AdaPlus: Integrating Momentum and Precise Stepsize Adjustment on AdamW Basis

Lei Guan

Department of Mathematics
National University of Defense Technology

Introduction

Popular first-order gradients methods:

- ✓ Accelerated schemes (e.g., SGD with momentum)
- ✓ Adaptive methods: Adam and its variants (e.g., AdamW and AdaBelief)

Three popular Adam-based optimizers

- ✓ AdamW: introduces decoupled weight decay into Adam;
- ✓ Nadam: incorporates Nesterov momentum into Adam;
- ✓ AdaBelief: adapts the stepsize according to the "belief" in the current gradient direction.

Introduction

AdamW: decouple weight decay from the gradient-base update

Algorithm 2 Adam with L_2 regularization and Adam with decoupled weight decay (AdamW)

given $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, \lambda \in \mathbb{R}$
initialize time step $t \leftarrow 0$, parameter vector $\boldsymbol{\theta}_{t=0} \in \mathbb{R}^n$, first moment vector $\boldsymbol{m}_{t=0} \leftarrow \boldsymbol{0}$, second momen vector $\boldsymbol{v}_{t=0} \leftarrow \boldsymbol{0}$, schedule multiplier $\eta_{t=0} \in \mathbb{R}$
repeat
  $t \leftarrow t + 1$
  $\nabla f_t(\boldsymbol{\theta}_{t-1}) \leftarrow \text{SelectBatch}(\boldsymbol{\theta}_{t-1})$ ▷ select batch and return the corresponding gradient
  $\boldsymbol{g}_t \leftarrow \nabla f_t(\boldsymbol{\theta}_{t-1}) + \lambda \boldsymbol{\theta}_{t-1}$
  $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$ ▷ here and below all operations are element-wise
  $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1 - \beta_2)\boldsymbol{g}_t^2$
  $\hat{\boldsymbol{m}}_t \leftarrow \boldsymbol{m}_t / (1 - \beta_1^t)$ ▷ β_1 is taken to the power of t
  $\hat{\boldsymbol{v}}_t \leftarrow \boldsymbol{v}_t / (1 - \beta_2^t)$ ▷ β_2 is taken to the power of t
  $\eta_t \leftarrow \text{SetScheduleMultiplier}(t)$ ▷ can be fixed, decay, or also be used for warm restarts
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta_t \left(\alpha \hat{\boldsymbol{m}}_t / (\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon) + \lambda \boldsymbol{\theta}_{t-1} \right)$
until *stopping criterion is met*
return optimized parameters $\boldsymbol{\theta}_t$

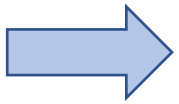
Introduction

Nadam: incorporate Nesterov momentum into Adam

- ✓ Nesterov's accelerated gradient (NAG) has a provably better bound than gradient descent.
- ✓ Nadam combines Adam and NAG.

Algorithm 7 NAG rewritten

$$\mathbf{g}_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1})$$
$$\mathbf{m}_t \leftarrow \mu_t \mathbf{m}_{t-1} + \mathbf{g}_t$$
$$\bar{\mathbf{m}}_t \leftarrow \mathbf{g}_t + \mu_{t+1}\mathbf{m}_t$$
$$\theta_t \leftarrow \theta_{t-1} - \eta \bar{\mathbf{m}}_t$$



Algorithm 8 Nesterov-accelerated adaptive moment estimation

$$\mathbf{g}_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1})$$
$$\hat{\mathbf{g}} \leftarrow \frac{\mathbf{g}_t}{1-\prod_{i=1}^{t}\mu_i}$$
$$\mathbf{m}_t \leftarrow \mu \mathbf{m}_{t-1} + (1-\mu)\mathbf{g}_t$$
$$\hat{\mathbf{m}}_t \leftarrow \frac{\mathbf{m}_t}{1-\prod_{i=1}^{t+1}\mu_i}$$
$$\mathbf{n}_t \leftarrow \nu \mathbf{n}_{t-1} + (1-\nu)\mathbf{g}_t^2$$
$$\hat{\mathbf{n}}_t \leftarrow \frac{\mathbf{n}_t}{1-\nu^t}$$
$$\bar{\mathbf{m}}_t \leftarrow (1-\mu_t)\hat{\mathbf{g}}_t + \mu_{t+1}\hat{\mathbf{m}}_t$$
$$\theta_t \leftarrow \theta_{t-1} - \eta \frac{\bar{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{n}}_t}+\varepsilon}$$

Precise Stepsize Adjustment

AdaBelief: achieve precise stepsize adjustment with “belief”

Algorithm 1: Adam Optimizer

Initialize θ_0, $m_0 \leftarrow 0$, $v_0 \leftarrow 0$, $t \leftarrow 0$

While θ_t not converged

$t \leftarrow t+1$

$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$

$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$

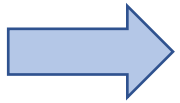
$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$

Bias Correction

$\widehat{m_t} \leftarrow \frac{m_t}{1-\beta_1^t}$, $\widehat{v_t} \leftarrow \frac{v_t}{1-\beta_2^t}$

Update

$\theta_t \leftarrow \prod_{\mathcal{F},\sqrt{\widehat{v_t}}} \left(\theta_{t-1} - \frac{\alpha \widehat{m_t}}{\sqrt{\widehat{v_t}}+\epsilon} \right)$



Algorithm 2: AdaBelief Optimizer

Initialize θ_0, $m_0 \leftarrow 0$, $s_0 \leftarrow 0$, $t \leftarrow 0$

While θ_t not converged

$t \leftarrow t+1$

$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$

$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$

$s_t \leftarrow \beta_2 s_{t-1} + (1-\beta_2)(g_t - m_t)^2 + \epsilon$

Bias Correction

$\widehat{m_t} \leftarrow \frac{m_t}{1-\beta_1^t}$, $\widehat{s_t} \leftarrow \frac{s_t}{1-\beta_2^t}$

Update

$\theta_t \leftarrow \prod_{\mathcal{F},\sqrt{\widehat{s_t}}} \left(\theta_{t-1} - \frac{\alpha \widehat{m_t}}{\sqrt{\widehat{s_t}}+\epsilon} \right)$

Introduction

Three popular Adam-based optimizers

- ✓ AdamW: introduces decoupled weight decay into Adam;
- ✓ Nadam: incorporates Nesterov momentum into Adam;
- ✓ AdaBelief: adapts the stepsize according to the "belief" in the current gradient direction.

Remarkable features of AdamW, Nadam, and AdaBelief:

- ✓ Build based on Adam;
- ✓ Enjoy different advantages in terms of boosting adaptive methods.

Guess: Can we combine the benefits of these three popular optimizers?

Contributions

- We propose a new optimizer AdaPlus.
  - ✓ AdaPlus = AdamW + Nesterov momentum (Nadam) + "Belief" (AdaBelief)
  - ✓ This is the first adaptive method that simultaneously combines the adavantages of decoupled weight decay, Nesterov momentum, and precise stepsize adjustment.
- We conducted extensive experimental evaluations to validate the effectiveness of AdaPlus.
  - ✓ AdaPlus performs the best in simultaneously achieving the goal of fast convergence, good generalization ability, and high stability.
  - ✓ For example, on the image classification task, AdaPlus yields an average test accuracy improvement of 1.97% (up to 2.36%), 1.85% (up to 2.0%), and 0.52% (up to 0.89%) over AdamW, Nadam, and AdaBelief , respectively.

Modifying AdamW's Momentum

Classical momentum

$$\mathbf{g}_t \leftarrow \nabla_{\boldsymbol{\theta}-1} f(\boldsymbol{\theta}_{t-1}),$$
$$\mathbf{m}_t \leftarrow u\mathbf{m}_{t-1} + a\mathbf{g}_t,$$
$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \mathbf{m}_t.$$

$$\mathbf{g}_t \leftarrow \nabla_{\boldsymbol{\theta}-1} f(\boldsymbol{\theta}_{t-1}),$$
$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - (u\mathbf{m}_{t-1} + a\mathbf{g}_t).$$

Classical momentum vs. NAG

$$\mathbf{g}_t \leftarrow \nabla_{\boldsymbol{\theta}-1} f(\boldsymbol{\theta}_{t-1}),$$
$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - (u\mathbf{m}_{t-1} + a\mathbf{g}_t).$$

classical momentum

$$\mathbf{g}_t \leftarrow \nabla_{\boldsymbol{\theta}-1} f(\boldsymbol{\theta}_{t-1}),$$
$$\mathbf{m}_t \leftarrow u\mathbf{m}_{t-1} + a\mathbf{g}_t,$$
$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - (u\mathbf{m}_t + a\mathbf{g}_t).$$

reformulation of NAG

NAG updates the parameter with m_t rather than m_{t-1}

Modifying AdamW's Momentum

AdamW's update step:

$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \frac{a}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon}\left(\frac{\beta_1 \mathbf{m}_{t-1}}{1-\beta_1^t} + \frac{(1-\beta_1)\mathbf{g}_t}{1-\beta_1^t}\right). \quad (2)$$

Replace m_{t-1} with m_t:

$$\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \frac{a}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon}\left(\frac{\beta_1 \mathbf{m}_t}{1-\beta_1^t} + \frac{(1-\beta_1)\mathbf{g}_t}{1-\beta_1^t}\right). \quad (3)$$

$$\begin{aligned} \bar{\mathbf{m}}_t &\leftarrow \beta_1 \mathbf{m}_t + (1-\beta_1)\mathbf{g}_t, \\ \hat{\mathbf{m}}_t &\leftarrow \frac{\bar{\mathbf{m}}_t}{1-\beta_1^t}, \\ \boldsymbol{\theta}_t &\leftarrow \boldsymbol{\theta}_{t-1} - \frac{a\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon}. \end{aligned} \quad (4)$$

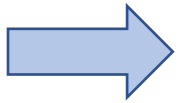
Modifying AdamW's Momentum

AdamW vs. AdamW + Nesterov Momentum

Algorithm 1 The AdamW Optimizer

Require: initial learning rate $a = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay factor $\lambda \in \mathbb{R}$

Initialize time step $t \leftarrow 0$, $\boldsymbol{\theta}_0$, $\mathbf{m}_0 \leftarrow 0$, $\mathbf{v}_0 \leftarrow 0$, $t \leftarrow 0$.
while θ_t not converged **do**
  $t \leftarrow t + 1$
  $\mathbf{g}_t \leftarrow \nabla_\theta f_t(\boldsymbol{\theta}_{t-1})$
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \gamma\lambda\boldsymbol{\theta}_{t-1}$
  $\mathbf{m}_t \leftarrow \beta_1\mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t$
  $\mathbf{v}_t \leftarrow \beta_2\mathbf{v}_{t-1} + (1-\beta_2)\mathbf{g}_t^2$
  $\hat{\mathbf{m}}_t \leftarrow \frac{\mathbf{m}_t}{1-\beta_1^t}$, $\hat{\mathbf{v}}_t \leftarrow \frac{\mathbf{v}_t}{1-\beta_2^t}$
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \frac{a\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t}+\epsilon}$
end while



Algorithm 2 AdamW + Nesterov momentum

Require: initial learning rate $a = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay factor $\lambda \in \mathbb{R}$

Initialize time step $t \leftarrow 0$, $\boldsymbol{\theta}_0$, $\mathbf{m}_0 \leftarrow 0$, $\mathbf{v}_0 \leftarrow 0$, $t \leftarrow 0$.
while θ_t not converged **do**
  $t \leftarrow t + 1$
  $\mathbf{g}_t \leftarrow \nabla_\theta f_t(\boldsymbol{\theta}_{t-1})$
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \gamma\lambda\boldsymbol{\theta}_{t-1}$
  $\mathbf{m}_t \leftarrow \beta_1\mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t$
  $\mathbf{v}_t \leftarrow \beta_2\mathbf{v}_{t-1} + (1-\beta_2)\mathbf{g}_t^2$
  $\bar{\mathbf{m}}_t \leftarrow \beta_1\mathbf{m}_t + (1-\beta_1)\mathbf{g}_t$
  $\hat{\mathbf{m}}_t \leftarrow \frac{\bar{\mathbf{m}}_t}{1-\beta_1^t}$, $\hat{\mathbf{v}}_t \leftarrow \frac{\mathbf{v}_t}{1-\beta_2^t}$
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \frac{a\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t}+\epsilon}$
end while

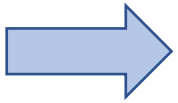
Precise Stepsize Adjustment

AdamW+Nesterov momentum vs. AdaPlus

Algorithm 2 AdamW + Nesterov momentum

Require: initial learning rate $a = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay factor $\lambda \in \mathbb{R}$

Initialize time step $t \leftarrow 0$, $\boldsymbol{\theta}_0$, $\mathbf{m}_0 \leftarrow 0$, $\mathbf{v}_0 \leftarrow 0$, $t \leftarrow 0$.
while θ_t not converged **do**
  $t \leftarrow t + 1$
  $\mathbf{g}_t \leftarrow \nabla_\theta f_t(\boldsymbol{\theta}_{t-1})$
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \gamma\lambda\boldsymbol{\theta}_{t-1}$
  $\mathbf{m}_t \leftarrow \beta_1\mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t$
  $\mathbf{v}_t \leftarrow \beta_2\mathbf{v}_{t-1} + (1-\beta_2)\mathbf{g}_t^2$
  $\bar{\mathbf{m}}_t \leftarrow \beta_1\mathbf{m}_t + (1-\beta_1)\mathbf{g}_t$
  $\hat{\mathbf{m}}_t \leftarrow \frac{\bar{\mathbf{m}}_t}{1-\beta_1^t}$, $\hat{\mathbf{v}}_t \leftarrow \frac{\mathbf{v}_t}{1-\beta_2^t}$
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \frac{a\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t}+\epsilon}$
end while



Algorithm 3 The AdaPlus Optimizer

Require: initial learning rate $a = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay factor $\lambda \in \mathbb{R}$

Initialize time step $t \leftarrow 0$, $\boldsymbol{\theta}_0$, $\mathbf{m}_0 \leftarrow 0$, $\mathbf{v}_0 \leftarrow 0$, $t \leftarrow 0$.
while θ_t not converged **do**
  $t \leftarrow t + 1$
  $\mathbf{g}_t \leftarrow \nabla_\theta f_t(\boldsymbol{\theta}_{t-1})$
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \gamma\lambda\boldsymbol{\theta}_{t-1}$
  $\mathbf{m}_t \leftarrow \beta_1\mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t$
  $\mathbf{s}_t \leftarrow \beta_2\mathbf{s}_{t-1} + (1-\beta_2)(\mathbf{g}_t - \mathbf{m}_t)^2 + \epsilon$
  $\bar{\mathbf{m}}_t \leftarrow \beta_1\mathbf{m}_t + (1-\beta_1)\mathbf{g}_t$
  $\hat{\mathbf{m}}_t \leftarrow \frac{\bar{\mathbf{m}}_t}{1-\beta_1^t}$, $\hat{\mathbf{s}}_t \leftarrow \frac{\mathbf{s}_t}{1-\beta_2^t}$
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \frac{a\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{s}}_t}+\epsilon}$
end while

Precise Stepsize Adjustment

The main advantage of AdaBelief over Adam lies in "large gradient, small curvature" case

- ✓ AdaBelief increase the stepsize as an ideal optimizer does;
- ✓ Adam decreases the stepsize.

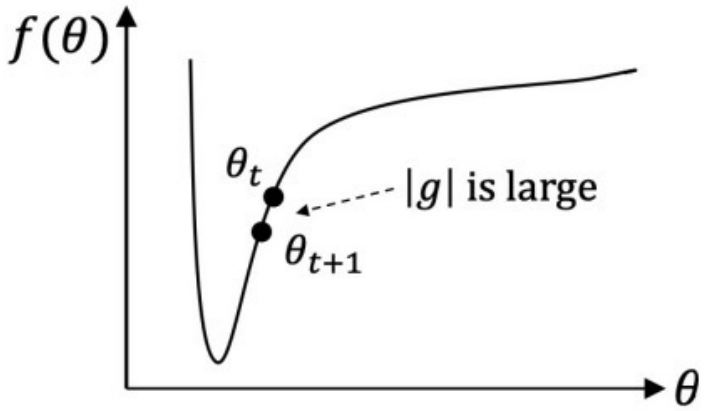


Fig. 1: Illustration of "large gradient, small curvature" case where current stepsize is small and $|g(\boldsymbol{\theta}_t) - g(\boldsymbol{\theta}_{t+1})|$ is small. An ideal optimizer should increase the stepsize.

$$\Delta\boldsymbol{\theta}_t^{\text{AdamW, Nadam}} = -\frac{a\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon},$$
$$\Delta\boldsymbol{\theta}_t^{\text{AdaBelief, AdaPlus}} = -\frac{a\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{s}}_t} + \epsilon} \quad (5)$$

- ✓ AdamW and Nadam decrease the stepsize
- ✓ AdaBelief and AdaPlus increase the stepsize as the "ideal" optimizer does

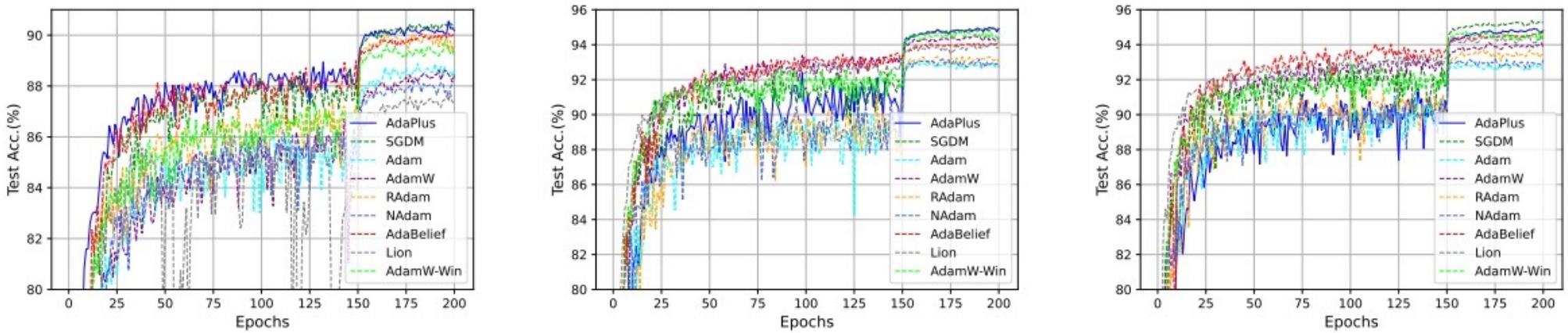
Experiment settings

- Baseline optimizers:
  - ✓ SGDM, Adam, Nadam, AdamW, Radam, AdaBelief, AdamW-Win, Lion.
- Computing platform:
  - ✓ A multi-GPU machine equipped with four NVIDIA Tesla GPUs
- Machine-learning tasks:
  - ✓ Image classification: VGG-11, ResNet-34 and DenseNet-121 on CIFAR-10
  - ✓ Language modeling: 1-layer, 2-layer, and 3-layer LSTMs on Penn TreeBank
  - ✓ GANs: Wasserstein GAN (WGAN) and WGAN-GP on CIFAR-10

Experiments for Image Classification

Table 1: Maximum test accuracy on CIFAR-10. **Higher** is better.

Models	AdaPlus	SGDM	Adam	Nadam	AdamW	RAdam	AdaBelief	Lion	AdamW-Win
VGG-11	**90.55%**	90.48%	88.89%	88.19%	88.64%	90.05%	90.07%	87.71%	89.72%
ResNet-34	**94.99%**	94.96%	92.99%	93.19%	94.50%	93.33%	94.10%	94.10%	94.72%
DenseNet-121	94.91%	**95.37%**	93.02%	93.17%	94.11%	93.70%	94.71%	94.54%	94.75%



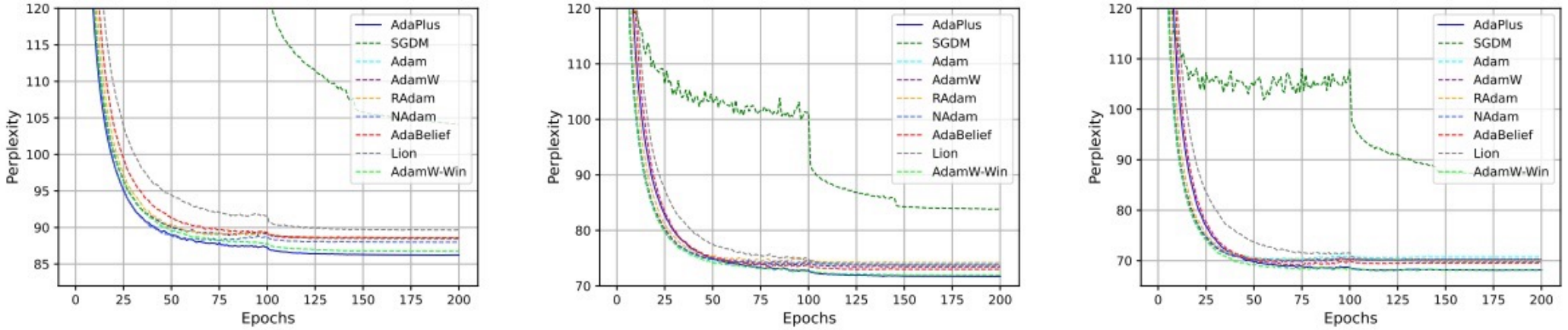
(a) VGG-11 on CIFAR-10 (b) ResNet-34 on CIFAR-10 (c) DenseNet-121 on CIFAR-10

Fig. 2: Validation accuracy vs. epochs of training VGG-11, ResNet-34, and DenseNet-121 on CIFAR-10.

Experiments for Language Modeling

Table 2: Minimum perplexity on Penn TreeBank. **Lower** is better.

LSTM	AdaPlus	SGDM	Adam	Nadam	AdamW	RAdam	AdaBelief	Lion	AdamW-Win
1 layer	**86.22**	104.13	88.54	87.98	88.49	88.56	88.59	89.64	86.73
2 layers	**71.72**	83.80	73.72	73.91	73.43	74.20	72.97	73.65	71.93
3 layers	68.08	86.93	70.24	69.82	69.67	70.01	69.10	69.77	**68.03**



(a) 1-layer LSTM (b) 2-layer LSTM (c) 3-layer LSTM

Fig. 3: Perplexity vs. epochs of training LSTM on Penn TreeBank.

Experiments for GANs on CIFAR-10

Table 3: FID (lower is better) of WGAN and WGAN-GP on CIFAR-10.

Model	AdaPlus	SGDM	Adam	Nadam	AdamW	RAdam	AdaBelief	Lion	AdamW-Win
WGAN	82.96	299.88	94.15	95.17	93.72	108.09	86.92	77.48	**60.10**
WGAN-GP	**63.70**	257.67	76.60	76.54	68.85	94.29	66.63	249.58	64.40

Conclusions

- We propose a novel and efficient adaptive method AdaPlus.
  - ✓ We construct efficient scheme to combine the advantages of AdamW, Nadam, and Adabelief.
  - ✓ The experimental results validate that AdaPlus outperforms the popular eight state-of-the-art optimizers in terms of simultaneously considering convergence traits, generalization ability, and training stability.
- AdaPlus provides new idea for designing deep learning optimizers.

Thank you!

Feel free to contact me!

guanleimath@163.com