Globally Optimizing the Learning Rate of the Heavy Ball Method

Yue Sun

1 Introduction

People have been using gradient based algorithm to do optimization. Although it's proved that accelerated gradient descent [1–3] hits the optimal convergence rate bound, it is hoped that, for the specific problem or structure, there exists a better algorithm not based on worst case behavior among a too big collection of functions. Thus people have been using meta-learning, i.e., learning to learn, a special kind of reinforcement learning strategy, before training deep neural networks. The meta learning problem, in principal, is the following: given the objective function f and hypothesis class of algorithms $\mathscr{A}$, the best algorithm is

$$\underset{\mathcal{A}\in\mathscr{A}}{\operatorname{argmin}} \boldsymbol{E}_{x_0\sim P_0} L(x,f) \text{ s.t. } x_{t+1} = \mathcal{A}(x_{[t]}, f),$$

where L is a loss function indicating progress of algorithm. An example is $L(x,f) = \sum_{t=0}^{T} f(x_t)$.

Empirically, it usually returns a good optimization algorithm, e.g., [4] is a very interesting start point. However, people have few understanding about the convergence guarantee, neither about meta-learning, nor any reinforcement learning or recurrent neural nets' training. In this project, we try to give a very much simplification of [4], and analyze the landscape of the objective function.

We consider the linear regression problem

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{2}x^T Ax - b^T x \tag{1}$$

where $A \in \boldsymbol{S}_{++}^{d\times d}$ is a positive definite matrix. The algorithm class $\mathscr{A}$ we consider is gradient descent and heavy ball algorithm,

$$\begin{aligned}&(\text{GD}) \quad \mathscr{A} = \{x_t - \eta\nabla f(x_t) |\, \eta \in (0, 2/\lambda_{\max}(A))\}\\ &(\text{HB}) \quad \mathscr{A} = \{x_t - \eta\nabla f(x_t) + \gamma(x_t - x_{t-1}) |\, \eta \in (0, 2/\lambda_{\max}(A)), \gamma \in [0,1)\}\,.\end{aligned}$$

And we apply such an optimization problem as the learning to learn problem

$$\underset{\mathcal{A}\in\mathscr{A}}{\operatorname{argmin}} \boldsymbol{E}_{x_0\sim P_0} \sum_{t=0}^{\infty} (f(x_t) - f(x^*)) \text{ s.t. } x_{t+1} = \mathcal{A}(x_{[t]}, f).$$

The infinite sum can be approximated by truncated sum if converge. Applying GD and HB,

$$(\text{GD}) \quad \underset{\eta\in(0,2/\lambda_{\max}(A))}{\operatorname{argmin}} \boldsymbol{E}_{x_0\sim P_0} \sum_{t=0}^{\infty} (f(x_t) - f(x^*)) \text{ s.t. } x_{t+1} = x_t - \eta\nabla f(x_t),$$

$$(\text{HB}) \quad \underset{\eta\in(0,2/\lambda_{\max}(A)),\gamma\in[0,1)}{\operatorname{argmin}} \boldsymbol{E}_{x_1=x_0\sim P_0} \sum_{t=0}^{\infty} (f(x_t) - f(x^*)) \text{ s.t. } x_{t+1} = x_t - \eta\nabla f(x_t) + \gamma(x_t - x_{t-1}),$$

where $f(x) = \frac{1}{2}x^T Ax - b^T x$. With that small hypothesis class, we only have (at most) two scalar variables, η and γ. The question is, for such a simple problem, what if we optimize by gradient descent? Is there any spurious local minimum? Is there any high order saddle points/vanishing gradients? Is it convex? We will analyze the objective function geometry both via visualization and algebra, and argue that we can use gradient descent to solve the problem. As (1) can be regarded as a quadratic approximation of any smooth convex function, we hope that learning to learn strategy, at least for convex optimization pretraining, works well.

2 Main theorems

We first form the objective function into a desired way. $f(x) = \frac{1}{2}x^T Ax - b^T x$, so $x^* = A^{-1}b$. The loss at each iteration is

$$f(x) - f(x^*) = \frac{1}{2}\|A^{\frac{1}{2}}x - A^{-\frac{1}{2}}b\|^2. \tag{2}$$

Denote $z_t = A^{\frac{1}{2}}x_t - A^{-\frac{1}{2}}b$.

2.1 Warm up: gradient descent

Let's start from gradient descent, where $x_{t+1} = x_t - \eta\nabla f(x_t)$. The iteration is

$$\begin{aligned} z_{t+1} &= A^{\frac{1}{2}}x_{t+1} - A^{-\frac{1}{2}}b && \text{(3a)}\\ &= A^{\frac{1}{2}}(x_t - \eta Ax_t + \eta b) - A^{-\frac{1}{2}}b && \text{(3b)}\\ &= (I - \eta A)z_t. && \text{(3c)} \end{aligned}$$

So the loss function is

$$\begin{aligned} L(\eta; z_0) = \sum_{t=0}^{\infty}\|z_t\| &= z_0^T\sum_{t=0}^{\infty}(I - \eta A)^{2t}z_0 && \text{(4a)}\\ &= z_0^T(I - (I - \eta A)^2)^{-1}z_0 && \text{(4b)} \end{aligned}$$

Denote v_i as the eigenvectors of A, corresponding to eigenvalue λ_i. $z_{0,i} = \|\mathcal{P}_{v_i}z_0\|$. So

$$L(\eta; z_0) = \sum_{i=1}^{d} z_{0,i}^2(1 - (1 - \eta\lambda_i)^2)^{-1}. \tag{5}$$

It's monotonically decreasing when $\eta \leq 1/\lambda_{\max}(A)$ (for any z_0, thus also true for L with a distribution of z_0). But when $1/\lambda_{\max}(A) \leq \eta \leq 2/\lambda_{\max}(A)$ it's not clear. We will postpone it until subsection 2.3.

2.2 Heavy ball – momentum variable

If the step size η is fixed, we consider the problem of

$$\operatorname*{argmin}_{\gamma\in[0,1)} L_0(\gamma) = L(\gamma; x_0 = x_1, \eta) := \sum_{t=0}^{\infty}(f(x_t) - f(x^*)) \text{ s.t. } x_{t+1} = x_t - \eta\nabla f(x_t) + \gamma(x_t - x_{t-1}) \tag{6}$$

The iteration is given by

$$\begin{aligned} z_{t+1} &= A^{\frac{1}{2}}x_{t+1} - A^{-\frac{1}{2}}b && \text{(7a)}\\ &= A^{\frac{1}{2}}(x_t - \eta Ax_t + \eta b + \gamma(x_t - x_{t-1})) - A^{-\frac{1}{2}}b && \text{(7b)}\\ &= ((1 + \gamma)I - \eta A)z_t - \gamma z_{t-1}. && \text{(7c)} \end{aligned}$$

Let $z_0 = z_1$, we have

$$z_t = ((D_2 - D_1)^{-1}(D_2 - 1)D_1^t + (D_1 - D_2)^{-1}(D_1 - 1)D_2^t)z_0, \tag{8}$$

where D_1, D_2 are matrices with same eigenvector as A, and eigenvalues are solution of $x^2 - ((1 + \gamma)I - \eta\lambda_i) + \gamma^2 = 0$. Then we can write out the close form of loss function

$$L_0(\gamma) = -\sum_{i=1}^{d}\frac{\nu_i^3 + (2 + \bar{\lambda}_i)\nu_i^2 + 2\bar{\lambda}_i\nu_i + 2\bar{\lambda}_i}{\bar{\lambda}_i(\bar{\lambda}_i + \nu_i)(\bar{\lambda}_i + 2\nu_i + 4)}z_{0,i}^2, \tag{9}$$

where

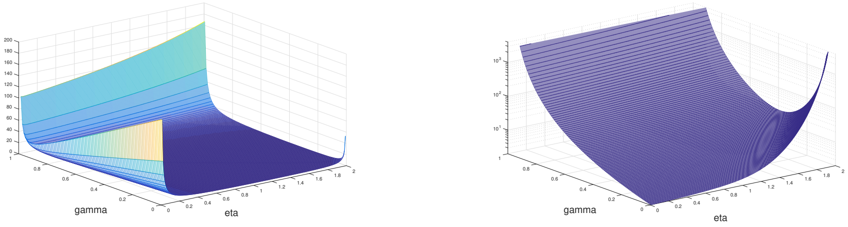
$$\bar{\lambda}_i = \eta\lambda_i, \nu_i = \gamma - \bar{\lambda}_i - 1. \tag{10}$$

Take the second derivative, we have

$$\nabla^2 L_{0,i}(\gamma) = \frac{4(\bar{\lambda}_i \nu_i - 8)}{(\bar{\lambda}_i + 2\nu_i + 4)^3} - \frac{2\bar{\lambda}_i}{(\bar{\lambda}_i + 2\nu_i + 4)^2} + \frac{4}{(\bar{\lambda}_i + \nu_i)^3} \tag{11a}$$

$$- \frac{64}{(\bar{\lambda}_i + \nu_i)(\bar{\lambda}_i + 2\nu_i + 4)^3} - \frac{32}{(\bar{\lambda}_i + \nu_i)^2(\bar{\lambda}_i + 2\nu_i + 4)^2} - \frac{16}{(\bar{\lambda}_i + \nu_i)^3(\bar{\lambda}_i + 2\nu_i + 4)} \tag{11b}$$

Figure 1 is a visualization, from it we see the second derivative is positive. It can be proved that when $0 \leq \nu_i < 1, 0 < \bar{\lambda}_i < 2$, we always have $\nabla^2 L_{0,i}(\gamma) > 0$, so the loss is convex with respect to γ. See appendix for details.



(a)

(b)

Figure 1: x_0 are standard normal around optimizer (not practical, but just for simplicity). $A = diag(0.4, 1)$. (a) Loss function $L(\gamma; \eta)$ versus η and γ, (b) Hessian wrt γ, $\nabla^2 L(\gamma; \eta)$ versus η and γ.

2.3 Heavy ball

However, for heavy ball method, the loss function is not jointly convex with respect to η and γ, see Figure 2. But we can show that, *all critical points* of loss function are *global minimum* when the distribution of z_0 fully exploits the space. We use the technique in [5]. Denote $u_t = [z_{t+1}^T, z_t^T]^T$, $w = [\eta, \gamma]^T$, we have $u_t = Qu_{t-1}$ where

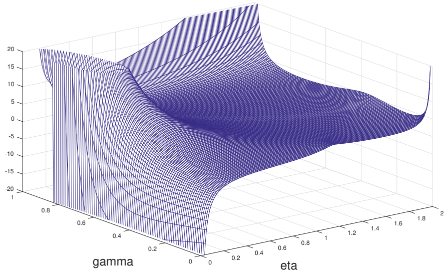


Figure 2: Minimum eigenvalue of $\nabla^2 L(\eta, \gamma)$. x_0 are standard normal around optimizer (not practical, but just for simplicity). $A = diag(0.4, 1)$.

$$Q(\eta, \gamma) = \begin{bmatrix} 0 & I \\ -\gamma I & (1+\gamma)I - \eta A \end{bmatrix} \tag{12}$$

When there's no confusion, we write $Q := Q(\eta, \gamma)$. So

$$L_0(w) := 2L(\eta, \gamma; z_0, z_1) - \|z_0\|^2 = \sum_{t=0}^{\infty} \|u_t\|^2 = u_0^T P_w u_0, \tag{13}$$

where $P_v \in \boldsymbol{S}_{++}$ does not depend on u_0. The reason is that u_t is always some matrix independent of $u_{[t]}$ times u_0. So

$$L_0(w) = u_0^T u_0 + u_1^T P_w u_1 = u_0^T u_0 + u_0^T Q^T P_w Q u_0 \tag{14}$$

Denote

$$D_\eta = \begin{bmatrix} 0 & 0 \\ 0 & -A \end{bmatrix}^T, D_\gamma = \begin{bmatrix} 0 & 0 \\ -I & I \end{bmatrix}^T, \tag{15}$$

then

$$\nabla_\eta L_0(w) = \langle D_\eta P_w Q, u_0 u_0^T \rangle + \nabla_\eta 2L(\eta, \gamma; z_1, z_2) \tag{16a}$$

$$= \langle D_\eta P_w Q, \sum_{t=0}^{\infty} u_t u_t^T \rangle \tag{16b}$$

$$\nabla_\gamma L_0(w) = \langle D_\gamma P_w Q, \sum_{t=0}^{\infty} u_t u_t^T \rangle \tag{16c}$$

On the other hand, denote $A_Q(u, Q') = u^T Q'^T P_w Q' u - u^T Q^T P_w Q u$, and for shorthand $Q' = Q(\eta', \gamma')$. So that

$$L_0(w') - L_0(w) = \sum_{t=0}^{\infty} A_Q(u_t, Q'). \tag{17}$$

And

$$A_Q(u, Q') = 2u^T \Delta^T P_w Q u + u^T \Delta^T P_w \Delta u \tag{18}$$

where $\Delta = Q' - Q$. Denote $Q = q_1 D_\eta^T + q_2 D_\gamma^T + Q_3$ where $\langle D_\eta P_w Q_3, \sum_{t=0}^{\infty} u_t u_t^T \rangle = 0$, $\langle D_\gamma P_w Q_3, \sum_{t=0}^{\infty} u_t u_t^T \rangle = 0$. Denote $\mathcal{P}_\Delta Q = q_1 D_\eta^T + q_2 D_\gamma^T$, then (16b) and (16c) becomes

$$\nabla_\eta L_0(w) = \langle D_\eta P_w \mathcal{P}_\Delta Q, \sum_{t=0}^{\infty} u_t u_t^T \rangle \tag{19a}$$

$$\nabla_\gamma L_0(w) = \langle D_\gamma P_w \mathcal{P}_\Delta Q, \sum_{t=0}^{\infty} u_t u_t^T \rangle. \tag{19b}$$

Let $\Delta = Q - Q^*$, $w^* = [\eta^*, \gamma^*]^T$.

$$L_0(w) - L_0(w^*) = -\sum_{t=0}^{\infty} A_Q(u^*, Q^*) \tag{20a}$$

$$= -\langle \Delta^T P_w Q + \Delta^T P_w \Delta, \sum_{t=0}^{\infty} u_t^* u_t^{*T} \rangle \tag{20b}$$

$$\leq \langle (\mathcal{P}_\Delta Q)^T P_w (\mathcal{P}_\Delta Q), \sum_{t=0}^{\infty} u_t^* u_t^{*T} \rangle \tag{20c}$$

$$\leq \lambda_{\max}(\sum_{t=0}^{\infty} u_t^* u_t^{*T}) \mathrm{Tr}((\mathcal{P}_\Delta Q)^T P_w (\mathcal{P}_\Delta Q)) \tag{20d}$$

If gradient (19a) and (19b) are 0, then (20d) is 0. So all critical points are global minimums.

3 Summary

In this project we consider the learning to learn problem with a quadratic function to minimize over. We restrict our perspective to gradient descent and heavy ball method, and give the best algorithm parameter for optimization. It is actually a policy gradient method applied in a two dimensional space, and we can see that, when projected to the momentum space, the function is actually convex. Even if joint in step size and momentum space, it is not convex, we can still show that the critical points are all minimizers. Thus we can still use gradient based algorithm to solve this learning to learn problem.

Currently the model is very simple, with two variables, so this work is only a start point. We are interested in that, if the algorithm class to learn is more broad, or the objective function is not simple quadratic, is it still possible to do this and land on a global minimum? It's not clear, but our conjecture is, at least for convex objective functions, if locally it can be approximated by quadratic, then learning to learn may not give a too bad optimization algorithm.

References

[1] Y. Nesterov, “A method of solving a convex programming problem with convergence rate o (1/k2).”

[2] H. Lin, J. Mairal, and Z. Harchaoui, “Catalyst acceleration for first-order convex optimization: from theory to practice,” *arXiv preprint arXiv:1712.05654*, 2017.

[3] C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui, “Catalyst acceleration for gradient-based non-convex optimization,” *arXiv preprint arXiv:1703.10993*, 2017.

[4] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, and N. de Freitas, “Learning to learn by gradient descent by gradient descent,” in *Advances in Neural Information Processing Systems*, 2016, pp. 3981–3989.

[5] M. Fazel, R. Ge, S. M. Kakade, and M. Mesbahi, “Global convergence of policy gradient methods for linearized control problems,” *arXiv preprint arXiv:1801.05039*, 2018.

A Positive semidefiniteness of heavy ball second derivative with respect to γ

We prove the Hessian

$$\nabla^2 L_{0,i}(\gamma) = \frac{4(\bar\lambda_i \nu_i - 8)}{(\bar\lambda_i + 2\nu_i + 4)^3} - \frac{2\bar\lambda_i}{(\bar\lambda_i + 2\nu_i + 4)^2} + \frac{4}{(\bar\lambda_i + \nu_i)^3} \tag{21a}$$

$$- \frac{64}{(\bar\lambda_i + \nu_i)(\bar\lambda_i + 2\nu_i + 4)^3} - \frac{32}{(\bar\lambda_i + \nu_i)^2(\bar\lambda_i + 2\nu_i + 4)^2} - \frac{16}{(\bar\lambda_i + \nu_i)^3(\bar\lambda_i + 2\nu_i + 4)} \tag{21b}$$

is positive when $\gamma \in [0, 1)$, $\bar\lambda_i \in (0, 2)$. Note that $\bar\lambda_i + \nu_i < 0$, $\bar\lambda_i + 2\nu_i + 4 > 0$, so it suffices to prove that

$$(\bar\lambda_i + \nu_i)^3(\bar\lambda_i + 2\nu_i + 4)^3 \nabla^2 L_{0,i}(\gamma) < 0 \tag{22}$$

The left hand side is equal to

$$\begin{aligned}
&(\bar\lambda_i + \nu_i)^3(\bar\lambda_i + 2\nu_i + 4)^3 \nabla^2 L_{0,i}(\gamma) && (23a)\\
=& -4\bar\lambda_i^3 + 6\bar\lambda_i^2 + 30\bar\lambda_i^2\gamma - 6\bar\lambda_i^2\gamma^2 + 8\bar\lambda_i\gamma^3 + 2\bar\lambda_i^2\gamma^3 - 72\bar\lambda_i\gamma + (192 - 40\bar\lambda_i + 64\gamma^2)(\gamma - 1) && (23b)\\
\le& (8\bar\lambda_i + 2\bar\lambda_i^2)\gamma^3 - 6\bar\lambda_i^2\gamma^2 + (30\bar\lambda_i^2 - 72\bar\lambda_i)\gamma + (6\bar\lambda_i^2 - 4\bar\lambda_i^3) + 112(\gamma - 1) && (23c)\\
\le& (8\bar\lambda_i + 2\bar\lambda_i^2) - 6\bar\lambda_i^2 + (30\bar\lambda_i^2 - 72\bar\lambda_i) + (6\bar\lambda_i^2 - 4\bar\lambda_i^3) && (23d)\\
=& \bar\lambda_i(-4\bar\lambda_i^2 + 32\bar\lambda_i - 64) && (23e)\\
=& \bar\lambda_i(-4(\bar\lambda_i - 4)^2) && (23f)\\
<& 0. && (23g)
\end{aligned}$$

(23d) is because $0 \le \gamma < 1$, the gradient with respect to γ is

$$\begin{aligned}
&3(8\bar\lambda_i + 2\bar\lambda_i^2)\gamma^2 - 12\bar\lambda_i^2\gamma + (30\bar\lambda_i - 72)\bar\lambda_i + 112 && (24a)\\
\ge& -12\bar\lambda_i^2 + (30\bar\lambda_i - 72)\bar\lambda_i + 112 && (24b)\\
=& 18\bar\lambda_i^2 - 72\bar\lambda_i + 112 && (24c)\\
=& 18(\bar\lambda_i - 2)^2 + 40 && (24d)\\
>& 0. && (24e)
\end{aligned}$$

So it’s increasing and maximizer occurs at $\gamma = 1$. (23g) is strict as $\bar\lambda_i \le 2$.