Sharpness-Aware Minimization For Efficiently Improving Generalization

Kyeong Ryeol, Go

M.S. candidate of OSI Lab

Contents

1. Motivation
2. Preliminary
3. SAM optimizer
4. Experiment
5. Critique

Motivation

Problem setting

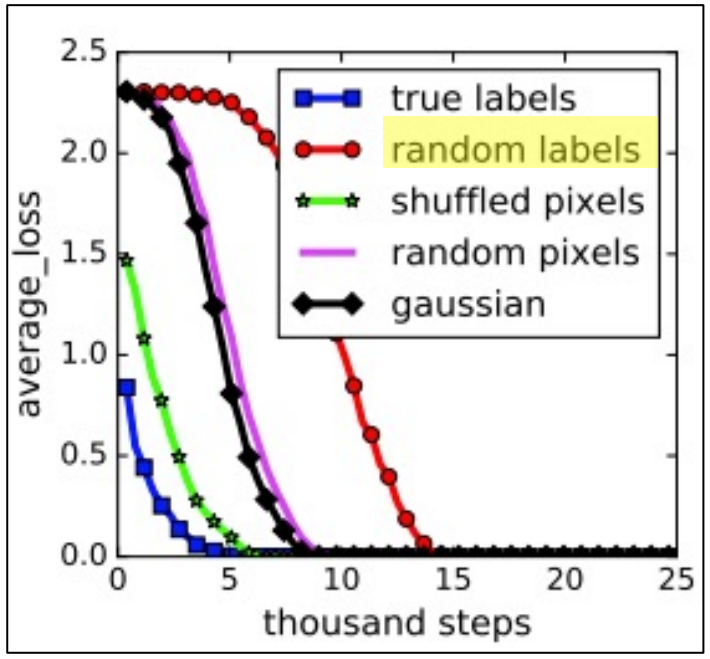
- Notation
  - Training dataset : $S = \cup_{i=1}^{n} \{(x_i, y_i)\} \sim \mathcal{D}$
  - Model parameter : $w \in \mathcal{W} \subseteq \mathbb{R}^k$
  - Per-data-point loss function : $l: \mathcal{W} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$
  - Empirical loss function : $L_S(w) = \frac{1}{n} \sum_{i=1}^{n} l(w, x_i, y_i)$
  - Population loss function : $L_{\mathcal{D}}(w) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[l(w, x, y)]$

- Objective
  - Select model parameter w having low population loss $L_{\mathcal{D}}(w)$

  * Conventional approach : $\min_{\mathrm{w}} L_S(w) + \lambda \|w\|_2^2$

Minimizing training loss is not sufficient

- Deep neural networks easily fit random labels
- Explicit regularization is not sufficient for controlling generalization error



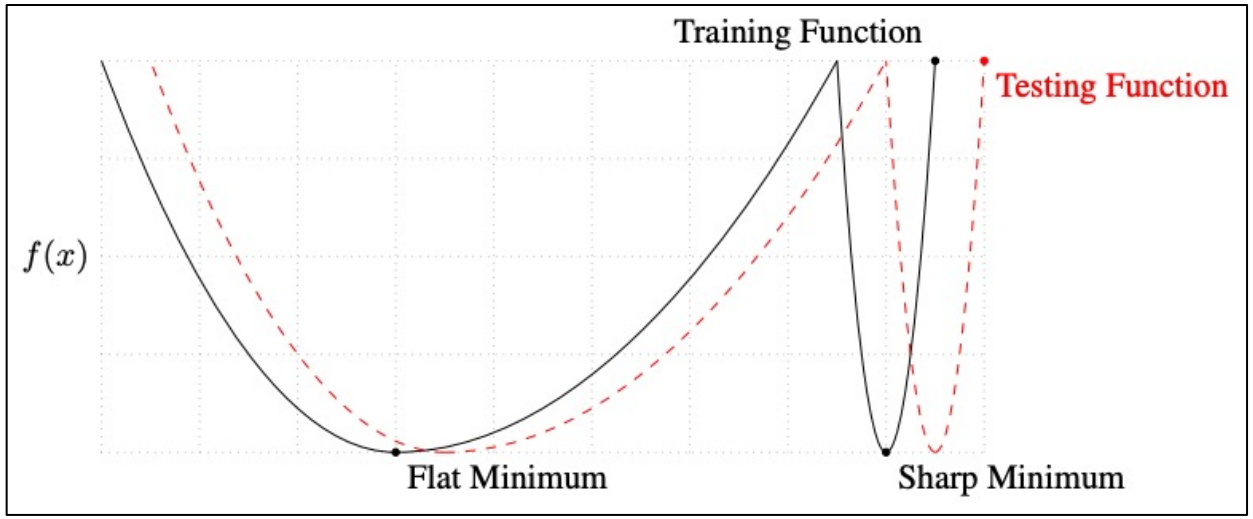
Training loss decaying with steps

data aug	dropout	weight decay	top-1 train	top-5 train	top-1 test	top-5 test
ImageNet 1000 classes with the original labels						
yes	yes	yes	92.18	99.21	77.84	93.92
yes	no	no	92.33	99.17	72.95	90.43
no	no	yes	90.60	100.0	67.18 (72.57)	86.44 (91.31)
no	no	no	99.53	100.0	59.80 (63.16)	80.38 (84.49)
Alexnet (Krizhevsky et al., 2012)			-	-	-	83.6
ImageNet 1000 classes with random labels						
no	yes	yes	91.18	97.95	0.09	0.49
no	no	yes	87.81	96.15	0.12	0.50
no	no	no	95.20	99.14	0.11	0.56

Performance on ImageNet with true labels and random labels

Zhang, Chiyuan, et al. "Understanding deep learning (still) requires rethinking generalization." *ICLR 2017.*

Loss Landscape v.s. Generalization



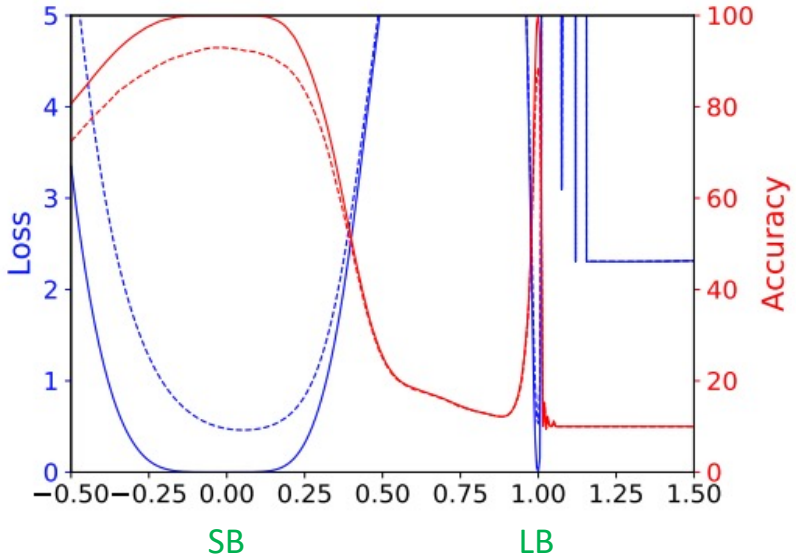
If there exists a shift from empirical loss to population loss, flat minimum is more robust than sharp minimum.

- Smoothening loss landscape : Skip connection, Batch normalization
- Escaping basins of sharp minima : Small-batch training
- Biasing towards flat side : Averaging SGD trajectories

Keskar, Nitish Shirish, et al. "On large-batch training for deep learning: Generalization gap and sharp minima." ICLR 2017.
Izmailov, Pavel, et al. "Averaging weights leads to wider optima and better generalization.", UAI 2018.
Santurkar, Shibani, et al. "How does batch normalization help optimization?." NeurIPS 2018.
He, Haowei, Gao Huang, and Yang Yuan. "Asymmetric valleys: Beyond sharp and flat local minima." NeurIPS 2019.

How to analyze sharpness?

1D linear interpolation

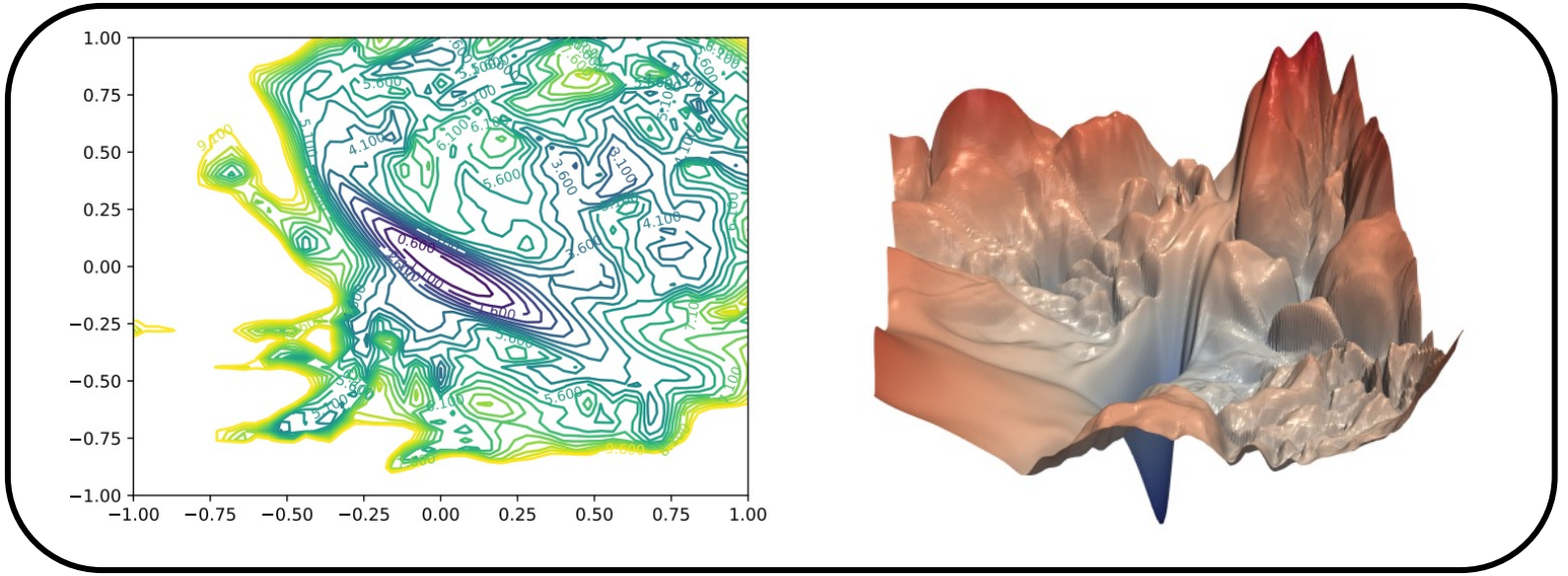


$L(\alpha w_l^* + (1-\alpha) w_s^*)$ with $-1 \leq \alpha \leq 1$

(w_s^* : converged w/ Small Batch (SB) training)

(w_l^* : converged w/ Large Batch (LB) training)

2D contour plot and its 3D visualization



$L(w + \alpha\delta + \beta\eta)$ with $-1 \leq \alpha, \beta \leq 1$

(δ and η are random weights of the same size of w)

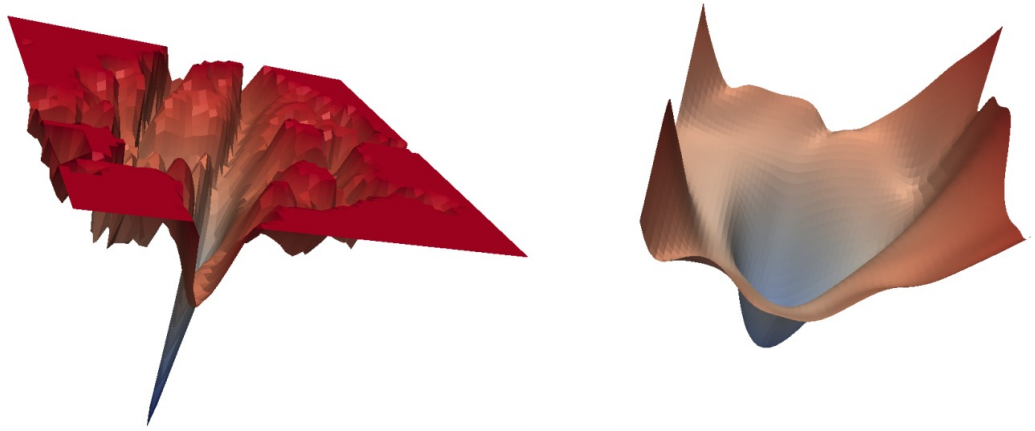
(Filter-wise normalization in aware of *scale invariance*)

Previous studies have shown better generalization, however, yet struggle to beat a range of state-of-the-art models

Li, Hao, et al. "Visualizing the loss landscape of neural nets.", *NeurIPS 2018.*

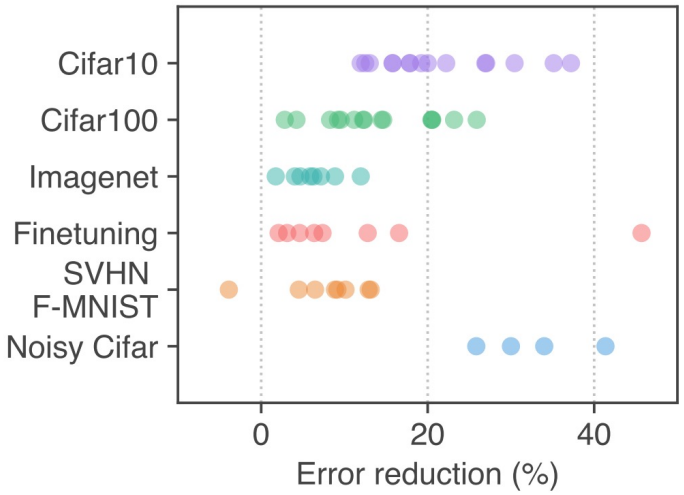
Sharpness-Aware Minimization (SAM)

- Find a wide and flat minima by simultaneously minimizing loss and sharpness
  1. Seeks parameters that lie in neighborhoods having uniformly low loss
  2. Conduct a rigorous empirical study across a range of widely studied computer vision tasks



(a) Without SAM (b) With SAM

3D visualization of the loss landscape around the converged solutions



Error rate reduction by switching to SAM (dataset / model / augmentation)

Preliminary

PAC versus PAC-bayes

- PAC (Probably Approximately Correct)

With probability greater than $1-\delta$,

$$L_{\mathcal{D}}(w) \leq L_S(w) + \sqrt{\frac{1}{2n}\left(\ln|\mathcal{H}| + \ln\left(\frac{2}{\delta}\right)\right)}$$

- PAC-bayes

With probability greater than $1-\delta$,

$$\mathbb{E}_{w\sim Q}[L_{\mathcal{D}}(w)] \leq \mathbb{E}_{w\sim Q}[L_S(w)] + \sqrt{\frac{1}{2(n-1)}\left(KL(Q|P) + ln\left(\frac{n}{\delta}\right)\right)}$$

Learning algorithm w/ PAC-bayes bound

1. Fix a probability $\delta > 0$ and a distribution $P \sim \mathcal{N}(\mu_P, \sigma_P^2 I)$
2. Collect an i.i.d. dataset S of size n
3. Compute the optimal distribution $Q \sim \mathcal{N}(\mu_Q, \sigma_Q^2 I)$ that minimizes

$$\mathbb{E}_{w \sim Q}[L_S(w)] + \sqrt{\frac{1}{2(n-1)}\left(KL(Q|P) + ln\left(\frac{n}{\delta}\right)\right)}$$

4. Return the randomized classifier given by Q

* *Closed form expression of KL divergence :* $KL(Q|P) = \frac{1}{2}\left[\frac{k\sigma_Q^2 + \|\mu_P - \mu_Q\|_2^2}{\sigma_P^2} - k + k \ln\left(\frac{\sigma_P^2}{\sigma_Q^2}\right)\right]$

Dziugaite, Gintare Karolina, and Daniel M. Roy. "Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data." *UAI 2016*

How can we set the prior P in advance?

- Predefine values for σ_P^2 and pick the best one in that set
  - Set $\delta_j = \frac{6\delta}{\pi^2 j^2}$ and $\sigma_P^2 \in T \coloneqq \left\{ c \exp\left(\frac{1-j}{k}\right) \middle| j \in \mathbb{N} \right\}$

    Let A_j is an event s.t. $\mathbb{E}[Z] > \bar{Z} + g(\delta_j)$ where $\Pr(A_j) < \delta_j$

    Then, $\Pr\left(\cup_{j\in\mathbb{N}} A_j\right) < \sum_{j\in\mathbb{N}} \delta_j = \frac{6\delta}{\pi^2} \sum_{j\in\mathbb{N}} \frac{1}{j^2} = \delta$

  ∴ With probability $1 - \delta$, PAC-bayes bound holds for all $\sigma_P^2 \in T$

Langford, John, and Rich Caruana. "(Not) bounding the true error." *NIPS 2001*

Component-wise level bound

$$\mathbb{E}_{w\sim Q}[L_{\mathcal{D}}(w)] \leq \mathbb{E}_{w\sim Q}[L_S(w)] + \sqrt{\frac{1}{2(n-1)}\left(KL(Q|P) + ln\left(\frac{n}{\delta_j}\right)\right)}$$

Reparameterization : $w+\epsilon$ where $\epsilon\sim\mathcal{N}(0,\sigma^2 I)$

Let $P\sim\mathcal{N}\left(0, c\exp\left(\frac{1-j}{k}\right)I\right)$ for $j\in\mathbb{N}$ and $Q\sim\mathcal{N}(w,\sigma^2 I)$

- $KL(Q|P) \leq \frac{1}{2}\left[1 + k\log\left(1+\frac{\|w\|_2^2}{k\sigma^2}\right)\right]$
- $ln\left(\frac{n}{\delta_j}\right) \leq \log\left(\frac{n}{\delta}\right) + 2\log(6n+3k)$

Please refer to Appendix

$$\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma^2 I)}[L_{\mathcal{D}}(w+\epsilon)] \leq \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\sigma^2 I)}[L_S(w+\epsilon)] + \sqrt{\frac{1}{2(n-1)}\left(\frac{1}{2}\left[1 + k\log\left(1+\frac{\|w\|_2^2}{k\sigma^2}\right)\right] + \log\left(\frac{n}{\delta}\right) + 2\log(6n+3k)\right)}$$

Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes.", *ICLR 2014*

How can we constrain the neighborhood?

- Directly constraining ϵ is not intuitive due to high-dimensional nature
- Rather propose to use *norm*

If $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, then $\|\epsilon\|_2^2 \sim \chi^2(k\sigma^2)$

$$\Pr\left(\|\epsilon\|_2^2 - k\sigma^2 \geq 2\sigma^2\sqrt{kt} + 2t\sigma^2\right) \leq \exp(-t)$$

With probability greater than $1 - 1/\sqrt{n}$,

$$\|\epsilon\|_2^2 \leq k\sigma^2 + 2\sigma^2\sqrt{kt} + 2t\sigma^2 = k\sigma^2\left(1 + 2\sqrt{t/k} + 2t/k\right) (:= \rho^2)$$

PAC-bayes Generalization Bound for SAM

$$L_D(w) \leq \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)}[L_{\mathcal{D}}(w+\epsilon)] \quad (by\ assumption)$$

$$\leq \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)}[L_S(w+\epsilon)] + \sqrt{\frac{1}{2(n-1)}\left(\frac{1}{2}\left[1 + k\log\left(1+\frac{\|w\|_2^2}{k\sigma^2}\right)\right] + \log\left(\frac{n}{\delta}\right) + 2\log(6n+3k)\right)}$$

$$\leq \left(1-\frac{1}{\sqrt{n}}\right)\max_{\|\epsilon\|_2 \leq \rho} L_S(w+\epsilon) + \frac{1}{\sqrt{n}} + \sqrt{\frac{1}{4(n-1)}\left(1 + k\log\left(1+\frac{\|w\|_2^2}{k\sigma^2}\right) + 2\log\left(\frac{n}{\delta}\right) + 4\log(6n+3k)\right)}$$

$$\leq \max_{\|\epsilon\|_2 \leq \rho} L_S(w+\epsilon) + \sqrt{\frac{1 + k\log\left(1+\frac{\|w\|_2^2}{\rho^2}\left(1+\sqrt{\frac{2\ln(n)}{k}}\right)^2\right) + 2\log\left(\frac{n}{\delta}\right) + 4\log(6n+3k)}{4(n-1)}}$$

SAM optimizer

Framing into min-max optimization problem

$$\min_{w} \max_{\|\epsilon\|_2 \leq \rho} L_S(w+\epsilon) + \sqrt{\frac{1 + k\log\left(1 + \frac{\|w\|_2^2}{\rho^2}\left(1+\sqrt{\frac{2\ln(n)}{k}}\right)^2\right) + 2\log\left(\frac{n}{\delta}\right) + 4\log(6n+3k)}{4(n-1)}}$$

$$= \min_{w} \max_{\|\epsilon\|_2 \leq \rho} L_S(w+\epsilon) + \mathrm{h}\left(\frac{\|w\|_2^2}{\rho^2}\right) \approx \min_{w} \max_{\|\epsilon\|_p \leq \rho} L_S(w+\epsilon) + \lambda\|w\|_2^2$$

$$= \min_{w} \max_{\|\epsilon\|_p \leq \rho} L_S(w+\epsilon) - L_S(w) + L_S(w) + \lambda\|w\|_2^2$$

Sharpness Training Loss Regularizer

How quickly the training loss can be increased by moving from w to a nearby parameter value

Solving min-max optimization problem

1. $\max_{\|\epsilon\|_p \le \rho} L_S(w+\epsilon)$

$$\epsilon^*(w) = \arg\max_{\|\epsilon\|_p \le \rho} L_S(w+\epsilon) \approx \arg\max_{\|\epsilon\|_p \le \rho} L_S(w) + \epsilon^T \nabla_w L_S(w) = \arg\max_{\|\epsilon\|_p \le \rho} \epsilon^T \nabla_w L_S(w)$$

$$\approx \rho \frac{sign\left(\nabla_w L_S(w)\right)|\nabla_w L_S(w)|^{\frac{q}{p}}}{\left(\|\nabla_w L_S(w)\|_q\right)^{\frac{q}{p}}} \ (:= \hat{\epsilon}(w)) \ \ where \ \ \frac{1}{p} + \frac{1}{q} = 1$$

If $p = 2$, then $\|\hat{\epsilon}(w)\|_2 = \rho \quad and \quad \hat{\epsilon}(w) = \nabla_w L_s(w) / \|\nabla_w L_S(w)\|_2$

Solving min-max optimization problem

2. $\boldsymbol{\min_{w} L_S(w + \hat{\epsilon}(w))}$

$$\nabla_w L_S(w_t + \hat{\epsilon}(w_t)) = \frac{d\left(w + \hat{\epsilon}(w)\right)}{dw} \nabla_w L_S(w)\Big|_{w_t + \hat{\epsilon}(w_t)}$$

$$= \nabla_w L_S(w)|_{w_t + \hat{\epsilon}(w_t)} + \frac{d\hat{\epsilon}(w)}{dw} \nabla_w L_S(w)|_{w_t + \hat{\epsilon}(w_t)}$$

Hessian computation

$$\approx \nabla_w L_S(w)|_{w_t + \hat{\epsilon}(w_t)}$$

By Stochastic Gradient Descent (SGD), $w_{t+1} = w_t - \eta \nabla_w L_S(w)|_{w_t + \hat{\epsilon}(w_t)}$

Experiment

Image classification

		CIFAR-10		CIFAR-100	
Model	Augmentation	SAM	SGD	SAM	SGD
WRN-28-10 (200 epochs)	Basic	**2.7**$_{\pm 0.1}$	3.5$_{\pm 0.1}$	**16.5**$_{\pm 0.2}$	18.8$_{\pm 0.2}$
WRN-28-10 (200 epochs)	Cutout	**2.3**$_{\pm 0.1}$	2.6$_{\pm 0.1}$	**14.9**$_{\pm 0.2}$	16.9$_{\pm 0.1}$
WRN-28-10 (200 epochs)	AA	**2.1**$_{\pm <0.1}$	2.3$_{\pm 0.1}$	**13.6**$_{\pm 0.2}$	15.8$_{\pm 0.2}$
WRN-28-10 (1800 epochs)	Basic	**2.4**$_{\pm 0.1}$	3.5$_{\pm 0.1}$	**16.3**$_{\pm 0.2}$	19.1$_{\pm 0.1}$
WRN-28-10 (1800 epochs)	Cutout	**2.1**$_{\pm 0.1}$	2.7$_{\pm 0.1}$	**14.0**$_{\pm 0.1}$	17.4$_{\pm 0.1}$
WRN-28-10 (1800 epochs)	AA	**1.6**$_{\pm 0.1}$	2.2$_{\pm <0.1}$	**12.8**$_{\pm 0.2}$	16.1$_{\pm 0.2}$
Shake-Shake (26 2x96d)	Basic	**2.3**$_{\pm <0.1}$	2.7$_{\pm 0.1}$	**15.1**$_{\pm 0.1}$	17.0$_{\pm 0.1}$
Shake-Shake (26 2x96d)	Cutout	**2.0**$_{\pm <0.1}$	2.3$_{\pm 0.1}$	**14.2**$_{\pm 0.2}$	15.7$_{\pm 0.2}$
Shake-Shake (26 2x96d)	AA	**1.6**$_{\pm <0.1}$	1.9$_{\pm 0.1}$	**12.8**$_{\pm 0.1}$	14.1$_{\pm 0.2}$
PyramidNet	Basic	**2.7**$_{\pm 0.1}$	4.0$_{\pm 0.1}$	**14.6**$_{\pm 0.4}$	19.7$_{\pm 0.3}$
PyramidNet	Cutout	**1.9**$_{\pm 0.1}$	2.5$_{\pm 0.1}$	**12.6**$_{\pm 0.2}$	16.4$_{\pm 0.1}$
PyramidNet	AA	**1.6**$_{\pm 0.1}$	1.9$_{\pm 0.1}$	**11.6**$_{\pm 0.1}$	14.6$_{\pm 0.1}$
PyramidNet+ShakeDrop	Basic	**2.1**$_{\pm 0.1}$	2.5$_{\pm 0.1}$	**13.3**$_{\pm 0.2}$	14.5$_{\pm 0.1}$
PyramidNet+ShakeDrop	Cutout	**1.6**$_{\pm <0.1}$	1.9$_{\pm 0.1}$	**11.3**$_{\pm 0.1}$	11.8$_{\pm 0.2}$
PyramidNet+ShakeDrop	AA	**1.4**$_{\pm <0.1}$	1.6$_{\pm <0.1}$	**10.3**$_{\pm 0.1}$	10.6$_{\pm 0.1}$

Comparison of SAM and SGD for the test error rate on CIFAR-{10, 100}
(Similar trend is observed on SVHN and Fashion-MNIST)

Image classification

Model	Epoch	SAM Top-1	SAM Top-5	Standard Training (No SAM) Top-1	Standard Training (No SAM) Top-5
ResNet-50	100	**22.5**$_{\pm 0.1}$	6.28$_{\pm 0.08}$	22.9$_{\pm 0.1}$	6.62$_{\pm 0.11}$
	200	**21.4**$_{\pm 0.1}$	5.82$_{\pm 0.03}$	22.3$_{\pm 0.1}$	6.37$_{\pm 0.04}$
	400	**20.9**$_{\pm 0.1}$	5.51$_{\pm 0.03}$	22.3$_{\pm 0.1}$	6.40$_{\pm 0.06}$
ResNet-101	100	**20.2**$_{\pm 0.1}$	5.12$_{\pm 0.03}$	21.2$_{\pm 0.1}$	5.66$_{\pm 0.05}$
	200	**19.4**$_{\pm 0.1}$	4.76$_{\pm 0.03}$	20.9$_{\pm 0.1}$	5.66$_{\pm 0.04}$
	400	**19.0**$_{\pm <0.01}$	4.65$_{\pm 0.05}$	22.3$_{\pm 0.1}$	6.41$_{\pm 0.06}$
ResNet-152	100	**19.2**$_{\pm <0.01}$	4.69$_{\pm 0.04}$	20.4$_{\pm <0.0}$	5.39$_{\pm 0.06}$
	200	**18.5**$_{\pm 0.1}$	4.37$_{\pm 0.03}$	20.3$_{\pm 0.2}$	5.39$_{\pm 0.07}$
	400	**18.4**$_{\pm <0.01}$	4.35$_{\pm 0.04}$	20.9$_{\pm <0.0}$	5.84$_{\pm 0.07}$

Comparison of SAM and SGD for the test error rate on ImageNet

“SAM continues to improve accuracy without overfitting as training proceeds”

Finetuning the pretrained model

Pretrained on ImageNet

Pretrained on ImageNet and unlabeled JFT

Dataset	EffNet-b7 + SAM	EffNet-b7	Prev. SOTA (ImageNet only)	EffNet-L2 + SAM	EffNet-L2	Prev. SOTA
FGVC_Aircraft	$6.80_{\pm 0.06}$	$8.15_{\pm 0.08}$	**5.3** (TBMSL-Net)	$\mathbf{4.82}_{\pm 0.08}$	$5.80_{\pm 0.1}$	5.3 (TBMSL-Net)
Flowers	$\mathbf{0.63}_{\pm 0.02}$	$1.16_{\pm 0.05}$	0.7 (BiT-M)	$\mathbf{0.35}_{\pm 0.01}$	$0.40_{\pm 0.02}$	0.37 (EffNet)
Oxford_IIIT_Pets	$\mathbf{3.97}_{\pm 0.04}$	$4.24_{\pm 0.09}$	4.1 (Gpipe)	$\mathbf{2.90}_{\pm 0.04}$	$3.08_{\pm 0.04}$	4.1 (Gpipe)
Stanford_Cars	$5.18_{\pm 0.02}$	$5.94_{\pm 0.06}$	**5.0** (TBMSL-Net)	$4.04_{\pm 0.03}$	$4.93_{\pm 0.04}$	**3.8** (DAT)
CIFAR-10	$\mathbf{0.88}_{\pm 0.02}$	$0.95_{\pm 0.03}$	1 (Gpipe)	$\mathbf{0.30}_{\pm 0.01}$	$0.34_{\pm 0.02}$	0.63 (BiT-L)
CIFAR-100	$\mathbf{7.44}_{\pm 0.06}$	$7.68_{\pm 0.06}$	7.83 (BiT-M)	$\mathbf{3.92}_{\pm 0.06}$	$4.07_{\pm 0.08}$	6.49 (BiT-L)
Birdsnap	$\mathbf{13.64}_{\pm 0.15}$	$14.30_{\pm 0.18}$	15.7 (EffNet)	$\mathbf{9.93}_{\pm 0.15}$	$10.31_{\pm 0.15}$	14.5 (DAT)
Food101	$7.02_{\pm 0.02}$	$7.17_{\pm 0.03}$	7.0 (Gpipe)	$\mathbf{3.82}_{\pm 0.01}$	$3.97_{\pm 0.03}$	4.7 (DAT)
ImageNet	$15.14_{\pm 0.03}$	15.3	**14.2** (KDforAA)	$\mathbf{11.39}_{\pm 0.02}$	11.8	11.45 (ViT)

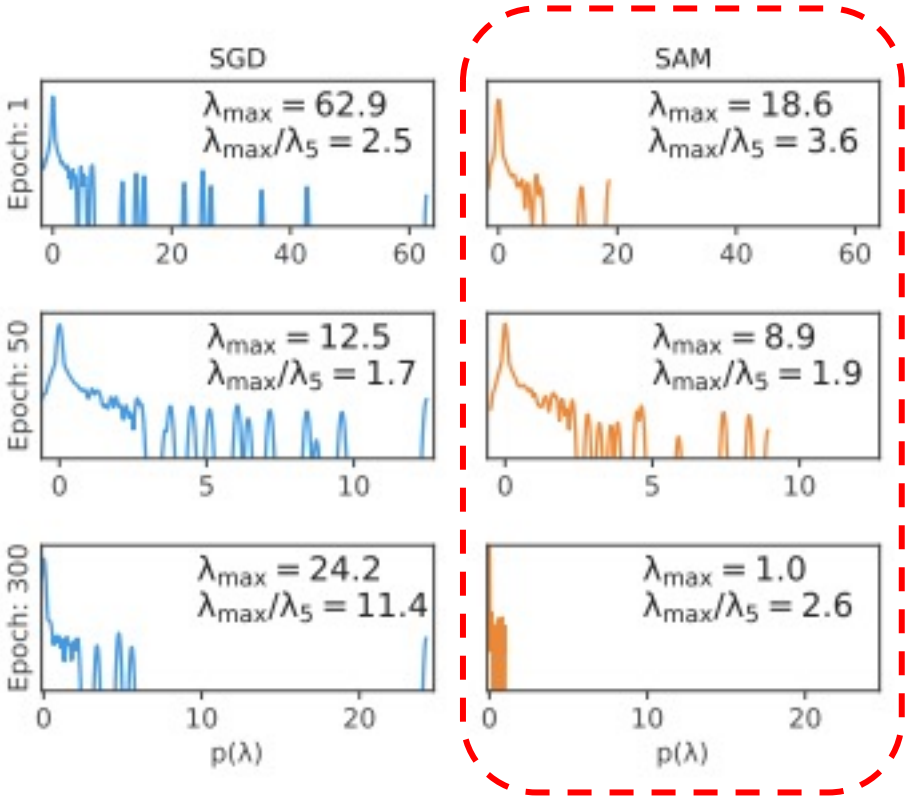
Top-1 error rates for finetuning EffNet-b7 or EffNet-L2 to various target datasets

Robustness to label noise

Method	Noise rate (%)			
	20	40	60	80
Sanchez et al. (2019)	94.0	92.8	90.3	74.1
Zhang & Sabuncu (2018)	89.7	87.6	82.7	67.9
Lee et al. (2019)	87.1	81.8	75.4	-
Chen et al. (2019)	89.7	-	-	52.3
Huang et al. (2019)	92.6	90.3	43.4	-
MentorNet (2017)	92.0	91.2	74.2	60.0
Mixup (2017)	94.0	91.5	86.8	76.9
MentorMix (2019)	**95.6**	**94.2**	91.3	**81.0**
SGD	84.8	68.8	48.2	26.2
Mixup	93.0	90.0	83.8	70.2
Bootstrap + Mixup	93.3	92.0	87.6	72.0
SAM	95.1	93.4	90.5	77.9
Bootstrap + SAM	95.4	**94.2**	**91.8**	79.9

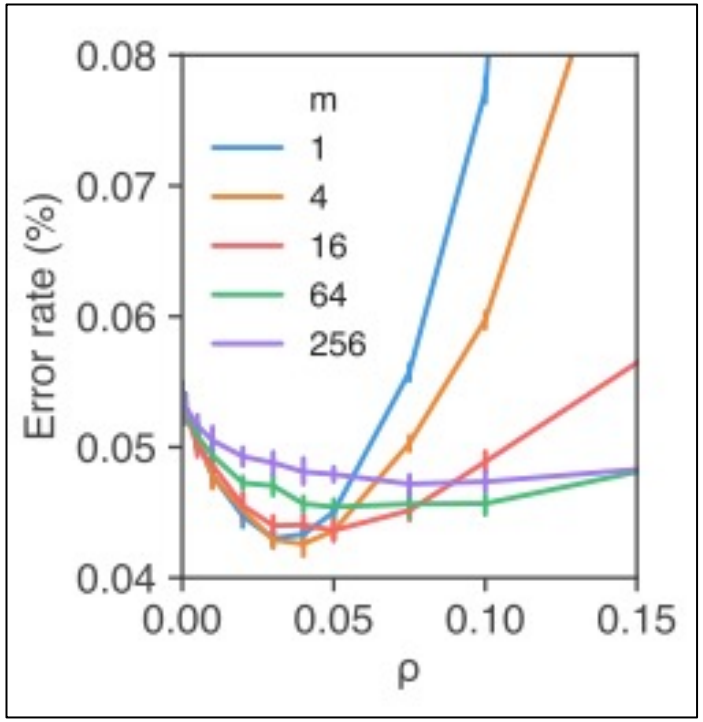
Test accuracy on clean test dataset while trained with noisy labels

Sharpness and Generalization



Evolution of <u>hessian spectrum</u> along epoch

- SGD : to sharp minima
- SAM : to flat minima



Effect of batch size(m) to <u>test error rate</u>

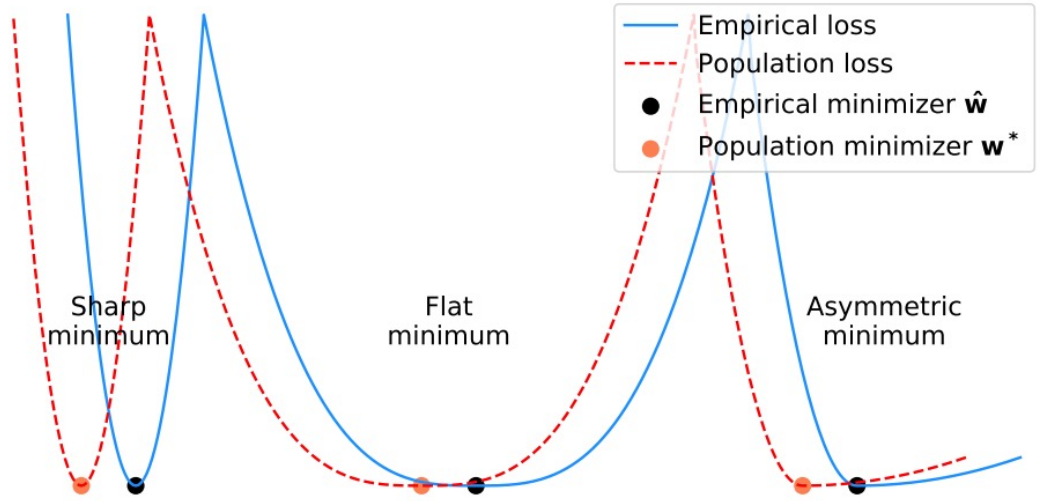
- Small batch training leads to flat minima
- Model gets less robustness to hyperparameter ρ

Critique

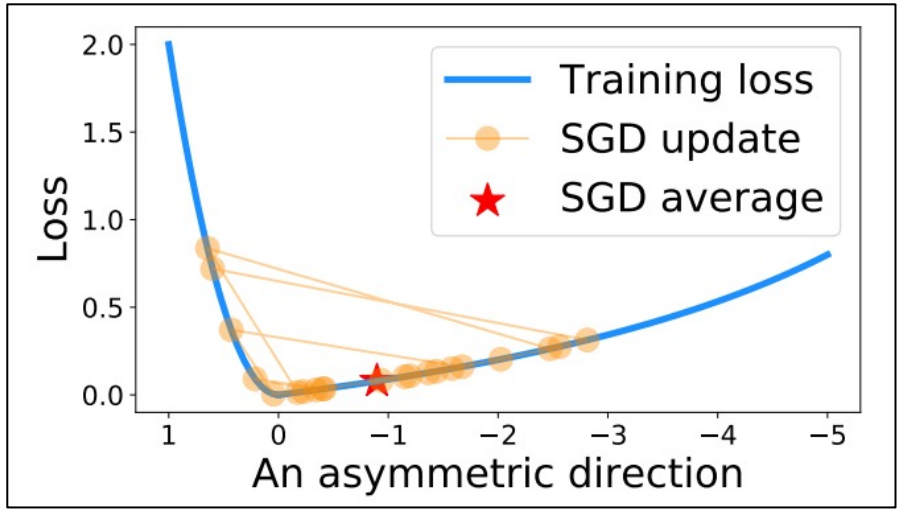
In-between sharpness and flatness?

Known to be in sharp minima

By sampling the loss function in a neighborhood of LB solutions, we observe that it rises steeply only along a small dimensional subspace (e.g. 5% of the whole space); on most other directions, the function is relatively flat



Notion of sharpness and flatness may be oversimplification

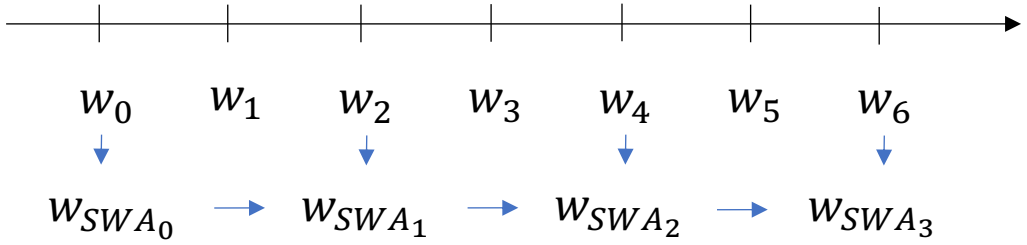


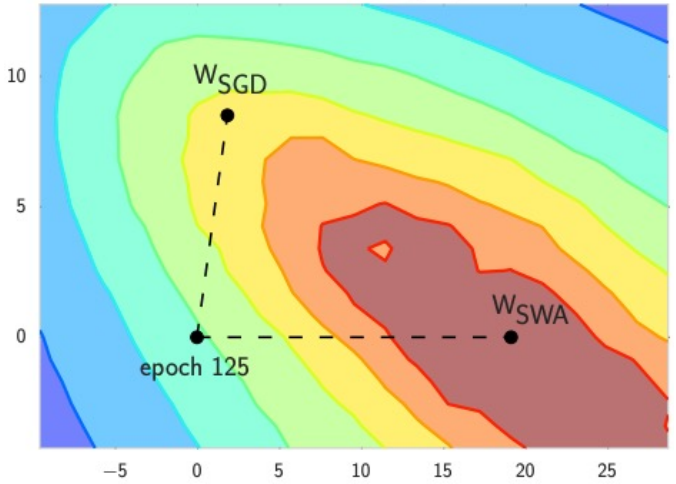
Averaging SGD trajectory leads to flat side

Keskar, Nitish Shirish, et al. "On large-batch training for deep learning: Generalization gap and sharp minima." ICLR 2017.
He, Haowei, Gao Huang, and Yang Yuan. "Asymmetric valleys: Beyond sharp and flat local minima." NeurIPS 2019.

How to free the choice of batch size?

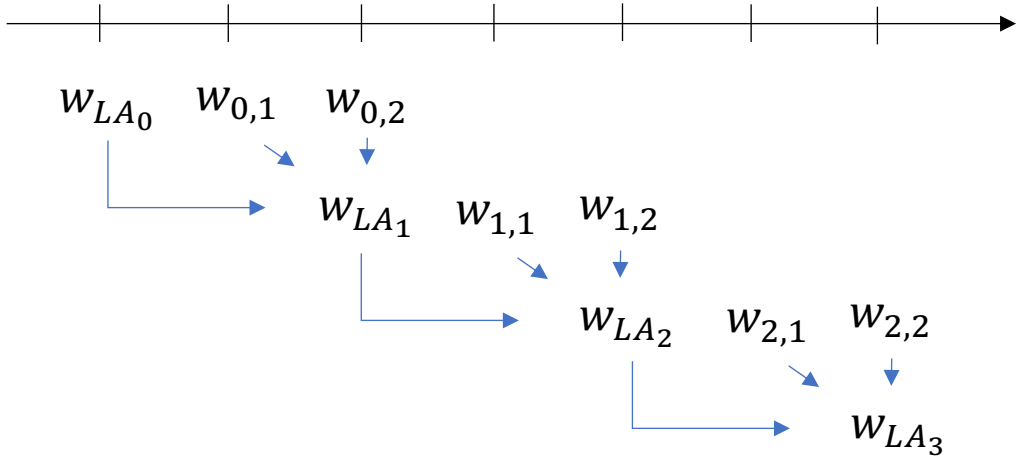
Stochastic Weight Averaging (SWA), *UAI 2018*

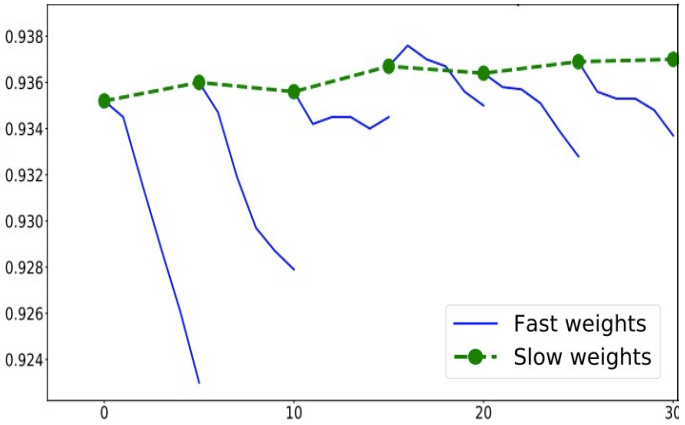




Contour plot of test loss surface

Look ahead(LA), *NeurIPS 2019*





Test accuracy along epoch

Izmailov, Pavel, et al. "Averaging weights leads to wider optima and better generalization.", UAI 2018.
Zhang, Michael R., et al. "Lookahead optimizer: k steps forward, 1 step back.", NeurIPS, 2019.