Asynchronous Parallel Iteration

Yiping Lu[1]

[1]School of Mathmetical Science
Peking University

Topics in Modern Information Processing, PKU, 2017

Outline

1. Coordinate Friendly Structure
   - Coordinate Update Algorithmic Framework
   - Coordinate Friendly Operator
   - Composite Coordinate Friendly Operators
   - Operator Splitting
2. Asynchronous Parallel Iteration
   - Arbitrary Delay Case
     - Converge results
   - True Delays

Outline

Coordinate update

The general framework can be written as

- set $k \leftarrow 0$ and initialize $x^0 \in \mathbb{H} = \mathbb{H}_1 \times \mathbb{H}_2 \times \cdots \mathbb{H}_m$
- while **not converged** to do
  - select an index $i_k \in [m]$;
  - update x_i^{k+1} for $i = i_k$ while keeping $x_i^{k+1} = x_i^k, \forall i \neq i_k$
  - $k \leftarrow k + 1$

There is a sequence of coordinate indices $i_1, i_2, \cdots, i_n$ chosen according to one of the following rules:

- cyclic
- cyclic permutation
- random
- greedy

Then update $\boxed{x_i^{k+1} = x_i^k - \eta_k (x^k - Tx_k)_i}$ for $i = i_k$ while keeping $x_i^{k+1} = x_i^k, \forall i \neq i_k$

Examples:

- Gauss-Seidel iteration
- alternating projection for finding a point in the intersection of two sets.
- ADMM for solving monotropic programs
- Douglas-Rachford Splitting(DRS) for finding a zero the sum of two operators.

Coordinate Update

There is a sequence of coordinate indices $i_1, i_2, \cdots, i_n$ chosen according to one of the following rules:

- cyclic
- cyclic permutation
- random
- greedy

Then update $\boxed{x_i^{k+1} = x_i^k - \eta_k(x^k - Tx_k)_i}$ for $i = i_k$ while keeping $x_i^{k+1} = x_i^k, \forall i \neq i_k$

Examples:

- Gauss-Seidel iteration
- alternating projection for finding a point in the intersection of two sets.
- ADMM for solving monotropic programs
- Douglas-Rachford Splitting(DRS) for finding a zero the sum of two operators.

Coordinate Update

There is a sequence of coordinate indices $i_1, i_2, \cdots, i_n$ chosen according to one of the following rules:

- cyclic
- cyclic permutation
- random
- greedy

Then update $\boxed{x_i^{k+1} = x_i^k - \eta_k (x^k - Tx_k)_i}$ for $i = i_k$ while keeping $x_i^{k+1} = x_i^k, \forall i \neq i_k$

Examples:

- Gauss-Seidel iteration
- alternating projection for finding a point in the intersection of two sets.
- ADMM for solving monotropic programs
- Douglas-Rachford Splitting(DRS) for finding a zero the sum of two operators.

Coordinate Descent

In optimization, we solve one of the following subproblems:

- $(Tx^k)_i = \arg\min_{x_i} f(x^k_{i_-}, x_i, x^k_{i_+})$
- $(Tx^k)_i = \arg\min_{x_i} f(x^k_{i_-}, x_i, x^k_{i_+}) + \frac{1}{2\eta_k}||x_i - x_i^k||^2$
- $(Tx^k)_i = \arg\min_{x_i} \langle \nabla_i f(x^k), x_i \rangle + \frac{1}{2\eta_k}||x_i - x_i^k||^2$
- $(Tx^k)_i = \arg\min_{x_i} \langle \nabla_i f^{diff}(x^k), x_i \rangle + f_i^{prox}(x_i) + \frac{1}{2\eta_k}||x_i - x_i^k||^2$

For the last setting, letting

$$f(x) = f^{diff}(x) + \sum_{i=1}^{m} f_i^{prox}(x_i)$$

Coordinate Descent

In optimization, we solve one of the following subproblems:

- $(Tx^k)_i = \arg\min_{x_i} f(x^k_{i_-}, x_i, x^k_{i_+})$
- $(Tx^k)_i = \arg\min_{x_i} f(x^k_{i_-}, x_i, x^k_{i_+}) + \frac{1}{2\eta_k}||x_i - x_i^k||^2$
- $(Tx^k)_i = \arg\min_{x_i} \langle \nabla_i f(x^k), x_i \rangle + \frac{1}{2\eta_k}||x_i - x_i^k||^2$
- $(Tx^k)_i = \arg\min_{x_i} \langle \nabla_i f^{diff}(x^k), x_i \rangle + f_i^{prox}(x_i) + \frac{1}{2\eta_k}||x_i - x_i^k||^2$

For the last setting, letting

$$f(x) = f^{diff}(x) + \sum_{i=1}^{m} f_i^{prox}(x_i)$$

In optimization, we solve one of the following subproblems:

- $(Tx^k)_i = \arg\min_{x_i} f(x^k_{i_-}, x_i, x^k_{i_+})$
- $(Tx^k)_i = \arg\min_{x_i} f(x^k_{i_-}, x_i, x^k_{i_+}) + \frac{1}{2\eta_k}||x_i - x_i^k||^2$
- $(Tx^k)_i = \arg\min_{x_i} \langle \nabla_i f(x^k), x_i \rangle + \frac{1}{2\eta_k}||x_i - x_i^k||^2$
- $(Tx^k)_i = \arg\min_{x_i} \langle \nabla_i f^{diff}(x^k), x_i \rangle + f_i^{prox}(x_i) + \frac{1}{2\eta_k}||x_i - x_i^k||^2$

For the last setting, letting

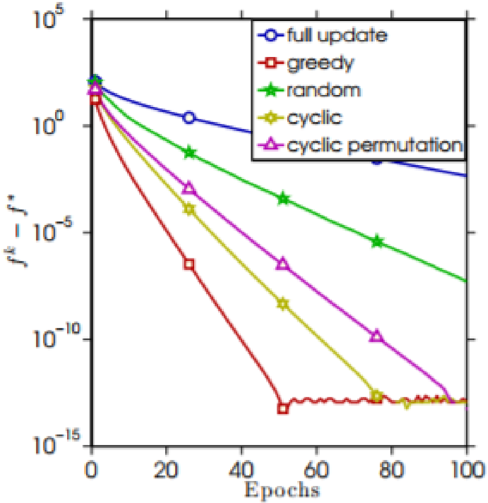
$$f(x) = f^{diff}(x) + \sum_{i=1}^{m} f_i^{prox}(x_i)$$

In optimization, we solve one of the following subproblems:

- $(Tx^k)_i = \arg\min_{x_i} f(x^k_{i_-}, x_i, x^k_{i_+})$
- $(Tx^k)_i = \arg\min_{x_i} f(x^k_{i_-}, x_i, x^k_{i_+}) + \frac{1}{2\eta_k}||x_i - x_i^k||^2$
- $(Tx^k)_i = \arg\min_{x_i} \langle \nabla_i f(x^k), x_i \rangle + \frac{1}{2\eta_k}||x_i - x_i^k||^2$
- $(Tx^k)_i = \arg\min_{x_i} \langle \nabla_i f^{diff}(x^k), x_i \rangle + f_i^{prox}(x_i) + \frac{1}{2\eta_k}||x_i - x_i^k||^2$

For the last setting, letting

$$f(x) = f^{diff}(x) + \sum_{i=1}^{m} f_i^{prox}(x_i)$$


full update
greedy
random
cyclic
cyclic permutation
10^5
10^0
10^{-5}
10^{-10}
10^{-15}
$f^k - f^*$
0
20
40
60
80
100
Epochs

Parallel Update

Sync-parallel(Jacobi) Update specifies a sequence of index subsets $\mathbb{I}_1, \mathbb{I}_2, \cdots \subset [m]$, and at each iteration k the coordinates in $\mathbb{I}_k$ are updated in parallel by multiple agents: $\boxed{x_i^{k+1} = x_i^k - \eta_k(x^k - Tx^k)_i}$

Async-parallel Update a set of agents still perform parallel updates, but synchronization is eliminated or weaked.Hence, each agent continuously applies update, wich reads x from and writes x_i back to the shared memory. k **increases whenever any agent completes an update.** Formally $\boxed{x_i^{k+1} = x_i^k - \eta_k((I - T)x^{k-d_k})_i}$

The lack of synchronization often results in computation with out-of-date information.

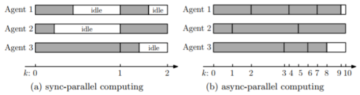


Figure 2: Sync-parallel computing (left) versus async-parallel computing (right). On the left, all the agents must wait at idle (white boxes) until the slowest agent has finished.

Parallel Update

Sync-parallel(Jacobi) Update specifies a sequence of index subsets $\mathbb{I}_1, \mathbb{I}_2, \cdots \subset [m]$, and at each iteration k the coordinates in $\mathbb{I}_k$ are updated in parallel by multiple agents: $\boxed{x_i^{k+1} = x_i^k - \eta_k(x^k - Tx^k)_i}$

Async-parallel Update a set of agents still perform parallel updates, but synchronization is eliminated or weaked.Hence, each agent continuously applies update, wich reads x from and writes x_i back to the shared memory. k **increases whenever any agent completes an update.** Formally $\boxed{x_i^{k+1} = x_i^k - \eta_k((I - T)x^{k-d_k})_i}$

The lack of synchronization often results in computation with out-of-date information.

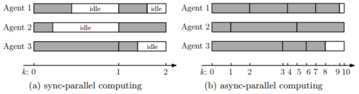


Figure 2: Sync-parallel computing (left) versus async-parallel computing (right). On the left, all the agents must wait at idle (white boxes) until the slowest agent has finished.

Outline

1. Coordinate Friendly Structure
   - Coordinate Update Algorithmic Framework
   - Coordinate Friendly Operator
   - Composite Coordinate Friendly Operators
   - Operator Splitting

2. Asynchronous Parallel Iteration
   - Arbitrary Delay Case
     - Converge results
   - True Delays

Notation

We assume our variable *x* consist of *m* coordinates:

$$x^0 \in \mathbb{H} = \mathbb{H}_1 \times \mathbb{H}_2 \times \cdots \mathbb{H}_m$$

For simplicity we assume that $\mathbb{H}_i$ are finite dimensional real Hilbert spaces.

Definition

We let $m[a \to b]$ denote the number of basic operations that it takes to compute the quantity *b* from the input *a*

least square1

Example

Consider the least square problem

$$\min f(x) := \frac{1}{2}||Ax - b||_2^2$$

Here $A \in \mathbb{R}^{p\times m}$, $b \in \mathbb{R}^p$

The full update can be written as

$$Tx := x - \eta\nabla f(x) = x - \eta A^T Ax + \eta A^T b$$

For the $i-$th coordinate:

$$(Tx)_i = (A^T A)_{i,:} \cdot x - (A^T b)_i$$

Assuming $A^T A$ and $A^T b$ is already computed

$$m[x \to (Tx)_i] = O(m) = O(\frac{1}{m} x \to (Tx)_i)$$

Example

Consider the least square problem

$$\min f(x) := \frac{1}{2}||Ax - b||_2^2$$

Here $A \in \mathbb{R}^{p\times m}$, $b \in \mathbb{R}^p$

The full update can be written as

$$Tx := x - \eta\nabla f(x) = x - \eta A^T Ax + \eta A^T b$$

For the $i-$th coordinate:

$$(Tx)_i = (A^T A)_{i,:} \cdot x - (A^T b)_i$$

Assuming $A^T A$ and $A^T b$ is already computed

$$m[x \to (Tx)_i] = O(m) = O(\frac{1}{m} x \to (Tx)_i)$$

Example

Consider the least square problem

$$\min f(x) := \frac{1}{2}||Ax - b||_2^2$$

Here $A \in \mathbb{R}^{p\times m}, b \in \mathbb{R}^p$

The full update can be written as

$$Tx := x - \eta\nabla f(x) = x - \eta A^T Ax + \eta A^T b$$

Suppose we have x, Tx and need to update Tx^{k+1}(For if we have Tx^k, it is easy to get x^{k+1})

$$Tx^{k+1} = Tx^k + x^{k+1} - x^k - \eta(x_{i_k}^{k+1} - x_{i_k}^k)(A^T A)_{:,i_k}$$

Example

Consider the least square problem

$$\min f(x) := \frac{1}{2}||Ax - b||_2^2$$

Here $A \in \mathbb{R}^{p\times m}, b \in \mathbb{R}^p$

The full update can be written as

$$Tx := x - \eta\nabla f(x) = x - \eta A^T Ax + \eta A^T b$$

Suppose we have x, Tx and need to update Tx^{k+1}(For if we have Tx^k, it is easy to get x^{k+1})

$$Tx^{k+1} = Tx^k + x^{k+1} - x^k - \eta(x_{i_k}^{k+1} - x_{i_k}^k)(A^T A)_{:,i_k}$$

Example

Consider the least square problem

$$\min f(x) := \frac{1}{2}||Ax - b||_2^2$$

Here $A \in \mathbb{R}^{p\times m}, b \in \mathbb{R}^p$

Suppose we have x, Tx and need to update Tx^{k+1}(For if we have Tx^k, it is easy to get x^{k+1})

$$Tx^{k+1} = Tx^k + x^{k+1} - x^k - \eta(x_{i_k}^{k+1} - x_{i_k}^k)(A^T A)_{:,i_k}$$

we have

$$m[\{x^k, Tx^k, x^{k+1}\} \to Tx^{k+1}] = O(\frac{1}{m}m[x^{k+1} \to Tx^{k+1}])$$

Corrdinate Friendly Operator

Definition

- **Type1 CF**

$$m[x \to (Tx)_i] = O(\frac{1}{m} x \to (Tx)_i)$$

- **Type2 CF** for any i, x and $x^+ := (x_1, \cdots, (Tx)_i, \cdots, x_m)$ we have

$$m[\{x, Tx, x^+\} \to Tx^+] = O(\frac{1}{m} m[x^+ \to Tx^+])$$

Example

Consider the least square problem

$$\min f(x) := \frac{1}{2}||Ax - b||_2^2$$

Here $A \in \mathbb{R}^{p\times m}$, $b \in \mathbb{R}^p$

When $p << m$ we should avoid computing $A^T A$, it is cheaper to compute $A^T(Ax)$

$$\begin{aligned}(Tx^k)_{i_k} &= x^k_{i_k} - \eta(A^T(Ax^k) - A^T b)_{i_k} \\ &= x^k_{i_k} - \eta(A^T_{i_k,:}(Ax^k) - A^T_{i_k,:}b)\end{aligned}$$

That is say we have

$$m[\{x^k, Ax^k\} \to \{x^{k+1}, Ax^{k+1}\}] = O(\frac{1}{m}m[x \to Tx^k])$$

Definition

CF Operator We say that an operator $T : \mathbb{H} \to \mathbb{H}$ is **CF** if for any i, x and $x^+ := (x_1, \cdots, (Tx)_i, \cdots, x_m)$, the following holds

$$m[\{x, M(x)\} \to \{x^+, M(x^+)\}] = O(\frac{1}{m} m[x \to Tx])$$

Theorem

Type1 and Type2 CF operator is CF operator!

Definition

- separable operator
- nearly-separable operator
- non-separable operator

Remark

Not all nearly-separable operators are Type2 CG operator. Indeed consider a sparse matrix $A \in \mathbb{R}^{m \times m}$ whose non-zero entries are only located in the last column. Let $Tx = Ax$, then $Tx^{+} = Tx + (x_m^{+} - x_m)A_{:,m}$ takes m operations. But $Tx^{+} = x_m^{+}A_{:,m}$ also takes m operation.

Example

- (diagonal matrix) $A = diag(a_{1,1}, \cdots, a_{m,m})$, $T : x \to Ax$ is separable
- Gradient and proximal maps of a separbale function$f = \sum_{i=1}^{m} f_i(x_i)$.
- projection to a box, indeed $(proj_B(x))_i = \max(b_i, \min(a_i, x_i))$
- squared hinge loss function, consider for $a, x \in \mathbb{R}^m$

$$f(x) := \frac{1}{2}(\max(0, 1 - \beta a^T x))^2$$

consider $Tx = \nabla f(x) = -\beta \max(0, 1 - \beta a^T x)a$
Let $M(x) = a^T x$, we can know it is a CF operator.

Outline

Combinations of operators

Example

Scalar map pre-composing affine function. Let $a_j \in \mathbb{R}^m, b_j \in \mathbb{R}$, and $\phi_j : \mathbb{R} \to \mathbb{R}$ be differentiable function, $j \in [p]$. Let

$$f(x) = \sum_{j=1}^{p} \phi_j(a_j^T x + b_j)$$

Then ∇f is CF

Let $T_1 y = A^T y, T_2 y := [\phi_1'(y_1), \cdots, \phi_p'(y_p)], T_3 x := Ax + b$, where $A = [a_1^T; a_2^T; \cdots; a_p^T], b = [b_1; b_2; \cdots, b_p]$. Then $\nabla f = T_1 \circ T_2 \circ T_3 x$ and let $M(x) := T_3 x$

Combinations of operators

Now $T_1 y = A^T y$, $T_2 y := [\phi_1'(y_1), \cdots, \phi_p'(y_p)]$, $T_3 x := Ax + b$, $\nabla f = T_1 \circ T_2 \circ T_3 x$ and let $M(x) := T_3 x$

- calculate $T_2 \circ T_3 x$ from $T_3 x$ for $O(p)$ operations.
- Compute $\nabla_i f(x)$(thus x^+) from $T_2 \circ T_3 x$ for $O(p)$ operations.
- update the $T_3 x^+$ by $O(p)$ operations.

Why effecient? T_1 Type1 CF, T_2 separable and T_3 type2, so that $T_1 \circ T_2$ still Type1 and $T_2 \circ T_3$ CF.
Attention: $T_2 \circ T_3$ is neither CF1 nor CF2.

Now $T_1 y = A^T y$, $T_2 y := [\phi_1'(y_1), \cdots, \phi_p'(y_p)]$, $T_3 x := Ax + b$, $\nabla f = T_1 \circ T_2 \circ T_3 x$ and let $M(x) := T_3 x$

- calculate $T_2 \circ T_3 x$ from $T_3 x$ for $O(p)$ operations.
- Compute $\nabla_i f(x)$(thus x^+) from $T_2 \circ T_3 x$ for $O(p)$ operations.
- update the $T_3 x^+$ by $O(p)$ operations.

Why effecient? T_1 Type1 CF, T_2 separable and T_3 type2, so that $T_1 \circ T_2$ still Type1 and $T_2 \circ T_3$ CF.
Attention: $T_2 \circ T_3$ is neither CF1 nor CF2.

Now $T_1 y = A^T y$, $T_2 y := [\phi_1'(y_1), \cdots, \phi_p'(y_p)]$, $T_3 x := Ax + b$, $\nabla f = T_1 \circ T_2 \circ T_3 x$ and let $M(x) := T_3 x$

- calculate $T_2 \circ T_3 x$ from $T_3 x$ for $O(p)$ operations.
- Compute $\nabla_i f(x)$(thus x^+) from $T_2 \circ T_3 x$ for $O(p)$ operations.
- update the $T_3 x^+$ by $O(p)$ operations.

Why effecient? T_1 Type1 CF, T_2 separable and T_3 type2, so that $T_1 \circ T_2$ still Type1 and $T_2 \circ T_3$ CF.

Attention: $T_2 \circ T_3$ is neither CF1 nor CF2.

Now $T_1 y = A^T y$, $T_2 y := [\phi_1'(y_1), \cdots, \phi_p'(y_p)]$, $T_3 x := Ax + b$, $\nabla f = T_1 \circ T_2 \circ T_3 x$ and let $M(x) := T_3 x$

- calculate $T_2 \circ T_3 x$ from $T_3 x$ for $O(p)$ operations.
- Compute $\nabla_i f(x)$(thus x^+) from $T_2 \circ T_3 x$ for $O(p)$ operations.
- update the $T_3 x^+$ by $O(p)$ operations.

Why effecient? T_1 Type1 CF, T_2 separable and T_3 type2, so that $T_1 \circ T_2$ still Type1 and $T_2 \circ T_3$ CF.
Attention: $T_2 \circ T_3$ is neither CF1 nor CF2.

Now $T_1 y = A^T y$, $T_2 y := [\phi_1'(y_1), \cdots, \phi_p'(y_p)]$, $T_3 x := Ax + b$, $\nabla f = T_1 \circ T_2 \circ T_3 x$ and let $M(x) := T_3 x$

- calculate $T_2 \circ T_3 x$ from $T_3 x$ for $O(p)$ operations.
- Compute $\nabla_i f(x)$(thus x^+) from $T_2 \circ T_3 x$ for $O(p)$ operations.
- update the $T_3 x^+$ by $O(p)$ operations.

Why effecient? T_1 Type1 CF, T_2 separable and T_3 type2, so that $T_1 \circ T_2$ still Type1 and $T_2 \circ T_3$ CF.
Attention: $T_2 \circ T_3$ is neither CF1 nor CF2.

Definition

(Cheap Operator). For a composite operator $T = T_1 \circ T_2 \circ \cdots \circ T_p$, an operator $T_i : \mathbb{H} \to \mathbb{G}$ is cheap if $m[x \to T_i x]$ is less than or equal to the number of remaining coordinate-update operations, in order of magnitude.

Definition

(Easy-to-maintain Operator). For a composite operator $T = T_1 \circ T_2 \circ \cdots \circ T_p$, the operator $T_p : \mathbb{H} \to \mathbb{G}$ is easy-to maintain, if for any x, i, x^+ satisfying $m[\{x, T_p x, x^+\} \to T_p x^+]$ is less than or equal to the number of remaining coordinate-update operations, in order of magnitude, or belongs to $O(\frac{1}{dim\mathbb{G}})m[x^+ \to Tx^+]$

Outline

Operator Splitting

Definition

A common firmly-nonexpansive operator is the resolvent of a maximally monotone map T, written as

$$J_A := (I + A)^{-1}$$

A reflective resolvent is

$$R_A := 2J_A - I$$

Example

$$prox_{\gamma f} = (I + \gamma \partial f)^{-1}$$

Outline

Notation and assumptions

We consider a block-structructured optimization problem

$$\min_{x \in \mathbb{R}^n} F(x) = f(x_1, \cdots, x_m) + \sum_{i=1}^{m} r_i(x_i) \tag{1}$$

Definition

A point x^* is called critical point of(1) if $0 \in \nabla f(x^*) + \partial R(x^*)$

Every time we use the proximal gradient to do the update

$$x_i^{k+1} \leftarrow prox_{\eta r_i}(x_i^k - \eta \nabla_i f(\hat{x}^k))$$

i is choosen random uniformly every time.

- Problem(1) has at least one solution, the solution set is denote as X^*
- ∇f is Lipschitz continuous with constant L_f. For each $i \in [m]$, fixing all block coordinates but the $i-$th one, $\nabla f(x)$ and $\nabla_i f(x)$ are Lipschitz continuous with x_i with constants L_r and L_c, the condition number is denoted as $\kappa = \frac{L_r}{L_c}$
- For each $k \geq 1$, the reading $\hat{x}^k$ is consistent and delayed by j_k, namely $\hat{x}^k = x^{x-j_k}$, and delay follows an identical distribution

$$Prob(j_k) = t = q_t, t = 0, 1, 2, \cdots, \forall k$$

Nonconvex case

Theorem

Convergence for the nonconvex smooth case. *let* $\{x^k\}_{k\geq 1}$ *be generated from the algorithm. Assume*

$$T := \mathbb{E}[j_k] < \infty$$

If the stepsize is take as $0 < \eta < \frac{1/L_c}{1+2\kappa T/\sqrt{m}}$, *then*

$$\lim_{k\to\infty} \mathbb{E}||\nabla f(x^k)|| = 0$$

and any limit point of $\{x^k\}_{k\geq 1}$ *is almost surely a critical point.*

Outline

Continuous-time Analysis

Let t be time in this section, consider the ODE

$$\dot{x}(t) = -\eta \nabla f(\hat{x}(t))$$

If there is no delay, easily set $\hat{x}(t) = x(t)$, the ODE describe a gradient flow, which monotonically decreases $f(x(t))$ for $\frac{d}{dt} f(x(t)) = \langle \nabla f(x(t)), \dot{x}(t) \rangle = -\frac{1}{\eta} ||\dot{x}(t)||_2^2$ Instead, we allow delays and impose the bound $c > 0$ on the delays:

$$||\hat{x}(t) - x(t)||_2 \leq \int_{t-c}^{t} ||\dot{x}(s)||_2 ds$$

Continuous-time Analysis

Let t be time in this section, consider the ODE

$$\dot{x}(t) = -\eta \nabla f(\hat{x}(t))$$

If there is no delay, easily set $\hat{x}(t) = x(t)$, the ODE describe a gradient flow, which monotonically decreases $f(x(t))$ for $\frac{d}{dt} f(x(t)) = \langle \nabla f(x(t)), \dot{x}(t) \rangle = -\frac{1}{\eta} ||\dot{x}(t)||_2^2$ Instead, we allow delays and impose the bound $c > 0$ on the delays:

$$||\hat{x}(t) - x(t)||_2 \leq \int_{t-c}^{t} ||\dot{x}(s)||_2 ds$$

Continuous-time Analysis

Let t be time in this section, consider the ODE

$$\dot{x}(t) = -\eta \nabla f(\hat{x}(t))$$

If there is no delay, easily set $\hat{x}(t) = x(t)$, the ODE describe a gradient flow, which monotonically decreases $f(x(t))$ for $\frac{d}{dt} f(x(t)) = \langle \nabla f(x(t)), \dot{x}(t) \rangle = -\frac{1}{\eta} ||\dot{x}(t)||_2^2$ Instead, we allow delays and impose the bound $c > 0$ on the delays:

$$||\hat{x}(t) - x(t)||_2 \leq \int_{t-c}^{t} ||\dot{x}(s)||_2 ds$$

Let t be time in this section, consider the ODE

$$\dot{x}(t) = -\eta \nabla f(\hat{x}(t))$$

We lose monotonicity

Proof.

$$\begin{aligned}
\frac{d}{dt} f(x(t)) &= \langle \nabla f(\hat{x}(t)), \dot{x}(t) \rangle + \langle \nabla f(x(t)) - \nabla f(\hat{x}(t)), \dot{x}(t) \rangle \\
&\leq -\frac{1}{\eta} ||\dot{x}(t)||_2^2 + L||x(t) - \hat{x}(t)||_2 \cdot ||\dot{x}(t)||_2 \\
&\leq -\frac{1}{2\eta} ||\dot{x}(t)||_2^2 + \frac{\eta c L^2}{2} \int_{t-c}^{t} ||\dot{x}(s)||_2^2 ds
\end{aligned}$$

□

Let t be time in this section, consider the ODE

$$\dot{x}(t) = -\eta \nabla f(\hat{x}(t))$$

We design an **Energy function** with both f and a weighted total keinetic term, where $\gamma > 0$.

$$\xi(t) = f(x(t)) + \gamma \int_{t-c}^{t} (s - (t - c))||\dot{x}(s)||_2^2 ds \tag{2}$$

$\xi(t)$ has the time derivative

$$\begin{aligned}\dot{\xi}(t) &= \frac{d}{dt} f(x(t)) + \gamma c ||x(t)||_2^2 - \gamma \int_{t-c}^{t} ||\dot{x}(s)||_2^2 ds \\ &\leq -(\frac{1}{\eta} - \gamma)||\dot{x}(t)||_2^2 - (\gamma - \frac{ncL^2}{2}) \int_{t-c}^{t} ||\dot{x}(s)||_2^2 ds\end{aligned}$$

Discrete Analysis

We can define the Lyapunov function

$$\xi_k := f(x^k) + \frac{L}{2\epsilon} \sum_{i=k-\tau}^{k-1} (i - (k-\tau) + 1)||\Delta^i||_2^2$$

The proof is like the one given before

- $f(x^{k+1}) - f(x^k) \leq \frac{L}{2\epsilon} \sum_{i=k-\tau}^{k-1} ||\Delta^i||_2^2 + [\frac{L(\tau\epsilon+1)}{2} - \frac{L}{\gamma}]||\Delta^k||_2^2$
- $\xi_k - \xi_{k+1} \geq \frac{1}{2}(\frac{1}{\gamma} - \frac{1}{2} - \tau)L \cdot ||\Delta^k||_2^2$

Theorem

Converge Rate.

$$\lim_k ||\nabla f(x^k)||_2 = 0, \lim_{1\leq i\leq k} ||\nabla f(x^k)||_2 = o(1/\sqrt{k})$$

The same magnitude as standard gradient descent

Discrete Analysis

We can define the Lyapunov function

$$\xi_k := f(x^k) + \frac{L}{2\epsilon} \sum_{i=k-\tau}^{k-1} (i - (k-\tau) + 1)||\Delta^i||_2^2$$

The proof is like the one given before

- $f(x^{k+1}) - f(x^k) \leq \frac{L}{2\epsilon} \sum_{i=k-\tau}^{k-1} ||\Delta^i||_2^2 + [\frac{L(\tau\epsilon+1)}{2} - \frac{L}{\gamma}]||\Delta^k||_2^2$
- $\xi_k - \xi_{k+1} \geq \frac{1}{2}(\frac{1}{\gamma} - \frac{1}{2} - \tau)L \cdot ||\Delta^k||_2^2$

Theorem

Converge Rate.

$$\lim_k ||\nabla f(x^k)||_2 = 0, \lim_{1\leq i\leq k} ||\nabla f(x^k)||_2 = o(1/\sqrt{k})$$

The same magnitude as standard gradient descent

Discrete Analysis

We can define the Lyapunov function

$$\xi_k := f(x^k) + \frac{L}{2\epsilon}\sum_{i=k-\tau}^{k-1}(i-(k-\tau)+1)||\Delta^i||_2^2$$

The proof is like the one given before

- $f(x^{k+1}) - f(x^k) \leq \frac{L}{2\epsilon}\sum_{i=k-\tau}^{k-1}||\Delta^i||_2^2 + [\frac{L(\tau\epsilon+1)}{2} - \frac{L}{\gamma}]||\Delta^k||_2^2$
- $\xi_k - \xi_{k+1} \geq \frac{1}{2}(\frac{1}{\gamma} - \frac{1}{2} - \tau)L\cdot||\Delta^k||_2^2$

Theorem

Converge Rate.

$$\lim_k ||\nabla f(x^k)||_2 = 0, \lim_{1\leq i\leq k} ||\nabla f(x^k)||_2 = o(1/\sqrt{k})$$

The same magnitude as standard gradient descent