Averaging Weights Leads to Wider Optima and Better Generalization

Sacha Elkoubi
sacha.elkoubi@hec.edu

Michael Ein Dor
michael.ein-dor@polytechnique.edu

Abstract

The paper that we analysed introduces the Stochastic Weight Averaging method (SWA), a new optimization strategy that diverges from the traditional paradigm of minimizing the loss function through a simple trajectory [3]. This strategy consists in averaging the weights of the models obtained through a Stochastic Gradient Descent. According to the authors, this strategy enables highly efficient fine-tuning of model parameters, and is equivalent to the fast ensembling techniques that have proved their worth in many contexts (e.g. Kaggle challenge). In addition, their SWA technic have the advantage of not requiring excessive weight storage. In this work we explain the authors' intuitions, we reimplemente the algorithm in a simple and controllable case, we analyze the results obtained with regard to the article, and we compare it to a recent improvement of the technique called Periodic SWA.

1 Introduction

Optimizing neural networks is a trade-off between achieving high training accuracy and robust generalization to unseen data. When it comes to the later epochs of the training of a neural network, optimization strategy becomes very important. Most optimization strategies lead the model to favor minimums that are too narrow, or to increase too much the variance of the models, which leads to overfitting.

In their paper the authors introduce Stochastic Weight Averaging (SWA), an algorithm challenging the conventional idea of decreasing learning rates in favor of averaging model weights along the trajectory of Stochastic Gradient Descent (SGD) employing either cyclical or constant learning rates. The authors explain that such a technique converges to a larger minimum than the minima reached with SGD, that also has better generalization capabilities.

In their research, the authors evaluate Stochastic Weight Averaging (SWA) against Stochastic Gradient Descent (SGD) across various models (Resnet, VGG, WRN, PyramidNet) and datasets. However, they apply uniform learning rates, momentum, and other learning parameters for all models and datasets. While this approach simplifies implementation, **it diverges significantly from real-world application scenarios**, where models or specific tasks often necessitate tailored adjustments to the learning rate. In our study, we focus exclusively on the Resnet-50 model for the classification task on the MNIST-10 dataset, **opting to fine-tune the parameters for each method individually rather than employing a one-size-fits-all learning rate**. This approach mirrors the conditions under which models are more typically trained.

First, we explore the efficacy of SWA under these more realistic conditions by comparing it to traditional fine-tuning methods like SGD with a decaying learning rate. Then, we examine the geometrical implications on the loss function, interpret the outcomes, and suggest testing an enhancement proposed by other researchers, named PSWA (Periodic Stochastic Weight Averaging).

Preprint. Under review.

2 Why should we use SWA ?

2.1 Intuition

The main idea behind SWA is to average multiple points in the parameter space to find a solution that is better than any individual point. This approach **can lead to better generalization and reduce the variance of the model's predictions**.

The initial hypothesis supporting improved performance through Stochastic Weight Averaging (SWA) draws on the principle of variance reduction, akin to the concept behind Ensembling. Ensembling combines multiple models to create a single predictive model. The central intuition is that if the models make independent errors, the averaged prediction is likely to be closer to the true value, reducing the variance of the predictions. However, training and maintaining multiple deep learning models can be computationally expensive.

SWA emerges as a resource-efficient alternative to traditional Ensembling in the deep learning context. Instead of training multiple separate models, SWA periodically saves the weights of a single model during its training phase (often at the end of each epoch) and computes the average of these weights. By averaging weights over different stages of the training process, SWA effectively simulates the ensembling of different models without having to save many weights, which can be memory-intensive.

2.2 The implementation

Stochastic Weight Averaging (SWA) consists in averaging weights from samples proposed by Stochastic Gradient Descent (SGD). Moreover, the authors couple this method with a learning rate schedules that improves exploration within the weight space of the network.

Algorithm 1 Stochastic Weight Averaging

Require: weights $\mathbf{w}$, LR bounds α_1, α_2, cycle length c (for constant learning rate $c = 1$), number of iterations n

Ensure: $\mathbf{w}_{\text{SWA}}$

$\mathbf{w}_{\text{SWA}} \leftarrow \mathbf{w}$
$\mathbf{w}_{\text{init}} \leftarrow \mathbf{w}$
for $i \leftarrow 1, 2, \ldots, n$ **do**
  $\alpha(i) \leftarrow$ Calculate LR for the iteration i
  $\mathbf{w} \leftarrow \mathbf{w} - \alpha(i)\nabla C(\mathbf{w})$ ▷ Stochastic gradient update
  if $\text{mod}(i, c) == 0$ **then**
    $n_{\text{models}} \leftarrow i/c$ ▷ Number of models
    $\mathbf{w}_{\text{SWA}} \leftarrow \frac{\mathbf{w}_{\text{SWA}} \cdot n_{\text{models}} + \mathbf{w}}{n_{\text{models}} + 1}$ ▷ Update average
  end if
end for
Compute BatchNorm statistics for $\mathbf{w}_{\text{SWA}}$ weights

In terms of computational complexity, SWA adds a negligible overhead compared to conventional SGD, as it requires holding an additional model in memory to store the running average of weights. The actual computation of the weight average is a simple operation, performed at the end of each cycle or epoch, and does not significantly impact the computational cost. Thus, while SWA introduces conceptual and procedural complexity through its learning rate schedule and averaging mechanism, **it remains computationally efficient, making it a practical extension of SGD for improving model generalization**.

2.3 Explanation of the performances improvements according to the authors

In their research, the authors present an analysis where Stochastic Weight Averaging (SWA) demonstrates superior performance over Stochastic Gradient Descent (SGD), offering multiple rationales for this observation.

As we explained previously, the initial argument draws from the concept of model ensembling, suggesting that by averaging models that are not fully correlated, SWA can decrease the model's sensitivity to minor input variations, akin to traditional Ensembling methods.

Furthermore, they associate the practice of weight averaging with the likelihood of finding solutions within broader and potentially more robust optima, which is inspired from [2]. Their last perspective is that an advantage of SWA is its ability to explore parameter space. SWA, by averaging weights, allows the model to explore a broader area of the parameter space, potentially converging to a broader, flatter minimum.

In the next part we will analyze these two first explanations and try to verify it in our implementation.

2.4 Challenges of Hyperparameter Tuning

In our implementation we were able to verify that the success of SWA is also highly dependent on the configuration of momentum and learning rate during the training process. These two hyperparameters play pivotal roles in determining the trajectory of the model's exploration of the parameter space and, consequently, the diversity and utility of the weights being averaged.

Our heuristic was as follows: for each of the procedures, we chose to manually tune the parameters in order to obtain the best results for the given procedure. For that, we launched many training and we have only kept the best one for each algorithm.

3 Results of the SWA

3.1 Experiment

To examine the authors' results, we chose to evaluate the various fine-tuning methodologies utilizing the MNIST-10 dataset for a classification task with a ResNet-50 model. Our methodology is outlined as follows:

- Initially, the ResNet-50's final layers were trained on the MNIST-10 dataset using the Adam optimizer.
- Upon completing the initial training phase, the ResNet-50 is fine-tunes through a range of methods including SGD, SWA, and SWA with a cyclical learning rate, among others.

For the SGD strategy involving a multiplicative reduction of the learning rate, we initiated with a learning rate (LR) of 0.01, reducing it by a factor of 4 every 5 iterations.

We opted for a momentum of 0.9 to facilitate the fastest convergence. In the case of standard SGD, the learning rate was held constant at 0.005, a parameter choice derived from achieving optimal results in this setup. For the SWA algorithm, a cyclical learning rate fluctuating between 0.005 and 0.01 was employed.

3.2 Results obtained with our implementation.

The results obtained by SWA finetuning during training can be seen in the table 3.2.

Method	**Final test set accuracy**
Simple SGD **with adjusted learning rate**	93.01
SGD with Multiplicative Scheduler	93.0
SWA with Cyclical LR	**93.04**
SWA with Constant LR	92.90
Model without finetuning (only ADAM training)	92.31

Table 1: Comparison of Loss and Accuracy for Different Finetuning Strategies

As we can see from this table of results, the SWA is indeed an effective strategy for fine-tuning the model. However, because we adapt the learning rate of the SGD, **the results of SWA are extremely close to those of SGD**.

In comparison with the authors' article, the difference in results between the SGD and the SWA is much smaller because the authors seem to have favoured an extremely high learning rate for the SGD, which does not allow convergence. **This learning rate is probably appropriate for some models and in some cases, but for our model, it does not lead to satisfactory results for the SGD and must be adjusted.** While a high learning rate makes sense for the SWA (since the averaging operation can be likened to a reduction in the learning rate), it does not produce good results for the SGD. In our configuration (learning rate equal to 0.005 for SGD), the benefits of SWA are smaller.

One advantage of SWA not mentioned by the authors is its stability. This stability can be seen in figure 1. Even with a cyclic learning rate scheduler that takes very high values for the learning rate and that induces an instability, the SWA procedure allows to keep a stable accuracy. The loss function is very smooth.

Figure 1: Training curve of the SWA weights with cyclical SGD

3.3 Performance interpretation

We also wished to verify the authors' results on this particular model, which consisted in explaining that the SWA reaches a thicker local minimum than the classic SGD. Our strategy involved defining an alpha direction—a vector direction—between each fine-tuned model (obtained through either SWA or SGD) and its corresponding non-fine-tuned counterpart. This step was critical for mapping out the local geometry of the loss landscape in relation to the models under consideration. This procedure is directly inspired from Guo & all [1]. The results can be visualize in the figure 2.

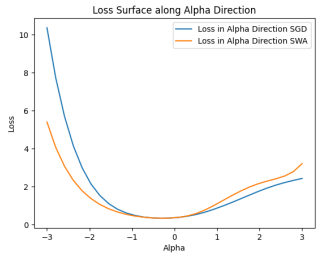

Figure 2: Loss geometry in the neighbourhood of the final model

Here too, we did not find a particular difference between SWA and the classic SGD. On one side of the loss, SWA is better, and on the other side SGD is better, but no conclusion can be drawn.

To explain why SWA performs slightly better than SGD (especially visible in figure 1), we can assume that the averaging effect reduces the variance of the model without, however, allowing to place it in a better minimum. So, **in our case, SWA is very close to ensembling**.

4 To go further : Periodic SWA

A fairly obvious limitation of the SWA presented by the authors in their paper is that it averages all the models without taking their performance into account. PSWA was proposed in [1] in order to enhance SWA by periodically applying the weight averaging process throughout the training, rather than a single averaging step at the end. This allows **only the latest epochs to be taken into account, which improves the SWA results without compromising the stability of the model on the loss**.

Assume that we have a budget of N epochs, the training process is divided into P periods, each consisting of M epochs, where $N = P \times M$. At the end of each period, a weight averaging operation is performed, similar to SWA, but on a subset of the weight vectors corresponding to that period.

Algorithm 2 PSWA

Require: Initial weights $\tilde{w}$, learning-rate schedule, cycle length c, total number of iterations n, number of phases p
Ensure: Optimized weights w_{PSWA}
1: **for** $i = 1$ **to** p **do**
2: Run SWA procedure with input $\tilde{w}$, c, n/p.
3: Update $\tilde{w}$ with the output from SWA.
4: **end for**
5: **return** $\tilde{w}$ as w_{PSWA}

4.1 Results

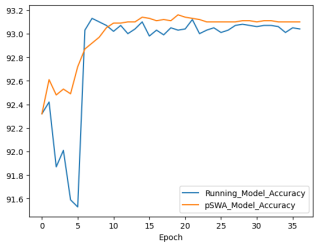

Figure 3: Different models accuracy during training, $p = 3$

As we can see in 3, the Periodic SWA algorithm with $p = 3$ outperforms the SWA. Moreover, the PSWA offers the same smoothness of the loss than the SWA. This shows the interest to take into account only the last values and to update gradually the model used to disseminate the parameters.

4.2 Opening on possible improvements

The main problem of PSWA is that this algorithm takes into account only the last epochs to perform the average. A solution could be to take inspiration from stacking methods and average models with a weighted sum based on their score on a validation set. These techniques generally provide excellent results and are widely used in machine learning competitions to create the best ensembing of several models.

5 Conclusion

To conclude, the SWA algorithm proposed by the authors is indeed an effective way of finetuning models, which seems to perform slightly better than SGD while providing much better loss stability over time.

However, by tuning the SGD parameter in the context of our specific task, we observe that the results between SGD and SWA are ultimately **much closer than those obtained by the authors in their papers**. This result is not surprising **since their learning rate was extremely high for SGD**. Furthermore, the two minima reached with this optimal learning rate seem to have the same width, which makes the geometric explanation less plausible than the variance reduction explanation.

It's critical, however, to acknowledge the scope of our analysis is limited to a single case study. Thus, there's a possibility that the specific conditions of our experiment might not be the most favorable to the SWA.

Finally, we were able to test a recent PSWA model that uses only the latest epochs and by adapting the model that allows exploration. **This model outperforms the SWA and looks promising for high-precision applications (e.g. Kaggle competition)**.

References

[1] Liu Guo, Jin. Stochastic weight averaging revisited. *Applied Sciences*, 2023. Academic Editor: Sangtae Ahn.

[2] Jorge Nocedal Mikhail Smelyanskiy Ping Tak Peter Tang Nitish Shirish Keskar, Dheevatsa Mudigere. On large-batch training for deep learning: Generalization gap and sharp minima.

[3] Dmitry Vetrov Andrew Gordon Wilson1 Pavel Izmailov Dmitrii Podoprikhin, Timur Garipov. Averaging weights leads to wider optima and better generalization.