Lookahead Optimizer:
k steps forward, 1 step back

Tao Shen

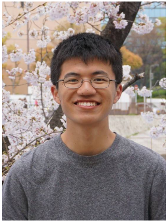
Zhejiang University

August 11, 2019

Overview

Authors

Department of Computer Science,
University of Toronto,
Vector Institute



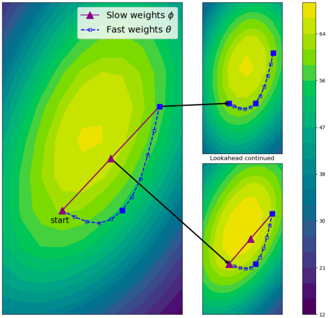
Michael Zhang

James Lucas

Geoffrey Hinton

Jimmy Ba

The Lookahead Algorithm



Visualize

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

Pseudocode

Features

Slow weights trajectory

$$
\begin{aligned}
\phi_{t+1} &= \phi_t + \alpha\left(\theta_{t,k} - \phi_t\right) \\
&= \alpha\left[\theta_{t,k} + (1-\alpha)\theta_{t-1,k} \ldots + (1-\alpha)^{t-1}\theta_{0,k}\right] + (1-\alpha)^t\phi_0
\end{aligned}
$$

Fast weights trajectory

$$\theta_{t,i+1} = \theta_{t,i} + A\left(L, \theta_{t,i-1}, d\right)$$

Computational complexity

The number of operations is $\mathcal{O}\left(\frac{k+1}{k}\right)$ times that of the inner optimizer

Derived Stochastic Gradient Descent (SGD)

Adaptive learning rate schemes

AdaGrad, Adam

Accelerated schemes

Polyak heavy-ball, Nesterov momentum

- Make use of the accumulated past gradient information
- Costly hyperparameter tuning

Advantages

1. Generalization
   - Improve generalization performance
2. Convergence
   - Improve convergence over the inner optimizer
3. Robustness
   - Be robust to hyperparameter changes
     - changes in the inner loop optimizer
     - changes in the outer loop

Generalization

Example (Model)

$$\hat{\mathcal{L}}(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\mathbf{c})^T\mathbf{A}(\mathbf{x}-\mathbf{c})$$

$$\mathbf{c} \sim \mathcal{N}(\mathbf{x}^*, \Sigma) \qquad \mathbf{x}^* = 0$$

SGD:

$$\mathbb{V}\left[\mathbf{x}^{(t+1)}\right] = (\mathbf{I}-\gamma\mathbf{A})^2\mathbb{V}\left[\mathbf{x}^{(t)}\right] + \gamma^2\mathbf{A}^2\Sigma$$

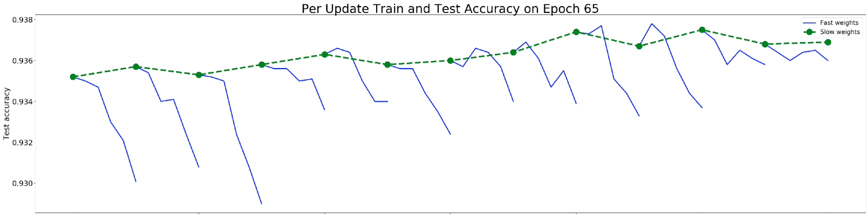
Lookahead with SGD:

$$\mathbb{V}[\phi_{t+1}] = \left[1-\alpha+\alpha(\mathbf{I}-\gamma\mathbf{A})^k\right]^2\mathbb{V}[\phi_t] + \alpha^2\sum_{i=0}^{k-1}(\mathbf{I}-\gamma\mathbf{A})^{2i}\gamma^2\mathbf{A}^2\Sigma$$

$$V^*_{SGD} = \frac{\gamma^2\mathbf{A}^2\Sigma^2}{\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^2}$$

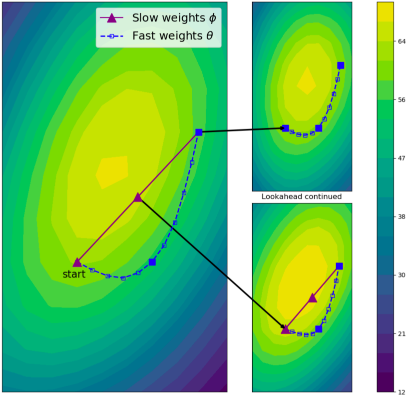
$$V^*_{LA} = \frac{\alpha^2\left(\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^{2k}\right)}{\alpha^2\left(\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^{2k}\right)+2\alpha(1-\alpha)\left(\mathbf{I}-(\mathbf{I}-\gamma\mathbf{A})^k\right)^k}V^*_{SGD}$$

Generalization



Within each inner loop the fast weights may lead to substantial degradation in task performance. The slow weights step recovers the outer loop variance and restores the test accuracy.

Convergence



When oscillating in the high curvature direction, the fast weights updates make rapid progress along the low curvature direction. The slow weights help smooth out the oscillation through the parameter interpolation. The combination of fast weights and slow weights improves learning in high curvature directions, reduces variance, and enables Lookahead to converge rapidly in practice.

Convergence

Example (Model)

$$\hat{\mathcal{L}}(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\mathbf{c})^T\mathbf{A}(\mathbf{x}-\mathbf{c})$$

$$\mathbf{c} \sim \mathcal{N}(\mathbf{x}^*, \Sigma) \qquad \mathbf{x}^* = 0$$

SGD:

$$\mathbb{E}\left[\mathbf{x}^{(t+1)}\right] = (\mathbf{I}-\gamma\mathbf{A})\mathbb{E}\left[\mathbf{x}^{(t)}\right]$$

Lookahead with SGD:

$$(\mathbf{I}-\gamma\mathbf{A})^k < 1-\alpha+\alpha(\mathbf{I}-\gamma\mathbf{A})^k$$

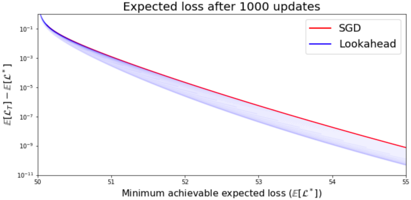
$$\begin{aligned}
&\mathbb{E}[\phi_{t+1}] \\
&= (1-\alpha)\mathbb{E}[\phi_t] + \alpha\mathbb{E}[\boldsymbol{\theta}_{t,k}] \\
&= (1-\alpha)\mathbb{E}[\boldsymbol{\phi}_t] + \alpha(\mathbf{I}-\gamma\mathbf{A})^k\mathbb{E}[\boldsymbol{\phi}_t] \\
&= \left[1-\alpha+\alpha(\mathbf{I}-\gamma\mathbf{A})^k\right]\mathbb{E}[\boldsymbol{\phi}_t]
\end{aligned}$$

Convergence

Example (Model)

$$\hat{\mathcal{L}}(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\mathbf{c})^T\mathbf{A}(\mathbf{x}-\mathbf{c})$$

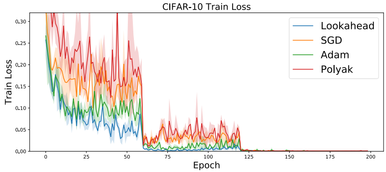
$$\mathbf{c} \sim \mathcal{N}(\mathbf{x}^*, \Sigma) \qquad \mathbf{x}^* = 0$$



We speculate that the learning rate for the inner optimizer is set sufficiently high such that the variance reduction term is more important

$$\mathbb{E}\left[\hat{\mathcal{L}}\left(\theta^{(t)}\right)\right] = \frac{1}{2}\sum_i a_i\left(\mathbb{E}\left[\theta_i^{(t)}\right]^2 + \mathbb{V}\left[\theta_i^{(t)}\right] + \sigma_i^2\right)$$

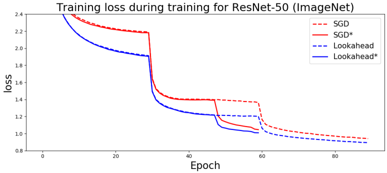
Convergence



OPTIMIZER	CIFAR-10	CIFAR-100
SGD	$95.23 \pm .19$	$78.24 \pm .18$
POLYAK	$95.26 \pm .04$	$77.99 \pm .42$
ADAM	$94.84 \pm .16$	$76.88 \pm .39$
LOOKAHEAD	$95.27 \pm .06$	$78.34 \pm .05$

Table 1: CIFAR Final Validation Accuracy.

CIFAR

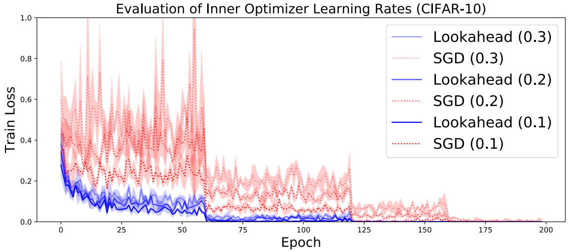


OPTIMIZER	LA	SGD
EPOCH 50 - TOP 1	75.13	74.43
EPOCH 50 - TOP 5	92.22	92.15
EPOCH 60 - TOP 1	75.49	75.15
EPOCH 60 - TOP 5	92.53	92.56

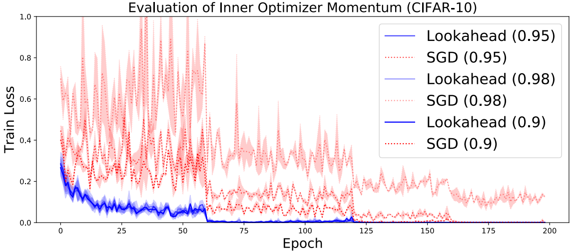
Table 2: Top-1 and Top-5 single crop validation accuracies on ImageNet.

ImageNet

Inner Robustness

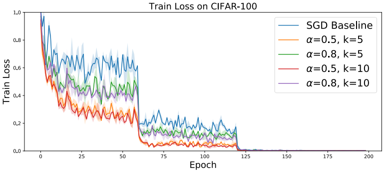


(a) CIFAR-10 Train Loss: Different LR



(b) CIFAR-10 Train Loss: Different momentum

Outner Robustness



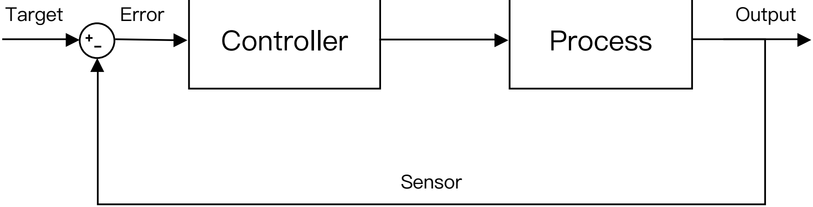
K \ α	0.5	0.8
5	$78.24 \pm .02$	$78.27 \pm .04$
10	$78.19 \pm .22$	$77.94 \pm .22$

Table 5: All settings have higher validation accuracy than SGD (77.72%)

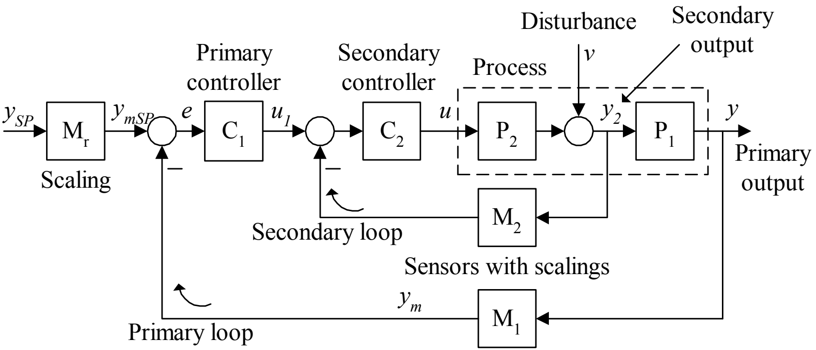
Conclusion

1. Generalization
   - Improve generalization performance
2. Convergence
   - Improve convergence over the inner optimizer
3. Robustness
   - Be robust to hyperparameter changes
     - changes in the inner loop optimizer
     - changes in the outer loop

Discuss



Cascade Control



Comparison

1. Generalization
2. Robustness
3. Larger Learning Rate

1. Anti-disturbance
2. Robustness
3. Larger P

The End