Polyak [11, 12] and Ruppert [13]. This optimization technique, known as Polyak-Ruppert averaging, sets the final parameters to an equally weighted average of past iterates. More recently, Izmailov et al. [7] propose Stochastic Weight Averaging (SWA), which uses an equal average of stochastic gradient descent (SGD) iterates paired with a cyclical or constant learning rate to improve generalization in deep neural networks.

In this report, we focus exclusively on a variant of Polyak-Ruppert averaging in which an exponential moving average (EMA) of past iterates is maintained. We refer to this averaged network as the *EMA network*. Consider a deep neural network trained with SGD (or any of its variants), which we refer to as the *SGD / No EMA network*. After each training step, we perform the following update:

$$\theta_{\text{EMA}} \leftarrow \lambda \cdot \theta_{\text{EMA}} + (1 - \lambda) \cdot \theta_{\text{SGD}}$$

where θ_{SGD} and θ_{EMA} are the parameters of the SGD and EMA networks respectively, and λ is the decay rate. A lower value of λ discounts older iterates faster. In recent applications, λ commonly ranges from 0.99 to 0.9999.

3. Experiments

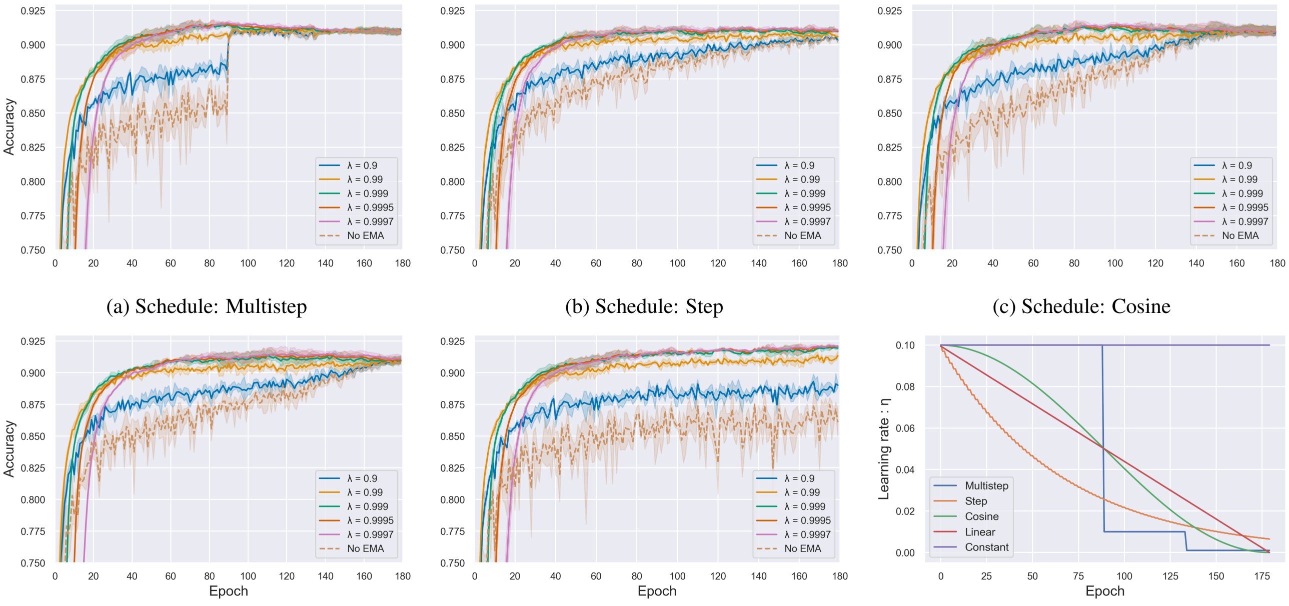
For our experiments, we train a ResNet-20 [6] on CIFAR-10. We use the training procedure specified in He et al. [6] to which we also add an EMA network. In our experiments, we modify the EMA decay rate, initial learning rate and learning rate schedule, but leave all other hyperparameters intact. More details can be found in Appendix A.

3.1. Impact of decay rate and learning rate schedule

In this section, we study the performance of EMA with five popular learning rate schedules, namely, multistep, step, cosine, linear and constant. See Appendix A for details on the choice of schedulers, and Figure 1f for the value of the learning rate at each epoch for each of these schedules. For each of these schedules, we average models using EMA decay rates $\lambda \in \{0.9, 0.99, 0.999, 0.9995, 0.9997\}$.

Figure 1 shows the effect of decay rate for a selection of learning rate schedules. First, we observe that in the early stages of training, EMA networks outperform SGD for all schedules, and that this gap closes when the learning rate is lowered. Evidently, this does not happen with the

[*] Equal contribution.



(a) Schedule: Multistep

(b) Schedule: Step

(c) Schedule: Cosine

(d) Schedule: Linear

(e) Schedule: Constant

(f) Schedules

Figure 1: **Training EMA networks.** The mean validation accuracy over 3 trials for various learning rate schedules and decay rates λ.

constant schedule, and we see significant gains with EMA. We also remark in Appendix B that norm of the difference in weights can track this dynamic between EMA and SGD. Second, we observe that using higher decay rates, which discount older iterates slower, is usually beneficial.

Table 1 shows the final test accuracies obtained with different schedules and EMA decay rates. When using schedules that strongly reduce the learning rate, EMA and SGD perform nearly identically. This is in stark contrast with the results obtained with the constant schedule. Despite SGD performing poorly when keeping the learning rate constant, EMA with a high decay rate surpasses all other schedules by a significant margin.

		Decay rate (λ)				
Schedule	No EMA	0.9	0.99	0.999	0.9995	0.9997
Multistep	**91.4**	**91.4**	**91.4**	91.4	91.3	91.4
Step	90.5	90.4	90.7	90.9	90.9	90.9
Cosine	**91.4**	**91.4**	**91.4**	91.4	91.4	91.4
Linear	91.1	91.1	91.1	91.1	91.1	91.2
Constant	86.4	89.3	**91.4**	**92.2**	**92.3**	**92.2**

Table 1: **Performance of EMA.** Performance of EMA networks for different schedules and decay rates. We report the mean test accuracy (in %) over 3 trials. Cells in **bold** correspond to the best schedule for each decay rate, and the highlighted cell corresponds to the best accuracy overall.

3.2. Impact of learning rate

Encouraged by the success of the EMA network, we report the choice of learning rate for the constant schedule in Table 2. We observe that regardless of the chosen learning rate, EMA outperforms SGD, and that the impact of EMA increases as the learning rate increases. However, it is crucial to find a right balance for the learning rate, as the performance of the SGD network deteriorates strongly when the learning rate is too high, leading to poorer results even when averaging. We further remark on the potential of EMA with constant learning rate in Section 4.

	No EMA		EMA		
η	Train	Test	Train	Test	Increase (%)
0.01	**96.8**	87.7	**99.7**	90.6	3.4
0.02	96.2	87.9	**99.7**	90.8	3.3
0.05	94.9	**87.9**	99.4	91.7	4.3
0.1	91.2	86.6	98.3	**92.2**	6.5
0.2	89.7	85.9	96.4	91.9	7.0
0.5	81.3	78.8	92.3	89.9	**14.1**

Table 2: **Impact of learning rate.** The train, test, EMA train and EMA test accuracy (in %) for 6 different learning rates (η) over 1 trial, all with a constant schedule. A decay rate of 0.9995 was used. Cells in **bold** correspond to the best learning rate for each of the columns. Increase corresponds to the EMA Test / No EMA Test ratio.

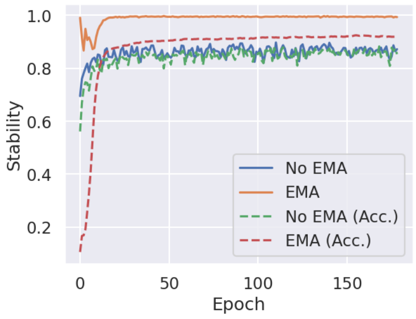


Figure 2: Stability in predictions over training. For reference, we also plot the validation accuracies over iterates.

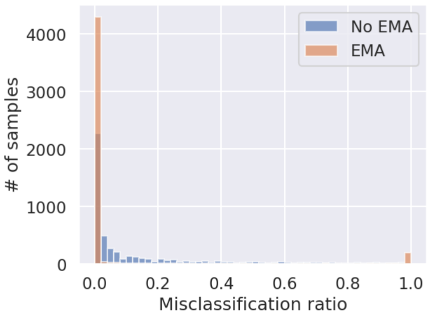


Figure 3: Histogram of misclassification ratios after training (average taken over predictions in epochs 90 to 180, per sample).

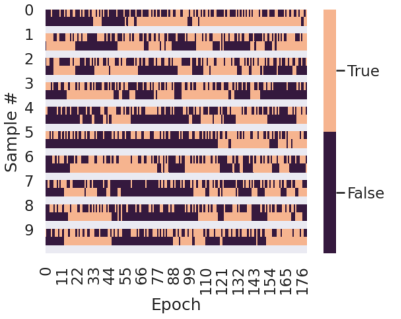


Figure 4: Predictions on individual samples visualized as barcodes for SGD (top row) and EMA (bottom row).

3.3. Further explanatory analysis

To understand the dynamics of EMA better, we conduct additional explanatory experiments. Izmailov et al. [7] show that SGD with cyclic and constant learning rate traverses around a local optimal weight space but is unable to move towards its central point. Likewise, we would like to study this rotational behavior of SGD around solutions. However, as detecting loops in high-dimensional space is hard, we look for cyclical behaviors in predictions of SGD and EMA networks. By examining correct and wrong classifications over iterates, we aim to extract information about how models traverse along on the "quotient set" formed with the equivalence characterized by equal predictive correctness on all samples in the validation set. For all the experiments in this section, we use a constant learning rate of 0.1 with an exponential decay rate of 0.9995.

Figure 2 shows how stable the predictions are on the validation set from epoch to epoch. That is, in what percentage of the samples, the correctness of prediction did not change. We observe that EMA is stable except for initial iterates, whereas the stability of SGD fluctuates.

Figure 3 shows the histogram of per sample cumulative misclassifications for the latter half of training (epoch 90 to 180). For the EMA network, the predictions are stable: each sample is either almost always correctly classified (large peak at 0), or almost always incorrectly classified (smaller peak at 1). For the SGD network, while around half of the samples are almost always correctly classified, the predictions over the remaining samples fluctuate during training. Together with the results of Figure 1, it seems as though while most of these samples are correctly classified on average, specific versions of the SGD network never correctly classify all of them at once, resulting in worsened accuracy. This behavior does not occur when using an EMA network with a high enough decay rate.

Figure 4 displays the predictions of SGD and EMA on the top ten samples that have resulted in maximum correctness flips. We observe that the correctness of the EMA model is much less volatile, whereas SGD is more likely to switch predictions in consecutive epochs.

4. Discussion

Early stopping. As we have seen in Section 3.1, EMA networks converge to the final value much more quickly than SGD networks. Considering the low computational overhead, EMA is particularly promising for training deep neural networks faster.

Choosing a learning rate schedule. Interestingly, EMA with a constant learning rate surpasses other more sophisticated learning rate schedules. We postulate that EMA benefits from the cyclical behavior of SGD, which is likely to be more pronounced at high learning rates than in schedules with an ever-decreasing learning rate. Therefore, EMA could potentially simplify the choice of learning rate schedule and allow for reliable optimization.

5. Conclusion

Taking an exponential moving average of model weights is a simple and cost-efficient technique used in several areas of machine learning. In this report, we experiment with decay rates and learning rate schedules to illuminate how and when EMA on weights is helpful in training. In particular, we observe impressive gains with a constant learning rate. Furthermore, our explanatory analysis shows that the EMA model is more stable in its predictions over iterates, suggestive of how EMA utilizes cyclical behaviors of SGD with over-parameterized deep neural networks to improve performance.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. “Emerging Properties in Self-Supervised Vision Transformers”. In: *arXiv:2104.14294 [cs]* (May 24, 2021). arXiv: 2104.14294.

[2] Terrance DeVries and Graham W. Taylor. “Improved Regularization of Convolutional Neural Networks with Cutout”. In: *arXiv:1708.04552 [cs]* (Nov. 29, 2017). arXiv: 1708.04552.

[3] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *arXiv:2010.11929 [cs]* (June 3, 2021). arXiv: 2010.11929.

[4] Jean-Bastien Grill et al. “Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 21271–21284.

[5] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. “Momentum Contrast for Unsupervised Visual Representation Learning”. In: *arXiv:1911.05722 [cs]* (Mar. 23, 2020). arXiv: 1911.05722.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, pp. 770–778.

[7] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. “Averaging Weights Leads to Wider Optima and Better Generalization”. In: *arXiv:1803.05407 [cs, stat]* (Feb. 25, 2019). arXiv: 1803.05407.

[8] Alex Krizhevsky and Geoffrey Hinton. “Learning Multiple Layers of Features from Tiny Images”. In: (2009), p. 60.

[9] Kuang Liu. *kuangliu/pytorch-cifar*. original-date: 2017-01-21T05:43:20Z. Dec. 15, 2020.

[10] Ilya Loshchilov and Frank Hutter. *SGDR: Stochastic Gradient Descent with Warm Restarts*. 2017. arXiv: 1608.03983 [cs.LG].

[11] B. T. Polyak. “New stochastic approximation type procedures”. In: *Automat. i Telemekh* 7.98-107 (1990), p. 2.

[12] B. T. Polyak and A. B. Juditsky. “Acceleration of Stochastic Approximation by Averaging”. In: *SIAM Journal on Control and Optimization* 30.4 (July 1, 1992). Publisher: Society for Industrial and Applied Mathematics, pp. 838–855.

[13] David Ruppert. “Efficient estimators from a slowly converging robbins-monro process”. In: (Feb. 1988).

[14] Mingxing Tan and Quoc V. Le. “EfficientNetV2: Smaller Models and Faster Training”. In: *arXiv:2104.00298 [cs]* (May 12, 2021). arXiv: 2104.00298.

[15] Antti Tarvainen and Harri Valpola. “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results”. In: *arXiv:1703.01780 [cs, stat]* (Apr. 16, 2018). version: 6. arXiv: 1703.01780.

[16] Yasin Yazıcı, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, Georgios Piliouras, and Vijay Chandrasekhar. “The Unusual Effectiveness of Averaging in GAN Training”. In: International Conference on Learning Representations. Sept. 27, 2018.

[17] Sergey Zagoruyko and Nikos Komodakis. *Wide Residual Networks*. 2017. arXiv: 1605.07146 [cs.CV].

Appendix

A. Experiment details

Training procedure. We conduct our experiments on CIFAR-10 [8], which consists of 50k training images and 10k test images from 10 distinct classes. We further split the training set into 45k/5k train/val sets.

We follow the training procedure from He et al. [6]: the dataset is normalized using per-channel mean and standard deviation, and the standard data augmentation is applied. That is, images are zero-padded with 4 pixels on each side to obtain a 40×40 image, and then a random 32×32 crop is extracted and flipped horizontally with 50% probability. We then train a ResNet-20 using SGD with momentum 0.9 and weight decay 0.0001, with a mini-batch size of 128 and initial learning rate of 0.1. Our implementation slightly differs in the training length. He et al. [6] train their model for 64k iterations, while we train ours for 63.4k iterations (exactly 180 epochs). We do not observe any difference in results with this slight change. We adapt the default multistep schedule accordingly, by dividing the learning rate by 10 at the 90th and 135th epoch.

Learning rate schedules. The learning rate schedules we use are either adapted from SOTA image classification models or conceptually very simple. The multistep schedule is the one used for CIFAR-10 training in the original ResNet paper [6]. The step schedule is similar to the one used by Tan and Le [14], although we adapt it for CIFAR-10 training by multiplying the learning rate by 0.97 every 2 epochs. The cosine schedule was proposed by Loshchilov and Hutter [10]. We use the version without warm restarts, for which the learning rate decreases from 0.1 to 0 following a cosine curve. The linear schedule similarly goes from 0.1 to 0, although the learning rate is decreased linearly. These two schedules are commonly used to train Vision Transformers [3].

B. Norm of weight difference

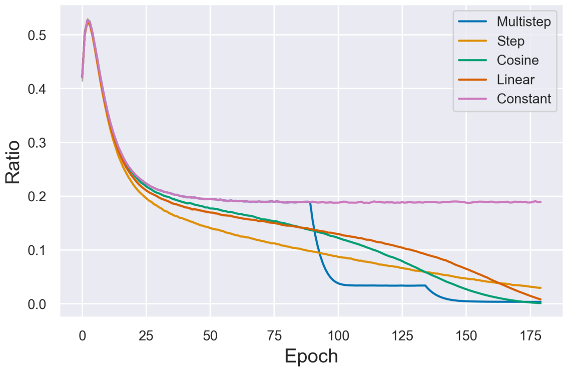


Figure 5: Norm of the difference between EMA and SGD weights divided by the norm of SGD weights for different schedules.

Figure 5 plots the norm of the difference in weights between the SGD and EMA networks relative to the norm of the weights of SGD network. As expected, the difference vanishes as the learning rate decays to zero and EMA starts to follow SGD. However, when using a constant learning rate, differences in norms stay on the same relative ratio and indeed, Section 3 shows that the impact of EMA is more pronounced with a constant learning rate.

C. Large model experiment

We conduct an additional experiment in which we train a ResNet-50 to check if our findings hold when using a much larger model (25M parameters for ResNet-50, compared to 0.27M for ResNet-20). We use a ResNet-50 as implemented in Liu [9], which differs from the one described by He et al. [6] to make it more appropriate for classification on CIFAR-10. Our training procedure is very similar to the one described Appendix A, except that we train the model for 200 epochs. We choose an EMA decay rate of 0.9995 and consider three schedules: a cosine schedule and constant schedule, both with a linear warm-up for 5000 iterations, as well as a multistep schedule identical to the one in Zagoruyko and Komodakis [17] and DeVries and Taylor [2], which divides the learning rate by 5 at the 60th, 120th and 160th epoch. We report our results in Table 3. Our findings are consistent with the ones in Section 3.1: when using a constant schedule, the EMA network outperforms everything else even though the SGD network performs poorly.

	No EMA		EMA	
Schedule	Train	Test	Train	Test
Multistep	100.0	93.4	100.0	93.3
Cosine	100.0	94.8	100.0	94.8
Constant	97.7	91.6	100.0	95.3

Table 3: **ResNet-50 results.** We report the train, test, EMA train and EMA test accuracy (in %) for 3 different schedules over 1 trial. For EMA, a decay rate of 0.9995 was used.

D. Stability of multi-step scheduling

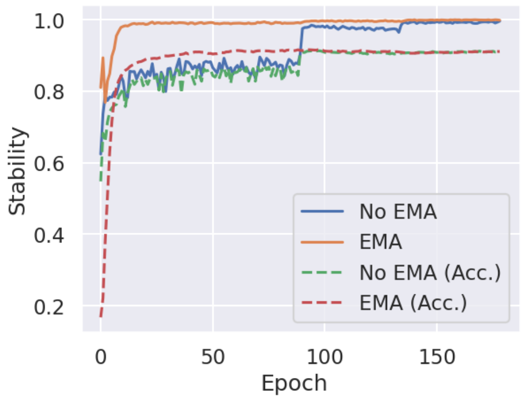


Figure 6: Stability in predictions over training for multi-step scheduling. For reference, we also plot the validation accuracies of over iterates.

Figure 6 shows how stable the predictions are for multi-step scheduling. Similar to Figure 2, we observe that stability correlates strongly with the validation accuracy on the first part of the training. We consider this curious result as an implication that studying stability is potentially beneficial to understand model behavior during training.

E. Unstable samples

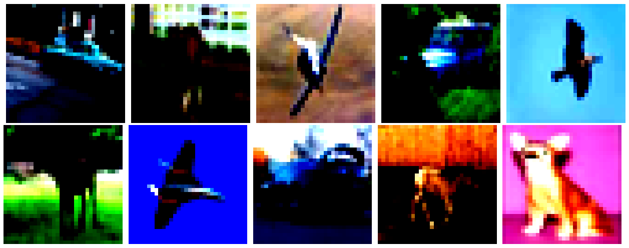


Figure 7: Top ten samples resulting in maximum correctness flips, used in Figure 4.