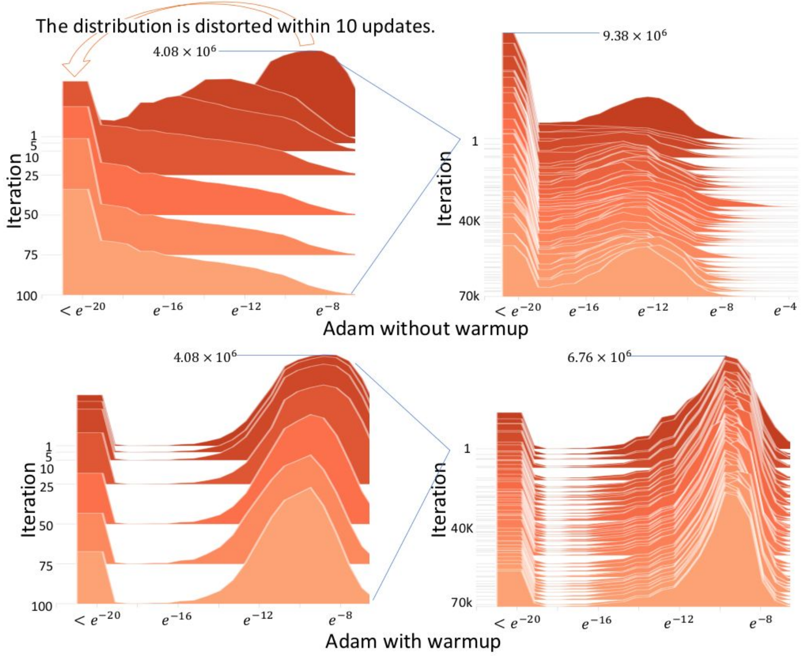
RAdam - Rectified Adam

- learning rate warmup stabilizes training, accelerates convergence and improves generalization

- problem of adaptive learning rates:
  - problematically large variance in the early stage of training

  → *suggests warmup works as a variance reduction technique!*

Absolute gradient histogram:



- RAdam introduces a term to rectify the variance of the adaptive learning rate!

https://arxiv.org/pdf/1908.03265 & https://medium.com/@lessw/new-state-of-the-art-ai-optimizer-rectified-adam-radam-5d854730807b

Vienna Deep Learning Meetup

RAdam - Update

Algorithm 2: Rectified Adam. All operations are element-wise.

Input: $\{\alpha_t\}_{t=1}^{T}$: step size, $\{\beta_1, \beta_2\}$: decay rate to calculate moving average and moving 2nd moment, θ_0: initial parameter, $f_t(\theta)$: stochastic objective function.

Output: θ_t: resulting parameters

$m_0, v_0 \leftarrow 0, 0$ (Initialize moving 1st and 2nd moment)

$\rho_\infty \leftarrow 2/(1-\beta_2) - 1$ (Compute the maximum length of the approximated SMA)

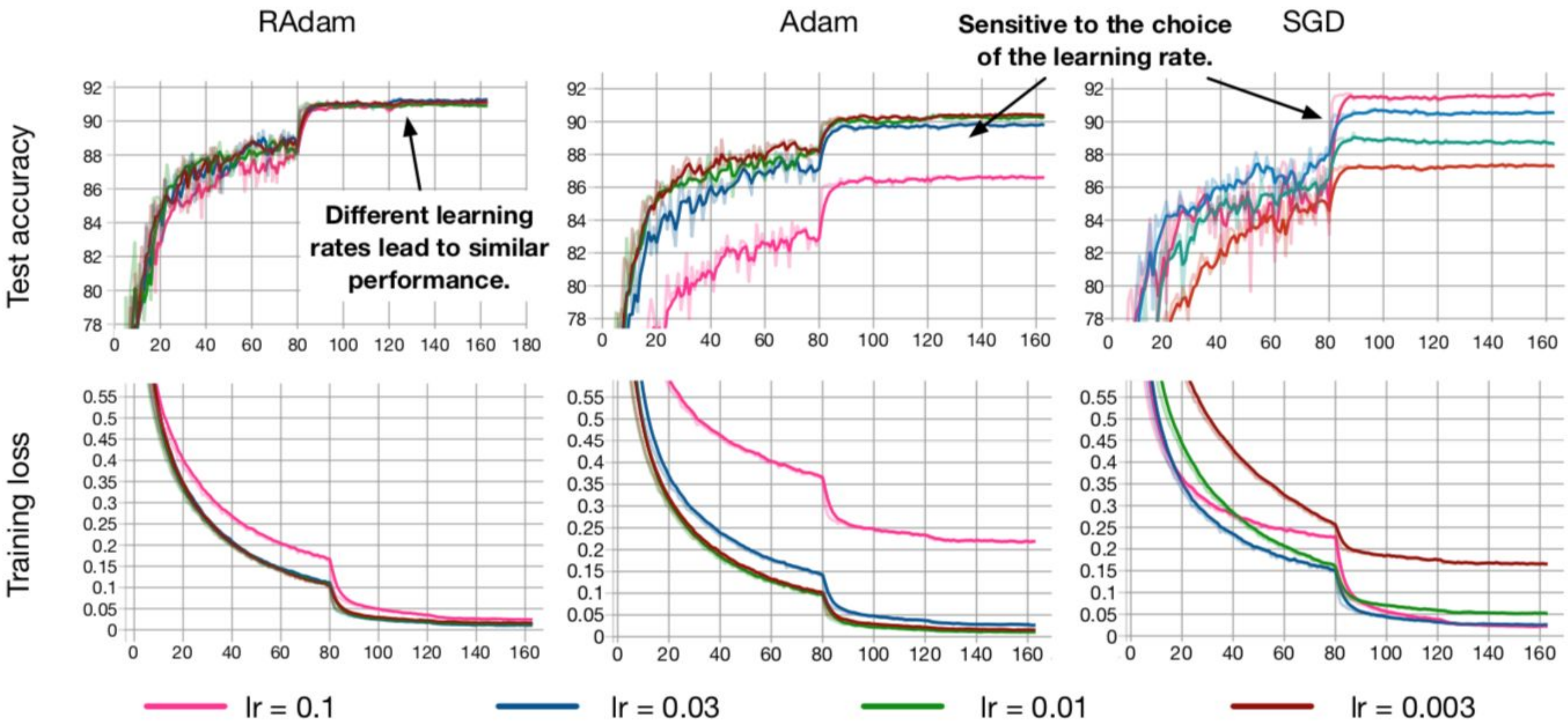
while $t = \{1, \cdots, T\}$ **do**

- $g_t \leftarrow \Delta_\theta f_t(\theta_{t-1})$ (Calculate gradients w.r.t. stochastic objective at timestep t)
- $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$ (Update exponential moving 2nd moment)
- $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$ (Update exponential moving 1st moment)
- $\widehat{m_t} \leftarrow m_t/(1-\beta_1^t)$ (Compute bias-corrected moving average)
- $\rho_t \leftarrow \rho_\infty - 2t\beta_2^t/(1-\beta_2^t)$(Compute the length of the approximated SMA)
- **if** *the variance is tractable, i.e.,* $\rho_t > 4$ **then**
  - $\widehat{v_t} \leftarrow \sqrt{v_t/(1-\beta_2^t)}$ (Compute bias-corrected moving 2nd moment)
  - $r_t \leftarrow \sqrt{\frac{(\rho_t-4)(\rho_t-2)\rho_\infty}{(\rho_\infty-4)(\rho_\infty-2)\rho_t}}$ (Compute the variance rectification term)
  - $\theta_t \leftarrow \theta_{t-1} - \alpha_t r_t \widehat{m_t}/\widehat{v_t}$ (Update parameters with adaptive momentum)
- **else**
  - $\theta_t \leftarrow \theta_{t-1} - \alpha_t \widehat{m_t}$ (Update parameters with un-adapted momentum)

return θ_T

https://arxiv.org/pdf/1908.03265

RAdam, Adam, and SGD

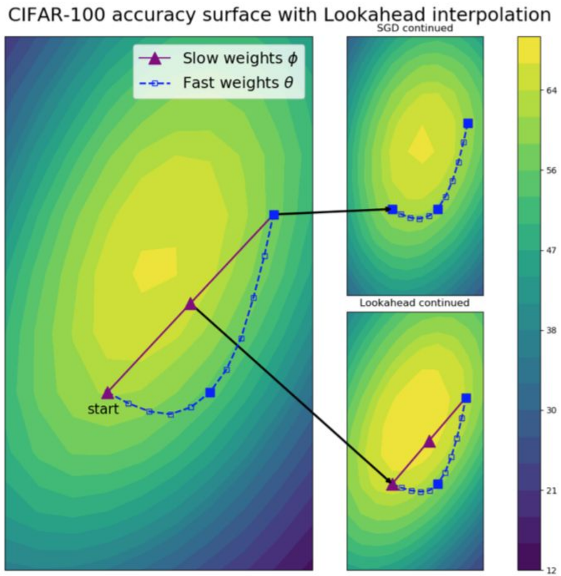


https://arxiv.org/pdf/1908.03265

Vienna Deep Learning Meetup

LookAhead

- Iteratively updates two sets of weights:
  - “slow weights” get updated by looking ahead at the sequence of “fast weights” generated by another optimizer

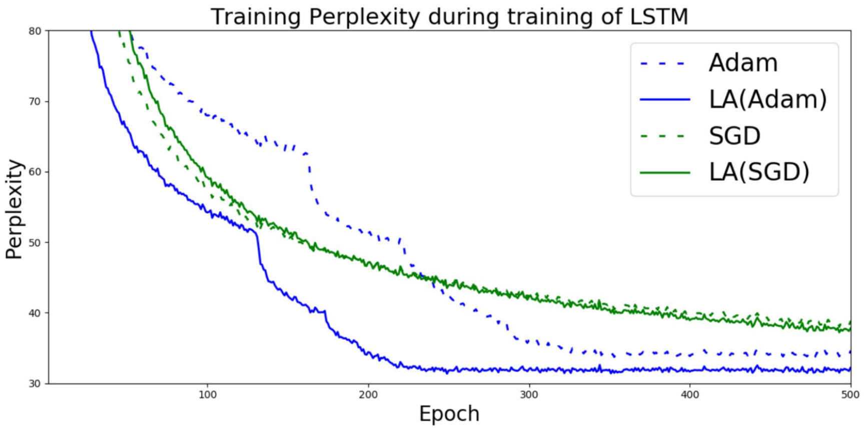


Algorithm 1 Lookahead Optimizer:

Require: Initial parameters ϕ_0, objective function L
Require: Synchronization period k, slow weights step size α, optimizer A
  for $t = 1, 2, \ldots$ **do**
    Synchronize parameters $\theta_{t,0} \leftarrow \phi_{t-1}$
    for $i = 1, 2, \ldots, k$ **do**
      sample minibatch of data $d \sim \mathcal{D}$
      $\theta_{t,i} \leftarrow \theta_{t,i-1} + A(L, \theta_{t,i-1}, d)$
    end for
    Perform outer update $\phi_t \leftarrow \phi_{t-1} + \alpha(\theta_{t,k} - \phi_{t-1})$
  end for
  return parameters ϕ

https://arxiv.org/pdf/1907.08610.pdf

LookAhead + ?



A better combination?

→ Ranger = LookAhead + RAdam

https://arxiv.org/pdf/1907.08610.pdf &
https://medium.com/@lessw/new-deep-learning-optimizer-ranger-synergistic-combination-of-radam-lookahead-for-the-best-of-2dc83f79a48d

Vienna Deep Learning Meetup