Adaptive Optimization

Hyperparameter tuning is a big cost of deep learning.
Momentum: a key hyperparameter to SGD and variants.
Adaptive methods, e.g. Adam, don’t tune momentum.

YellowFin optimizer

- Based on the robustness properties of momentum.
- Auto-tuning of momentum and learning rate in SGD.
- Closed-loop momentum control for async. training.

Robustness of Momentum

Momentum operation
- SGD step: \( x_{t+1} = x_t - \alpha \nabla f(x_t) + \mu (x_t - x_{t-1}) \).
- In a 1-D case, matrix form with \( \nabla f(x_t) = h(x_t)(x_t - x^*) \):
  \[
  (x_{t+1} - x^*) = \left[ 1 - \alpha h(x_t) + \mu - \mu h(x_t) \right] (x_t - x^*) = A_t (x_t - x^*)
  \]

Robust region

\[
\frac{(1 - \sqrt{\mu})^2}{h_{\text{max}}} \leq \alpha \leq \frac{(1 + \sqrt{\mu})^2}{h_{\text{min}}}, \text{ given } h_{\text{min}} \leq h(x_t) \leq h_{\text{max}}.
\]

Spectral radius \( \rho(A_t) = \sqrt{\mu} \to \text{linear convergence}^3 \)

\(^3\) Not guaranteed for non-quadratics

- Linear rate robust to curvature variance (middle).
- Linear rate robust to a range of learning rates (left).

YellowFin

Noisy quadratic model

- Model stochastic setting with gradient variance \( C \).
- Local quadratic approximation\(^3\): 1-D case.

In robust region, distance to optimum is approx. by

\[
\mathbb{E}(x_t - x^*)^2 \approx \mu^t (x_0 - x^*)^2 + (1 - \mu^t) \frac{\alpha^2 C}{1 - \mu}
\]

Greedy tuning strategy

- Principle I: Stay in the robust region
- Principle II: Minimize \( \mathbb{E}(x_0 - x^*)^2 \) after one step \((t=1)\)
- Solve learning rate and momentum in closed-form sol.

Experiments

ResNet and LSTM

YellowFin runs with no tuning.
Adam, mom. SGD etc. are tuned on learning rate grids.
YellowFin can outperform tuned SoA on train/val. metrics.

Facebook Convolutional Seq-to-seq model

IWSLT 2014 German-English translation
YellowFin outperforms the hand-tuned default optimizer

<table>
<thead>
<tr>
<th></th>
<th>Val. loss</th>
<th>Val. BLEU@4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Nesterov Momentum</td>
<td>2.86</td>
<td>30.75</td>
</tr>
<tr>
<td>YellowFin</td>
<td>2.75</td>
<td>31.59</td>
</tr>
</tbody>
</table>

Github for PyTorch: https://github.com/JianGoForIt/YellowFin_Pytorch
Github for TensorFlow: https://github.com/JianGoForIt/YellowFin

“Send us your bug reports!”

Extension: Closed-loop YellowFin

Async. distributed training: fast, no sync. barrier.
However, Asynchrony induces additional \( \bar{\mu} \) to \( \mu \).
Can we auto-match total momentum \( \mu_T \) to YF-tuned \( \mu^* \)?

Closed-loop momentum control

Async. induced \( \bar{\mu} \)

Algorithmic \( \mu \to \text{Effective total value} \)

\[
\mu_T = \mu + \bar{\mu}
\]

Feedback control \( \mu \leftarrow \mu + \gamma \cdot (\mu^* - \mu_T) \)

Closed-loop mechanism improves YellowFin in async.