Positive-Negative Momentum: Manipulating Stochastic Gradient Noise to Improve Generalization

Zeke Xie[1, 2], Li Yuan[3], Zhanxing Zhu[4], and Masashi Sugiyama[2, 1]

[1]The University of Tokyo
[2]RIKEN Center for AIP
[3]National University of Singapore
[4]Beijing Institute of Big Data Research

ICML2021, July, 2021

東京大学 THE UNIVERSITY OF TOKYO AIP Center for Advanced Intelligence Project

Positive-Negative Momentum (PNM)

- Motivation:
  As Stochastic Gradient Noise (SGN) is important, we want to manipulate SGN to improve deep learning.

- Constraint:
  We prefer not to change the learning rate or batch size due to multiple limitations.

- Our work:
  PNM, an alternative to conventional Momentum, can manipulate SGN without changing the learning rate or batch size.

Manipulating SGN by changing $\frac{\eta}{B}$.

(Mandt et al., 2017): The magnitude of SGN is proportional to $\frac{\eta}{B}$, where η is the learning rate and B is the batch size.

- (He et al., 2019) argued that increasing $\frac{\eta}{B}$ which corresponds larger SGN may improve generalization.
- (Xie et al., 2021) argued that $\frac{\eta}{B}$ exponentially matters to learning flat minima.

Stephan, M., Hoffman, M. D., and Blei, D. M. (2017). Stochastic Gradient Descent as Approximate Bayesian Inference. JMLR.
He, F., Liu, T., and Tao, D. (2019). Control batch size and learning rate to generalize well: Theoretical and empirical evidence. In NeurIPS.
Zeke Xie, Issei Sato, and Masashi Sugiyama. (2021). A Diffusion Theory For Deep Learning Dynamics: Stochastic Gradient Descent Exponentially Favors Flat Minima. In ICLR.

Increasing $\frac{\eta}{B}$ has at least three limitations.

Three Limitations:

1. Training with a too small batch size is computationally expensive per epoch and often requires more epochs for convergence.
2. Increasing the learning rate only works in a narrow range, since too large initial learning rates may lead to optimization divergence or bad convergence.
3. Decaying the ratio $\frac{\eta}{B}$ during training (via learning rate decay) is almost necessary for the convergence/training of deep networks.

Basic idea: Positive-Negative Weighted Average

Suppose that $g^{(a)}$ and $g^{(b)}$ are two independent unbiased noisy gradients of $\nabla f(\theta)$, where $f(\theta)$ is the loss. Then their weighted average is

$$\begin{aligned}\bar{g} =& (1+\beta_0)g^{(a)} - \beta_0 g^{(b)} \\ =& \nabla f(\theta) + \bar{\xi},\end{aligned}$$

where the gradient noise $\bar{\xi} = (1+\beta_0)\xi^{(a)} - \beta_0\xi^{(b)}$. If $\beta_0 > 0$ and $\sigma^2 = \mathrm{Var}(\xi^{(a)}) = \mathrm{Var}(\xi^{(a)})$, we have the same gradient expectation $\mathbb{E}[\bar{\xi}] = 0$ but larger gradient noise

$$\mathrm{Var}(\bar{\xi}) = [(1+\beta_0)^2 + \beta_0^2]\sigma^2.$$

β_0 controls the noise magnitude!

From Momentum to Positive-Negative Momentum

Momentum: uses the (positive only) weighted average of past gradients to update parameters as

$$m_t = \sum_{k=0}^{t} \beta_3 \beta_1^{t-k} g_k.$$

Then we approximately have $\mathbb{E}[m] \approx \frac{\beta_3}{1-\beta_1} \nabla f(\theta)$. The SGN in momentum is given by

$$\xi_t^{\mathrm{m}} = \sum_{k=0}^{t} \beta_3 \beta_1^{t-k} \xi_k.$$

Algorithm 1: Momentum

$m_t = \beta_1 m_{t-1} + \beta_3 g_t$;
$\theta_{t+1} = \theta_t - \eta m_t$;

PNM can manipulate SGN

PNM: uses the (positive-negative) weighted average of past gradients to update parameters.
PNM maintains two independent momentum terms as

$$\begin{cases} m_t^{(\mathrm{odd})} = \sum_{k=1,3,\ldots,t} \beta_3 \beta_1^{t-k} g_k, \\ m_t^{(\mathrm{even})} = \sum_{k=0,2,\ldots,t-1} \beta_3 \beta_1^{t-k} g_k, \\ m_t = (1+\beta_0) m_t^{(\mathrm{odd})} - \beta_0 m_t^{(\mathrm{even})}, \end{cases}$$

by using two alternate sequences of past gradients, respectively. Then we approximately have $\mathbb{E}[m] \approx \frac{\beta_3}{1-\beta_1} \nabla f(\theta)$. The SGN in PNM is given by

$$\mathrm{Var}(\xi^{\mathrm{pnm}}) \approx [(1+\beta_0)^2 + \beta_0^2] \xi^{\mathrm{m}}.$$

β_0 controls the noise magnitude!

PNM: Algorithm and Theory

Algorithm 2: (Stochastic) PNM

$$m_t = \beta_1^2 m_{t-2} + (1-\beta_1^2) g_t;$$
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{(1+\beta_0)^2+\beta_0^2}}[(1+\beta_0)m_t - \beta_0 m_{t-1}];$$

- Theorem 1: PNM has the convergence guarantee similar to Momentum.
- Theorem 4: Stochastic PNM may have a tighter generalization bound than SGD within the PAC-Bayesian framework.

Experiment: PNM and AdaPNM

Table: PNM versus conventional Momentum. We report the mean and the standard deviations (as the subscripts) of the optimal test errors computed over three runs of each experiment.

Dataset	Model	PNM	AdaPNM	SGD	Adam	AMSGrad	AdamW	AdaBound	Padam	Yogi	RAdam
CIFAR-10	ResNet18	$\mathbf{4.48}_{0.09}$	$4.94_{0.05}$	$5.01_{0.03}$	$6.53_{0.03}$	$6.16_{0.18}$	$5.08_{0.07}$	$5.65_{0.08}$	$5.12_{0.04}$	$5.87_{0.12}$	$6.01_{0.10}$
	VGG16	$6.26_{0.05}$	$\mathbf{5.99}_{0.11}$	$6.42_{0.02}$	$7.31_{0.25}$	$7.14_{0.14}$	$6.48_{0.13}$	$6.76_{0.12}$	$6.15_{0.06}$	$6.90_{0.22}$	$6.56_{0.04}$
CIFAR-100	ResNet34	$20.59_{0.29}$	$\mathbf{20.41}_{0.18}$	$21.52_{0.37}$	$27.16_{0.55}$	$25.53_{0.19}$	$22.99_{0.40}$	$22.87_{0.13}$	$22.72_{0.10}$	$23.57_{0.12}$	$24.41_{0.40}$
	DenseNet121	$\mathbf{19.76}_{0.28}$	$20.68_{0.11}$	$19.81_{0.33}$	$25.11_{0.15}$	$24.43_{0.09}$	$21.55_{0.14}$	$22.69_{0.15}$	$21.10_{0.23}$	$22.15_{0.36}$	$22.27_{0.22}$
	GoogLeNet	$20.38_{0.31}$	$\mathbf{20.26}_{0.21}$	$21.21_{0.29}$	$26.12_{0.33}$	$25.53_{0.17}$	$21.29_{0.17}$	$23.18_{0.31}$	$21.82_{0.17}$	$24.24_{0.16}$	$22.23_{0.15}$

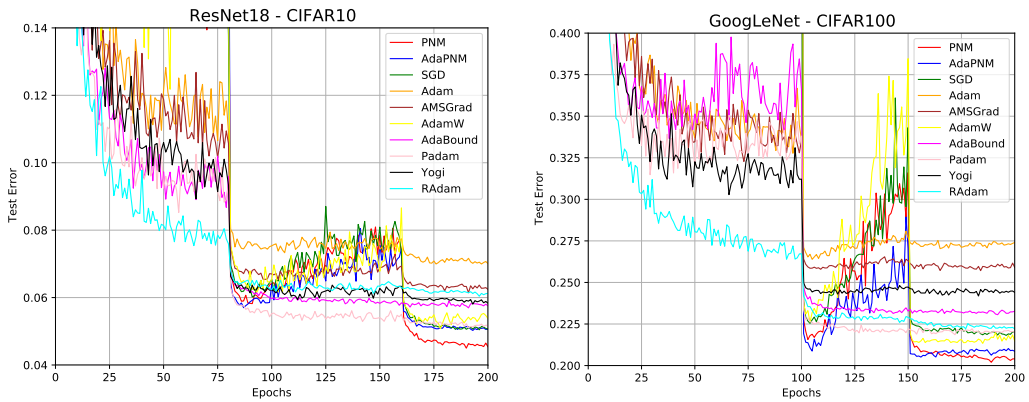

Figure: The learning curves of popular models on CIFAR-10 and CIFAR-100, respectively. It demonstrates that PNM and AdaPNM yield significantly better test results.

Experiment: PNM prevents overfitting noisy labels

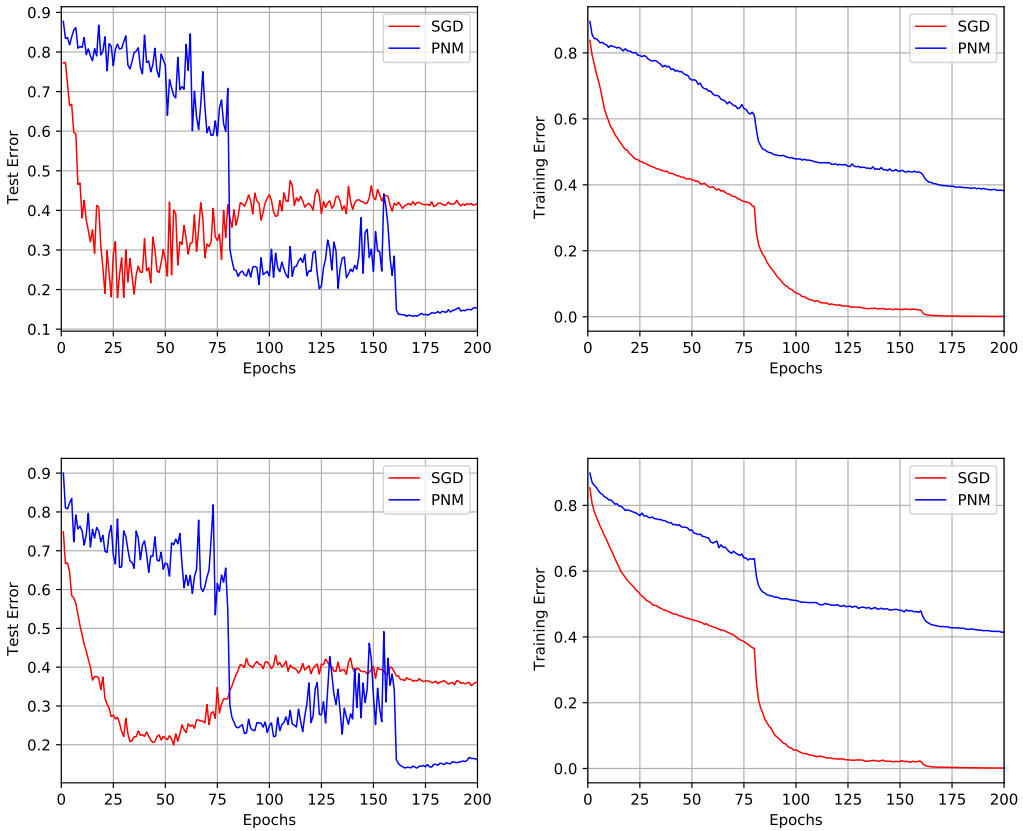

Figure: ResNet34 on CIFAR-10 with 40% asymmetric label noise (Top Row) and 40% symmetric label noise (Bottom Row). Left: Test Curve. Right: Training Curve. PNM with a large β_0 may effectively relieve memorizing noisy labels and almost only learn clean labels, while SGD almost memorizes all noisy labels and has a nearly zero training error.

Experiment: Robustness to the new hyperparameter β_0.

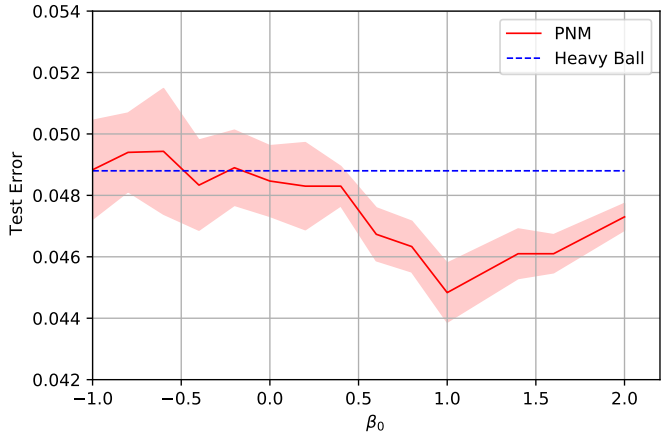

Figure: We train ResNet18 on CIFAR-10 under various β_0 choices. It demonstrates that PNM may achieve significantly better generalization with a wide range of $\beta_0 > 0$, which corresponds to a positive-negative momentum pair for enhancing SGN as we expect. With any $\beta_0 \in [-1, 0]$, the test performance does not sensitively depend on β_0, because this case cannot enhance SGN.

Summary

Three contributions:

1. PNM can manipulate SGN without changing the learning rate or batch size.
2. PNM can replace conventional Momentum in popular optimizers at low costs.
3. PNM can have a tighter generalization bound than SGD.

Future directions:

- Rethinking the importance and the popularity of Momentum.
- Designing optimization dynamics by manipulating SGN.