Исследование и применение метода оптимизации Lookahead

Манджиев Данил Б05-120

Научный руководитель: Безносиков Александр Николаевич

Lookahead: k шагов вперёд, один шаг назад

- Lookahead - метаалгоритм, являющийся обёрткой над базовыми алгоритмами оптимизациями.
- Каждые k шагов базовый алгоритм обновляет "лёгкие веса"
- На k + 1 шаге обновляются "тяжёлые веса" с шагом α

Lookahead работат в команде с другими алгоритмами

+ Lookahead + Adam
+ Lookahead + SGD
+ Lookahead + GD
+ Основным плюсом при таком подходе является уменьшение разброса весов и стабильная сходимость к оптимому

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

Гиперпараметры Lookahead

- Как видно, k и α являются гиперпараметрами Lookahead, которые нужно подбирать до начала оптимизации
- Авторы статьи советуют перебирать k от 1 до 20
- По умолчанию берутся k = 5 и α = 0.8 (также советуют устанавливать на уровне 0.5)

D-adaptation and Progidy

- Подбор гиперпараметров является известной проблемой стандартных алгоритмов оптимизации
- Идея: не устанавливать гиперпараметры до обучения, а настраивать их в процессе, чтобы алгоритм сам учился на своём опыте

Algorithm 1 Dual Averaging with D-Adaptation

Input: x_0, $d_0 > 0$

$s_0 = 0$, $g_0 \in \partial f(x_0)$, $\gamma_0 = 1/\|g_0\|$

If $g_0 = 0$, exit with $\hat{x}_n = x_0$

for $k = 0$ **to** n **do**

$g_k \in \partial f(x_k)$

$s_{k+1} = s_k + d_k g_k$

$$\gamma_{k+1} = \frac{1}{\sqrt{\sum_{i=0}^{k} \|g_i\|^2}}$$

$$\hat{d}_{k+1} = \frac{\gamma_{k+1} \|s_{k+1}\|^2 - \sum_{i=0}^{k} \gamma_i d_i^2 \|g_i\|^2}{2\|s_{k+1}\|} \qquad \text{Option II: } \hat{d}_{k+1} = \frac{\sum_{i=0}^{k} d_i \gamma_i \langle g_i, s_i \rangle}{\|s_{k+1}\|}$$

$d_{k+1} = \max(d_k, \hat{d}_{k+1})$

$x_{k+1} = x_0 - \gamma_{k+1} s_{k+1}$

end for

Return $\hat{x}_n = \frac{1}{\sum_{k=0}^{n} d_k} \sum_{k=0}^{n} d_k x_k$

Улучшение D-adaptation and Progidy

- Верхние алгоритмы оптимизации появились недавно(2023 и 2024), и разрабатывались со спонсорством Meta AI и Samsung AI
- В своей работе я хочу расширить данные методы на Lookahead
- То есть, я хочу добиться комбинации D-adaptation(1 порядка) + Lookahead + D-adaptation(2 порядка) + SGD/ADAM/...

Подбор гиперпараметров

- Подбор α(который заменяет собой "шаг градиентного спуска") будет реализовываться, используя стандартные приёмы из AdaGrad и других адаптивных алгоритмов(с приставкой Ada) в том числе D – adaptation и Progidy
- К последним относится создание обновляемой переменной d - монотонной последовательности, (сходящейся) к истинному D
- Подбор k является более сложной задачей

k

- Максимальное k при котором норма разности нового веса со старым монотонно возрастает
- Стохастический алгоритм при котором k выбирается с весом пропорционально нормы разности
- Установление пороговых значении для d, при котором шаг алгоритма будет реализовываться после их достижении

Спасибо за внимание!

Литература

+ Lookahead Optimizer: k steps forward, 1 step back
+ A Variational Inequality Perspective on Generative Adversarial Networks
+ Learning-Rate-Free Learning by D-Adaptation
+ Prodigy: An Expeditiously Adaptive Parameter-Free Learner