Gradient Flow. Accelerated gradient flow.

Daniil Merkulov

Optimization methods. MIPT

Gradient Flow

Gradient Flow intuition

- Antigradient $-\nabla f(x)$ indicates the direction of steepest descent at the point x.

Gradient Flow intuition

- Antigradient $-\nabla f(x)$ indicates the direction of steepest descent at the point x.
- Note also, that the antigradient solves the problem of minimization the Taylor linear approximation of the function on the Euclidian ball

$$
\min_{\delta x \in \mathbb{R}^n} \nabla f(x_0)^\top \delta x
$$
$$
\text{s.t. } \delta x^\top \delta x = \varepsilon^2
$$

Gradient Flow intuition

- Antigradient $-\nabla f(x)$ indicates the direction of steepest descent at the point x.
- Note also, that the antigradient solves the problem of minimization the Taylor linear approximation of the function on the Euclidian ball
$$\min_{\delta x \in \mathbb{R}^n} \nabla f(x_0)^\top \delta x$$
$$\text{s.t. } \delta x^\top \delta x = \varepsilon^2$$
- The gradient descent is the most classical iterative algorithm to minimize differentiable functions. It comes with a plenty of forms: steepest, stochastic, pre-conditioned, conjugate, proximal, projected, accelerated, etc.

Gradient Flow intuition

- Antigradient $-\nabla f(x)$ indicates the direction of steepest descent at the point x.
- Note also, that the antigradient solves the problem of minimization the Taylor linear approximation of the function on the Euclidian ball

$$
\min_{\delta x \in \mathbb{R}^n} \nabla f(x_0)^\top \delta x
$$

$$
\text{s.t. } \delta x^\top \delta x = \varepsilon^2
$$

- The gradient descent is the most classical iterative algorithm to minimize differentiable functions. It comes with a plenty of forms: steepest, stochastic, pre-conditioned, conjugate, proximal, projected, accelerated, etc.

$$
x_{k+1} = x_k - \alpha_k \nabla f(x_k)
$$

Gradient Flow intuition

- Antigradient $-\nabla f(x)$ indicates the direction of steepest descent at the point x.
- Note also, that the antigradient solves the problem of minimization the Taylor linear approximation of the function on the Euclidian ball

$$
\begin{aligned}
&\min_{\delta x \in \mathbb{R}^n} \nabla f(x_0)^\top \delta x \\
&\text{s.t. } \delta x^\top \delta x = \varepsilon^2
\end{aligned}
$$

- The gradient descent is the most classical iterative algorithm to minimize differentiable functions. It comes with a plenty of forms: steepest, stochastic, pre-conditioned, conjugate, proximal, projected, accelerated, etc.

$$
\begin{aligned}
x_{k+1} &= x_k - \alpha_k \nabla f(x_k) \\
x_{k+1} - x_k &= -\alpha_k \nabla f(x_k)
\end{aligned}
$$

Gradient Flow intuition

- Antigradient $-\nabla f(x)$ indicates the direction of steepest descent at the point x.
- Note also, that the antigradient solves the problem of minimization the Taylor linear approximation of the function on the Euclidian ball

$$
\min_{\delta x \in \mathbb{R}^n} \nabla f(x_0)^\top \delta x
$$

$$
\text{s.t. } \delta x^\top \delta x = \varepsilon^2
$$

- The gradient descent is the most classical iterative algorithm to minimize differentiable functions. It comes with a plenty of forms: steepest, stochastic, pre-conditioned, conjugate, proximal, projected, accelerated, etc.

$$
\begin{aligned}
x_{k+1} &= x_k - \alpha_k \nabla f(x_k) \\
x_{k+1} - x_k &= -\alpha_k \nabla f(x_k) \\
\frac{x_{k+1} - x_k}{\alpha_k} &= -\nabla f(x_k)
\end{aligned}
$$

Gradient Flow intuition

- Antigradient $-\nabla f(x)$ indicates the direction of steepest descent at the point x.
- Note also, that the antigradient solves the problem of minimization the Taylor linear approximation of the function on the Euclidian ball

$$
\min_{\delta x \in \mathbb{R}^n} \nabla f(x_0)^\top \delta x
$$
$$
\text{s.t. } \delta x^\top \delta x = \varepsilon^2
$$

- The gradient descent is the most classical iterative algorithm to minimize differentiable functions. It comes with a plenty of forms: steepest, stochastic, pre-conditioned, conjugate, proximal, projected, accelerated, etc.

$$
\begin{aligned}
x_{k+1} &= x_k - \alpha_k \nabla f(x_k) \\
x_{k+1} - x_k &= -\alpha_k \nabla f(x_k) \\
\frac{x_{k+1} - x_k}{\alpha_k} &= -\nabla f(x_k)
\end{aligned}
$$

- The gradient flow is essentially the limit of gradient descent when the step-size α_k tends to zero

Gradient Flow intuition

- Antigradient $-\nabla f(x)$ indicates the direction of steepest descent at the point x.
- Note also, that the antigradient solves the problem of minimization the Taylor linear approximation of the function on the Euclidian ball

$$
\begin{aligned}
&\min_{\delta x \in \mathbb{R}^n} \nabla f(x_0)^\top \delta x \\
&\text{s.t. } \delta x^\top \delta x = \varepsilon^2
\end{aligned}
$$

- The gradient descent is the most classical iterative algorithm to minimize differentiable functions. It comes with a plenty of forms: steepest, stochastic, pre-conditioned, conjugate, proximal, projected, accelerated, etc.

$$
\begin{aligned}
x_{k+1} &= x_k - \alpha_k \nabla f(x_k) \\
x_{k+1} - x_k &= -\alpha_k \nabla f(x_k) \\
\frac{x_{k+1} - x_k}{\alpha_k} &= -\nabla f(x_k)
\end{aligned}
$$

- The gradient flow is essentially the limit of gradient descent when the step-size α_k tends to zero

Gradient Flow intuition

- Antigradient $-\nabla f(x)$ indicates the direction of steepest descent at the point x.
- Note also, that the antigradient solves the problem of minimization the Taylor linear approximation of the function on the Euclidian ball

$$\min_{\delta x \in \mathbb{R}^n} \nabla f(x_0)^\top \delta x$$
$$\text{s.t. } \delta x^\top \delta x = \varepsilon^2$$

- The gradient descent is the most classical iterative algorithm to minimize differentiable functions. It comes with a plenty of forms: steepest, stochastic, pre-conditioned, conjugate, proximal, projected, accelerated, etc.

$$
\begin{aligned}
x_{k+1} &= x_k - \alpha_k \nabla f(x_k) \\
x_{k+1} - x_k &= -\alpha_k \nabla f(x_k) \\
\frac{x_{k+1} - x_k}{\alpha_k} &= -\nabla f(x_k)
\end{aligned}
$$

- The gradient flow is essentially the limit of gradient descent when the step-size α_k tends to zero

!

$$\frac{dx}{dt} = -\nabla f(x)$$

Gradient Flow

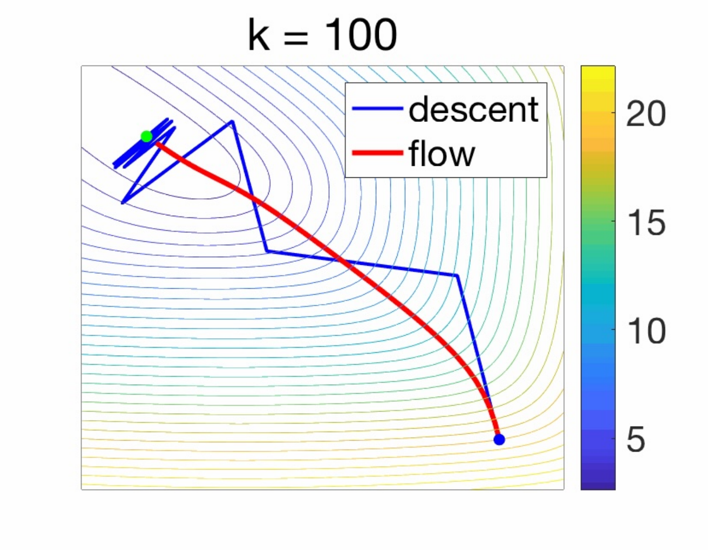

Рис. 1: Source

- **Simplified analyses.** The gradient flow has no step-size, so all the traditional annoying issues regarding the choice of step-size, with line-search, constant, decreasing or with a weird schedule are unnecessary.

Gradient Flow

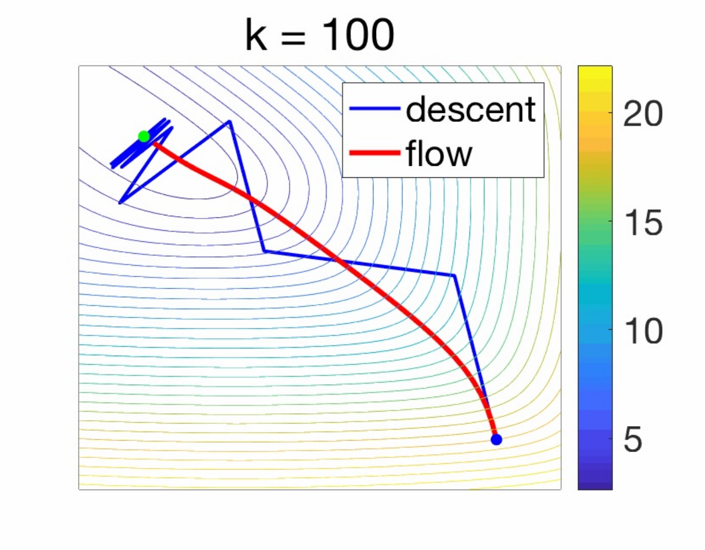

Рис. 1: Source

- **Simplified analyses.** The gradient flow has no step-size, so all the traditional annoying issues regarding the choice of step-size, with line-search, constant, decreasing or with a weird schedule are unnecessary.
- **Analytical solution in some cases.** For example, one can consider quadratic problem with linear gradient, which will form a linear ODE with known exact formula.

Gradient Flow

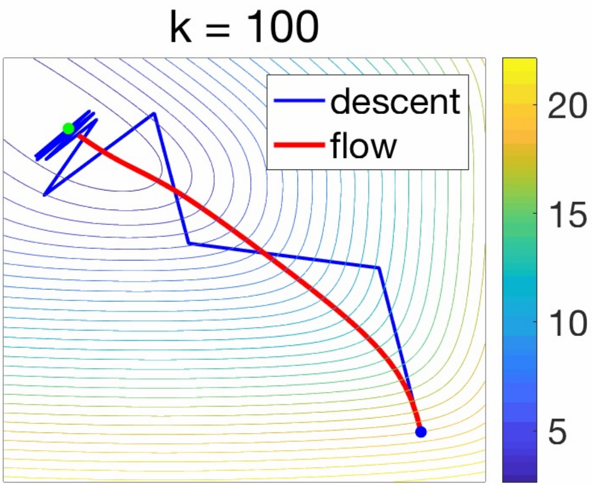

Рис. 1: Source

- **Simplified analyses.** The gradient flow has no step-size, so all the traditional annoying issues regarding the choice of step-size, with line-search, constant, decreasing or with a weird schedule are unnecessary.
- **Analytical solution in some cases.** For example, one can consider quadratic problem with linear gradient, which will form a linear ODE with known exact formula.
- **Different discretization leads to different methods.** We will see, that the continuous-time object is pretty rich in terms of the variety of produced algorithms. Therefore, it is interesting to study optimization from this perspsective.

Gradient Flow discretization

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

Gradient Flow discretization

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

Gradient Flow discretization

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

$$\boxed{x_{k+1} = x_k - \alpha \nabla f(x_k)} \qquad \text{(GD)}$$

Gradient Flow discretization

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

$$\boxed{x_{k+1} = x_k - \alpha \nabla f(x_k)} \qquad \text{(GD)}$$

Implicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_{k+1})$$

Gradient Flow discretization

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

$$\boxed{x_{k+1} = x_k - \alpha \nabla f(x_k)} \quad \text{(GD)}$$

Implicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_{k+1})$$

$$\frac{x_{k+1} - x_k}{\alpha} + \nabla f(x_{k+1}) = 0$$

Gradient Flow discretization

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

$$\boxed{x_{k+1} = x_k - \alpha \nabla f(x_k)} \qquad \text{(GD)}$$

Implicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_{k+1})$$

$$\frac{x_{k+1} - x_k}{\alpha} + \nabla f(x_{k+1}) = 0$$

$$\frac{x - x_k}{\alpha} + \nabla f(x)\Big|_{x = x_{k+1}} = 0$$

Gradient Flow discretization

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

$$\boxed{x_{k+1} = x_k - \alpha \nabla f(x_k)} \qquad \text{(GD)}$$

Implicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_{k+1})$$

$$\frac{x_{k+1} - x_k}{\alpha} + \nabla f(x_{k+1}) = 0$$

$$\left. \frac{x - x_k}{\alpha} + \nabla f(x) \right|_{x = x_{k+1}} = 0$$

$$\left. \nabla \left[\frac{1}{2\alpha} \|x - x_k\|_2^2 + f(x) \right] \right|_{x = x_{k+1}} = 0$$

Gradient Flow discretization

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

$$\boxed{x_{k+1} = x_k - \alpha \nabla f(x_k)} \qquad \text{(GD)}$$

Implicit Euler discretization:

$$\begin{aligned}
\frac{x_{k+1} - x_k}{\alpha} &= -\nabla f(x_{k+1}) \\
\frac{x_{k+1} - x_k}{\alpha} + \nabla f(x_{k+1}) &= 0 \\
\left.\frac{x - x_k}{\alpha} + \nabla f(x)\right|_{x = x_{k+1}} &= 0 \\
\left.\nabla \left[\frac{1}{2\alpha}\|x - x_k\|_2^2 + f(x)\right]\right|_{x = x_{k+1}} &= 0 \\
x_{k+1} = \text{arg}\min_{x \in \mathbb{R}^n} \left[f(x) + \frac{1}{2\alpha}\|x - x_k\|_2^2\right]
\end{aligned}$$

Gradient Flow discretization

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

$$\boxed{x_{k+1} = x_k - \alpha \nabla f(x_k)} \qquad \text{(GD)}$$

Implicit Euler discretization:

$$\begin{aligned}
\frac{x_{k+1} - x_k}{\alpha} &= -\nabla f(x_{k+1}) \\
\frac{x_{k+1} - x_k}{\alpha} + \nabla f(x_{k+1}) &= 0 \\
\left. \frac{x - x_k}{\alpha} + \nabla f(x) \right|_{x = x_{k+1}} &= 0 \\
\left. \nabla \left[\frac{1}{2\alpha} \|x - x_k\|_2^2 + f(x) \right] \right|_{x = x_{k+1}} &= 0 \\
x_{k+1} = \text{arg}\min_{x \in \mathbb{R}^n} \left[f(x) + \frac{1}{2\alpha} \|x - x_k\|_2^2 \right]
\end{aligned}$$

Gradient Flow discretization

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

$$\boxed{x_{k+1} = x_k - \alpha \nabla f(x_k)} \tag{GD}$$

Implicit Euler discretization:

$$\begin{aligned}
\frac{x_{k+1} - x_k}{\alpha} &= -\nabla f(x_{k+1}) \\
\frac{x_{k+1} - x_k}{\alpha} + \nabla f(x_{k+1}) &= 0 \\
\left. \frac{x - x_k}{\alpha} + \nabla f(x) \right|_{x = x_{k+1}} &= 0 \\
\left. \nabla \left[\frac{1}{2\alpha} \|x - x_k\|_2^2 + f(x) \right] \right|_{x = x_{k+1}} &= 0 \\
x_{k+1} = \text{arg}\min_{x \in \mathbb{R}^n} \left[f(x) + \frac{1}{2\alpha} \|x - x_k\|_2^2 \right]
\end{aligned}$$

$$\boxed{x_{k+1} = \text{prox}_{\alpha f}(x_k)} \tag{PPM}$$

Convergence analysis. Convex case.

1. Simplest proof of monotonic decrease of GF:

$$\frac{d}{dt}f(x(t)) = \nabla f(x(t))^\top \frac{dx(t)}{dt} = -\|\nabla f(x(t))\|_2^2 \leqslant 0.$$

   If f is bounded from below, then $f(x(t))$ will always converge as a non-increasing function which is bounded from below. It is straightforward, that GF converges to the stationary point, where $\nabla f = 0$ (potentailly including minima, maxima and saddle points).

Convergence analysis. Convex case.

1. Simplest proof of monotonic decrease of GF:

$$\frac{d}{dt}f(x(t)) = \nabla f(x(t))^\top \frac{dx(t)}{dt} = -\|\nabla f(x(t))\|_2^2 \leqslant 0.$$

   If f is bounded from below, then $f(x(t))$ will always converge as a non-increasing function which is bounded from below. It is straightforward, that GF converges to the stationary point, where $\nabla f = 0$ (potentailly including minima, maxima and saddle points).

2. If we additionaly have convexity:

$$f(x) \geqslant f(y) + \nabla f(y)^\top (x - y) \qquad \Rightarrow \qquad \nabla f(y)^\top (x - y) \leqslant f(x) - f(y)$$

Convergence analysis. Convex case.

1. Simplest proof of monotonic decrease of GF:

$$
\frac{d}{dt} f(x(t)) = \nabla f(x(t))^\top \frac{dx(t)}{dt} = -\|\nabla f(x(t))\|_2^2 \leqslant 0.
$$

   If f is bounded from below, then $f(x(t))$ will always converge as a non-increasing function which is bounded from below. It is straightforward, that GF converges to the stationary point, where $\nabla f = 0$ (potentailly including minima, maxima and saddle points).

2. If we additionaly have convexity:

$$
f(x) \geqslant f(y) + \nabla f(y)^\top (x - y) \qquad \Rightarrow \qquad \nabla f(y)^\top (x - y) \leqslant f(x) - f(y)
$$

3. Finally, using convexity:

$$
\frac{d}{dt}\left[\|x(t) - x^*\|^2\right] = -2(x(t) - x^*)^\top \nabla f(x(t)) \leqslant -2\left[f(x(t)) - f^*\right]
$$

Convergence analysis. Convex case.

1. Simplest proof of monotonic decrease of GF:

$$\frac{d}{dt}f(x(t)) = \nabla f(x(t))^\top \frac{dx(t)}{dt} = -\|\nabla f(x(t))\|_2^2 \leqslant 0.$$

   If f is bounded from below, then $f(x(t))$ will always converge as a non-increasing function which is bounded from below. It is straightforward, that GF converges to the stationary point, where $\nabla f = 0$ (potentailly including minima, maxima and saddle points).

2. If we additionaly have convexity:

$$f(x) \geqslant f(y) + \nabla f(y)^\top (x-y) \qquad \Rightarrow \qquad \nabla f(y)^\top (x-y) \leqslant f(x) - f(y)$$

3. Finally, using convexity:

$$\frac{d}{dt}\left[\|x(t) - x^*\|^2\right] = -2(x(t) - x^*)^\top \nabla f(x(t)) \leqslant -2\left[f(x(t)) - f^*\right]$$

4. Leading to, by integrating from 0 to t, and using the monotonicity of $f(x(t))$:

$$f(x(t)) - f^* \leqslant \frac{1}{t}\int_0^t [f(x(u)) - f^*]du \leqslant \frac{1}{2t}\|x(0) - x^*\|^2 - \frac{1}{2t}\|x(t) - x^*\|^2 \leqslant \frac{1}{2t}\|x(0) - x^*\|^2.$$

Convergence analysis. Convex case.

1. Simplest proof of monotonic decrease of GF:

$$\frac{d}{dt} f(x(t)) = \nabla f(x(t))^\top \frac{dx(t)}{dt} = -\|\nabla f(x(t))\|_2^2 \leqslant 0.$$

   If f is bounded from below, then $f(x(t))$ will always converge as a non-increasing function which is bounded from below. It is straightforward, that GF converges to the stationary point, where $\nabla f = 0$ (potentailly including minima, maxima and saddle points).

2. If we additionaly have convexity:

$$f(x) \geqslant f(y) + \nabla f(y)^\top (x - y) \qquad \Rightarrow \qquad \nabla f(y)^\top (x - y) \leqslant f(x) - f(y)$$

3. Finally, using convexity:

$$\frac{d}{dt} [\|x(t) - x^*\|^2] = -2(x(t) - x^*)^\top \nabla f(x(t)) \leqslant -2[f(x(t)) - f^*]$$

4. Leading to, by integrating from 0 to t, and using the monotonicity of $f(x(t))$:

$$f(x(t)) - f^* \leqslant \frac{1}{t} \int_0^t [f(x(u)) - f^*] du \leqslant \frac{1}{2t} \|x(0) - x^*\|^2 - \frac{1}{2t} \|x(t) - x^*\|^2 \leqslant \frac{1}{2t} \|x(0) - x^*\|^2.$$

Convergence analysis. Convex case.

1. Simplest proof of monotonic decrease of GF:

$$\frac{d}{dt}f(x(t)) = \nabla f(x(t))^\top \frac{dx(t)}{dt} = -\|\nabla f(x(t))\|_2^2 \leqslant 0.$$

   If f is bounded from below, then $f(x(t))$ will always converge as a non-increasing function which is bounded from below. It is straightforward, that GF converges to the stationary point, where $\nabla f = 0$ (potentailly including minima, maxima and saddle points).

2. If we additionaly have convexity:

$$f(x) \geqslant f(y) + \nabla f(y)^\top (x - y) \qquad \Rightarrow \qquad \nabla f(y)^\top (x - y) \leqslant f(x) - f(y)$$

3. Finally, using convexity:

$$\frac{d}{dt}\left[\|x(t) - x^*\|^2\right] = -2(x(t) - x^*)^\top \nabla f(x(t)) \leqslant -2\left[f(x(t)) - f^*\right]$$

4. Leading to, by integrating from 0 to t, and using the monotonicity of $f(x(t))$:

$$f(x(t)) - f^* \leqslant \frac{1}{t}\int_0^t [f(x(u)) - f^*]du \leqslant \frac{1}{2t}\|x(0) - x^*\|^2 - \frac{1}{2t}\|x(t) - x^*\|^2 \leqslant \frac{1}{2t}\|x(0) - x^*\|^2.$$

   We recover the usual rates in $\mathcal{O}\left(\frac{1}{k}\right)$, with $t = \alpha k$.

Convergence analysis. PL case.

1. The analsysis is straightforward. Suppose, the function satisfies PL-condition:

$$\|\nabla f(x)\|^2 \geq 2\mu(f(x) - f^*) \quad \forall x$$

Convergence analysis. PL case.

1. The analsysis is straightforward. Suppose, the function satisfies PL-condition:

$$\|\nabla f(x)\|^2 \geq 2\mu(f(x) - f^*) \quad \forall x$$

2. Then

$$\frac{d}{dt}[f(x(t)) - f(x^*)] = \nabla f(x(t))^\top \dot{x}(t) = -\|\nabla f(x(t))\|_2^2 \leqslant -2\mu[f(x(t)) - f^*]$$

Convergence analysis. PL case.

1. The analsysis is straightforward. Suppose, the function satisfies PL-condition:

$$
\|\nabla f(x)\|^2 \geq 2\mu(f(x) - f^*) \quad \forall x
$$

2. Then

$$
\frac{d}{dt}\left[f(x(t)) - f(x^*)\right] = \nabla f(x(t))^\top \dot{x}(t) = \; -\|\nabla f(x(t))\|_2^2 \leqslant \; -2\mu\left[f(x(t)) \; - f^*\right]
$$

3. Finally,

$$
f(x(t)) - f^* \leqslant \exp(-2\mu t)\left[f(x(0)) - f^*\right],
$$

Accelerated Gradient Flow

Accelerated Gradient Flow

Remember one of the forms of Nesterov Accelerated Gradient

$$
\begin{aligned}
x_{k+1} &= y_k - \alpha \nabla f(y_k) \\
y_k &= x_k + \frac{k-1}{k+2}(x_k - x_{k-1})
\end{aligned}
$$

The corresponding [1] ODE is:

$$
\ddot{X}_t + \frac{3}{t}\dot{X}_t + \nabla f(X_t) = 0
$$

[1] A Differential Equation for Modeling Nesterov's Accelerated Gradient Method: Theory and Insights, Weijie Su, Stephen Boyd, Emmanuel J. Candes

Accelerated Gradient Flow

Define the *energy*

$$E(t) = t^2\big(f(X(t)) - f^*\big) + 2\Big\|X(t) - x^* + \tfrac{t}{2}\dot{X}(t)\Big\|^2.$$

A direct differentiation using the ODE yields $\dot{E}(t) \le 0$ for all $t > 0$; hence $E(t)$ is non-increasing. Because the second term is non-negative we obtain the *convergence theorem*

$$\boxed{f(X(t)) - f^* \;\le\; \frac{2\,\|x_0 - x^*\|^2}{t^2}}. \tag{AGF-rate}$$

Thus AGF enjoys the same $\mathcal{O}(1/t^2)$ rate that discrete NAG achieves in $\mathcal{O}(1/k^2)$ iterations. A similar argument with a *restarted* ODE gives an exponential rate for μ-strongly convex f.

Stochastic Gradient Flow

Stochastic Gradient Flow

How to model stochasticity in the continuous process? A simple idea would be: $\frac{dx}{dt} = -\nabla f(x) + \xi$ with variety of options for ξ, for example $\xi \sim \mathcal{N}(0, \sigma^2) \sim \sigma^2 \mathcal{N}(0, 1)$.

Therefore, one can write down Stochastic Differential Equation (SDE) for analysis:

$$dx(t) = -\nabla f\left(x(t)\right) dt + \sigma dW(t)$$

Here $W(t)$ is called Wiener process. It is interesting, that one could analyze the convergence of the stochastic process above in two possible ways:

- Watching the trajectories of $x(t)$

Stochastic Gradient Flow

How to model stochasticity in the continuous process? A simple idea would be: $\frac{dx}{dt} = -\nabla f(x) + \xi$ with variety of options for ξ, for example $\xi \sim \mathcal{N}(0, \sigma^2) \sim \sigma^2 \mathcal{N}(0, 1)$.

Therefore, one can write down Stochastic Differential Equation (SDE) for analysis:

$$dx(t) = -\nabla f\left(x(t)\right) dt + \sigma dW(t)$$

Here $W(t)$ is called Wiener process. It is interesting, that one could analyze the convergence of the stochastic process above in two possible ways:

- Watching the trajectories of $x(t)$
- Watching the evolution of distribution density function of $\rho(t)$

Stochastic Gradient Flow

How to model stochasticity in the continuous process? A simple idea would be: $\frac{dx}{dt} = -\nabla f(x) + \xi$ with variety of options for ξ, for example $\xi \sim \mathcal{N}(0, \sigma^2) \sim \sigma^2 \mathcal{N}(0, 1)$.

Therefore, one can write down Stochastic Differential Equation (SDE) for analysis:

$$dx(t) = -\nabla f\left(x(t)\right) dt + \sigma dW(t)$$

Here $W(t)$ is called Wiener process. It is interesting, that one could analyze the convergence of the stochastic process above in two possible ways:

- Watching the trajectories of $x(t)$
- Watching the evolution of distribution density function of $\rho(t)$

Stochastic Gradient Flow

How to model stochasticity in the continuous process? A simple idea would be: $\frac{dx}{dt} = -\nabla f(x) + \xi$ with variety of options for ξ, for example $\xi \sim \mathcal{N}(0, \sigma^2) \sim \sigma^2 \mathcal{N}(0,1)$.

Therefore, one can write down Stochastic Differential Equation (SDE) for analysis:

$$dx(t) = -\nabla f\left(x(t)\right) dt + \sigma dW(t)$$

Here $W(t)$ is called Wiener process. It is interesting, that one could analyze the convergence of the stochastic process above in two possible ways:

- Watching the trajectories of $x(t)$
- Watching the evolution of distribution density function of $\rho(t)$

! Fokker-Planck equation

$$\frac{\partial \rho}{\partial t} = \nabla\left(\rho(t)\nabla f\right) + \frac{\sigma^2}{2}\Delta \rho(t)$$

Sources

- Francis Bach blog

Sources

- Francis Bach blog
- Off convex Path blog

Sources

- Francis Bach blog
- Off convex Path blog
- Stochastic gradient algorithms from ODE splitting perspective

Sources

- Francis Bach blog
- Off convex Path blog
- Stochastic gradient algorithms from ODE splitting perspective
- NAG-GS: Semi-Implicit, Accelerated and Robust Stochastic Optimizer

Sources

- Francis Bach blog
- Off convex Path blog
- Stochastic gradient algorithms from ODE splitting perspective
- NAG-GS: Semi-Implicit, Accelerated and Robust Stochastic Optimizer
- Introduction to Gradient Flows in the 2-Wasserstein Space

Sources

- Francis Bach blog
- Off convex Path blog
- Stochastic gradient algorithms from ODE splitting perspective
- NAG-GS: Semi-Implicit, Accelerated and Robust Stochastic Optimizer
- Introduction to Gradient Flows in the 2-Wasserstein Space
- Stochastic Modified Equations and Dynamics of Stochastic Gradient Algorithms I: Mathematical Foundations

Sources

- Francis Bach blog
- Off convex Path blog
- Stochastic gradient algorithms from ODE splitting perspective
- NAG-GS: Semi-Implicit, Accelerated and Robust Stochastic Optimizer
- Introduction to Gradient Flows in the 2-Wasserstein Space
- Stochastic Modified Equations and Dynamics of Stochastic Gradient Algorithms I: Mathematical Foundations
- Understanding Optimization in Deep Learning with Central Flows