Converge or Diverge? A Story of Adam

Yushun Zhang

School of Data Science,
The Chinese University of Hong Kong, Shenzhen

Sharing at Tsinghua University. Thanks Prof. Jian Li for the invitation!

Sep 28th, 2023

Adam: Adaptive Moment Estimation

2015
Birth of Adam
Claim: converges

Kingma and Ba. Adam: A method for stochastic optimization. ICLR 2015.
Cited by > 150k times

2018
"Adam diverges"

Reddi, Kale, & Kumar, On the convergence of Adam and beyond. ICLR 2018.
ICLR 2018 best paper

2022
"Adam converges"

Zhang, Chen, Shi, Sun, & Luo, Adam can converge without any modification on update rules. *NeurIPS 2022*

What to expect from this talk?

- **Question: Adam** converges or not? How to tune it?

- **For practitioners:**
  - ➢ Story of Adam: what it is, popularity, convergence
  - ➢ how to tune hyperparameters of Adam

- **For optimization theorists:**
  - ➢ Different meanings of "algorithm convergence"
  - ➢ Divergence-convergence phase transition
  - ➢ A method to analyze stochastic non-linear iterations

Empirical Guidance: Hyperparameter Tuning

- We prove that Adam can converge without ANY modification.

- Hyperparameter tunning suggestions:
  - First, tune up β_2.

    Then, try different β_1 with $\beta_1 < \sqrt{\beta_2}$
  - **Detailed suggestions:** end of talk

Tip for professors:

If DL experiments failed, ask students one more question:

have you tuned Adam hyperparameters?

(many think Adam is tuning-free)

Contents

Story of Adam: More Complete Version

1840-2010

Pre-ML Stage:
Development of 1st and 2nd order methods

2011-2015

Stage 1: ML Development Stage: Adagrad, RMSProp, Adam

2015-2022

Emp-Stage 2:
Popularity in AI

2018

Theo-Stage 2: Adam does not converge
variants: AMSGrad, Adashift, Adabound…

2022

Theo-Stage 3: Adam can converge

2022-?

Still dominating in ChatGPT
New algorithm?

Pre-ML Stage: Classical Algorithms (1840-2010)

- Central issue in (unconstrained) nonlinear optimization:

  information v.s. computation

1st order methods: gradient descent (1847, Cauchy), Accelerated 1st order method (Nesterov, 1983)

Second order methods: Newton method

Quasi-2nd order methods:

BFGS (1970s), LBFGS (1980s), BB (1980s)

Stage 1: Development of Adam (2011-2015)

2011: Adagrad, JMLR

Duchi, John, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research* 12.7 (2011).

2012: RMSProp, Lecture notes by Hinton

[CITATION] Lecture 6.5-**rmsprop**: Divide the gradient by a running average of its recent magnitude

T Tieleman, G **Hinton** - COURSERA: Neural networks for machine learning, 2012

Save Cite Cited by 6438 Related articles

2015: Adam, ICLR

Kingma,Ba. Adam: A method for stochastic optimization. ICLR 2015.

Let us start with SGD⋯

- Consider $\min_x f(x) := \sum_{i=1}^{n} f_i(x)$.

  n: number of samples (or mini-batches of samples)

  x: trainable parameters

- In the k-th iteration: Randomly sample τ_k from $\{1,2,\ldots,n\}$

SGD (Stochastic gradient descent): $x_{k+1} = x_k - \eta_k \nabla f_{\tau_k}(x_k)$

SGD with momentum (SGDM):

$$m_k = (1-\beta_1)\nabla f_{\tau_k}(x_k) + \beta_1 m_{k-1}$$

$$x_{k+1} = x_k - \eta_k m_k$$

⇐ 1st order momentum

⇐ Iterate update

Adagrad

$$\min_x f(x) \coloneqq \sum_{i=1}^{n} f_i(x) .$$

n: number of samples (or mini-batches of samples)

x: trainable parameters

In the k-th iteration: Randomly sample τ_k from $\{1,2,\dots,n\}$

Adagrad (Duchi et al.'11):

- $v_k = \left(\frac{k-1}{k}\right) \mathrm{v_{k-1}} + \frac{1}{k} \nabla \mathrm{f}_{\tau_\mathrm{k}}(\mathrm{x_k}) \circ \nabla \mathrm{f}_{\tau_\mathrm{k}}(\mathrm{x_k})$
- $x_{k+1} = x_k - \eta_k \frac{\nabla \mathrm{f}_{\tau_\mathrm{k}}(\mathrm{x_k})}{\sqrt{v_k}}$

2nd order momentum

Iterate update

Adagrad outperforms SGD significantly on language tasks

Becomes the default choice among NLPers, for ~5 years

Duchi, John, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research* 12.7 (2011).

RMSProp

AdaGrad: it treats all samples equally

RMSprop: use EMA (exponential moving average) to define v_k

RMSProp (Hinton '12):

- $v_k = (1-\beta_2)\nabla f_{\tau_k}(x_k) \circ \nabla f_{\tau_k}(x_k) + \beta_2 v_{k-1}$
- $x_{k+1} = x_k - \eta_k \frac{\nabla f_{\tau_k}(x_k)}{\sqrt{v_k}}$

2nd order momentum

Iterate update

Proposed in the lecture notes by Geoffrey Hinton

PyTorch default Choice: $\beta_2 = 0.99$

Adam

- $\min_x f(x) := \sum_{i=1}^{n} f_i(x)$. In the k-th iteration: Randomly sample τ_k from $\{1,2,\dots,n\}$

- Adam (Kingma and Ba'15):
- $m_k = (1-\beta_1)\nabla f_{\tau_k}(x_k) + \beta_1 m_{k-1}$
- $v_k = (1-\beta_2)\nabla f_{\tau_k}(x_k) \circ \nabla f_{\tau_k}(x_k) + \beta_2 v_{k-1}$
- $x_{k+1} = x_k - \eta_k \frac{\sqrt{1-\beta_2^k}}{1-\beta_1^k} \frac{m_k}{\sqrt{v_k}}$

1st order momentum

2nd order momentum

Iterate update

- β_1: Controls the 1st-order momentum m_k. Default setting: $\beta_1 = 0.9$
- β_2: Controls the 2nd-order momentum v_k. Default setting: $\beta_2 = 0.999$

Emp-Stage 2: Popularity in AI

- **Adam** becomes the most popular algorithms in deep learning (DL). (>150,000 citations, by August 2023)
- **Default in** LLM (large language models)

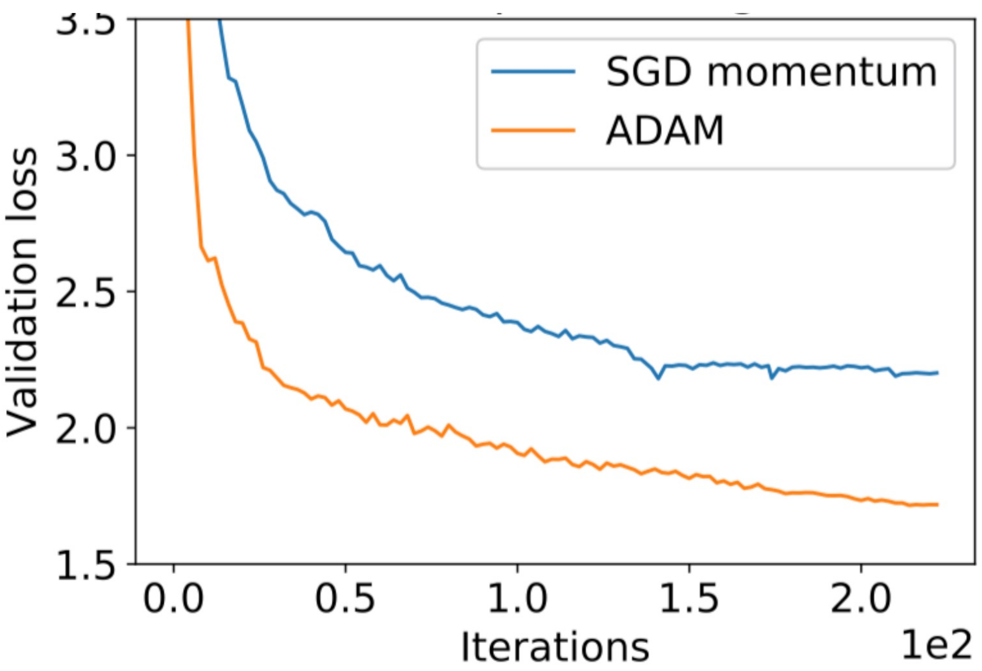
```
optimizer = optim.Adam(net.parameters(), lr=args.lr, betas=(args.beta1, args.beta2), eps=1e-08,
                       weight_decay=args.weightdecay, amsgrad=False)
```

- **Empirical fact (sad?): Adam seems to be the only choice for LLMs like ChatGPT**

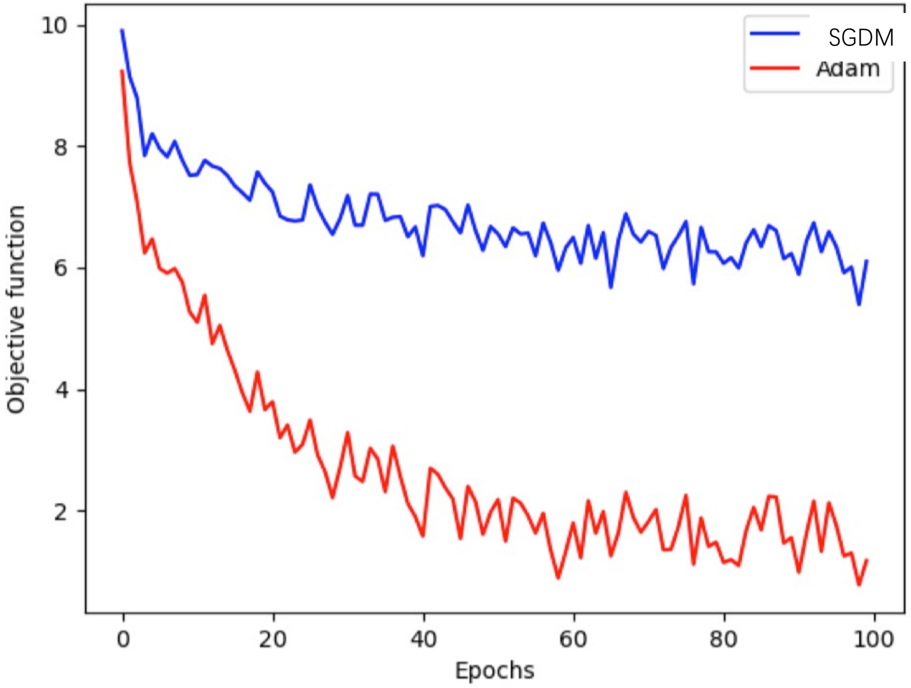
  --Recent new algorithms (Sophie, Lion, etc.)

  cannot beat Adam on 100 billion-parameter models.

Advantages of Adam



BERT (from [Zhang et al.19])

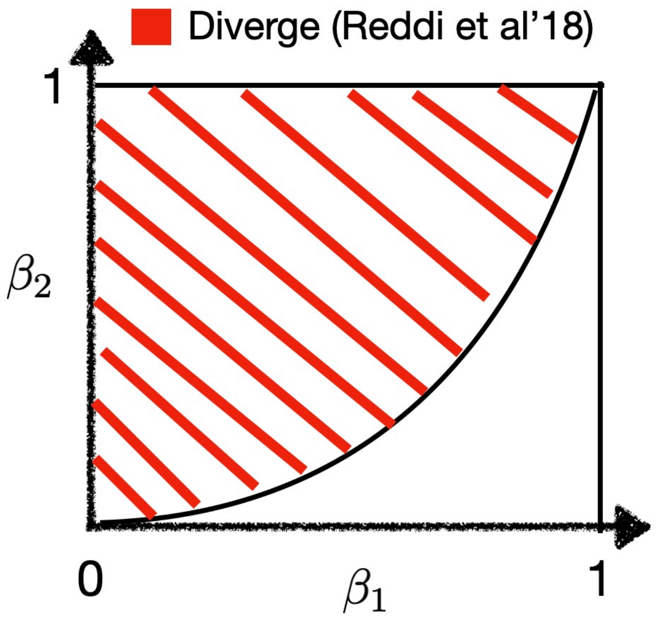


GPT (from [Wang et al.22])

Adam significantly outperforms SGDM in training large-AI models

Theo-Stage 2: "Adam does not converge"

Reddi et al.18 **(ICLR Best paper):**
For any β_1, β_2 s.t. $\beta_1 < \sqrt{\beta_2}$, there exists a problem such that Adam diverges



Debate on "convergence issue"

ICLR'18 paper reader's comment:

[−] **Is the problem with Adam, or with the theoretical framework used to analyse it?**

Jeremy Bernstein

26 Apr 2018 (modified: 26 Apr 2018) ICLR 2018 Conference Paper807 Public

Reader: "My claim is that…for any problem, a properly tuned-Adam will converge at least as well as SGD"

ICLR'18 paper authors reply:

[−] **TL;DR : Its with the algorithm**

ICLR 2018 Conference Paper807 Authors

01 Jun 2018 ICLR 2018 Conference Paper807 Official Comment Readers: Everyone

Comment: Dear Jeremy,

Thank you for your interest in the paper.

To answer your question "Is the problem with Adam ....?" : Our paper shows that the algorithm defined in the Adam paper (https://arxiv.org/pdf/1412.6980.pdf, Algorithm 1) (including one with decreasing step size alpha) has convergence issues. Specifically, for any setting of the Adam parameters (beta_1, beta_2, minibatch size, epsilon, etc) there is a convex optimization setting where Adam will not converge to the optimal solution, even if decreasing learning rates are used. This is in contrast to algorithms like SGD which, with decreasing learning rates, is guaranteed to converge.

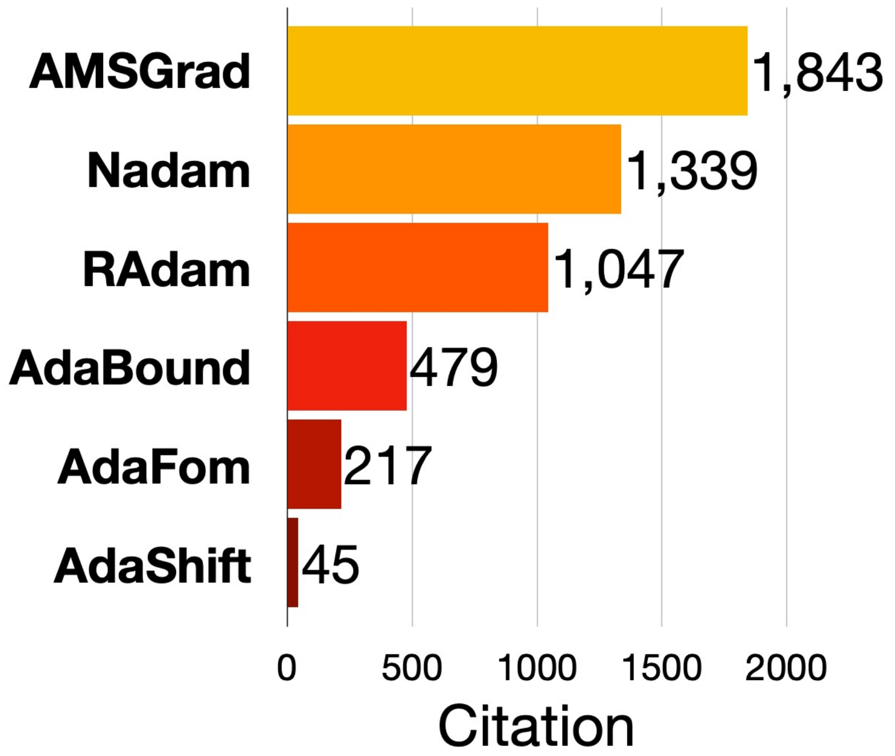
Authors: "Our paper shows that the algorithm defined in the Adam paper has convergence issues."

To Overcome Divergence, ⋯

- Modify Adam
  - **AMSGrad, AdaFom** [Reddi et al.'18, Chen et al.'18]: keep $v_k \geq v_{k-1}$
    - ➢ Slow convergence [Zhou et al.' 18]
  - **AdaBound** [Luo et al.' 19]: Impose constraint: $v_k \in [C_l, C_u]$
    - ➢ Need to tune two extra hyperparameters

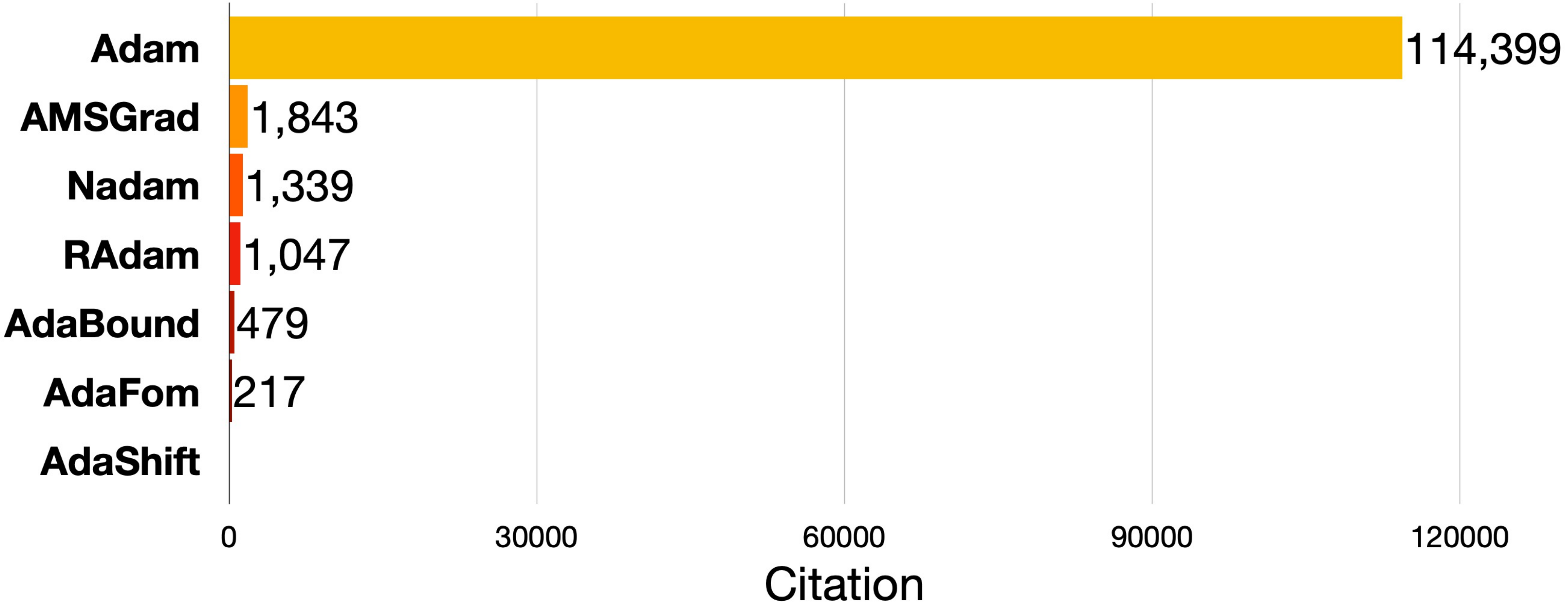
However, vanilla Adam works well for most practical applications!

Comparison: Adam vs its variants



- *Disclaimer: contribution is not proportional to citation.
  But citation might reflect the popularity among practitioners.

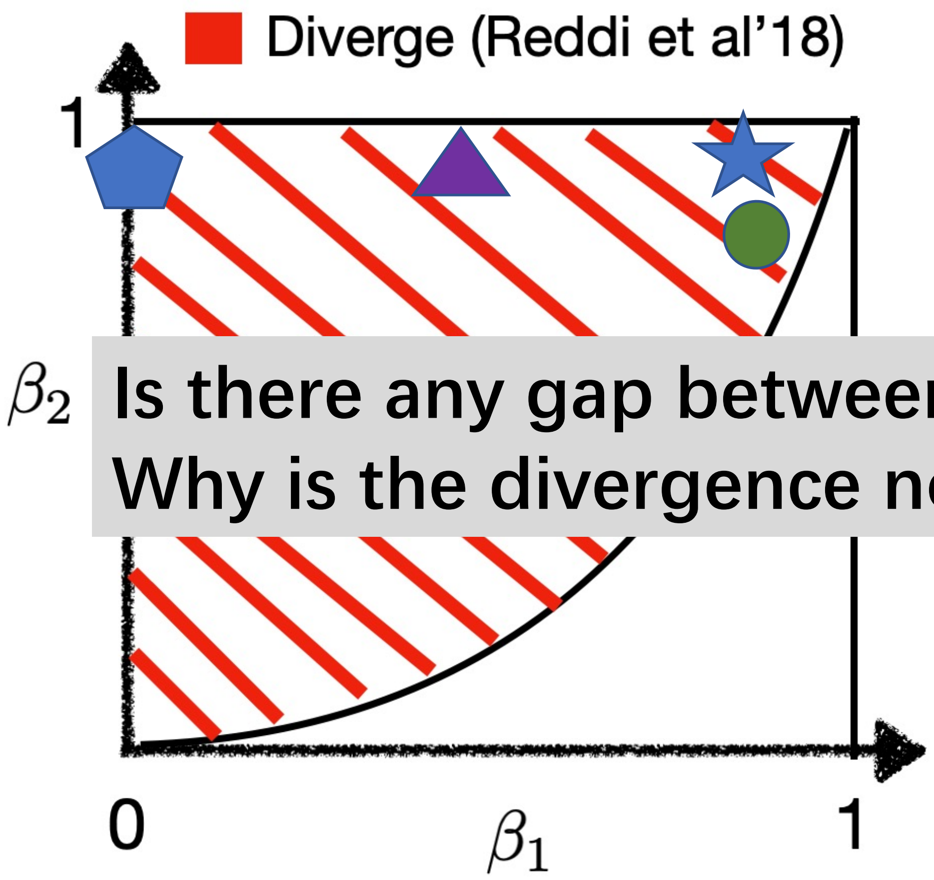
However, Adam remains overwhelmingly popular



- The attention Adam received is astonishing!
- Partially because many variants bring new issues (e.g., slow)

Divergence theory does not match practice

Observation: the reported (β_1, β_2) **actually satisfy divergence condition** $\beta_1 < \sqrt{\beta_2}$!



Is there any gap between theory and practice?
Why is the divergence not observed?

Most deep learning tasks
(e.g. RL, NLP, CV, GAN, etc.):
$\beta_1 = 0.9, \beta_2 = 0.999$

l, DCGAN, etc:
.999

uage models (e.g. GPT-3)
$\beta_1 = 0.9, \beta_2 = 0.95$

First-order GAN, MSG-GAN:
$\beta_1 = 0, \beta_2 = 0.999$

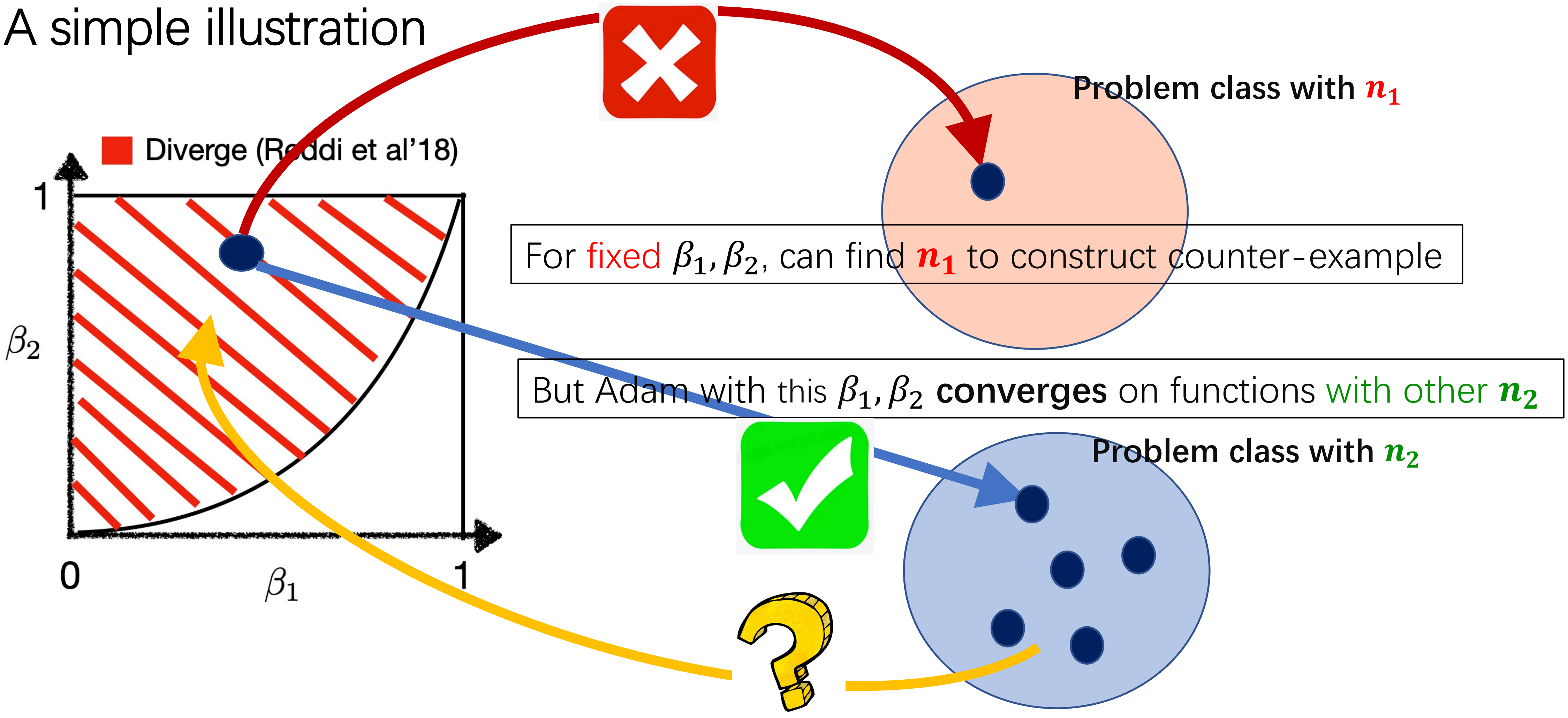
Why is divergence not observed?

- Reddi et al. 18 consider $\min_{x} f(x) := \sum_{i=1}^{n} f_i(x)$

Proof (Reddi et al. 18):
For any fixed β_1, β_2 s.t. $\beta_1 < \sqrt{\beta_2}$, we can find an n to construct the divergence example $f(x)$

- An important (but often ignored) feature: Reddi et al. fix β_1, β_2 **before picking the problem** (change n according to β_1, β_2)
- While in practice, parameters are often **problem-dependent** (e.g. tuning lr for GD)
- **Conjecture: Adam might converge for fixed problem (or fixed n)**

A simple illustration



Question: Does Adam converge for fixed problem class (fixed n)?

Contents

Assumptions

- Consider $\min_x f(x) \coloneqq \sum_{i=1}^{n} f_i(x)$
- **A1 (L-smooth):** assume $\nabla f_i(x)$ are L-Lipschitz continuous
- **A2 (Affine Variance):** $\frac{1}{n}\sum_{i=1}^{n} || \nabla f_i\,(\mathrm{x}) - \frac{1}{\mathrm{n}} \sum_{i=1}^{n} \nabla f_i(x)\,||_2^2 \leq D_1 || \nabla f\,(\mathrm{x}) ||_2^2 + D_0$
- **Remark:** A2 is quite general:
  - When $D_1 = 0$, **A2** becomes bounded variance, commonly used in SGD analysis
  - When $D_0 = 0$, **A2** becomes ``Strong Growth Condition (SGC)" [Vaswani et al., 19]

    -- Intuition: When $|| \nabla f\,(\mathrm{x}) ||$=0 $\Longrightarrow$ we have $|| \nabla f_i(\mathrm{x}) ||$=0.

    -- $\boldsymbol{D_0 = 0}$ **holds for overparametrized networks (Vaswani et al.19)**

- To our knowledge, **A1+ A2** are the mildest assumptions for stochastic opt algorithms **(we do not use bounded gradient assumption)**

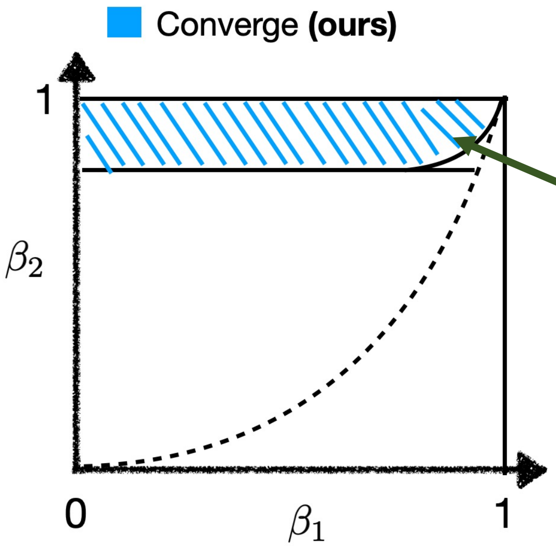
Convergence results for large $\boldsymbol{\beta_2}$

- **Theorem 1:** Consider the previous setting.
  When $\beta_2 \geq 1 - O\left(\frac{1-\beta_1^n}{n^{3.5}}\right)$ and $\beta_1 < \sqrt{\beta_2} < 1$, Adam with stepsize $\eta_k = \frac{1}{\sqrt{k}}$ converges to the neighborhood of stationary points:

$$\min_{k\in[1,T]} \mathbb{E}||\nabla f(x_k)||_2^2 = O\left(\frac{\log T}{(1-\beta_2)^2\sqrt{T}} + (1-\beta_2)D_0\right).$$

RK: When $D_0 = 0$ (e.g., for overparameterized models): Adam converges to stationary points

RK: Our result does not support $\beta_2 = 1$, so does not cover SGDM



We identify a safe region! **(UNKNOWN BEFORE!)**

Remark: Convergence to neighborhood

- When $D_0 > 0$: **converges to a neighborhood of stationary points** with size $O((1-\beta_2)D_0)$. (a.k.a. ``ambiguity zone”).
- **This** is common for

  --constant-step SGD [Yan et al., 2018; Yu et al., 2019]

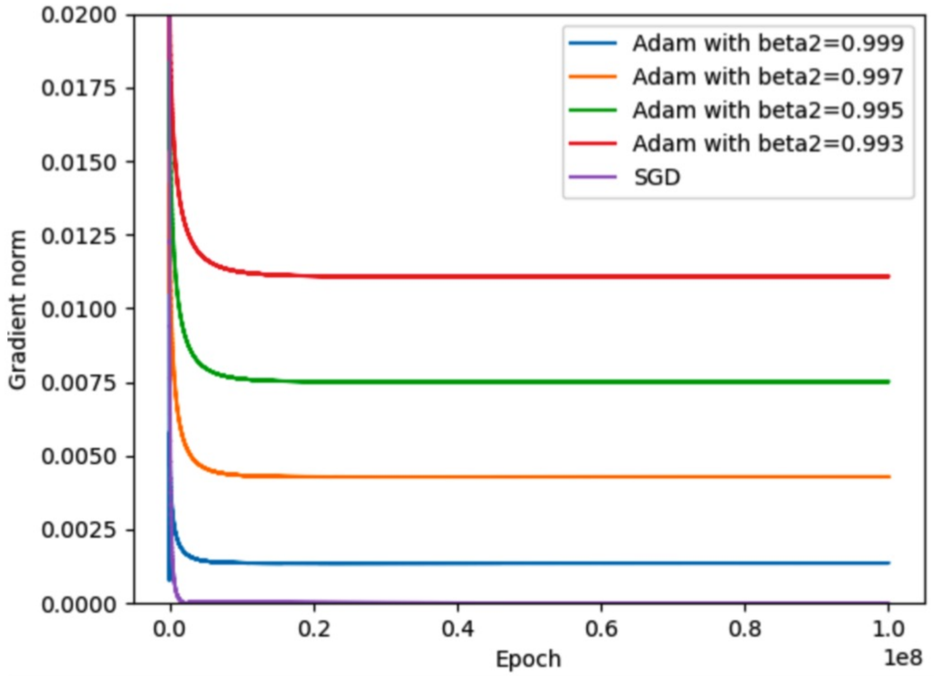
  --diminishing-lr adaptive gradient methods [Zaheer et al., 2018; Shi et al., 2020]:

$$x_{k+1} = x_k - \frac{\eta_k}{\sqrt{v_k}} m_k$$

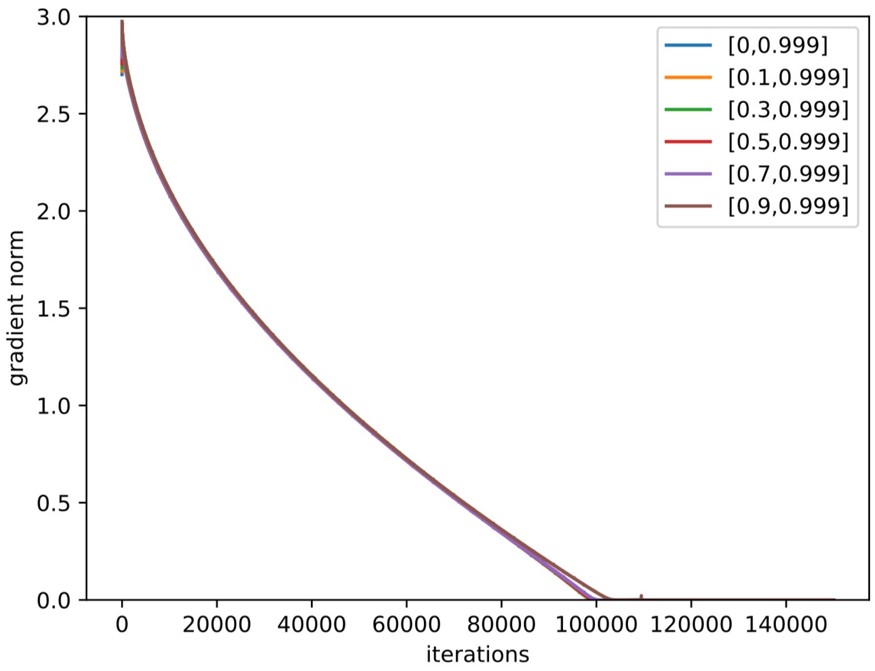
Intuition: Although η_k is diminishing, $\frac{\eta_k}{\sqrt{v_k}}$ may not decrease.

Remark: Convergence to neighborhood.

Left: A toy example with $D_0 > 0$



Right: A toy example with $D_0 = 0$



Setting: Adam & SGD with lr $\eta_k = \frac{1}{\sqrt{k}}$

Discussion: different meanings of convergence

$$\min_x f(x) := \sum_{i=1}^{n} f_i(x)$$

- **Pre-ML era:** n usually =1

  Meaning of Convergence:

  --Error term decays to 0 under certain rate (e.g., $\|\nabla f(x_k)\|^2 = O(\frac{1}{\sqrt{k}})$)

- **Post-ML era:** n usually >1, no access to the full gradient

  Meaning of convergence: only to the neighborhood of solution sets

  --For SGD: $\|\nabla f(x_k)\|^2 = O\left(\frac{1}{\sqrt{k}}\right) + O\left(\eta_k D_0\right)$

  --For Adam: $\|\nabla f(x_k)\|^2 = O\left(\frac{1}{\sqrt{k}}\right) + O\left((1-\beta_2)D_0\right)$

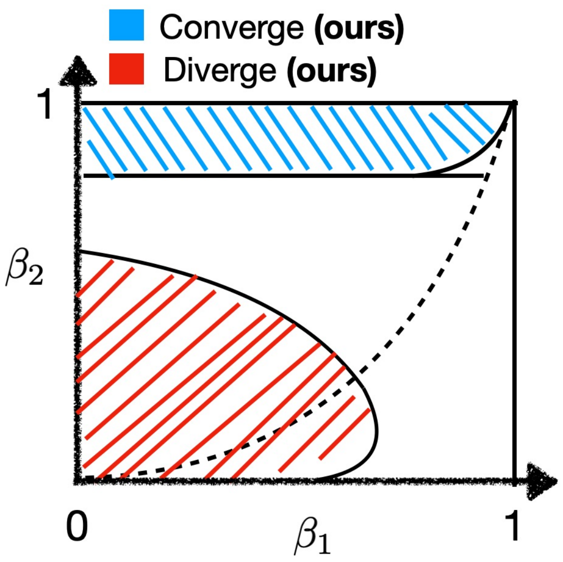
  → **Error floor!**

- The error floor might be acceptable because:
  -- $D_0 = 0$ for over-parameterized DNN (Vaswani et al.19)

  -- $\beta_2 \sim 0.999$ in practice, so the error is small

How does Adam behave in the rest of the region?

- When β_2 is large: we have shown a positive result.

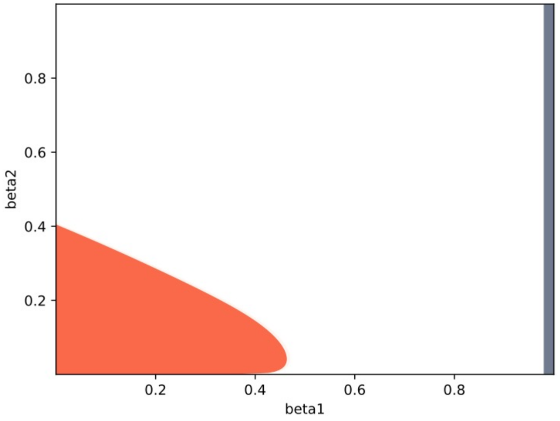


- When β_2 is small: we will show that Adam can still diverge! (even if the problem class is fixed)

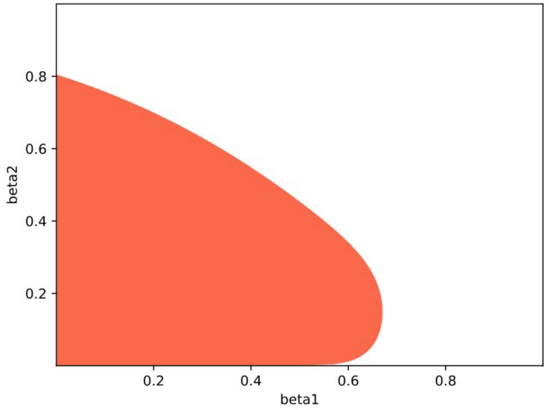
Divergence can happen when $\boldsymbol{\beta_2}$ is small

- **Thm 2:** For any fixed n, there exists a function $f(x)$ satisfying **A1** and **A2**, s.t. when (β_1, β_2) are chosen in the orange region (size depends on n), s.t. Adam's iterates and function values diverge to infinity

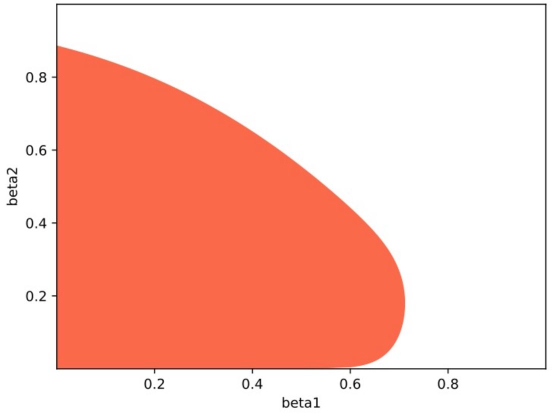
  - The region is **precisely drawn** (plotted by solving some analytic conditions)
  - region enlarges with n



(b) $n = 10$



(c) $n = 50$



(d) $n = 100$

Implication to practitioners

- **Case study:** Bob is using Adam to train NNs. However, Adam with default hyperparameter fails in his tasks.
- Bob heard there is a well-known result that Adam can diverge.
- So he wonders: shall I keep tuning hyperparameter to make it work?
- Or shall I just give up and switch to other algorithms like AdaBound (which has 2 extra hyperparameters)?

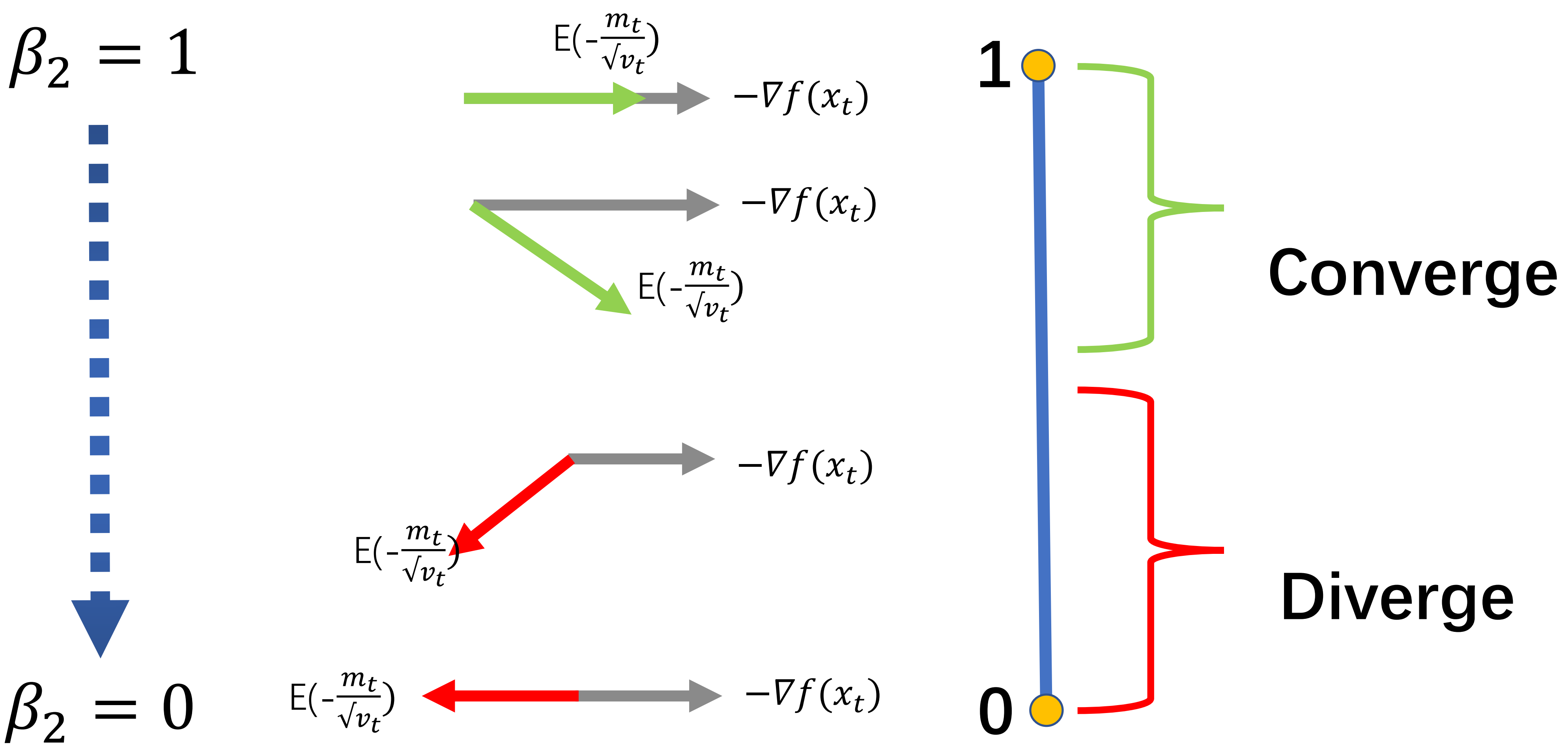
Our suggestions:

1. Adam is still a theoretically justified algorithm. **Please use it confidently!**
2. Suggestions for hyperparameter tunning:
   In one sentence: First, tune up β_2. Then, try different β_1 with $\beta_1 < \sqrt{\beta_2}$

Contents

Intuition behind convergence and divergence

Adam: $\boldsymbol{x^{t+1} = x^{t} - \eta_t \frac{m_t}{\sqrt{v_t}}}$



Proof Ideas for Convergence Analysis: An Overview

Want to show: $\mathbb{E}\left\langle \nabla f(x_k), \frac{m_k}{\sqrt{v_k}} \right\rangle = \mathbb{E}\left\langle \nabla f(x_k), \frac{(1-\beta_1)\nabla f_{\tau_k}(x_k)+\beta_1 m_{k-1}}{\sqrt{(1-\beta_2)\nabla f_{\tau_k}(x_k)\circ\nabla f_{\tau_k}(x_k)+\beta_2 v_{k-1}}} \right\rangle > 0$

Goal: want to prove: $\mathbb{E}\langle \nabla f(x_k), \frac{m_k}{\sqrt{v_k}} \rangle > constant * \mathbb{E}\langle \nabla f(x_k), \nabla f(x_k)\rangle > 0$

Major challenge 1: $\sqrt{v_k}$ appears in the denominator, may blow up.

Major challenge 2: momentum m_k contains history information.

Major challenge 3: both m_k and $\sqrt{v_k}$ are random

Solutions:

Step 1: $\mathbb{E}\left\langle \nabla f(x_k), \frac{m_k}{\sqrt{v_k}} \right\rangle = \mathbb{E}\left\langle \frac{\nabla f(x_k)}{\sqrt{v_k}}, m_k \right\rangle \approx \mathbb{E}\left\langle \frac{\nabla f(x_k)}{\sqrt{v_k}}, \nabla f(x_k) \right\rangle$ **(80% of the proof)**

Step 1-1: prove $\mathbb{E}(m_k) \approx \mathbb{E}(\nabla f(x_k))$ (to get idea and intuition)

Step 1-2: prove $\mathbb{E}\langle \frac{\nabla f(x_k)}{\sqrt{v_k}}, m_k \rangle \approx \mathbb{E}\langle \frac{\nabla f(x_k)}{\sqrt{v_k}}, \nabla f(x_k) \rangle$ (main part of Step 1)

Step 2: $\mathbb{E}\langle \nabla f(x_k), \frac{\nabla f(x_k)}{\sqrt{v_k}} \rangle \geq constant * \mathbb{E}\langle \nabla f(x_k), \nabla f(x_k)\rangle > 0$ **(20% of the proof)**

Step 1.1:Want to show: $\mathbb{E}(\nabla f(x_k) - m_k) \approx 0$

- **Want To Show:** $\mathbb{E}(\nabla f(x_k) - m_k) \approx 0$
- **What is the math problem here? Estimate difference of two sum' s.**

- Understanding Step (i): **Full-Batch case with n = 1**

  $\nabla f(x_k) = (1 - \beta_1)\,(\nabla f(x_k) + \beta_1 \nabla f(x_k) + \ \ldots \beta_1^{k-1} \nabla f(x_k))$

  m_k = weighted average of past gradients $=(1 - \beta_1)\,(\nabla f(x_k) + \beta_1 \nabla f(x_{k-1}) +$
 $\ldots \beta_1^{k-1} \nabla f(x_1))$

- **Math Problem:** Comparing weighted average over history v.g. current gradient
- **Traditional Solution:**
  analyze the spectrum of asymmetric update matrix (linear-algebra perspective)
  & construct potential function (optimization perspective)

Step 1.1:Want to show: $\mathbb{E}(\nabla f(x_k) - m_k) \approx 0$

- **Want To Show:** $\mathbb{E}(\nabla f(x_k) - m_k) \approx 0$
- **What is the math problem here? Estimate difference of two sum' s**

- Understanding Step (ii): **Stochastic case n =2**
- **More precisely:** need to compare the following difference per epoch:
- $\mathbb{E}(\nabla f(x_{k,0}) - (m_{k,0} + m_{k,1})) = \mathbb{E}(g_0(x_{k,0}) + g_1(x_{k,0}) - (m_{k,0} + m_{k,1})) \approx 0$

$$m_{k,0} = (1-\beta_1)(\, g_{k,0}(x_{k,0}) + \beta_1 g_{k-1,1}(x_{k-1,1}) + \beta_1^2 g_{k-1,0}(x_{k-1,0}) \ldots \beta_1^{2(k-1)-1} g_{1,1}(x_{1,1}) + \beta_1^{2(k-1)} g_{1,0}(x_{1,0}))$$

$$m_{k,1} = (1-\beta_1)(\, g_{k,1}(x_{k,1}) + \beta_1 g_{k,0}(x_{k,0}) + \beta_1^2 g_{k-1,1}(x_{k-1,1}) \ldots \beta_1^{2(k-1)} g_{1,1}(x_{1,1}) + \beta_1^{2(k-1)+1} g_{1,0}(x_{1,0}))$$

- **Math problem:** comparison of two "co-" growing exponentially-averaged sum' s
- **Our idea:** Find certain intrinsic properties of these sum' s under random permutation

Step 1.1: $\mathbb{E}(\nabla f(x_k) - m_k) \approx 0$, An overview

- **Want To Show:** $\mathbb{E}(\nabla f(x_k) - m_k) \approx 0$
- **Solution:** construct a simplified system by **assuming $\boldsymbol{x}$ fixed**
- **Color-ball of the 1st kind:** consider a box contains two balls labeled c_0 and c_1.
  In each round (epoch), we randomly sample balls from the box without replacement, then we put them both back.
  We denote the 1st sampled label in the k-th epoch as a_k and the 2nd sampled one as b_k.
- We define the following quantities: (These mimic momentum, but with fixed x)

$$m_1 := \underbrace{b_k + \beta_1 a_k}_{m_{1,k}} + \underbrace{\beta_1^2 b_{k-1} + \beta_1^3 a_{k-1}}_{m_{1,k-1}} + \cdots + \underbrace{\beta_1^{2(k-1)} b_1 + \beta_1^{2(k-1)+1} a_1}_{m_{1,1}}$$

$$m_0 := \underbrace{a_k}_{m_{0,k}} + \underbrace{\beta_1^1 b_{k-1} + \beta_1^2 a_{k-1}}_{m_{0,k-1}} + \cdots + \underbrace{\beta_1^{2(k-1)-1} b_1 + \beta_1^{2(k-1)} a_1}_{m_{0,1}}$$

(Notation: $m_{i,k}$ denotes the partial-sum of m_i in the k-th epoch)

Step 1.1: $\mathbb{E}(\nabla f(x_k) - m_k) \approx 0$, An overview

- **Want To Show:** $\mathbb{E}(\nabla f(x_k) - m_k) \approx 0$
- **Step 1-1 a):** construct a simplified system by **assuming x fixed**
- **Color-ball of the 1st kind:** We further define the following quantities: (These mimic gradient)

$$f_1 = c_1(\underbrace{1+\beta_1}_{f_{1,k}} + \underbrace{\beta_1^2+\beta_1^3}_{f_{1,k-1}} \cdots + \underbrace{\beta_1^{2(k-1)} + \beta_1^{2(k-1)+1}}_{f_{1,1}})$$

$$f_0 = c_0(\underbrace{1+\beta_1}_{f_{0,k}} + \underbrace{\beta_1^2+\beta_1^3}_{f_{0,k-1}} \cdots + \underbrace{\beta_1^{2(k-1)} + \beta_1^{2(k-1)+1}}_{f_{0,1}})$$

Lemma A.1. *In the color-ball model of the 1st kind, we have*

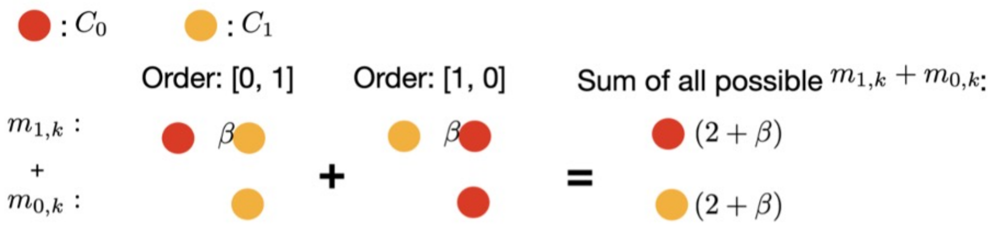
$$\left|\mathbb{E}\left[\sum_{i=0}^{1} m_i - \sum_{i=0}^{1} f_i\right]\right| = \boxed{\beta^{2(k-1)+1}}\left(\frac{c_0}{2} + \frac{c_1}{2}\right),$$

Key finding:
No low-order terms!

Due to random permutation.

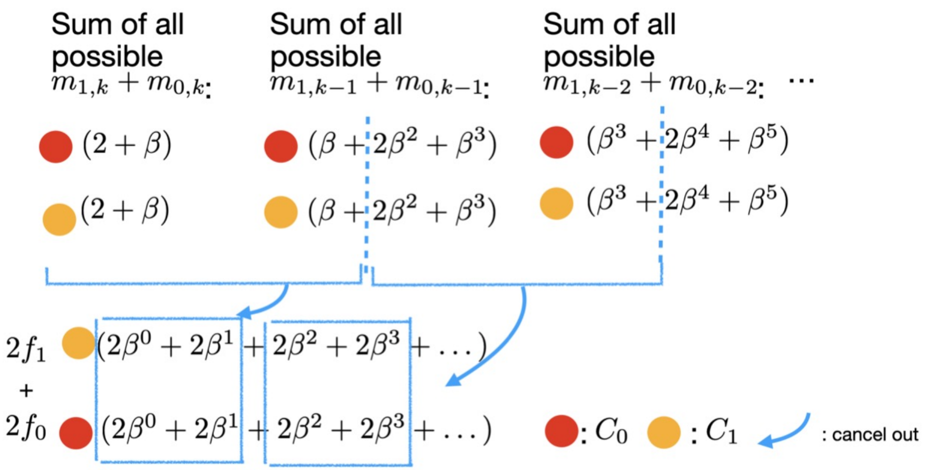
Step 1.1: $\mathbb{E}(\nabla f(x_k) - m_k) \approx 0$, An overview

Step 1: Take conditional expectation up to k-th epoch, calculate the partial sum $\mathbb{E}_k[\sum_i (m_{i,k} - f_{i,k})]$



Step 2: We move one step further to calculate

$$\mathbb{E}_{k-1}\mathbb{E}_k[\textstyle\sum_i (m_{i,k} - f_{i,k}) + [\sum_i (m_{i,k-1} - f_{i,k-1})]]$$



$$
\begin{aligned}
E_k\left[\sum_{i=0}^{1} m_{i,k} - \sum_{i=0}^{1} f_{i,k}\right] &= E_k\left[\sum_{i=0}^{1} m_{i,k}\right] - (1+\beta_1)(c_0 + c_1) \\
&= -\frac{\beta_1}{2}(c_0 + c_1)
\end{aligned}
$$

Observe repeated cancelation! We observe that only the highest order term remains in the calculation! Repeat this process until k=1. We complete the proof of **Lemma A.1**

Step 1.1: $\mathbb{E}(\nabla f(x_k) - m_k) \approx 0$, continued

- What we did so far: **Step 1-1 (a):** assume x fixed, find certain property

Lemma A.1. *In the color-ball model of the 1st kind, we have*

$$\mathbb{E}\left[\sum_{i=0}^{1} m_i - \sum_{i=0}^{1} f_i\right] = \beta_1^{2(k-1)+1}\left(-\frac{c_0}{2} - \frac{c_1}{2}\right)$$

- Continue **Step 1-1 (b):** consider x changing

$\mathbb{E}(\nabla f(x_k) - m_k) \xrightarrow{(1)\,(2)\,(3)} \mathbb{E}(\nabla f(x_k) - m_k)$ with "fixed x" $\xrightarrow{\text{Lemma A.1}} O(\beta_2^k) \rightarrow 0$

*: (1) Bounded Update Rule of Adam (2) diminishing stepsize (3) Lipschitz condition.

Combined we have $|g(x_k) - g(x_{k-1})| = O(1/\sqrt{k})$

(1) (2) (3) can only be applied to Adam, not SGD

Step 1.2: $\mathbb{E}\langle\frac{\nabla f_k}{\sqrt{v_k}}, m_k - \nabla f_k\rangle \approx 0$, an overview

- **Recall our goal:** $\mathbb{E}\langle\nabla f_k, \frac{m_k}{\sqrt{v_k}}\rangle > 0$

Greater than 0 Still unclear

- **Simple decomposition:** $\mathbb{E}\langle\nabla f_k, \frac{m_k}{\sqrt{v_k}}\rangle = \boxed{\mathbb{E}\left\langle\nabla f_k, \frac{\nabla f_k}{\sqrt{v_k}}\right\rangle} + \boxed{\mathbb{E}\left\langle\nabla f_k, \frac{\nabla f_k}{\sqrt{v_k}} - \frac{m_k}{\sqrt{v_k}}\right\rangle}$

- **Recall In Step 1-1, we have shown:** $\mathbb{E}(\nabla f(x_k) - m_k) \approx 0$
- **Now In Step 1-2, We will show:** $\mathbb{E}\left\langle\nabla f_k, \frac{g_k}{\sqrt{v_k}} - \frac{m_k}{\sqrt{v_k}}\right\rangle = \mathbb{E}\left\langle\frac{\nabla f_k}{\sqrt{v_k}}, g_k - m_k\right\rangle \approx 0$

  - **Idea in Step 1-2**: 1) control the movement of $\frac{\nabla f_k}{\sqrt{v_k}}$

    2) Extend the proof in Step 1-1

Step 1.2: $\mathbb{E}\langle \frac{\nabla f_k}{\sqrt{v_k}}, m_k - \nabla f_k \rangle \approx \mathbf{0}$, an overview

- **Want to show:** $\mathbb{E}\langle \frac{\nabla f_k}{\sqrt{v_k}}, m_k - \nabla f_k \rangle \approx 0$ when β_2 is large
- **Step I: Show that** $\| \frac{\nabla f_k}{\sqrt{v_k}} - \frac{\nabla f_{k-1}}{\sqrt{v_{k-1}}} \| = O\left(\frac{1}{\sqrt{k}}\right)$ when β_2 is large (requires several technical lemmas, omitted here)
- **Step II:** we construct another color-ball model
- **Color-ball of the 2nd kind:** Consider the same setting as the previous color ball. We further introduce a new seq of r.v. $\{r_j\}$ s.t. r_j is fixed when fixing the history up to j-th round and:

$$|r_j - r_{j-1}| = \frac{1}{\sqrt{j}}, \quad j = 1, \cdots k.$$

Now we define the following quantities:

$$r_k m_1 = r_k \left(\underbrace{b_k + \beta a_k}_{m_{1,k}} + \underbrace{\beta^2 b_{k-1} + \beta^3 a_{k-1}}_{m_{1,k-1}} + \cdots + \underbrace{\beta^{2(k-1)} b_1 + \beta^{2(k-1)+1} a_1}_{m_{1,1}} \right);$$

$$r_k m_0 = r_k \left(\underbrace{\qquad a_k}_{m_{0,k}} + \underbrace{\beta^1 b_{k-1} + \beta^2 a_{k-1}}_{m_{0,k-1}} + \cdots + \underbrace{\beta^{2(k-1)-1} b_1 + \beta^{2(k-1)} a_1}_{m_{1,1}} \right);$$

Step 1.2: $\mathbb{E}\langle \frac{\nabla f_k}{\sqrt{v_k}}, m_k - \nabla f_k \rangle \approx 0$, an overview

- **Step II:** we construct another color-ball model
- **Color-ball of the 2nd kind:** Now we define the following quantities:

$$r_k f_1 = r_k \left(c_1 \big(\underbrace{1 + \beta_1}_{f_{1,k}} + \underbrace{\beta_1^2 + \beta^3}_{f_{1,k-1}} \cdots + \underbrace{\beta_1^{2(k-1)} + \beta_1^{2(k-1)+1}}_{f_{1,1}} \big) \right)$$

$$r_k f_0 = r_k \left(c_0 \big(\underbrace{1 + \beta_1}_{f_{0,k}} + \underbrace{\beta_1^2 + \beta^3}_{f_{0,k-1}} \cdots + \underbrace{\beta_1^{2(k-1)} + \beta_1^{2(k-1)+1}}_{f_{0,1}} \big) \right)$$

Lemma A.2. *Consider the color-ball model of the 2nd kind, we have*

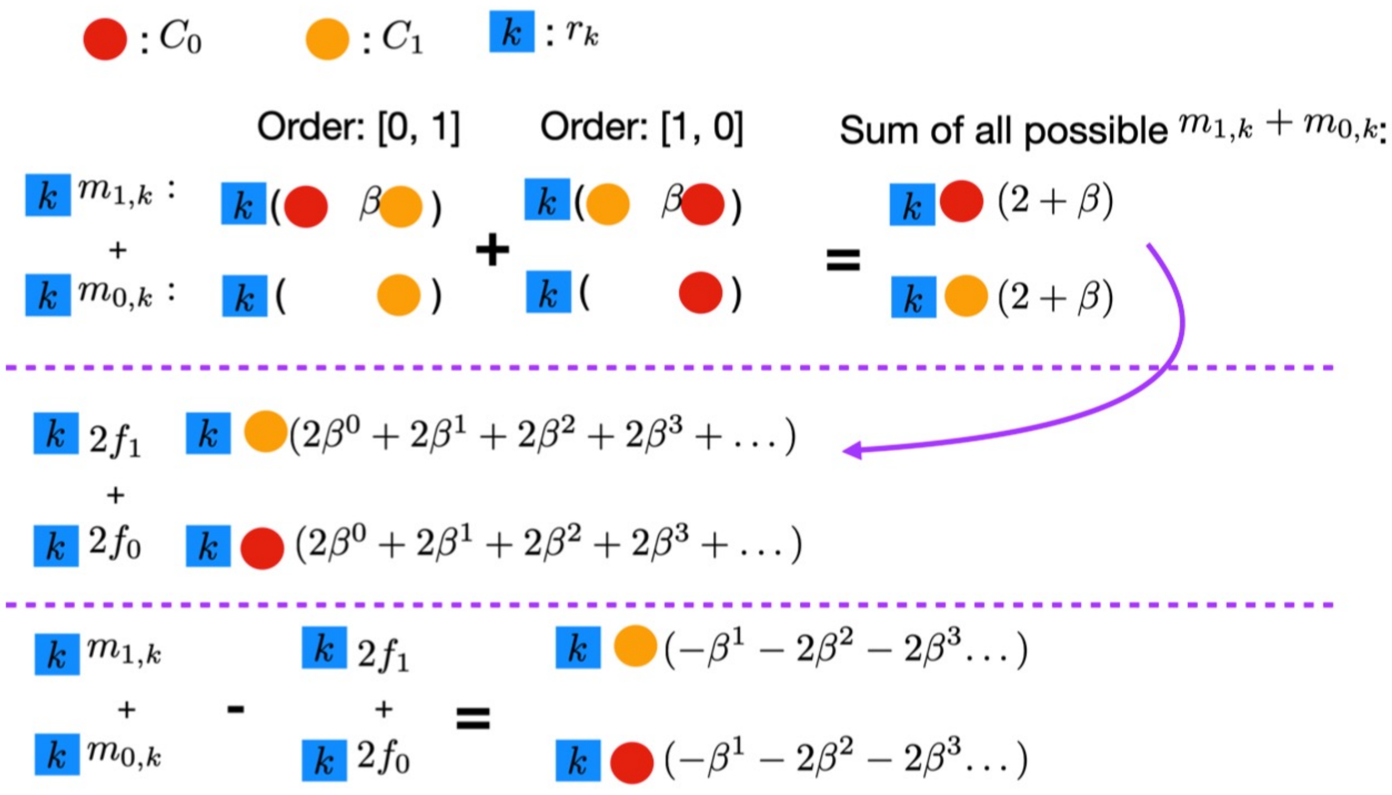
$$\mathbb{E}\left[\sum_{i=0}^{1} r_k m_i - \sum_{i=0}^{1} r_k f_i\right] = \boxed{\beta_1^{2(k-1)+1}\left(-\frac{c_0}{2} - \frac{c_1}{2}\right)} + \boxed{\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)}$$

Controllable error

Same as in Step 1-1

Step 1.2: $\mathbb{E}\langle\frac{\nabla f_k}{\sqrt{v_k}}, m_k - \nabla f_k\rangle \approx \mathbf{0}$, an overview

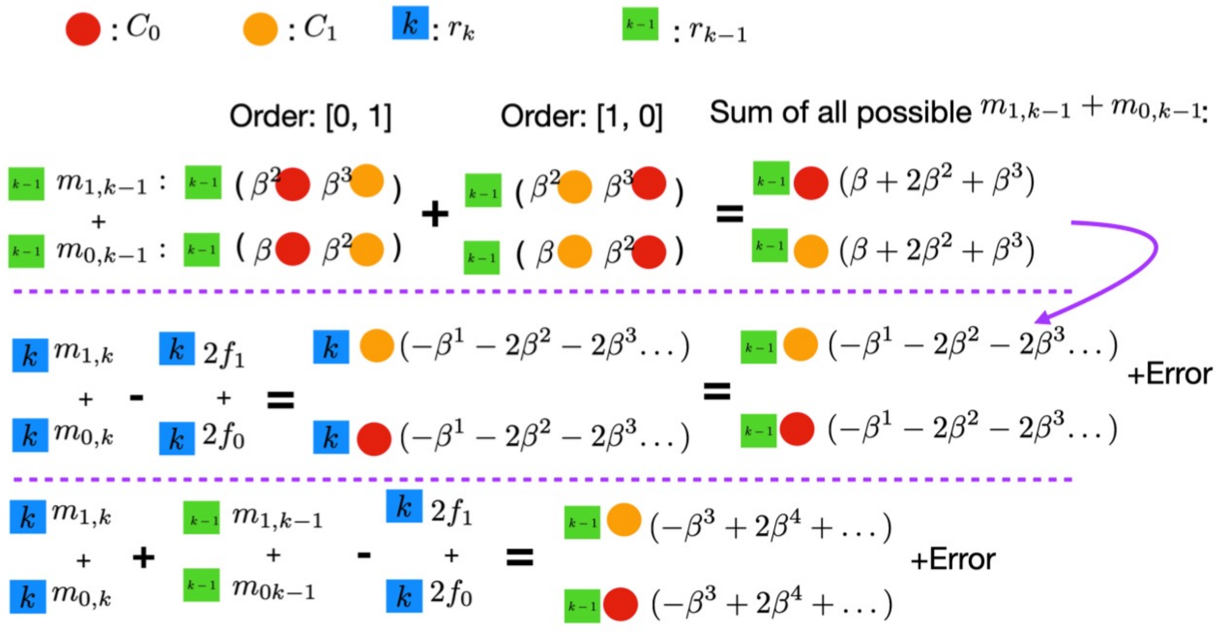
- We introduce 4 steps to prove Lemma A.2:
- **Step 1:** Take conditional exp and calculate $\mathbb{E}_k\left[r_k \sum_{i=0}^{1} m_{i,k} - r_k \sum_{i=0}^{1} f_i\right]$



Step 1.2: $\mathbb{E}\langle \frac{\nabla f_k}{\sqrt{v_k}}, m_k - \nabla f_k \rangle \approx 0$, an overview

- **Step 2:** change $\mathbb{E}_k\left[r_k \sum_{i=0}^{1} m_{i,k-1}\right]$ into $\mathbb{E}_k\left[r_{k-1} \sum_{i=0}^{1} m_{i,k-1}\right] + \textbf{Error}$

  where Error = $O\,(1/\sqrt{k})$
- **Step 3:** Take conditional exp : $\mathbb{E}_{k-1}\mathbb{E}_k\left[r_{k-1} \sum_{i=0}^{1} m_{i,k-1}\right] = \mathbb{E}_{k-1}\left[r_{k-1} \sum_{i=0}^{1} m_{i,k-1}\right]$
- **Step 4:** For the leftovers in Step 1: change all r_k into r_{k-1}



Repeat this process to the 1st epoch.
Complete the proof of Lemma A.2

Recap of the whole proof

Goal: want to prove: $\mathbb{E}\langle \nabla f(x_k), \frac{m_k}{\sqrt{v_k}} \rangle > constant * \ \mathbb{E}\langle \nabla f(x_k), \nabla f(x_k) \rangle > 0$

Preparation: $\mathbb{E}(m_k) \approx \mathbb{E}(\nabla f(x_k))$

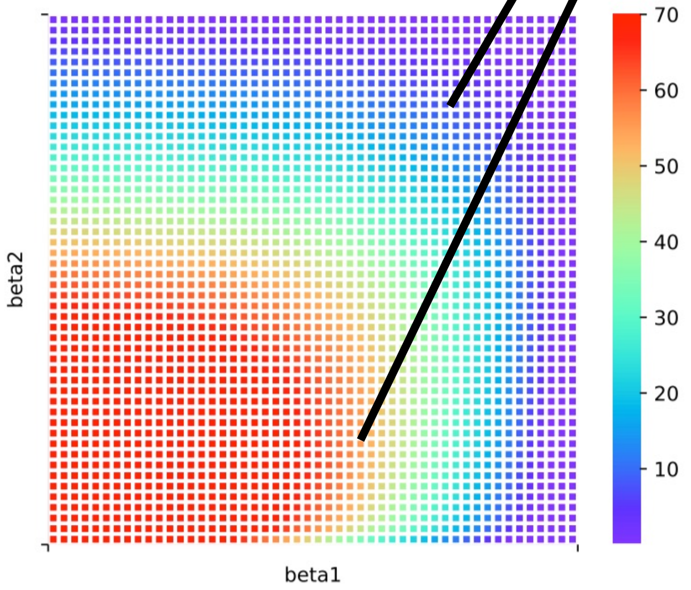
Step 1 (main part of the proof): $\mathbb{E}\langle \frac{\nabla f(x_k)}{\sqrt{v_k}}, m_k \rangle \approx \mathbb{E}\langle \frac{\nabla f(x_k)}{\sqrt{v_k}}, \nabla f(x_k) \rangle$

Step 2: $\mathbb{E}\langle \frac{\nabla f(x_k)}{\sqrt{v_k}}, \nabla f(x_k) \rangle \geq constant * \ \mathbb{E}\langle \nabla f(x_k), \nabla f(x_k) \rangle > 0$

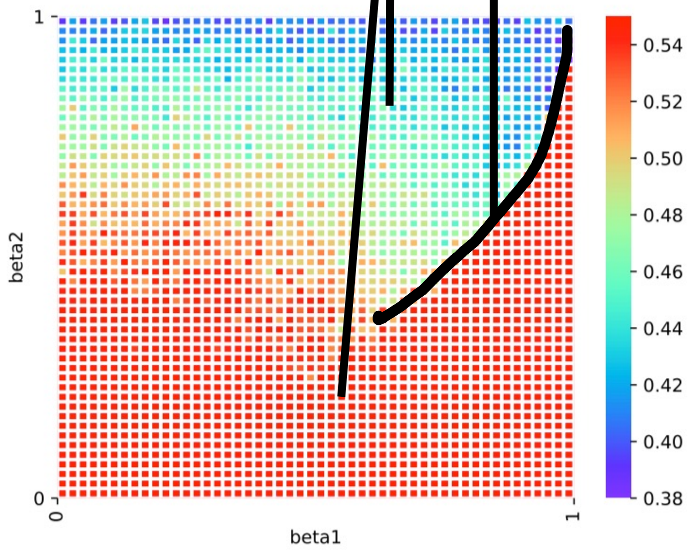
Contents

Our theory is consistent with experiments

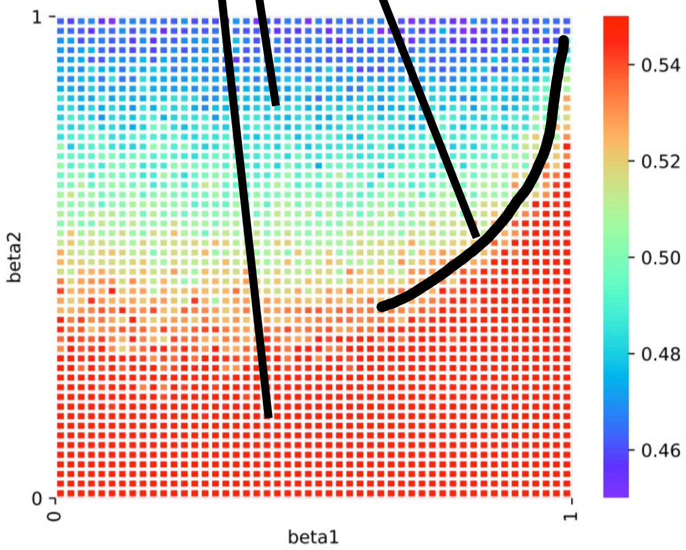
Optim Optimization error is Smooth boundaries ion



(a) Function (2)

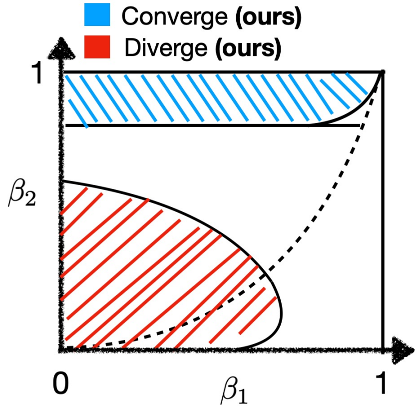


(b) MNIST



(c) CIFAR-10

Recipe for Adam hyperparameter-tuning



AdaShift? AdaBound?

No need!

Change algorithm?

Adam with default setting (β_1, β_2) = (0.9, 0.999) **fails** in your task

Tune up β_2.
Try 0.9995, 0.9999, 0.99995, etc

Does not work better?

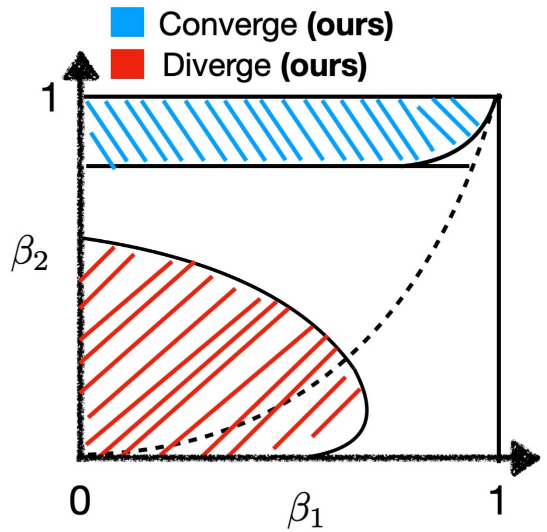
Try smaller β_1.
Try 0.7, 0.5, 0.3, 0.1, etc

Adam with default setting (β_1, β_2) = (0.9, 0.999) **works** in your task, but want better performance

Tune down β_2 a little bit
Try 0.995, 0.99, 0.95, etc

Does not work better?

Summary: the behavior of Adam changes dramatically under different hyperparameters



When increasing β_2:
There is a phase transition from divergence to convergence.

Setting	Hyperparameters	Adam's behavior
$\forall f(x)$ under **A1** and **A2** with $D_0 = 0$	β_2 is large and $\beta_1 < \sqrt{\beta_2}$	Converges to stationary points **(Ours)**
$\forall f(x)$ under **A1** and **A2** with $D_0 > 0$	β_2 is large and $\beta_1 < \sqrt{\beta_2}$	Converges to the neighborhood of stationary points **(Ours)**
$\exists\ f(x)$ under **A1** and **A2**	β_2 is small and a wide range of β_1	Diverges to infinity **(Ours)**

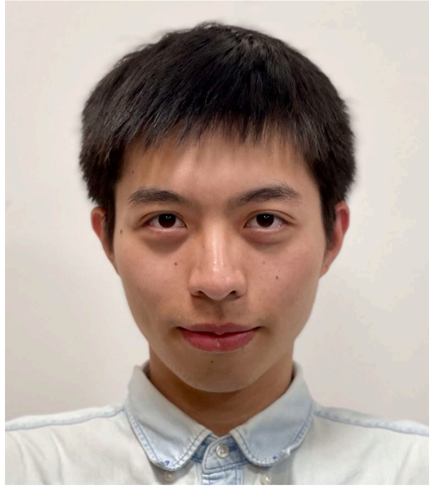
Our work is tweeted by **Dr. Kingma** (1st author of Adam)

Dr. Durk Kingma: inventor of Adam and VAE; co-founder of OpenAI; now a leader of Google Brain

Mainly based on:

- **Zhang**, Chen, Shi, Sun, & Luo, Adam can converge without any modification on update rules. *NeurIPS 2022*
- **Thanks to all the collaborators!**





Thank You!