Lookahead Optimizer: 
k steps forward, 1 step back

Michael R. Zhang, James Lucas, Geoffrey Hinton, Jimmy Ba
Related Work / Motivation
Polyak Averaging

- Taking arithmetic average of weights gives faster convergence in convex optimization.
- Weight averaging in neural networks has seen more interest recently.
**Stochastic Weight Averaging (2018)**

- Create an ensemble by averaging the *weights* of a neural network at different points in training
- Beats existing methods for ensembling in model space

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, Andrew Gordon Wilson
Regularized Nonlinear Acceleration (RNA)

- A related, more complicated algorithm that tries to find a point where the gradient is zero.
- It solves a linear system based on the most recent $k$ iterates.
- Achieves faster convergence and occasionally better generalization.
- Factor of $k$ times more memory and more compute.
  - $O(K^2d + K^3)$

Damien Scieur, Edouard Oyallon, Alexandre d’Aspremont, Francis Bach
Method
Lookahead Optimizer

Algorithm 1 Lookahead Optimizer:

Require: Initial parameters $\phi_0$, objective function $L$

Require: Synchronization period $k$, slow weights step size $\alpha$, 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 $\phi$
- Project parameters of neural network into 2-D for visualization
- Lighter colors represent regions of higher accuracy
Noisy Quadratic Analysis

- Simple proxy for neural network optimization (see work from James Martens, Roger Grosse, Tony Wu, Guodong Zhang et al.)

\[ L(\theta) = \frac{1}{2} \theta^T A \theta = \frac{1}{2} \sum_{i=1}^{d} a_i \theta_i^2 \triangleq \sum_{i=1}^{d} l(\theta_i). \]

We assume that the gradient we obtain is noisy: for each dimension, instead of receive the true gradient \( a_i \theta_i \), we get a noisy version \( a_i \theta_i + c_i \), where \( c_i \sim \mathcal{N}(0, \sigma_i^2) \).
Noisy Quadratic Analysis

**Proposition 2** (Lookahead variance reduction). Let $0 < \gamma < 2/L$ be the learning rate of SGD and Lookahead where $L = \max_i a_i$. In the noisy quadratic model, the iterates of SGD and Lookahead with SGD as its inner optimizer converge to 0 in expectation and the variances converge to the following fixed points:

\[
V_{SGD}^* = \frac{\gamma^2 A^2 \Sigma^2}{I - (I - \gamma A)^2}
\]  
\[
V_{LA}^* = \frac{\alpha^2 (I - (I - \gamma A)^{2k})}{\alpha^2 (I - (I - \gamma A)^{2k}) + 2\alpha(1 - \alpha)(I - (I - \gamma A)^k)} V_{SGD}^*
\]  

(6)  

(7)
Results
CIFAR-10

- Find best hyperparameters for inner optimizer, then perform small grid search on outer loop
Hyperparameter Robustness

Evaluation of Inner Optimizer Learning Rates (CIFAR-10)
Hyperparameter Robustness

Evaluation of Inner Optimizer Momentum (CIFAR-10)
Hyperparameter Robustness (LA hyperparams)

Train Loss on CIFAR-100

- SGD Baseline
- $\alpha=0.5, k=5$
- $\alpha=0.8, k=5$
- $\alpha=0.5, k=10$
- $\alpha=0.8, k=10$
Fast / Slow Weight Intuition

- Inner updates can degrade performance on both training and test set, while outer update restores performance.
ImageNet

- Standard benchmark for image classification: over 1.28 million training images and 50,000 validation images
- ResNet-50 has 25 million parameters, works for ResNet-152 as well
### Test time performance

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGD</td>
<td>95.23 ± .19</td>
<td>78.24 ± .18</td>
</tr>
<tr>
<td>POLYAK</td>
<td>95.26 ± .04</td>
<td>77.99 ± .42</td>
</tr>
<tr>
<td>ADAM</td>
<td>94.84 ± .16</td>
<td>76.88 ± .39</td>
</tr>
<tr>
<td>LOOKAHEAD</td>
<td>95.27 ± .06</td>
<td>78.34 ± .05</td>
</tr>
</tbody>
</table>

Table 1: CIFAR Final Validation Accuracy.

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>LA</th>
<th>SGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPOCH 50 - Top 1</td>
<td>75.13</td>
<td>74.43</td>
</tr>
<tr>
<td>EPOCH 50 - Top 5</td>
<td>92.22</td>
<td>92.15</td>
</tr>
<tr>
<td>EPOCH 60 - Top 1</td>
<td>75.49</td>
<td>75.15</td>
</tr>
<tr>
<td>EPOCH 60 - Top 5</td>
<td>92.53</td>
<td>92.56</td>
</tr>
</tbody>
</table>

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

- WMT14 English-to-German task with Transformer Model
- Lookahead is *robust*: achieve faster training convergence without using tuned, ramp-up learning rate schedules
Penn Tree Bank

- Benchmark to model prediction of next word given previous words
Summary

- Faster convergence with little hyperparameter tuning on a variety of datasets and models, big and small
- Fast weights can degrade test and training accuracy, but slow weights restore performance
- Extensions: learning rate scheduling, family of methods that maintain memory information
How to Use

Simple interface, in TensorFlow and PyTorch:

```python
optimizer = # {any optimizer} e.g. tf.train.AdamOptimizer
if args.lookahead:
    optimizer = Lookahead(optimizer, la_steps=args.la_steps, la_alpha=args.la_alpha)
```

Code: [https://github.com/michaelrzhang/lookahead](https://github.com/michaelrzhang/lookahead)
Contact: michael@cs.toronto.edu

Thank you!
Interaction with SWA

SWA WideResNet-28-10 CIFAR-100 Test Accuracy

- SGD + SWA
- LA(SGD) + SWA