Averaging Weights Leads to Wider Optima and Better Generalization

Pavel Izmailov[1] Dmitrii Podoprikhin[2,3] Timur Garipov[4,5]
Dmitry Vetrov[2,3] Andrew Gordon Wilson[1]

[1]Cornell University

[2]Higher School of Economics

[3]Samsung-HSE Laboratory

[4]Samsung AI Center in Moscow

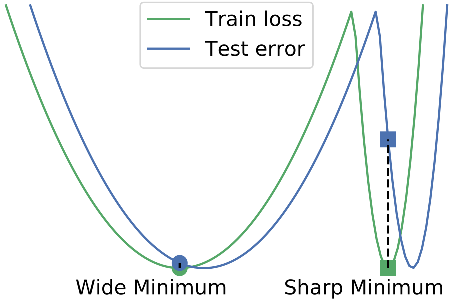
[5]Lomonosov Moscow State University

Uncertainty in Artificial Intelligence
Monterey, California, USA

August 9, 2018

Optima Width

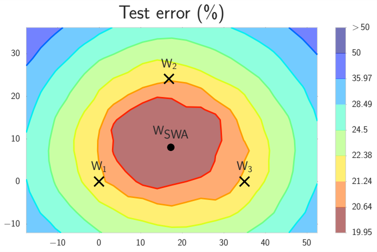
Optima width is conjectured to be correlated with generalization (Keskar et al. [2017], Hochreiter and Schmidhuber [1997])



Talk Outline

We propose Stochastic Weight Averaging (SWA) — an **equally weighted** running average of parameters (DNN weights) traversed by SGD with a modified learning (cyclical or high constant) rate schedule.

- Improves generalization
- No significant computational overhead
- Extremely easy to implement and use

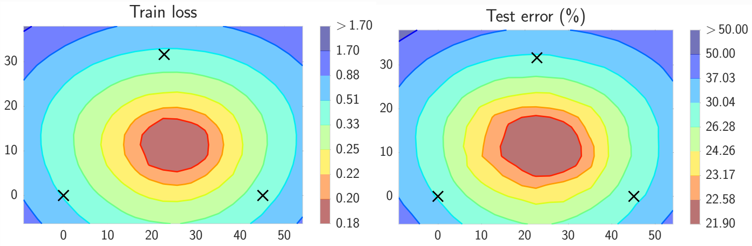


Explanation:

- Finds wider solutions centered in the set of high-performing networks
- Approximates ensembling

SGD Experiment: Constant Learning Rate

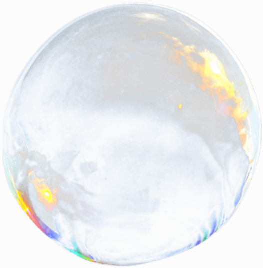
Run SGD with constant learning rate and visualize trajectory



- SGD iterates stay at the boundary of a high-quality region
- Averaging iterates improves performance
- Shift between train and test

Explanation: Soap Bubble

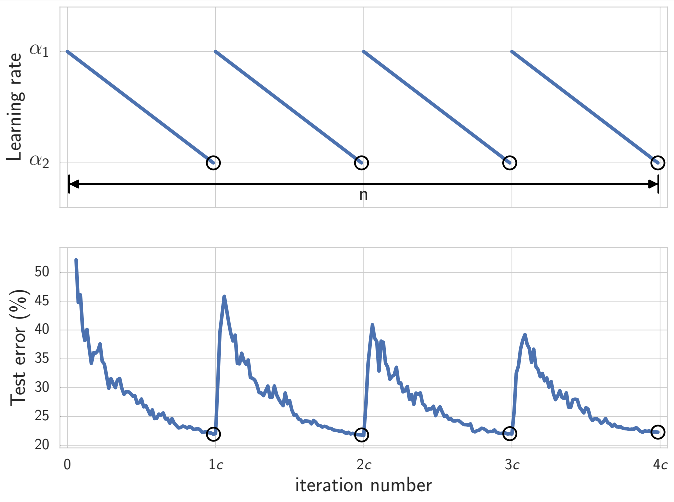
Mandt et al. [2017]: SGD with fixed learning rate samples from a Gaussian distribution centered at the minimum of the loss.



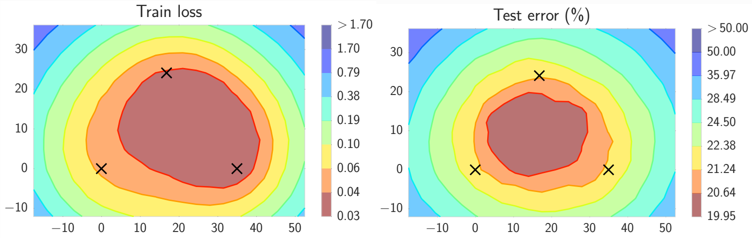
SGD iterates concentrate on a surface of an ellipsoid. Averaging lets us go inside the ellipsoid!

Cyclical Learning Rate

What if we use a cyclical learning rate?



SGD Experiment: Cyclical Learning Rate



Observations still hold:

- SGD iterates stay at the boundary of a high-quality region
- Averaging iterates improves performance
- Shift between train and test

Explanation: Ensemble Approximation

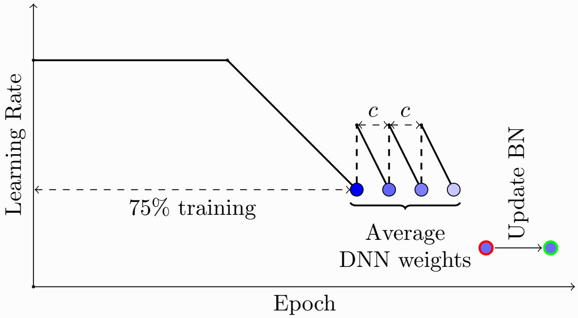
- SGD is taking small steps, so averaging weights $\approx$ ensembling by linearization

$$f\left(\frac{1}{n}\sum_{i=1}^{n} w_i\right) \approx \frac{1}{n}\sum_{i=1}^{n} f(w_i)$$

- Empirically, averaging weights and ensembling SGD iterates indeed lead to similar predictions

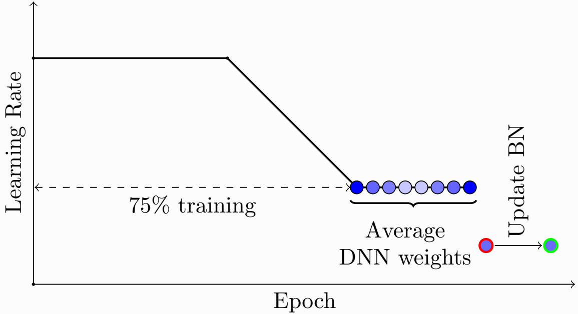
SWA details

- Use learning rate schedule that doesn't decay to zero (cyclical or constant)
- Average weights
  - Cyclical LR ⇒ at the end of each cycle
  - Constant LR ⇒ at the end of each epoch
- Recompute Batch Normalization statistics at the end of training; in practice do one additional forward pass on train data



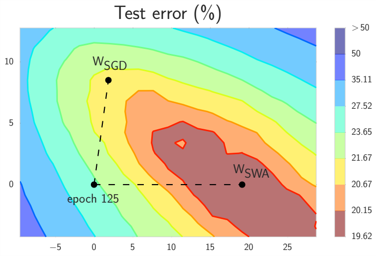
SWA details

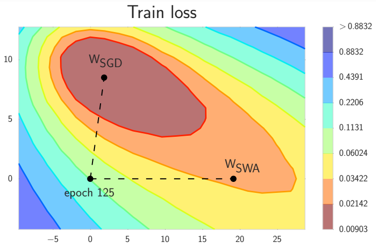
- Use learning rate schedule that doesn't decay to zero (cyclical or constant)
- Average weights
  - Cyclical LR $\Rightarrow$ at the end of each cycle
  - Constant LR $\Rightarrow$ at the end of each epoch
- Recompute Batch Normalization statistics at the end of training; in practice do one additional forward pass on train data



SWA vs SGD

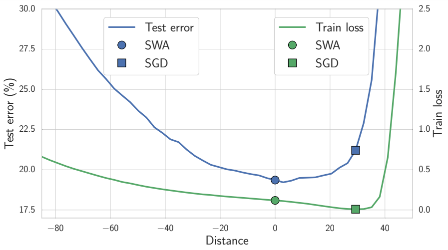
Run SGD and SWA from the same initialization (ResNet-164, CIFAR-100)

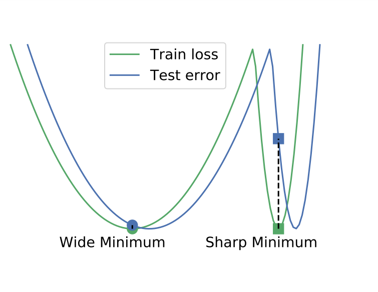




- SGD achieves better train loss
- SWA achieves better test accuracy
- Large shift between train and test

Connecting SWA and SGD Solutions



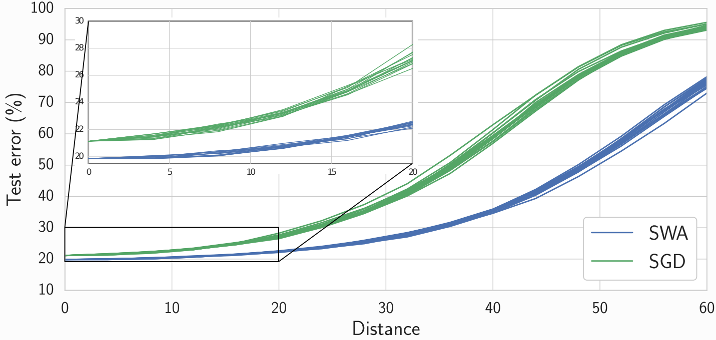


$$w(t) = t \cdot w_{\mathsf{SGD}} + (1 - t) \cdot w_{\mathsf{SWA}}$$

SWA Optima Width: Test Error

Width along random rays

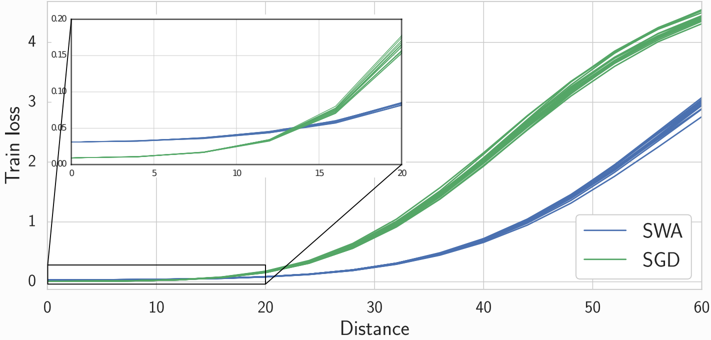
$$w(t) = \{w_{\textsf{SWA}}, w_{\textsf{SGD}}\} + t \cdot \frac{d}{\|d\|}, \quad d \sim \mathcal{N}(0, I)$$



SWA Optima Width: Train Loss

Width along random rays

$$w(t) = \{w_{\mathsf{SWA}}, w_{\mathsf{SGD}}\} + t \cdot \frac{d}{\|d\|}, \quad d \sim \mathcal{N}(0, I)$$



SWA Results

DNN (Budget)	SGD	SWA 1 Budget	SWA 1.5 Budget
CIFAR-100			
VGG-16 (200)	72.55 ± 0.10	73.91 ± 0.12	$\mathbf{74.27 \pm 0.25}$
ResNet-164 (150)	78.49 ± 0.36	79.77 ± 0.17	$\mathbf{80.35 \pm 0.16}$
WRN-28-10 (200)	80.82 ± 0.23	81.46 ± 0.23	$\mathbf{82.15 \pm 0.27}$
PyramidNet-272 (300)	83.41 ± 0.21	–	$\mathbf{84.16 \pm 0.15}$
CIFAR-10			
VGG-16 (200)	93.25 ± 0.16	93.59 ± 0.16	$\mathbf{93.64 \pm 0.18}$
ResNet-164 (150)	95.28 ± 0.10	95.56 ± 0.11	$\mathbf{95.83 \pm 0.03}$
WRN-28-10 (200)	96.18 ± 0.11	96.45 ± 0.11	$\mathbf{96.79 \pm 0.05}$
ShakeShake-2x64d (1800)	96.93 ± 0.10	–	$\mathbf{97.12 \pm 0.06}$
Imagenet			
DNN	SGD	SWA 5 epochs	SWA 10 epochs
ResNet-50	76.15	76.83 ± 0.01	$\mathbf{76.97 \pm 0.05}$
ResNet-152	78.31	78.82 ± 0.01	$\mathbf{78.94 \pm 0.07}$
DenseNet-161	77.65	78.26 ± 0.09	$\mathbf{78.44 \pm 0.06}$

Applications and Extensions

- Two papers at UDL workshop tomorrow!
  - Improving Stability in Deep Reinforcement Learning with Weight Averaging
  - Fast Uncertainty Estimates and Bayesian Model Averaging of DNNs
- Athiwaratkun et al. [2018]: use a modified version of SWA to get SOTA results in Semi-Supervised Learning

Summary

- SWA is a simple technique that consistently improves generalization with deep neural networks with virtually no computational overhead
- SWA is very easy to use and implement and proved useful in a range of applications
- Code is available, so we encourage you to try SWA for yourself!
  - PyTorch: `https://github.com/timgaripov/swa`
  - Chainer: `https://github.com/chainer/models/tree/master/swa`
  - fast.ai: `https://github.com/fastai/fastai`

References

Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Improving consistency-based semi-supervised learning with weight averaging. *arXiv preprint arXiv:1806.05594*, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations*, 2017.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *The Journal of Machine Learning Research*, 18(1):4873–4907, 2017.