AdaShift: Decorrelation and Convergence of Adaptive Learning Rate Methods

**Zhiming Zhou*, Qingru Zhang*, Guansong Lu,
Hongwei Wang, Weinan Zhang, Yong Yu**

Apex Data & Knowledge Management Lab
Shanghai Jiao Tong University

Content

- Introduction
- Non-convergence behavior of Adam
- Theoretic Analysis
- AdaShift, the Algorithm proposed
- Experiment Results

Introduction

Adaptive Optimization Algorithm:

- **General updating rule:** $\theta_{t+1} = \theta_t - \frac{\alpha_t}{\sqrt{v_t}} m_t$
- **Common choice of** m_t *and* v_t **is the exponential moving average of the gradients and squared gradients.**
- **Some state-of-art algorithms:**
  - **Adam, Adadelta, RMSProp, and Nadm.**
  - **Adam update rules:**

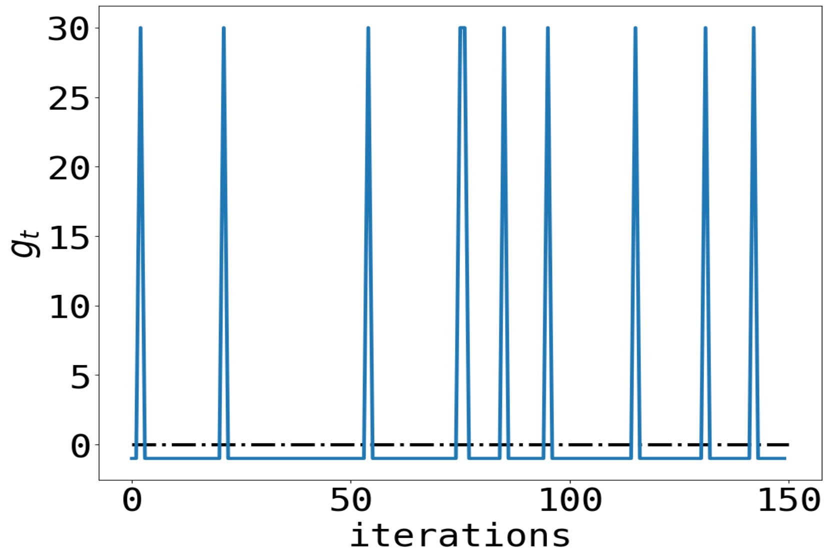
$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$
$$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot m_t / (\sqrt{v_t} + \epsilon)$$

Non-convergence Situations

"On the convergence of Adam and Beyond" **pointed out two type of non-convergence problems for Adam:**

- **Sequential Counterexample:**

$$f_t(\theta) = \begin{cases} C\theta, & \text{if t mod } d = 1; \\ -\theta, & \text{otherwise,} \end{cases}$$



- **Stochastic Counterexample:**

$$f_t(\theta) = \begin{cases} C\theta, & \text{with probability } p = \frac{1+\delta}{C+1}; \\ -\theta, & \text{with probability } 1-p = \frac{C-\delta}{C+1}, \end{cases}$$

APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Non-convergence Situations

Non-convergence Condition

- **Sequential Counterexample:**
  - **For any fixed β_1 *and* β_2 , C need to satisfy:**

$$(1-\beta_1)\beta_1^{C-1}C \le 1-\beta_1^{C-1}, \qquad \beta_2^{(C-2)/2}C^2 \le 1,$$

$$\frac{3(1-\beta_1)}{2\sqrt{1-\beta_2}}\left(1+\frac{\gamma(1-\gamma^{C-1})}{1-\gamma}\right)+\frac{\beta_1^{C/2-1}}{1-\beta_1} < \frac{C}{3},$$

- **Stochastic Counterexample:**
  - **For any fixed β_1 *and* β_2 ,**
  - **when C is large enough (as a function of β_1, β_2, δ),**
  - **the exception of update step will become non-negative**
- **Main Issue**
  - **Positive definiteness of Γ_{t+1}**

$$\Gamma_{t+1} = \left(\frac{\sqrt{V_{t+1}}}{\alpha_{t+1}} - \frac{\sqrt{V_t}}{\alpha_t}\right)$$

Non-convergence Situations

Two solutions proposed by Reddi et al.

- **AMSGrad**

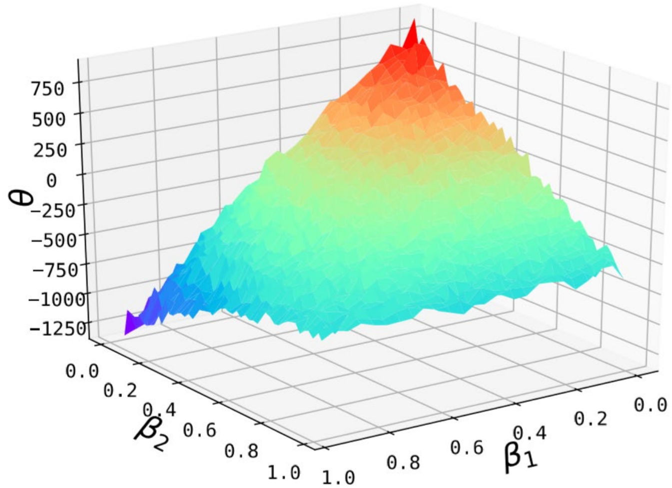
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$
$$\hat{v}_t = \max(\hat{v}_{t-1}, v_t)$$

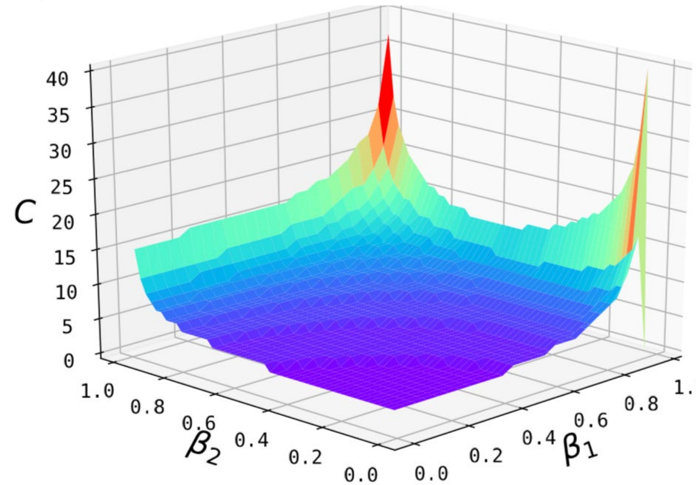
  - **Once a large gradient appears, it will maintain a very large v_t**
  - **and slow down the training process**

- **AdamNC**
  - **do not change the structure of Adam**
  - **use an increasing schedule of β_2, like** $\beta_{2t} = 1 - 1/t$
  - **v_t equal to the average of all history gradients squared**
  - **"long-term memory" but less flexibility**
  - **slightly violate the positive definiteness of Γ_{t+1}**

Non-convergence Condition

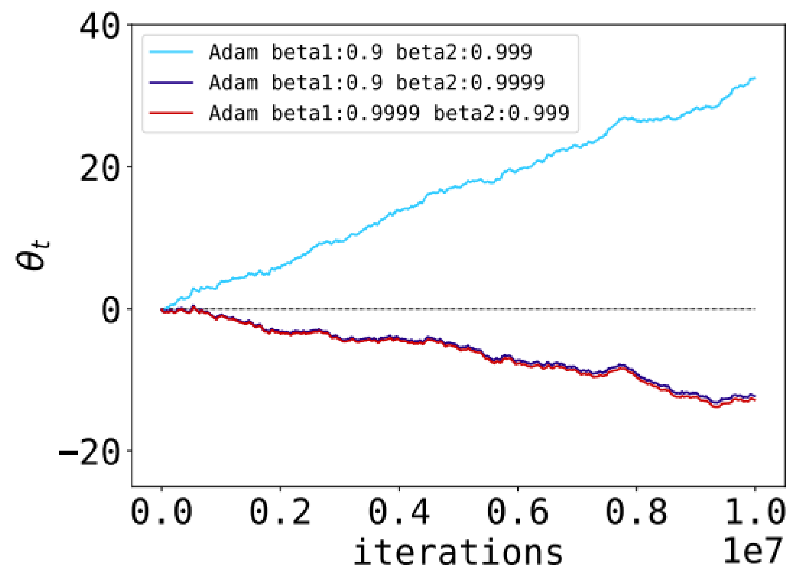
Stochastic Counterexample Experiments:





Conclusion:

- Both β_1 and β_2 influence the direction and speed of optimization
- Critical value of C_t, at which Adam gets into non-convergence, increases as β_1 and β_2 getting large.
- For any fixed C, as long as β_1 and β_2 large enough, non-convergence will disappear



APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

The Cause of Non-Convergence

Unbalanced Step Size

- v_t is positively correlated to the scale of gradient g_t
- It results in a small step size for a large gradient
- a large step size for a small gradient
- A common property of adaptive optimizer

Net Update Factor

To analyze the it we use a new perspective:

- **Consider the effect of every gradients on the whole optimization process.**

$$net(g_t) \triangleq \sum_{i=t}^{\infty} \frac{\alpha_i}{\sqrt{v_i}} [(1-\beta_1)\beta_1^{i-t} g_t] = k(g_t) \cdot g_t,$$

$$\text{where } \; k(g_t) = \sum_{i=t}^{\infty} \frac{\alpha_i}{\sqrt{v_i}} (1-\beta_1)\beta_1^{i-t}$$

Net Update Factor

Sequential Counterexample

- **Limit of v_t**

$$\lim_{n\to\infty} v_{nd+i} = \frac{1-\beta_2}{1-\beta_2^d}(C^2-1)\beta_2^{i-1}+1$$

- **Limit of net update factor**

$$\lim_{n\to\infty} k(g_{nd+i}) = \sum_{t=nd+i}^{\infty} \frac{(1-\beta_1)\beta_1^{t-nd-i}}{\sqrt{\frac{1-\beta_2}{1-\beta_2^d}(C^2-1)\beta_2^{(t-1) \bmod d}+1}}.$$

- **Conclusion: $k(C) < k(-1)$**

Stochastic Counterexample

- **For expectation of net update factor, $k(C) < k(-1)$**

⇒Unbalanced Step Size, combined with suitable β_1 and β_2, will cause the expectation of updates turn to non-negative

Decorrelation leads to convergence

Unbalanced step size is caused by the tight correlation between v_t and g_t

Decorrelation will lead to convergence.

- [Theorem] If v_t follows a fixed distribution and is independent of the current gradient g_t , then the expected net update factor for each gradient is identical.

Role of v_t

- v_t reflects the gradient scale, and adjusts learning rate dynamically
- In AdaShift, current v_t is independent with g_t, but the distribution of v_t is close to g_t's, and changes dynamically with g_t's.

AdaShift, Decorrelation Variant

- **Based on Adam, AdaShift adds two operations:**

  Temporal Shifting & Spatial Decorrelation

- **Algorithm:**

Algorithm 2 Block-wise Temporal-Spatial Decorrelation

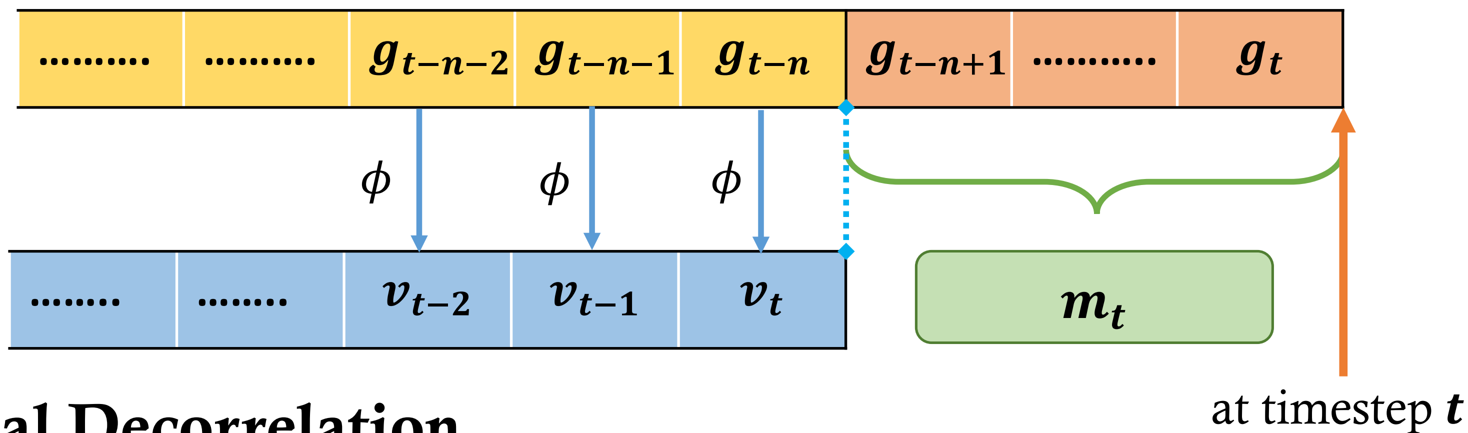
Input: θ_0, g_0, $\{f_t(\theta)\}_{t=1}^T$, $\{\alpha_t\}_{t=1}^T$ and β_2

set $v_0 = 0$
for $t = 1$ **to** T **do**
    $g_t = \nabla f_t(\theta_t)$
    for $i = 1$ **to** M **do**
        $v_t[i] = \beta_2 v_{t-1}[i] + (1 - \beta_2)[\phi(g_{t-1}[i])]^2$
        $\theta_t[i] = \theta_{t-1}[i] - \alpha_t / \sqrt{v_t[i]} \cdot g_t[i]$
    end for
end for

数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

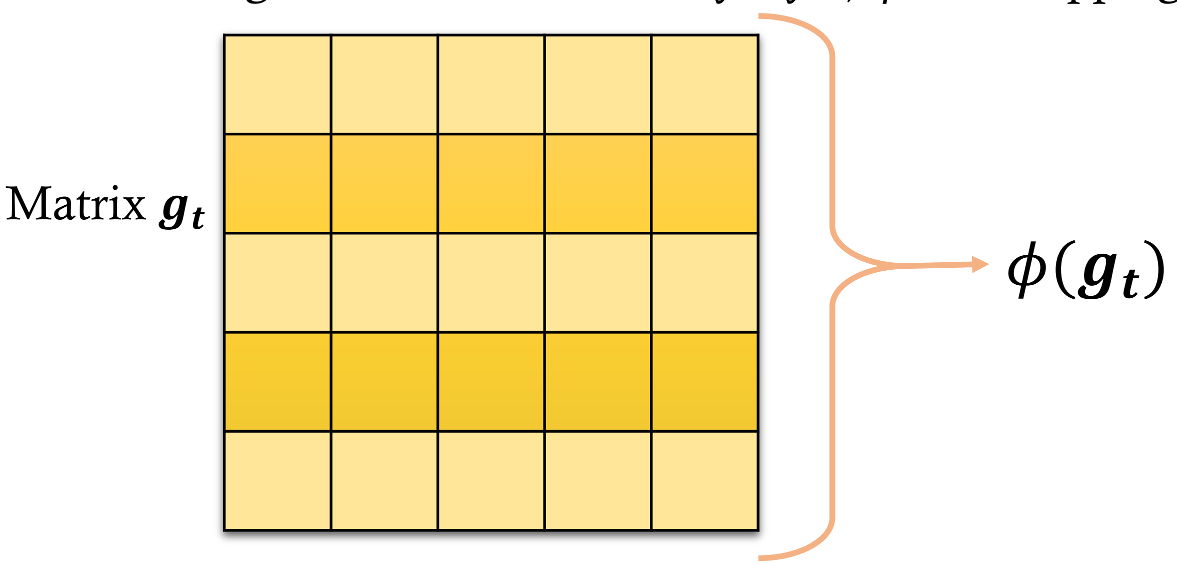
Intuitive Explanation

- **Temporal Shifting:**



- **Spatial Decorrelation**

For the gradient matrix of every layer, φ is a mapping function on it.



Future Work:
The design of ϕ

APEX 数据和知识管理实验室
Data & Knowledge Management Lab

Temporal Shifting

- Given the randomness of mini-batch, we assume that the mini-batch is independent of each other
- Thus, g_t is independent of each other in timeline
- The update rule for v_t now involves g_{t-1} (or g_{t-n}) instead of g_t

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_{t-1}^2$$

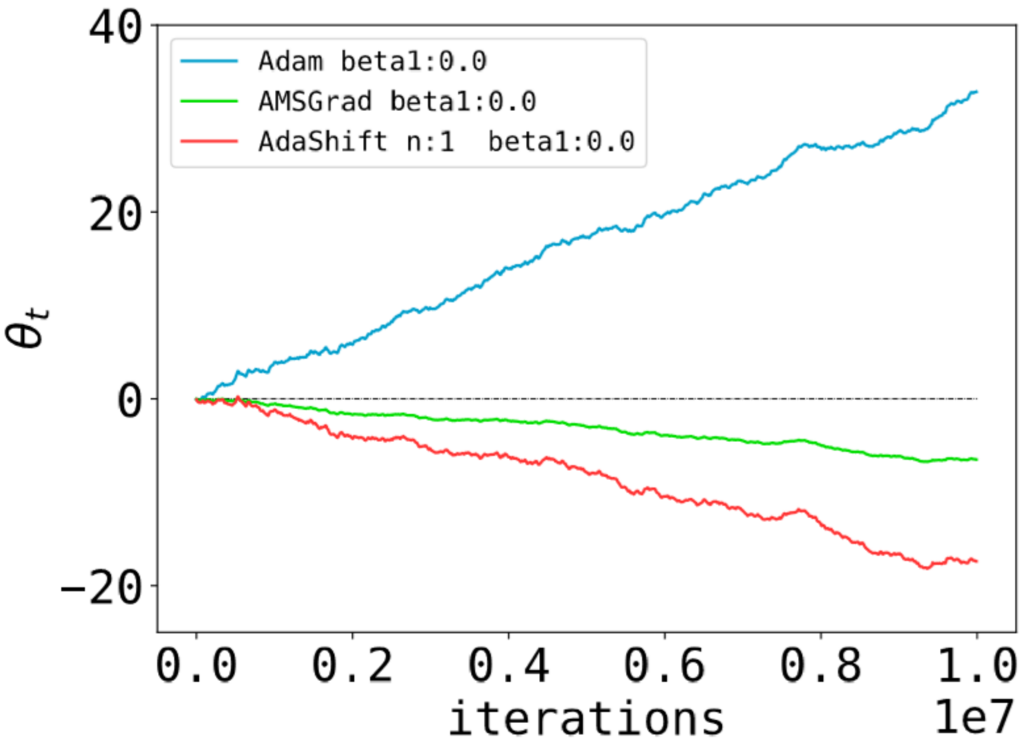
Spatial Decorrelation, Layer-wise Adaptive Learning Rate

for $i = 1$ **to** M **do**
$$v_t[i] = \beta_2 v_{t-1}[i] + (1 - \beta_2)[\phi(g_{t-1}[i])]^2$$
$$\theta_t[i] = \theta_{t-1}[i] - \alpha_t / \sqrt{v_t[i]} \cdot g_t[i]$$
end for

- no longer interpret v_t as the second moment of g_t
- v_t is a random variable, independent of g_t, while at the same time, reflects the overall gradient scale.
- Initialization methods somehow guarantee that the scale gradients in one layer are similar.
- Apply ϕ layer-wisely, outputs a shared adaptive learning rate scalar $v_t[i] \Rightarrow$ an adaptive learning rate SGD
- Adam sometimes does not generalize better than SGD, which might relate to the excessive learning rate adaptation in Adam

Experiment

Stochastic Counterexample



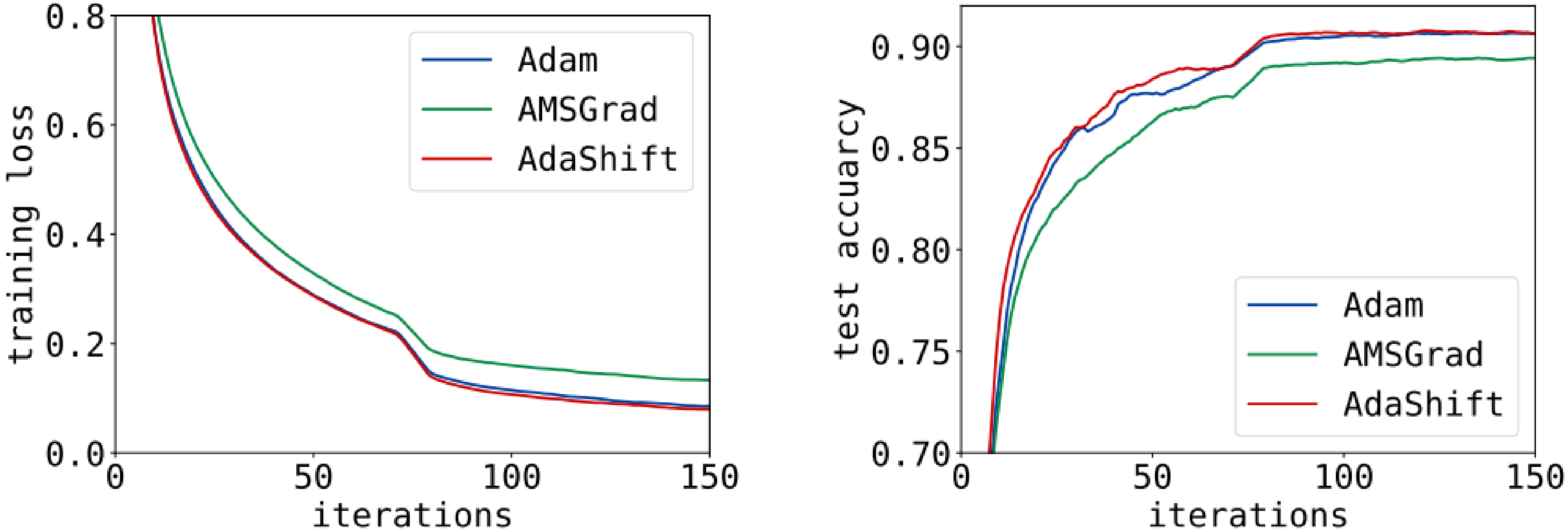
Conclusion:

- **AdaShift will converge on the correct direction and converge at the fastest speed**

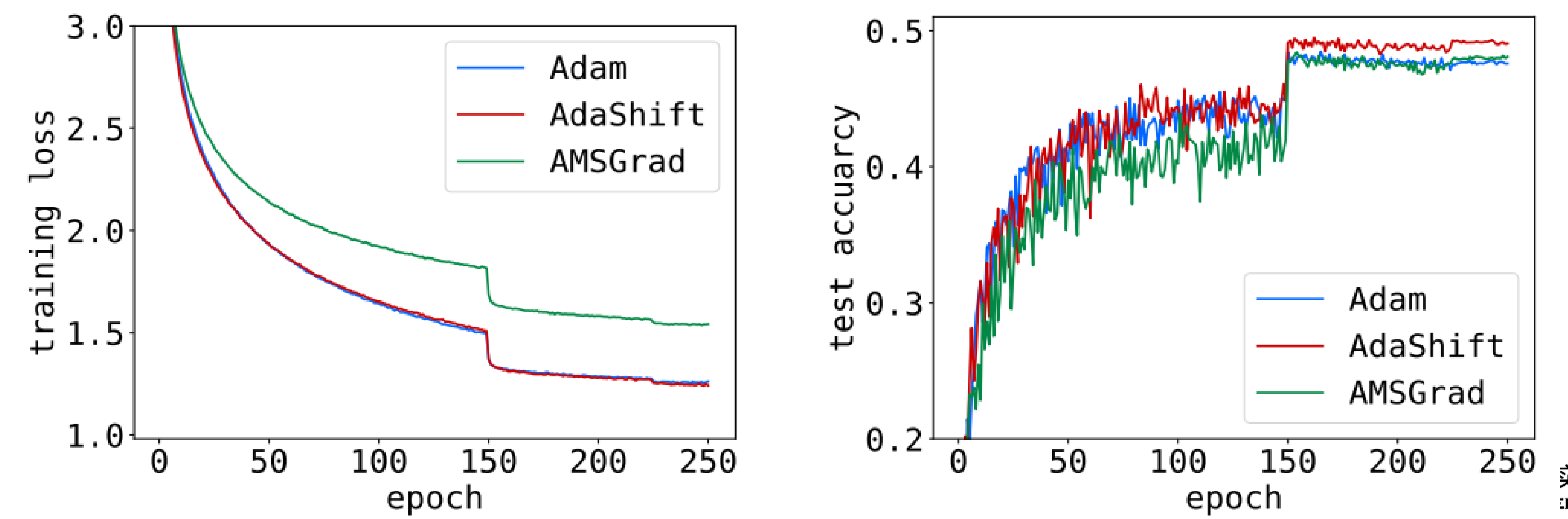
APEX 数据和知识管理实验室
Data & Knowledge Management Lab

Experiment

DenseNet with Cifar-10

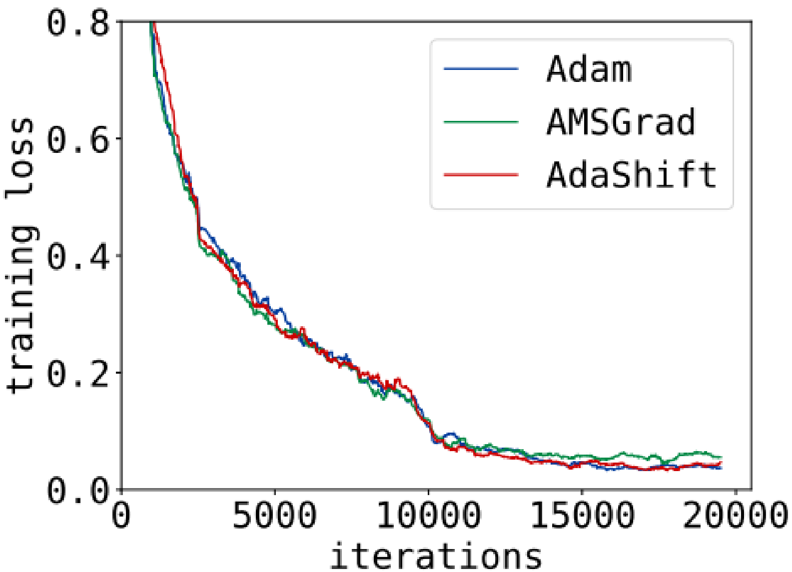


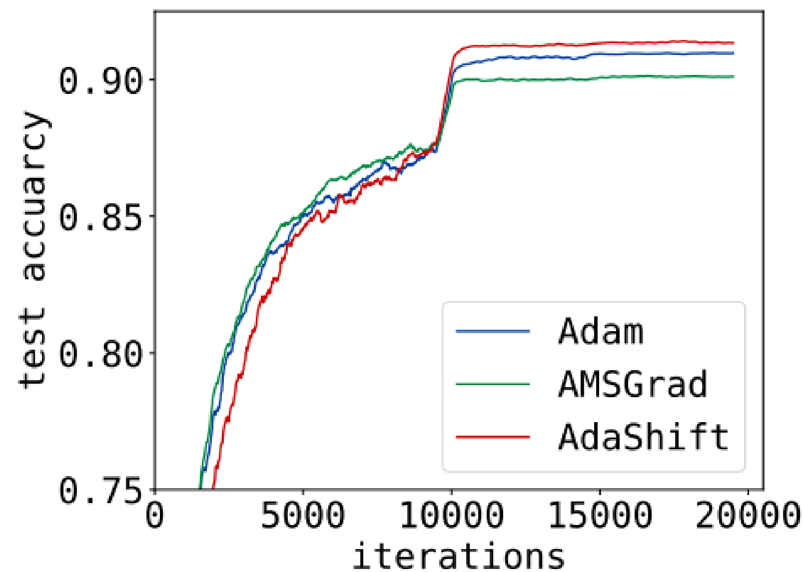
DenseNet with Tiny-ImageNet



Experiment

ResNet with Cifar-10





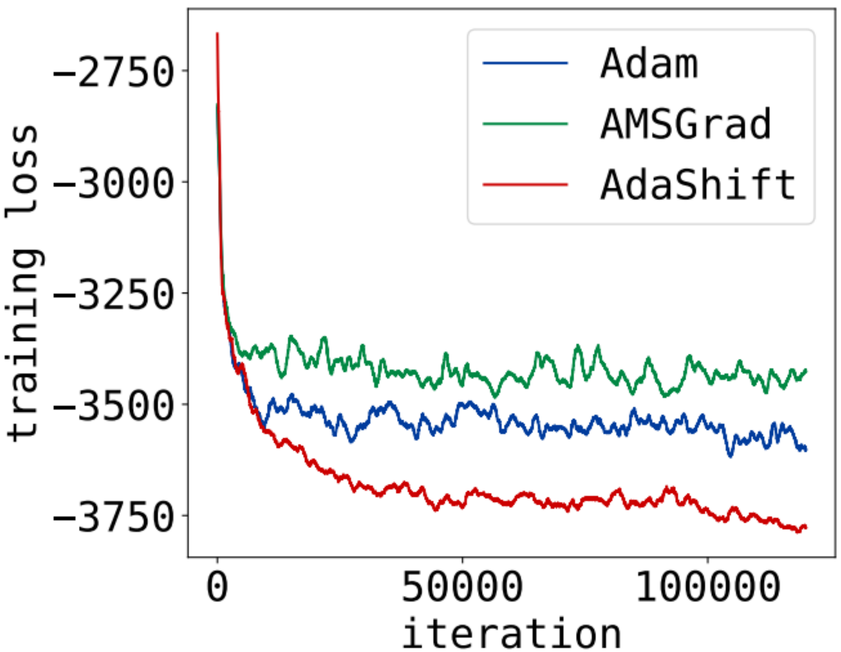
Conclusion:

AdaShift maintain a competitive performance with Adam in terms of both training speed and generalization

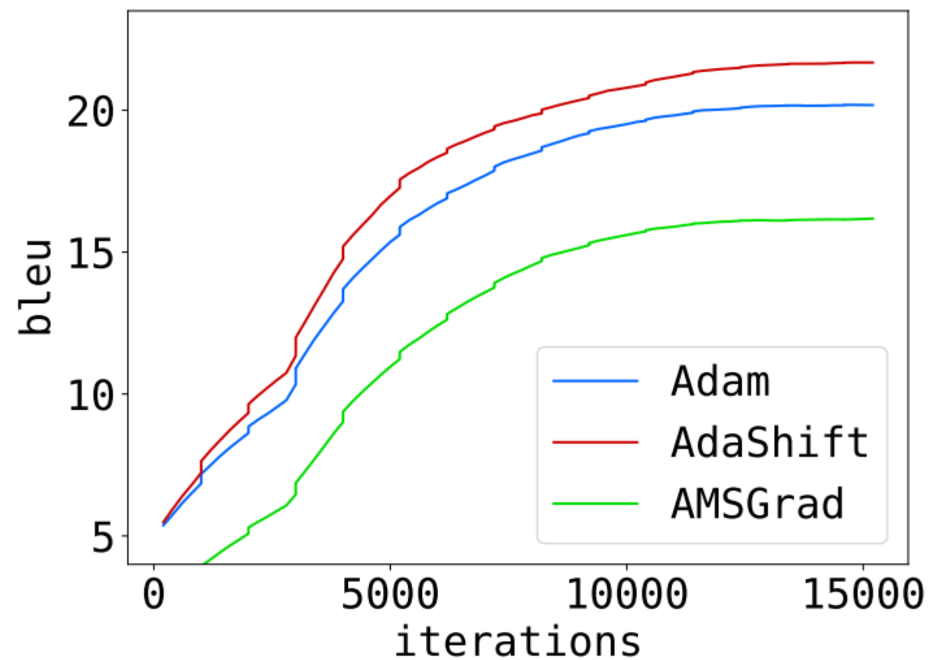
APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Experiment

Training WGAN Discriminator



Neural Machine Translation BLEU



APEX 数据和知识管理实验室
DATA & KNOWLEDGE MANAGEMENT LAB

Extension

- **Design of mapping function ϕ**
- **Understanding on generalization between SGD and Adam**
- **Understanding on layer-wise optimization**
- **Unit-wise adaptive learning rate method**

Q & A

Thanks for Listening!