PSGD Pytorch implementation updates

Xilin Li

December 11, 2023

Notations:

- H: Hessian
- v: a probe vector for Hessian estimation, typically $v \sim \mathcal{N}(0, I)$
- h: Hessian-vector product, i.e., $h = Hv + \epsilon$, where $\epsilon = 0$ if no gradient noise
- P: preconditioner
- Q: factor of P as $P = Q^T Q$

Preconditioner fitting loss given a single pair of (v, h),

$$\ell(v, h; P) = h^T P h + v^T P^{-1} v$$

Expectation of the above loss assuming $\epsilon = 0$,

$$\ell(P) = \text{tr}(PH^2) + \text{tr}(P^{-1})$$

The optimal P is $|H|^{-1}$. With $\epsilon \neq 0$, the stochastic gradient noise damps the preconditioner estimation.

1 Dec 2023

1.1 Added the gradient whitening preconditioner

When h is set to the stochastic gradient, we have the optimal P as

$$P = \left(E[gg^T]\right)^{-1/2}$$

This preconditioner whitens the gradient. Specifically, for this type, PSGD reduces to RMSProp and Adam (with momentum) when P is diagonal. By default, we assume the Newton type, i.e., $|H|^{-1}$ as the preconditioner.

1.2 Tighter lower bound for triangular matrix norm

We update Q on the group of triangular matrix as

$$Q \leftarrow (I + \mu A)Q$$

where the step size μ is small enough such that $\|\mu A\|_2 < 1$, and $-A \in \mathbb{R}^{N \times N}$ is the gradient for preconditioner fitting.

Possible cheap, i.e., $\mathcal{O}(N^2)$ complexity, lower bounds of $\|A\|_2$ for step size normalization are:

- Bound F: $\frac{1}{\sqrt{r}}\|A\|_F \leq \|A\|_2 \leq \|A\|_F$, where r is the rank of G.
- Bound max: $\|A\|_{\max} \leq \|A\|_2 \leq N\|A\|_{\max}$, where $\|A\|_{\max} = \max_{ij} a_{ij}$
- Bound 1: $\frac{1}{\sqrt{N}}\|A\|_1 \leq \|A\|_2 \leq \sqrt{N}\|A\|_1$, where $\|A\|_1 = \max_j \sum_i |a_{ij}|$

- Bound inf: $\frac{1}{\sqrt{N}}\|A\|_\infty \le \|A\|_2 \le \sqrt{N}\|A\|_\infty$, where $\|A\|_\infty = \max_i \sum_j |a_{ij}|$

Bound F is useful only when $r \ll N$. In general, none of the above lower bound is tighter than the rest. In the updated implementation, I replaced lower bound $\|A\|_{\max}$ with the following consistently tighter one.

Let

$$\beta = \sqrt{\max\left(\max_i \sum_j a_{ij}^2,\ \max_j \sum_i a_{ij}^2\right)}$$

Then we have

$$\beta \le \|A\|_2 \le \sqrt{N}\beta$$

Not difficult to prove the following desired properties:

- $\beta \le \|A\|_2$: since $\|A^T\|_2 = \|A\|_2 \ge \|Ax\|/\|x\|$ for any $x \ne 0$, we let x be the columns or rows of A to have this lower bound. This lower bound is tight, e.g., when $A = I$.
- $\beta \ge \frac{1}{\sqrt{N}}\|A\|_F$: obvious.
- $\beta \ge \|A\|_{\max}$: obvious.
- $\beta \ge \frac{1}{\sqrt{N}}\|A\|_1$: due to the Cauchy–Schwarz inequality, $N\sum_i a_{ij}^2 \ge (\sum_i |a_{ij}|)^2$.
- $\beta \ge \frac{1}{\sqrt{N}}\|A\|_\infty$: again, due to the Cauchy–Schwarz inequality.
- $\|A\|_2 \le \sqrt{N}\beta$: as $\|A\|_2 \le \|A\|_F \le \sqrt{N}\beta$. This upper bound is tight, e.g., when A is a matrix with all elements taking the same value.

1.3 Initial scale of the preconditioner can be set to None

I initialize Q as $Q = \alpha I$, where α is the initial scale. By letting

$$\alpha^2 \text{ learning\_rate}_{\text{PSGD}} = \text{learning\_rate}_{\text{SGD}}$$

we can map the initial settings of SGD to PSGD. In general, proper tuning of α is necessary.

If α is set to None, the first pair of (v, h) is used to initialize Q as

$$Q = \left(\frac{v^T v}{h^T h}\right)^{1/4} I$$

However, this can be risky as the Newton's method does not always converge faster than gradient descent out of the basin of attraction. For example, considering convex function $f(x) = x^2 - 4x^{1/2}, x \in \mathbb{R}^+$. We have $f'(x) = 2x - 2x^{-1/2}$, $f''(x) = 2 + x^{-3/2}$, and optimal soltion $x = 1$. The Newton's method can be arbitrarily slower than the gradient descent when $0 < x \ll 1$, as shown by

$$\lim_{x\to 0} \frac{f'(x)}{f''(x)} = \lim_{x\to 0} \frac{2x - 2x^{-1/2}}{2 + x^{-3/2}} = \lim_{x\to 0} \frac{-2x^{-1/2}}{x^{-3/2}} = -2\lim_{x\to 0} x = 0$$

Thus, for this example, PSGD will get stuck around $x = 0$ if we rely on this strategy to determine α.

1.4 More updates

- Changed class name UVd to LRA (low-rank approximation); keep UVd as an alias of LRA.
- Now LRA reduces to the diagonal preconditioner when the rank of approximation is set to 0.
- Updated hello_psgd.py to show how easy to apply PSGD on minimizing the Rosenbrock function.
- Added misc/preconditioner_fitting_rule_verification.py to show how fitting works on Lie groups.