Closing the Generalization Gap of Adaptive Gradient Methods in Training Deep Neural Networks

Quanquan Gu

Nov 2, 2018

Department of Computer Science
UCLA

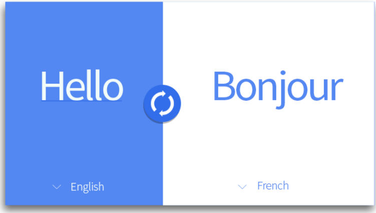
This is a joint work with Jinghui Chen and Dongruo Zhou

Outline

1. Background
2. Our Proposed Method
3. Theoretical Results
4. Experiments and Applications
5. Summary and Future Work

Deep Learning is Everywhere

Speech Recognition



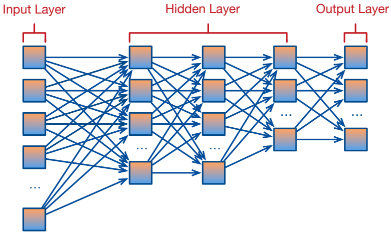
Machine Translation

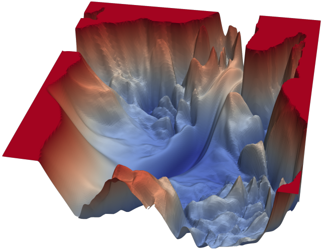
Image Recognition

Recommendation Systems

Deep Learning Problems Are HARD

- Various architectures, layer types, activation functions ...
- Highly nonconvex optimization loss surface





Right figure from https://www.cs.umd.edu/ tomg/projects/landscapes/

Optimization in Deep Learning

- The optimization problems in deep learning can be formalized as follows:

  $$\min f(\boldsymbol{\theta}) := \mathbb{E}_{\xi} f(\boldsymbol{\theta}; \xi), \tag{1}$$

  where $f(\theta; \xi)$ is nonconvex
- For example, $f(\boldsymbol{\theta}) = 1/n \sum_{i=1}^{n} f_i(\boldsymbol{\theta})$

Stochastic Gradient Descent

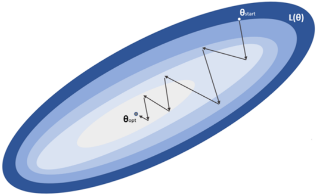
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \mathbf{g}_t \quad \text{where} \quad \mathbf{g}_t = \nabla f_t(\boldsymbol{\theta}_t)$$

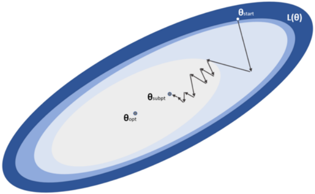
Advantages:

- Simple and efficient
- Low computation complexity

Disadvantages:

- Slow to converge
- Easily get "trapped" in suboptimal solutions

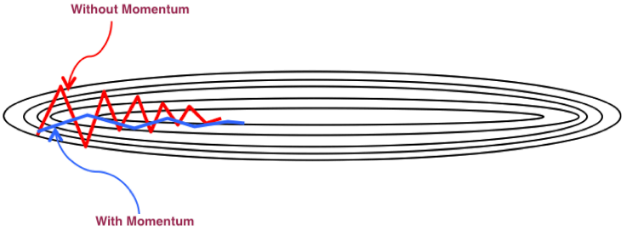




SGD + Momentum

Introduce a new momentum term, which accumulates the previous stochastic gradients, just like in Physics objects can accumulate momentum when moving.

$$
\begin{aligned}
\mathbf{m}_t &= \beta \mathbf{m}_{t-1} + \mathbf{g}_t \\
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \alpha_t \mathbf{m}_t
\end{aligned}
$$



AdaGrad (Duchi et al., 2012)[1]

AdaGrad modifies the universal learning rate into adaptive learning rate for each dimension, based on past gradients computed for each dimension[2] .

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \frac{\mathbf{g}_t}{\sqrt{\mathbf{v}_t}} \quad \text{where} \quad \mathbf{v}_t = \frac{1}{t}\sum_{j=1}^{t} \mathbf{g}_j^2$$

- First to use adaptive learning rate for each dimension
- Proved to converge faster than SGD in sparse gradient cases
- Also works for non-sparse cases empirically

[1] Duchi, John, Elad Hazan, and Yoram Singer. ”Adaptive subgradient methods for online learning and stochastic optimization.” Journal of Machine Learning Research 12.Jul (2011): 2121-2159.

[2] With slightly abuse of notation, the division between two vectors means element-wise division, and the square on vector means element-wise square.

RMSprop (Hinton et al., 2012)[1]

RMSprop uses the exponential moving average of the gradient to adjust the learning rate:

$$\begin{aligned} \mathbf{v}_t &= \beta \mathbf{v}_{t-1} + (1-\beta)\mathbf{g}_t^2 \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \alpha_t \frac{\mathbf{g}_t}{\sqrt{\mathbf{v}_t}} \end{aligned}$$

- Works well empirically

[1] Hinton, Geoffrey, Nitish Srivastava, and Kevin Swersky. "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent." Cited on (2012): 14.

Adam (Kingma & Ba, 2015)[1]

Adam combines RMSprop (Hinton et al., 2012) with momentum acceleration

$$\begin{aligned}
\mathbf{m}_t &= \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1)\mathbf{g}_t \\
\mathbf{v}_t &= \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2)\mathbf{g}_t^2 \\
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \alpha_t \frac{\mathbf{m}_t}{\sqrt{\mathbf{v}_t}}
\end{aligned}$$

- Easy to tune, default parameter setting works for most problems
- Empirically converge faster than most other adaptive methods on training dataset

[1] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

AmsGrad (Reddi et al., 2018)[1]

AmsGrad adds an extra max() step upon Adam to keep a “long term memory” (keep the largest value in history).

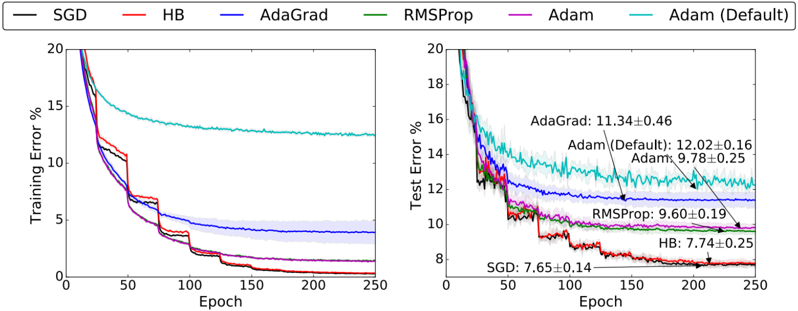
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \frac{\mathbf{m}_t}{\sqrt{\widehat{\mathbf{v}}_t}} \quad \text{where} \quad \begin{aligned} \mathbf{m}_t &= \beta \mathbf{m}_{t-1} + (1-\beta)\mathbf{g}_t \\ \mathbf{v}_t &= \beta \mathbf{v}_{t-1} + (1-\beta)\mathbf{g}_t^2 \\ \widehat{\mathbf{v}}_t &= \max(\widehat{\mathbf{v}}_{t-1}, \mathbf{v}_t) \end{aligned}$$

- Fixing the possible non-convergence issues found in Adam
- State-of-the-art in adaptive gradient method
- Empirically even faster than Adam

[1] Reddi, Sashank J., Satyen Kale, and Sanjiv Kumar. ”On the convergence of adam and beyond.” (2018).

Generalization Gap of Adam (Wilson et al.,2017)

Adam generalizes worse for largely over-parameterized problems, e.g., modern CNN architectures



(a) CIFAR-10 (Train) **(b)** CIFAR-10 (Test)

Figure from Wilson, Ashia C., et al. "The marginal value of adaptive gradient methods in machine learning." , Advances in Neural Information Processing Systems. 2017.

In Practice...

- Recent advances in designing neural network architectures (VGGNet, ResNet, WideResNet, DenseNet, ...) are all reporting their performances by training their models with SGD with momentum because of the generalization gap of Adam
- Adaptivity, on the other hand, gives faster convergence at early stages in training

A Question to Ask...

- Can we take the best from both Adam and SGD with momentum, i.e., design an algorithm that not only enjoys the fast convergence rate as Adam, but also generalizes as well as SGD with momentum?

Outline

1. Background
2. Our Proposed Method
3. Theoretical Results
4. Experiments and Applications
5. Summary and Future Work

What's Wrong with Adam?

- "Small Learning Rate Dilemma": Adam usually takes a much smaller learning rate than SGD. After several rounds of decaying, the learning rate of the Adam is too small to make any significant progress in the training process
- Reasons for the small learning rate in Adam? Adaptive terms based on previous gradients could be highly skewed, and we have to use smaller learning rate to keep some dimensions from exploding

Control the Adaptivity in Adam

- Solution: control the adaptivity in Adam
- Less adaptivity means larger learning rate is allowed and the "Small Learning Rate Dilemma" can be avoided

Partially Adaptive Momentum Estimation (Padam)

- Padam adds the partial adaptive parameter p to control the adaptivity in the denominator

$$\begin{aligned}\mathbf{m}_t &= \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t \\ \mathbf{v}_t &= \beta_2 \mathbf{v}_{t-1} + (1-\beta_2)\mathbf{g}_t^2 \\ \widehat{\mathbf{v}}_t &= \max(\widehat{\mathbf{v}}_{t-1}, \mathbf{v}_t) \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \alpha_t \frac{\mathbf{m}_t}{\widehat{\mathbf{v}}_t^{p}}\end{aligned}$$

$p \in (0, 1/2]$ is a tuning parameter

What p Should We Choose?

- For Padam, p can be chosen from $(0, 1/2]$
- Adam corresponds to Padam with $p = 1/2$, which is largest possible p to guarantee convergence
- Padam with a proper adaptive parameter p, will enable us to adopt a larger learning rate to avoid the "small learning rate dilemma"

Outline

1. Background
2. Our Proposed Method
3. Theoretical Results
4. Experiments and Applications
5. Summary and Future Work

Theory of Convergence for Convex Optimization [1]

- Regret: characterize the sum of all previous loss function values $f_t(\boldsymbol{\theta}_t)$ relative to the performance of the best fixed parameter $\boldsymbol{\theta}^*$ from a feasible set.

$$R_T = \sum_{t=1}^{T} \big(f_t(\boldsymbol{\theta}_t) - f_t(\boldsymbol{\theta}^*) \big), \text{ where } \boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathcal{X}} \sum_{t=1}^{T} f_t(\boldsymbol{\theta})$$

- Goal: to show that $R(T) = o(T)$, so that when $T \to \infty$, $R(T)/T$ converge to 0.
- Assumption: f is convex, i.e. $f_t(\mathbf{y}) \geq f_t(\mathbf{x}) + \nabla f_t(\mathbf{x})^\top (\mathbf{y} - \mathbf{x})$

[1] Chen, Jinghui, and Quanquan Gu. "Closing the Generalization Gap of Adaptive Gradient Methods in Training Deep Neural Networks." arXiv preprint arXiv:1806.06763 (2018).

Theory of Convergence for Convex Optimization

Theorem 1 (Padam, Convex)

Under convex assumption, suppose $\mathcal{X}$ has bounded diameter, and f_t has bounded gradient, we have

$$R_T \leq C_1 \sum_{i=1}^{d} \sqrt{T} \cdot \widehat{v}_{T,i}^{p} + C_2 \sum_{i=1}^{d} \|g_{1:T,i}\|_2 + C_3 d.$$

- Similar to Adam and Amsgrad, the regret of Padam can be considerably better than online gradient descent (which is known to have a regret bound of $O(\sqrt{dT})$) when $\sum_{i=1}^{d} \|g_{1:T,i}\|_2 \ll \sqrt{dT}$ and $\sum_{i=1}^{d} \widehat{v}_{T,i}^{p} \ll \sqrt{d}$.

Theory of Convergence for Convex Optimization

Corollary 2 (Padam, Convex)

The regret bound for Padam satisfies

$$R_T = \widetilde{O}(\sqrt{T}).$$

- It implies that Padam attains $R_T = o(T)$ for all situations (no matter whether the data features are sparse or not). This suggests that Padam indeed converges to the optimal solution when the loss functions are convex, as shown by the fact that $\lim_{T\to\infty} R_T/T \to 0$.

Theory of Convergence for Nonconvex Optimization[1]

- What about the convergence result for nonconvex functions?
- We first give the convergence result for Adam-type algorithms under nonconvex setting.

We need the following assumptions:

- f is L-smooth, where for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \le L\|\mathbf{x} - \mathbf{y}\|_2$.
- f has G_∞-bounded stochastic gradient, where for each $\mathbf{x}$ and ξ, $\|\nabla f(\mathbf{x}; \xi)\|_\infty \le G_\infty$.

[1] Zhou, Dongruo, et al. ”On the convergence of adaptive gradient methods for nonconvex optimization.” arXiv preprint arXiv:1808.05671 (2018).

Theory of Convergence for Nonconvex Optimization

Theorem 3 (Adam-type, Nonconvex)

Suppose that f is L-smooth and has G_∞-bounded stochastic gradient, $p \in [0, 1/2]$, $\beta_1 < \beta_2^{2p}$ and $\alpha_t = \alpha$ for $t = 1, \ldots, T$. Then for any $q \in [\max\{0, 4p-1\}, 1]$, the output $\boldsymbol{\theta}_{\text{out}}$ of Padam satisfies that

$$\mathbb{E}\Big[\big\|\nabla f(\boldsymbol{\theta}_{\text{out}})\big\|_2^2\Big] \leq \frac{M_1}{T\alpha} + \frac{M_2 d}{T} + \frac{M_3 \alpha d^q}{T^{(1-q)/2}} \mathbb{E}\bigg(\sum_{i=1}^{d} \|\mathbf{g}_{1:T,i}\|_2\bigg)^{1-q},$$

where M_1, M_2 and M_3 are constants independent of T, α and d.

- Similar to Adam and AmsGrad in convex cases, our analysis shows that the convergence results should be better when $\sum_{i=1}^{d} \|g_{1:T,i}\|_2 \ll \sqrt{dT}$.

Theory of Convergence for Nonconvex Optimization

Corollary 4 (Padam, Nonconvex)

Suppose that $p \in [0, 1/4]$. With the fact that $\sum_{i=1}^{d} \|\mathbf{g}_{1:T,i}\|_2 \ll \sqrt{dT}$ (Duchi et al., 2011) (Reddi et al., 2018) and specific choice of α, we have

$$\mathbb{E}\Big[\big\|\nabla f(\theta_{\text{out}})\big\|_2^2\Big] = O\bigg(\frac{d^{1/4}}{\sqrt{T}} + \frac{d}{T}\bigg).$$

- It implies that Padam attains $\mathbb{E}\|\nabla f(\boldsymbol{\theta}_{\text{out}})\|_2^2] = O(1/\sqrt{T})$, which is also attained by SGD and AmsGrad.

Proof Sketch

$$
\begin{aligned}
\mathbf{m}_t &= \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t \\
\mathbf{v}_t &= \beta_1 \mathbf{v}_{t-1} + (1-\beta_1)\mathbf{g}_t^2 \\
\widehat{\mathbf{v}}_t &= \max(\widehat{\mathbf{v}}_{t-1}, \mathbf{v}_t) \\
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \alpha_t \frac{\mathbf{m}_t}{\widehat{\mathbf{v}}_t^{p}}
\end{aligned}
$$

- Hard to analyze because of the momentum term $\mathbf{m}_t$ and adjusting learning rate $\alpha_t \widehat{\mathbf{v}}^{-p}$!
- How do we deal with them?

Deal with Momentum $\mathbf{m}_t$ via Auxiliary Sequence

- To deal with momentum: introduce an auxiliary sequence

$$\mathbf{z}_t = \boldsymbol{\theta}_t + \frac{\beta_1}{1-\beta_1}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}).$$

- We have

$$\mathbf{z}_{t+1} - \mathbf{z}_t = \frac{\beta_1}{1-\beta_1}\big(\alpha_{t-1}\widehat{\mathbf{V}}_{t-1}^{-p} - \alpha_t\widehat{\mathbf{V}}_t^{-p}\big)\mathbf{m}_{t-1} - \alpha_t\widehat{\mathbf{V}}_t^{-p}\mathbf{g}_t,$$

which involves both $\mathbf{m}_t$ and $\mathbf{g}_t$ into analysis!

Adjusting Learning Rate $\alpha_t \widehat{\mathbf{V}}^{-p}$

- Key property by the definition of α_t and $\widehat{\mathbf{V}}^{-p}$:

  $$\alpha_{t-1}\widehat{\mathbf{V}}_{t-1}^{-p} \succeq \alpha_t \widehat{\mathbf{V}}_t^{-p},$$

  which allows us to treat $\alpha_t \widehat{\mathbf{V}}_t^{-p}$ as a single decreasing scalar learning rate.
- Easy to analyze!

Final Proof Roadmap

We have

$$\begin{aligned} f(\mathbf{z}_{t+1}) &\leq f(\mathbf{z}_t) + \nabla f(\mathbf{z}_t)^\top(\mathbf{z}_{t+1} - \mathbf{z}_t) + \frac{L}{2}\|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2^2 \\ &= f(\mathbf{z}_t) + \underbrace{\nabla f(\mathbf{x}_t)^\top(\mathbf{z}_{t+1} - \mathbf{z}_t)}_{I_1} \\ &\quad + \underbrace{(\nabla f(\mathbf{z}_t) - \nabla f(\mathbf{x}_t))^\top(\mathbf{z}_{t+1} - \mathbf{z}_t)}_{I_2} + \underbrace{\frac{L}{2}\|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2^2}_{I_3}. \end{aligned}$$

- With the G_∞-bounded gradient assumption, we can bound $\mathbb{E}\nabla\|f(\boldsymbol{\theta}_{\text{out}})\|_2^2$ by bounding I_1, I_2, I_3 seprately!

Comparison of Convergence Rates

Algorithm	$\mathbb{E}\|\nabla f(\theta_{\mathrm{out}})\|_2^2$
Padam ($p \in [0, 1/4]$)	$O\left(\frac{d^{1/4}}{\sqrt{T}} + \frac{d}{T}\right)$
Adam/AmsGrad ($p = 1/2$)	$O\left(\sqrt{\frac{d}{T}} + \frac{d}{T}\right)$
SGD	$O\left(\sqrt{\frac{d}{T}}\right)$

The convergence rate of Padam has a better dependence on data dimension d compared with Adam and SGD[1].

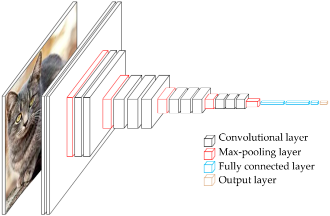
[1] Here we still assume that f has G_∞-bounded stochastic gradient.

Outline

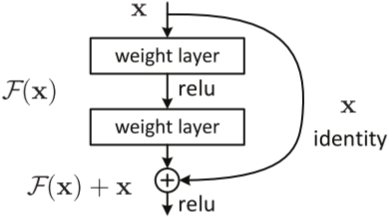
1. Background
2. Our Proposed Method
3. Theoretical Results
4. Experiments and Applications
5. Summary and Future Work

Experimental Setup

- CNN Architectures: VGGNet, ResNet, WideResnet
- Baseline Methods: SGD + Momentum, Adam, Amsgrad
- Datasets: CIFAR10, CIFAR100, ImageNet



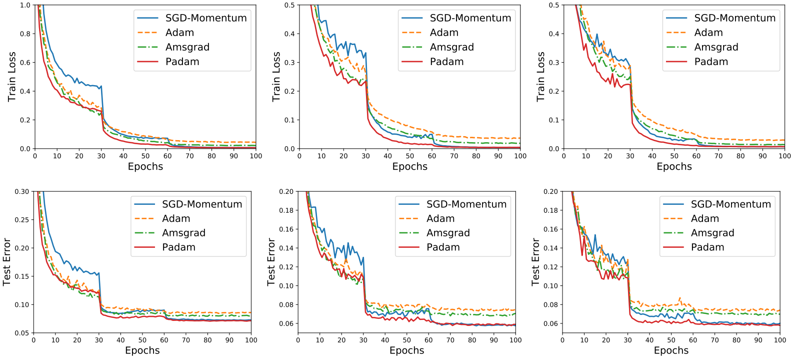
Sample VGGNet



Sample ResNet Block

CIFAR-10 Experiments

- Train loss and test error (top-1 error) of three CNN architectures on CIFAR-10.



VGGNet ResNet WideResNet

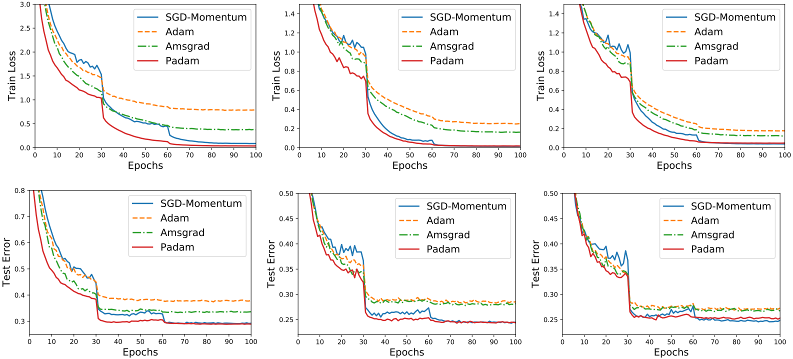
CIFAR-10 Experiments

- Test accuracy of VGGNet on CIFAR-10. Bold number indicates the best result.

Methods	Test Accuracy (%)			
	25th Epoch	50th Epoch	75th Epoch	100th Epoch
SGD Momentum	84.25 ± 0.59	91.01 ± 0.09	92.64 ± 0.14	92.72 ± 0.08
Adam	87.55 ± 0.48	90.74 ± 0.21	91.43 ± 0.36	91.41 ± 0.04
Amsgrad	**88.73 ± 0.41**	91.62 ± 0.12	91.87 ± 0.07	92.04 ± 0.06
Padam	87.73 ± 0.60	**92.11 ± 0.27**	**92.85 ± 0.23**	**92.86 ± 0.11**

CIFAR-100 Experiments

- Train loss and test error (top-1 error) of three CNN architectures on CIFAR-100.



VGGNet ResNet WideResNet

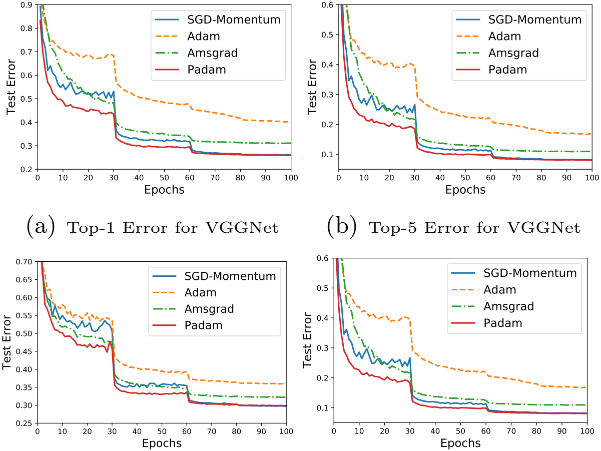
CIFAR-100 Experiments

- Test accuracy of VGGNet on CIFAR-100. Bold number indicates the best result.

Methods	Test Accuracy (%)			
	25th Epoch	50th Epoch	75th Epoch	100th Epoch
SGD Momentum	52.12 ± 0.65	66.53 ± 0.33	70.43 ± 0.24	70.78 ± 0.11
Adam	52.35 ± 0.47	61.73 ± 0.30	62.18 ± 0.15	62.20 ± 0.13
Amsgrad	58.39 ± 0.36	65.31 ± 0.31	66.32 ± 0.25	66.36 ± 0.14
Padam	$\mathbf{60.28 \pm 0.25}$	$\mathbf{69.69 \pm 0.30}$	$\mathbf{71.05 \pm 0.24}$	$\mathbf{71.10 \pm 0.08}$

ImageNet Experiments

- Top-1 and Top-5 error for VGGNet and ResNet on ImageNet dataset.



(a) Top-1 Error for VGGNet

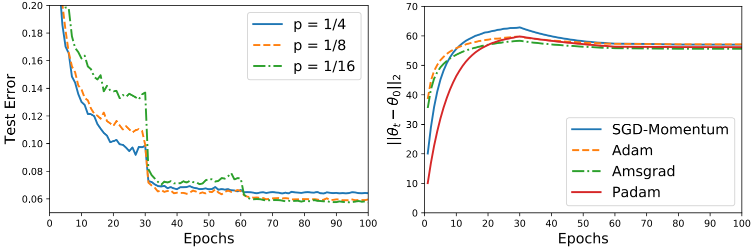
(b) Top-5 Error for VGGNet

(c) Top-1 Error for ResNet

(d) Top-5 Error for ResNet

Sensitivity Analysis

- Plots for sensitivity of p (Left) and $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0\|_2$ against training epochs (Right). Both experiments adopts ResNet on CIFAR-10 dataset.



(a) Sensitivity of p

(b) $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0\|_2$ ($p = 1/8$)

Outline

1. Background

2. Our Proposed Method

3. Theoretical Results

4. Experiments and Applications

5. Summary and Future Work

Summary

- We propose a new algorithm Padam, which generalizes the original Adam algorithm.
- We prove that under both convex and nonconvex setting, Padam achieves the state-of-art convergence result.

Future Work

- Optimality of dependence on dimension is still open.
- Only consider the convergence result for training loss, generalization result remains unknown.

Thank you!