Block Mean Approximation for Efficient Second Order Optimization

A short report on [Lu et al., 2018]

Nazarov Ivan

Skoltech

July 27, 2018

The update step

Gradient-based optimization usually use the following update

$$\theta \leftarrow \theta - G^{-1}\nabla_\theta f(\theta)\eta\,, \tag{1}$$

where $G \in \mathbb{R}^{d\times d}$ is non-singular matrix.

- Newton: G is the Hessian $H = \nabla_\theta^2 f(\theta)$
- Natural gradient: G is the Informaton Matrix $\nabla_\theta f(\theta)\nabla_\theta f(\theta)^{\mathrm{T}}$

the Idea of the paper

Inverting G is expensive $\mathcal{O}(d^3)$, so...

... approximate G so that it its cheap to invert

- $G = diag\ H$ adaptively scales the updates

... while also keeping correlation between the parameters

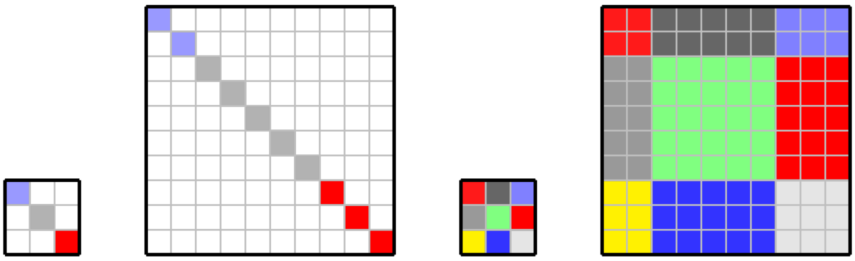
- the off-diagonal elements of G capture cross effects
- diagonal and block diagonal approximations neglect them

Block Mean Approximation

Approximate G with the **Block Mean Approximation**:

$$\mathtt{BMA}_{\mathbf{s}}(G) = \arg\min_{B,\Lambda} \left\| G - (\bar{\Lambda} + \bar{B}) \right\|_F^2 , \tag{2}$$

where $\bar{\Lambda}$ and $\bar{B}$ are block expansions of Λ and B w.r.t. partition $\mathbf{s}$.



(a) Λ (b) $\bar{\Lambda}$ (c) B (d) $\bar{B}$

Figure 1: Expansion matrices. (a) Diagonal matrix Λ. (b) Diagonal expansion of Λ. (c) Full matrix B. (d) Full expansion of B. The partition vector in both cases is $\mathbf{s} = (2, 5, 3)$.

Key operations

If $\bar{\Lambda}$ and $\bar{B}$ are block expansions of Λ and B w.r.t. partition $\mathbf{s}$, then

$$(\bar{\Lambda} + \bar{B})^{-1} = \bar{\Lambda}^{-1} + \bar{D}\,, \qquad (3)$$
$$D = (\Lambda S + SBS)^{-1} - (\Lambda S)^{-1}\,.$$

$$(\bar{\Lambda} + \bar{B})^{-\frac{1}{2}} = \bar{\Lambda}^{-\frac{1}{2}} + \bar{D}\,, \qquad (4)$$
$$D = S^{-\frac{1}{2}}\left((\Lambda + S^{\frac{1}{2}} B S^{\frac{1}{2}})^{-\frac{1}{2}} - \Lambda^{-\frac{1}{2}}\right) S^{-\frac{1}{2}}\,,$$

for $\mathbf{s} = (s_1, \ldots, s_L)$ and $S = diag(|\mathbf{s}_i|)_{i=1}^{L} \in mathbbR^{L \times L}$.

- requires $\mathcal{O}(L^3)$ instead of $\mathcal{O}(d^3)$
- no need to construct the expansions explicitly

Application to AdaGrad

Update (1): $\theta_{t+1} \leftarrow \theta_t - G_t^{-1} g_t \eta$ for $g_t = \nabla_\theta f_t(\theta_t)$ and

$$G_t = \hat{H}_t^{\frac{1}{2}}, \quad \text{with} \quad \hat{H}_t \approx H_t = \sum_{s \le t} g_s g_s^{\mathrm{T}}.$$

AdaGrad-Full has $\hat{H}_t = H_t$ with time-complexity $\mathcal{O}(d^3)$

Approximations use $\hat{H}_t = Z_t F_t Z_t$ with $Z_t = \left(diag\ H_t\right)^{\frac{1}{2}}$:

- **AdaGrad-Diag** has $F_t = I$ and $\mathcal{O}(d)$
- **AdaGrad-BMA** uses $F_t = \mathtt{BMA}_\mathbf{s}\left(Z_t^{-1} H_t Z_t^{-1}\right)$ and $\mathcal{O}(L^3 + d)$

AdaGrad-BMA

- Keep running estimates of the matrices needed for the BMA

$$u_{ti} = \sum_{j \in \mathbf{s}_i} g_{tj} \,, \text{ and } v_{ti} = \sum_{j \in \mathbf{s}_i} z_{tj} \,.$$

Then $Z_t^{-1} H_t Z_t^{-1}$ is approximated by the expansion matrices of

$$\Lambda_t = I \text{ and } B_t = S^{-\frac{1}{2}} \frac{U_t - diag\, U_t}{v_t v_t^{\mathrm{T}}} S^{-\frac{1}{2}} \,.$$

From (4) the inverse root is $I + \bar{D}$ where

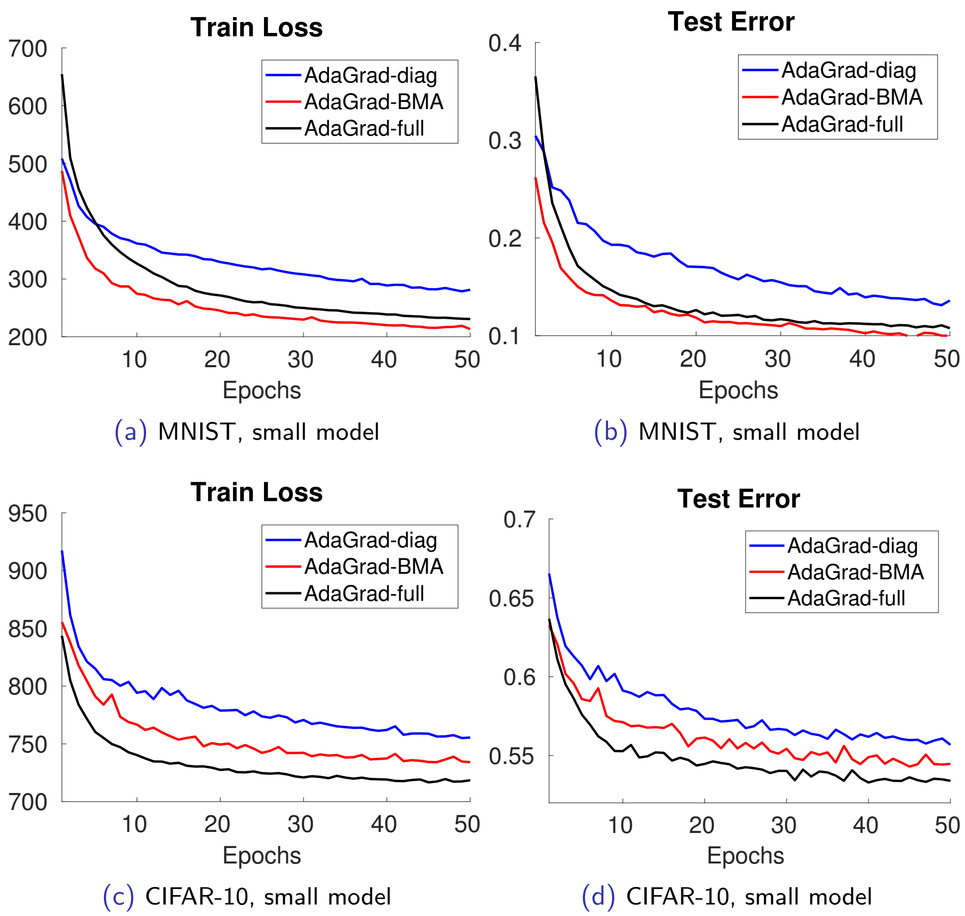
$$D = S^{-\frac{1}{2}} \left(\underbrace{\left(I + \frac{U_t - diag\, U_t}{v_t v_t^{\mathrm{T}}} \right)}_{M_T}^{-\frac{1}{2}} - I \right) S^{-\frac{1}{2}} \,,$$

Eigen-decompose M_t as RVR^{T} and get its root via $RV^{-\frac{1}{2}} R^{\mathrm{T}}$

Experiments

- Small and large conv-nets in MNIST and CIFAR-10
- partition **s** for BMA:
  - 1 block per layer of the NN
  - 2 subblocks for weights and bias in each layer
- $\eta \in \{10^{-k} \colon k = 0, \ldots, -4\}$ report the best performance

Some results



(a) MNIST, small model

(b) MNIST, small model

(c) CIFAR-10, small model

(d) CIFAR-10, small model

Figure 2: Performance of AdaGrad and its approximations on two standard datasets.

References

Lu, Y., Harandi, M., Hartley, R. I., and Pascanu, R. (2018).
Block Mean Approximation for Efficient Second Order Optimization.
ArXiv e-prints.