ICLR'2021

AdamP: Slowing Down the Slowdown for Momentum Optimizers on Scale-invariant Weight

Byeongho Heo*, Sanghyuk Chun*, Seong Joon Oh, Dongyoon Han,
Sangdoo Yun, Gyuwan Kim, Youngjung Uh, Jung-Woo Ha

* Equal contribution

NAVER AI LAB

NAVER CLOVA

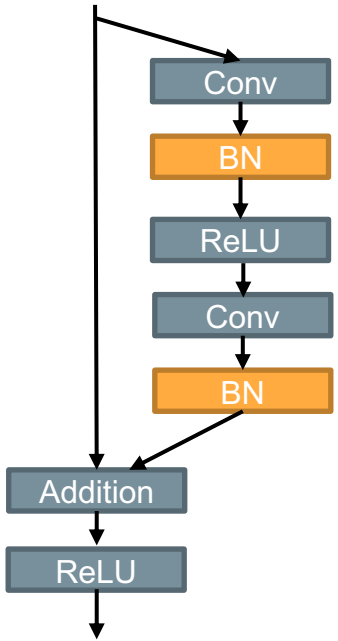
Scale-invariant weight

Normalization layer is widely-used technique

Batch Normalization (BN)

$$\mathrm{BN}(x_i) = \gamma \frac{x_i - \mu}{\sigma} + \beta,$$

$$\mu = \frac{1}{B}\sum_{i=1}^{B} x_i \text{ and } \sigma^2 = \frac{1}{B}\sum_{i=1}^{B}(x_i - \mu)^2$$



ResNet block

Scale-invariant weight

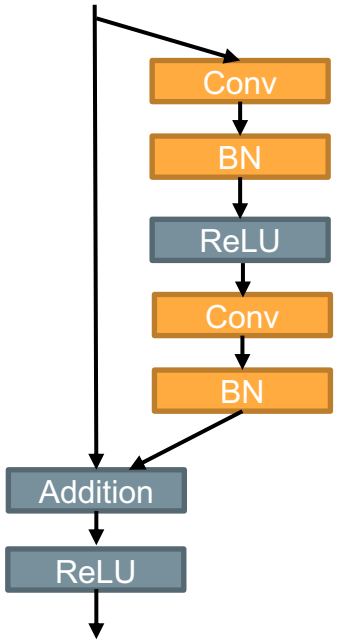
In network with the **normalization layers,**
many parameters are scale-invariant.

$$Norm(\boldsymbol{w}^T\boldsymbol{x}) = Norm\big((c\boldsymbol{w})^T\boldsymbol{x}\big) \quad \text{for any } c > 0$$

$$\widehat{\boldsymbol{w}} = \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$$

$$Norm(\|\boldsymbol{w}\|_2\widehat{\boldsymbol{w}}^T\boldsymbol{x}) = Norm(\widehat{\boldsymbol{w}}^T\boldsymbol{x})$$

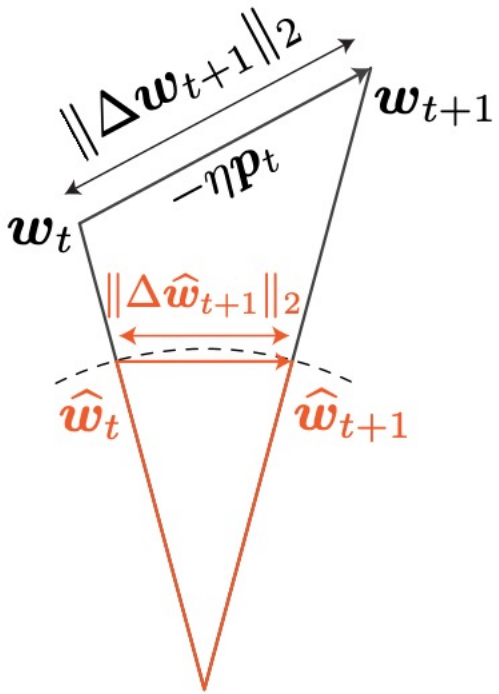
Since the normalization removes scale of inputs,
norm value of some parameters cannot affect output of network.



ResNet block

Scale-invariant weight

Norm value of scale-invariant params controls **effective step-size** of optimizers



Effective step-size

$$\|\Delta \widehat{\boldsymbol{w}}_{t+1}\|_2 := \left\| \frac{\boldsymbol{w}_{t+1}}{\|\boldsymbol{w}_{t+1}\|_2} - \frac{\boldsymbol{w}_t}{\|\boldsymbol{w}_t\|_2} \right\|_2 \approx \frac{\|\Delta \boldsymbol{w}_t\|_2}{\|\boldsymbol{w}_t\|_2}$$

Norm matters: efficient and accurate normalization schemes in deep networks (NIPS 2018)

Auto rate-tuning

Lemma 2.1 (Norm growth of gradient descent) For a scale-invariant parameter $\boldsymbol{w}$ and the vanilla gradient descent with learning rate η

where $\boldsymbol{p} = \nabla_{\boldsymbol{w}} f(\boldsymbol{w})$, gradient descent

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta \boldsymbol{p}_t, \qquad \boldsymbol{p}_t \leftarrow \nabla_{\boldsymbol{w}_t} f(\boldsymbol{w}_t)$$

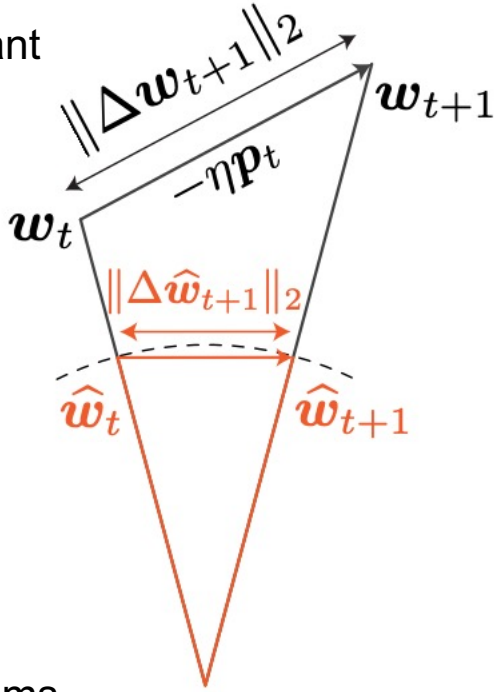
Orthogonality of a scale-invariant parameter

$$0 = \frac{\partial f(c\boldsymbol{w})}{\partial c} = \boldsymbol{w}^T \nabla_{\boldsymbol{w}} f(\boldsymbol{w})$$

Norm values of scale-invariant parameters slowly increase

$$\|\boldsymbol{w}_{t+1}\|_2^2 = \|\boldsymbol{w}_t\|_2^2 + \eta^2 \|\boldsymbol{p}_t\|_2^2$$

It automatically slowdowns effective step-size of scale-invariant params



Theoretical Analysis of Auto Rate-Tuning by Batch Normalization (ICLR 2019)

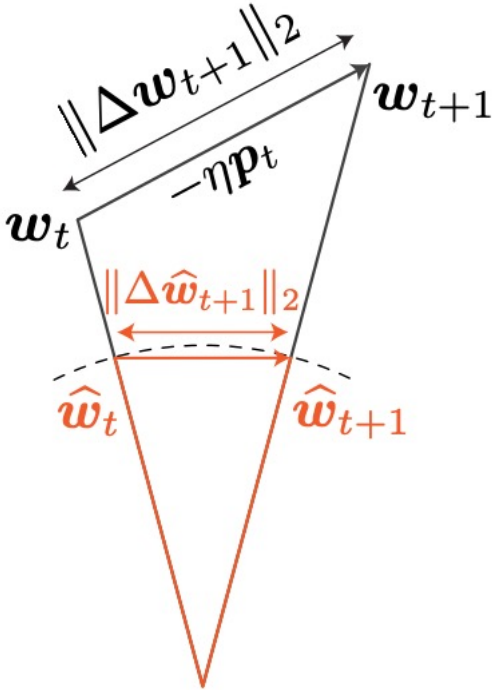
AdamP

A momentum update with parameter $\beta \in (0, 1)$ follows:

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta \boldsymbol{p}_t, \qquad \boldsymbol{p}_t \leftarrow \beta \boldsymbol{p}_{t-1} + \nabla_{\boldsymbol{w}_t} f(\boldsymbol{w}_t)$$

Lemma 2.2 (Norm growth by momentum) For a scale-invariant parameter $\boldsymbol{w}$ updated via momentum optimizer, we have

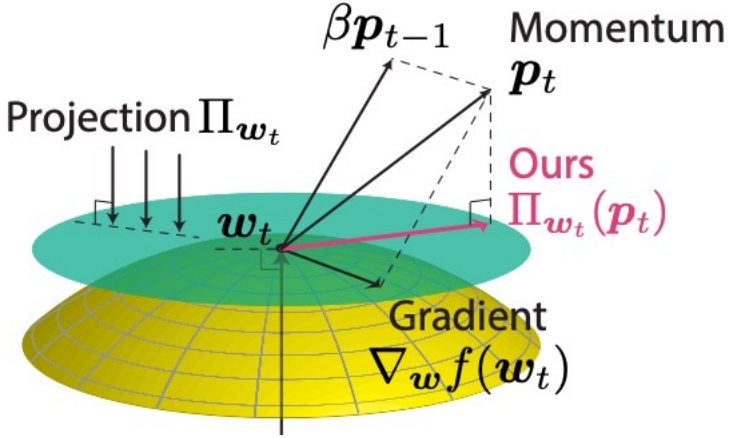
$$\|\boldsymbol{w}_{t+1}\|_2^2 = \|\boldsymbol{w}_t\|_2^2 + \eta^2 \|\boldsymbol{p}_t\|_2^2 + 2\eta^2 \sum_{k=0}^{t-1} \beta^{t-k} \|\boldsymbol{p}_k\|_2^2$$



Norm growth is larger than optimizer without momentum

⇒ Significantly slowdowns training of scale-invariant params

AdamP



Remove rapid norm growth with projection operation

$$\|\boldsymbol{w}_{t+1}\|_2^2 = \|\boldsymbol{w}_t\|_2^2 + \eta^2 \|\boldsymbol{p}_t\|_2^2 - 2\eta \boldsymbol{w}_t \cdot \boldsymbol{p}_t$$

Projection operation

$$\Pi_{\boldsymbol{w}}(\boldsymbol{x}) := \boldsymbol{x} - (\hat{\boldsymbol{w}} \cdot \boldsymbol{x})\hat{\boldsymbol{w}}.$$

$$\boldsymbol{q}_t = \begin{cases} \Pi_{\boldsymbol{w}_t}(\boldsymbol{p}_t) & \text{if } \boldsymbol{w}^\top \nabla_{\boldsymbol{w}} f(\boldsymbol{w}) < \delta \\ \boldsymbol{p}_t & \text{otherwise.} \end{cases}$$

AdamP

SGD version

Algorithm 1: SGDP

Require: Learning rate $\eta > 0$, momentum $\beta > 0$, thresholds $\delta, \varepsilon > 0$.

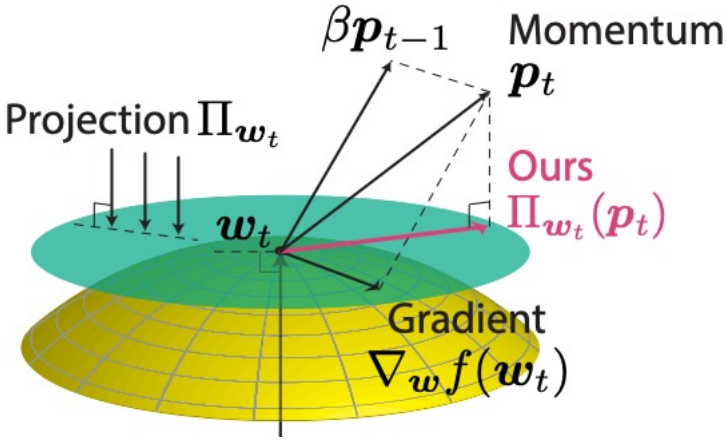
while $\boldsymbol{w}_t$ not converged **do**
  $\boldsymbol{p}_t \leftarrow \beta \boldsymbol{p}_{t-1} + \nabla_{\boldsymbol{w}} f_t(\boldsymbol{w}_t)$
  if $\boldsymbol{w}_t \cdot \nabla_{\boldsymbol{w}} f(\boldsymbol{w}_t) < \delta$ **then**
    $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta \, \Pi_{\boldsymbol{w}_t}(\boldsymbol{p}_t)$
  else
    $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta \, \boldsymbol{p}_t$
  end if
end while

Adam version

Algorithm 2: AdamP

Require: Learning rate $\eta > 0$, momentum $0 < \beta_1, \beta_2 < 1$, thresholds $\delta, \varepsilon > 0$.

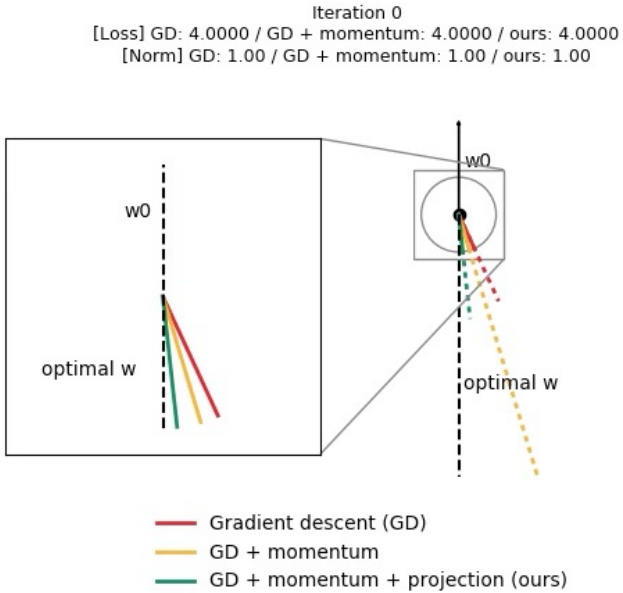
while $\boldsymbol{w}_t$ not converged **do**
  $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1) \nabla_{\boldsymbol{w}} f_t(\boldsymbol{w}_t)$
  $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2) (\nabla_{\boldsymbol{w}} f_t(\boldsymbol{w}_t))^2$
  $\boldsymbol{p}_t \leftarrow \boldsymbol{m}_t / (\sqrt{\boldsymbol{v}_t} + \varepsilon)$
  if $\boldsymbol{w}_t \cdot \nabla_{\boldsymbol{w}} f(\boldsymbol{w}_t) < \delta$ **then**
    $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta \, \Pi_{\boldsymbol{w}_t}(\boldsymbol{p}_t)$
  else
    $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta \, \boldsymbol{p}_t$
  end if
end while



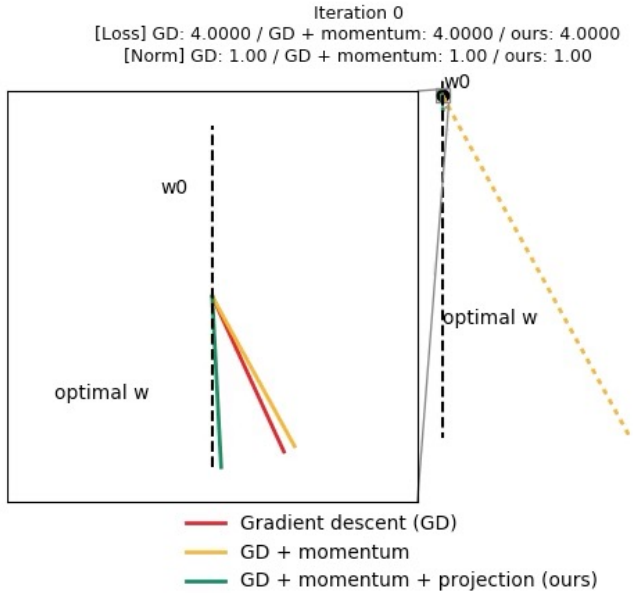
Toy problems

$$\min_{w} \frac{w}{\|w\|_2} \cdot \frac{w^\star}{\|w^\star\|_2}$$

Momentum = 0.9



Momentum = 0.99



- Momentum is beneficial, but norm growth slowdowns step-size
- Our projection enables to use momentum without slowdown problem

Experiments : 6 tasks and 14 datasets

Task	Dataset	#classes	#samples	Note
Image classification	ImageNet-1k	1,000	≈ 1.33M	
Object detection	MS-COCO	80	≈ 123k	
Robustness	CIFAR-10	10	≈ 60k	
	Biased-MNIST	10	≈ 60k	colors are injected to be biased
	9-Class ImageNet	9	≈ 57k	a subset of ImageNet-1k
	9-Class ImageNet-A	9	617	a subset of ImageNet-A
Audio classification	MagnaTagATune	50	≈ 21k	mutl-labeled dataset
	Speech Commands	35	≈ 106k	
	DCASE 2017 task 4	17	≈ 53k	mutl-labeled dataset
Language Modeling	WikiText-103	-	≈ 103M tokens	vocabulary size (267,735)
Image retrieval	CUB	200	≈ 12k	tr classes (100), te classes (100)
	Cars-196	196	≈ 16k	tr classes (98), te classes (98)
	In-Shop Clothes	7,982	≈ 53k	tr classes (3,997), te classes (3985)
	SOP	22,634	≈ 120k	tr classes (11,318), te classes (11,316)

Image recognition

ImageNet classification (Accuracy)

Architecture	# params	SGD	SGDP (ours)	Adam	AdamW	AdamP (ours)
MobileNetV2	3.5M	71.55	**72.09** (+0.54)	69.32	71.21	**72.45** (+1.24)
ResNet18	11.7M	70.47	**70.70** (+0.23)	68.05	70.39	**70.82** (+0.43)
ResNet50	25.6M	76.57	**76.66** (+0.09)	71.87	76.54	**76.92** (+0.38)
ResNet50 + CutMix	25.6M	77.69	**77.77** (+0.08)	76.35	78.04	**78.22** (+0.18)

MS-COCO object detection (mAP)

Model	Initialize	Adam	AdamP (ours)
CenterNet	Scratch	26.57	**27.11** (+0.54)
CenterNet	ImageNet	28.29	**29.05** (+0.76)
SSD	Scratch	27.10	**27.97** (+0.87)
SSD	ImageNet	28.39	**28.67** (+0.28)

Minimax optimizations

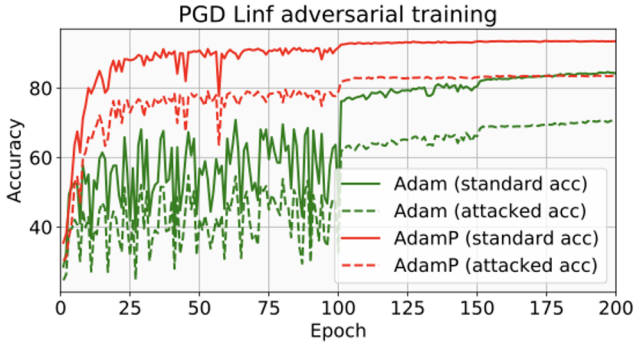
ReBias training (Biased-MNIST)

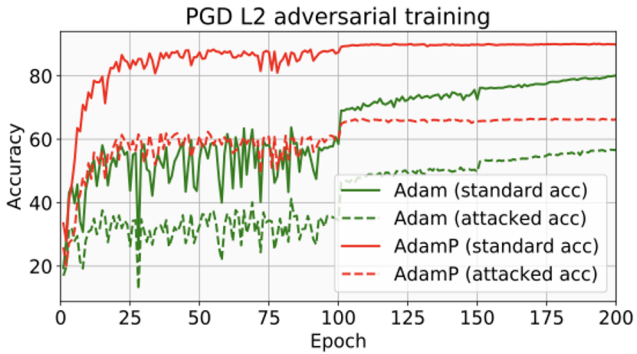
ρ	.999	.997	.995	.990	avg.
Adam	22.9	63.0	74.9	87.0	61.9
AdamP (ours)	**30.5** (+7.5)	**70.9** (+7.9)	**80.9** (+6.0)	**89.6** (+2.6)	**68.0** (+6.0)

ReBias training (9 class ImageNet)

Optimizer	Biased accuracy	UnBiased accuracy	ImageNet-A accuracy
Adam	93.8	92.6	31.2
AdamP (ours)	**95.2** (+1.4)	**94.5** (+1.8)	**32.9** (+1.7)

Adversarial training (CIFAR-10)





More experiments

Audio classification tasks.

Optimizer	Music Tagging		Keyword Spotting	Sound Event Tagging
	ROC-AUC	PR-AUC	Accuracy	F1 score
Adam + SGD (Won et al., 2019b)	91.27	45.67	96.08	54.60
AdamW	91.12	45.61	96.47	55.24
AdamP (ours)	**91.35** (+0.23)	**45.79** (+0.18)	**96.89** (+0.42)	**56.04** (+0.80)

Image retrieval

Optimizer	CUB		Cars-196		InShop		SOP	
	Triplet	PA	Triplet	PA	Triplet	PA	Triplet	PA
AdamW	57.9	69.3	59.8	86.7	62.7	85.2	62.0	76.5
AdamP (ours)	**58.2** (+0.3)	**69.5** (+0.2)	**59.9** (+0.2)	**86.9** (+0.2)	**62.8** (+0.0)	**87.4** (+2.2)	**62.6** (+0.6)	**78.0** (+1.5)

More experiments

Language model (WikiText-103)

Model	AdamW	AdamP (ours)
Transformer-XL	23.33	**23.26** (-0.07)
Transformer-XL + WN	23.90	**22.73** (-1.17)

Large-batch training (ImageNet)

Batch-size	AdamW	AdamP (ours)
4k	72.41	**73.75** (+1.34)
8k	69.74	**71.36** (+1.62)
16k	63.79	**66.89** (+3.1)

EfficientNet & ReXNet (ImageNet)

Network	Image size	# of params	Paper	Reproduce	AdamP
EfficientNet-B0	224	5.3M	77.1	77.7	**78.1** (+0.4)
EfficientNet-B1	240	7.8M	79.1	78.7	**79.9** (+1.2)
EfficientNet-B2	260	9.1M	80.1	80.4	**80.5** (+0.1)
EfficientNet-B3	300	12.2M	81.6	81.5	**81.9** (+0.4)
ReXNet-×1.0	224	4.8M	77.9	77.9	**78.1** (+0.2)
ReXNet-×1.3	224	6.4M	79.5	79.5	**79.7** (+0.2)

ClovaAI/AdamP: Open source for SGDP / AdamP.

```
pip install adamp
```

```
from adamp import AdamP, SGDP

# Usage is exactly same as torch.optim library!
optimizer = AdamP(lr=0.001, betas=(0.9, 0.999), weight_decay=1e-2)
optimizer = SGDP(lr=0.1, weight_decay=1e-5, momentum=0.9, nesterov=True)
```

Github

- https://github.com/clovaai/adamp
- Easy to use (just replace your Adam to AdamP)
- Works well with the same parameters as default optimizer

- Question & discussion

Poster session 10: May 6, 1 am (PDT)