Stochastic gradient methods

Pierre Ablin
CNRS – Université Paris-Dauphine

Solving the Finite Sum Training Problem

General machine learning framework

Dataset: $x_1, \ldots, x_n \in \mathbb{R}^d$

Parameters: $w \in \mathbb{R}^p$

Risk functions: $\ell_i(w, x_i) = f_i(w) \in \mathbb{R}$

General machine learning framework

Dataset: $x_1, \ldots, x_n \in \mathbb{R}^d$

Parameters: $w \in \mathbb{R}^p$

Risk functions: $\ell_i(w, x_i) = f_i(w) \in \mathbb{R}$

Example:

- regression $\ell_i(w, x_i) = \frac{1}{2}(\langle w, x_i \rangle - y_i)^2$

- binary regression $\ell_i(w, x_i) = \log(1 + \exp(-y_i \langle w, x_i \rangle))$

- multinomial regression $\ell_i(w, x_i) = \mathrm{CrossEntropy}(wx_i, y_i)$

General machine learning framework

Dataset: $x_1, \ldots, x_n \in \mathbb{R}^d$

Parameters: $w \in \mathbb{R}^p$

Risk functions: $\ell_i(w, x_i) = f_i(w) \in \mathbb{R}$

Empirical risk minimization (ERM):

$$\text{Find } w \text{ by minimizing } F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

General machine learning framework

Dataset: $x_1, \ldots, x_n \in \mathbb{R}^d$

Parameters: $w \in \mathbb{R}^p$

Risk functions: $\ell_i(w, x_i) = f_i(w) \in \mathbb{R}$

Empirical risk minimization (ERM):

$$\text{Find } w \text{ by minimizing } F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

99 % of machine learning optimization problems are of this form

Optimization of a sum of terms

Empirical risk minimization (ERM):

$$\text{Find } w \text{ by minimizing } F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

Can we use this sum structure?

The Training Problem

Solving the *training problem*: $\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(w)$

Reference method: Gradient descent

$$\nabla \left(\frac{1}{n} \sum_{i=1}^{n} f_i(w) \right) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(w)$$

Gradient Descent Algorithm

Set $w^0 = 0$, choose $\alpha > 0$.
for $t = 0, 1, 2, \ldots, T-1$
$$w^{t+1} = w^t - \frac{\alpha}{n} \sum_{i=1}^{n} \nabla f_i(w^t)$$
Output w^T

The Training Problem

Solving the *training problem*:

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

Problem with Gradient Descent:
Each iteration requires computing a gradient $\nabla f_i(w)$ for each data point. One gradient for each cat on the internet!

Gradient Descent Algorithm

Set $w^0 = 0$, choose $\alpha > 0$.
for $t = 0, 1, 2, \ldots, T$
$$w^{t+1} = w^t - \frac{\alpha}{n} \sum_{i=1}^{n} \nabla f_i(w^t)$$
Output w^T

Gradient descent

Problem with Gradient Descent:
Each iteration requires computing a gradient $\nabla f_i(w)$ for each data point.

One iteration costs $O(n)$:
cannot scale to a large scale setting.

Gradient descent

Problem with Gradient Descent:
Each iteration requires computing a gradient $\nabla f_i(w)$ for each data point.

One iteration costs $O(n)$:
cannot scale to a large scale setting.

We need a method with better scaling !
Can we progress on the training problem by looking at just a few samples at a time?

Stochastic Gradient Descent

Is it possible to design a method that uses only the gradient of a **single** data function $f_i(w)$ at each iteration?

Stochastic Gradient Descent

Is it possible to design a method that uses only the gradient of a **single** data function $f_i(w)$ at each iteration?

Unbiased Estimate

Let j be a random index sampled from $\{1, \ldots, n\}$ selected uniformly at random. Then

$$\mathbb{E}_j[\nabla f_j(w)] = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w) = \nabla f(w)$$

Stochastic Gradient Descent

Is it possible to design a method that uses only the gradient of a **single** data function $f_i(w)$ at each iteration?

Unbiased Estimate

Let j be a random index sampled from $\{1, ..., n\}$ selected uniformly at random. Then

$$\mathbb{E}_j[\nabla f_j(w)] = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w) = \nabla f(w)$$

Use $\nabla f_j(w) \approx \nabla f(w)$

Stochastic Gradient Descent

SGD, Constant stepsize

Set $w^0 = 0$, choose $\alpha > 0$

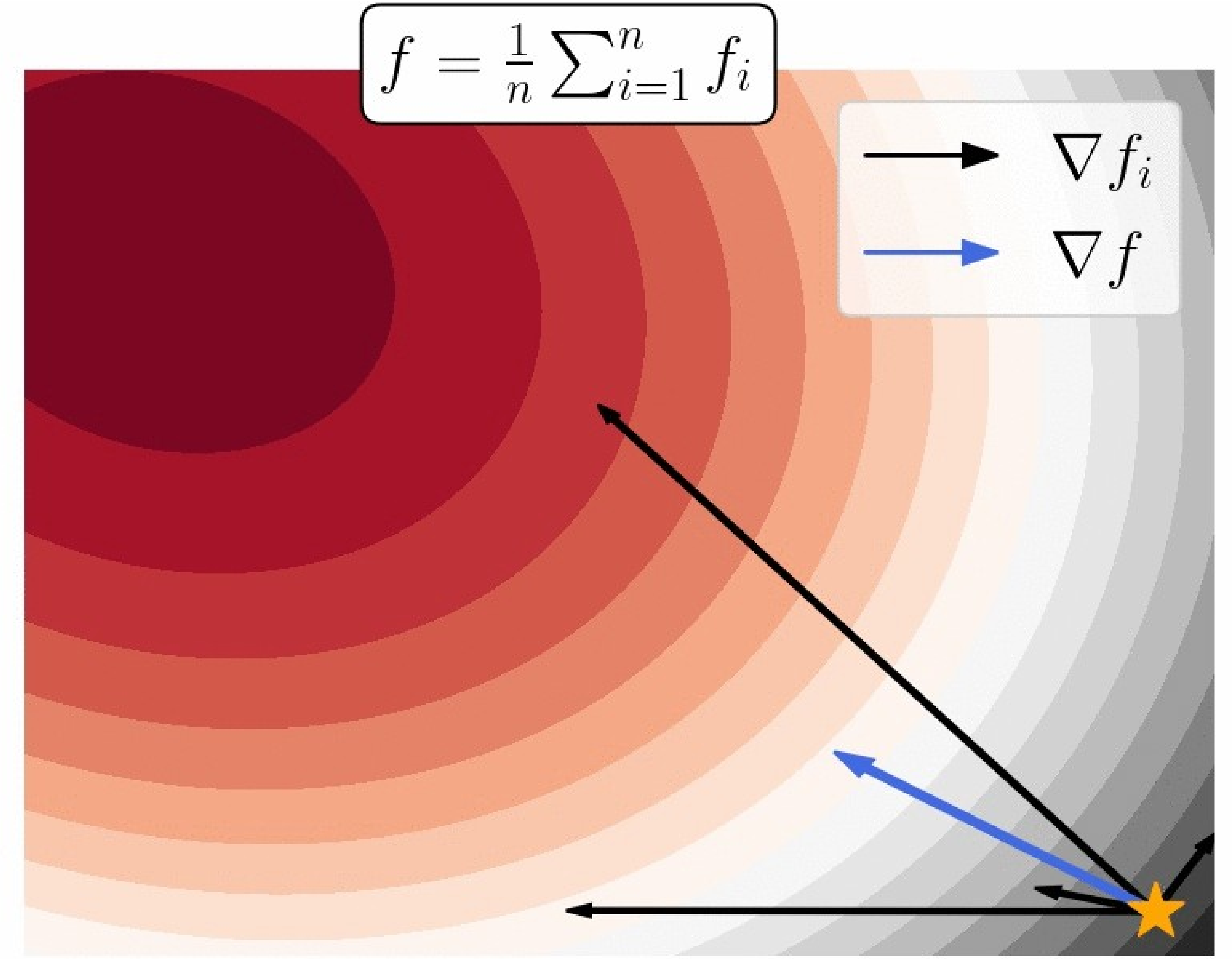
for $t = 0, 1, 2, \ldots, T-1$

  sample $j \in \{1, \ldots, n\}$

  $w^{t+1} = w^t - \alpha \nabla f_j(w^t)$

Output w^T

Intuition about SGD



SGD: intuition

$$\nabla f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

- When far from the optimum ($\nabla f(w)$ large), it is likely that $\nabla f_i(w)$ is a descent direction.

SGD: intuition

$$\nabla f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

- When far from the optimum ($\nabla f(w)$ large), it is likely that $\nabla f_i(w)$ is a descent direction.

Question: What is a quantity that measures whether $\nabla f_i(w)$ is a descent direction ? What is its average value ?

SGD: intuition

$$\nabla f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

- When far from the optimum ($\nabla f(w)$ large), it is likely that $\nabla f_i(w)$ is a descent direction.

Question: What is a quantity that measures whether $\nabla f_i(w)$ is a descent direction ? What is its average value ?

Answer

Scalar product: $\langle \nabla f(w), \nabla f_i(w) \rangle$

On average: $\mathbb{E}_i\left[\langle \nabla f(w), \nabla f_i(w) \rangle\right] = \frac{1}{n}\sum_{i=1}^{n} \langle \nabla f(w), \nabla f_i(w) \rangle = \|\nabla f(w)\|^2$

SGD: intuition

$$\nabla f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

- When far from the optimum ($\nabla f(w)$ large), it is likely that $\nabla f_i(w)$ is a descent direction.

- At the optimum, we do **not** have $\nabla f_i(w^*) = 0$, hence it is a bad estimate of the gradient : it is zero on average, but it has some **variance**

SGD: intuition

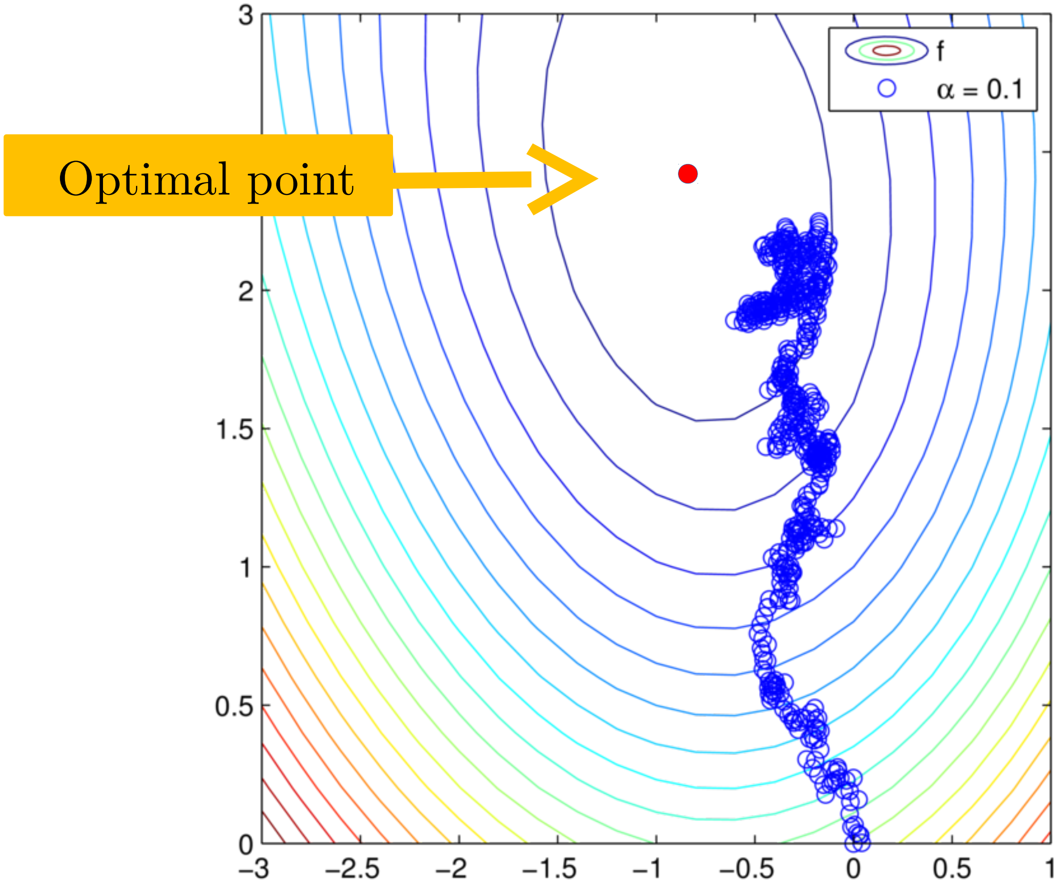
$$\nabla f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

- When far from the optimum ($\nabla f(w)$ large), it is likely that $\nabla f_i(w)$ is a descent direction.

- At the optimum, we do **not** have $\nabla f_i(w^*) = 0$, hence it is a bad estimate of the gradient : it is zero on average, but it has some **variance**

Question: Consider the least squares problem $f_i(w) = \frac{1}{2}(\langle x_i, w\rangle - y_i)^2$ where $\|x_i\|^2 = 1$
Let $r_i = \langle x_i, w\rangle - y_i$ the residuals. What is the variance of $\nabla f_i(w)$ at the optimum? Can it be 0?

Stochastic Gradient Descent



Convergence Strongly Convex and Bounded Gradient

Theorem If f is $\mu-$strongly convex and $\mathbb{E}[\|\nabla f_i(w)\|^2] \leq B^2$

If $0 < \alpha \leq \frac{1}{\mu}$ then the iterates of the SGD method satisfy

$$\mathbb{E}\left[||w^t - w^*||_2^2\right] \leq (1 - \alpha\mu)^t ||w^0 - w^*||_2^2 + \frac{\alpha}{\mu} B^2$$

Shows that $\alpha \approx \frac{1}{\mu}$

Shows that $\alpha \approx 0$

Proof: $$w^{t+1} = w^t - \alpha \nabla f_j(w^t), \quad j \sim [1, \ldots, n]$$

1) Show that

$$||w^{t+1} - w^*||_2^2 \quad = \quad ||w^t - w^*||_2^2 - 2\alpha \langle \nabla f_j(w^t), w^t - w^* \rangle + \alpha^2 ||\nabla f_j(w^t)||_2^2.$$

2) Show that

$$\mathbb{E}_j \left[||w^{t+1} - w^*||_2^2 \right] \leq ||w^t - w^*||_2^2 - 2\alpha \langle \nabla f(w^t), w^t - w^* \rangle + \alpha^2 B^2$$

3) Using strong convexity, demonstrate that

$$\mathbb{E}_j \left[||w^{t+1} - w^*||_2^2 \right] \leq (1 - \alpha\mu) ||w^t - w^*||_2^2 + \alpha^2 B^2$$

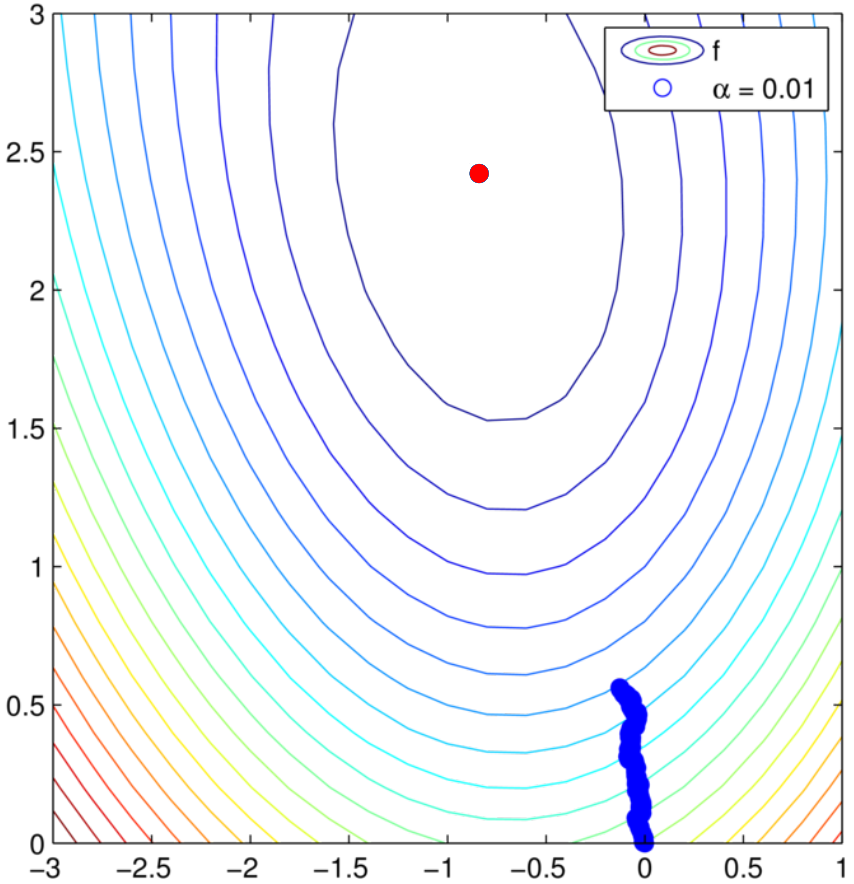
4) Show that

$$\mathbb{E} \left[||w^{t+1} - w^*||_2^2 \right] \leq (1 - \alpha\mu) \mathbb{E} \left[||w^t - w^*||_2^2 \right] + \alpha^2 B^2$$

Where the expectation is taken w.r.t. the whole past. Conclude.

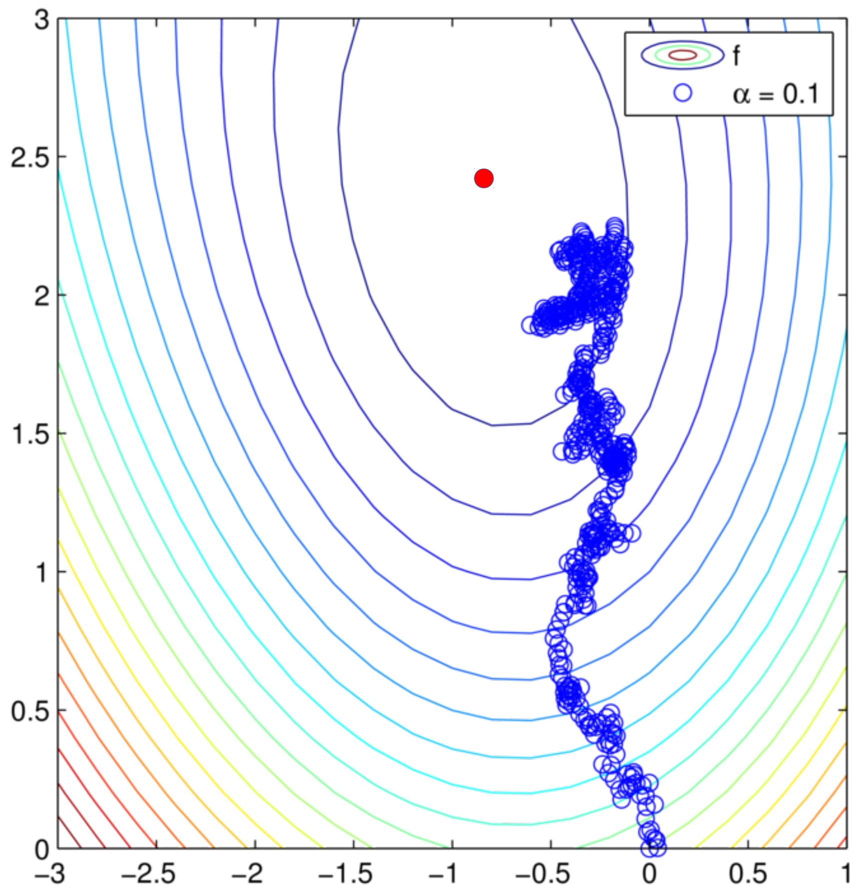
Stochastic Gradient Descent

α =0.01



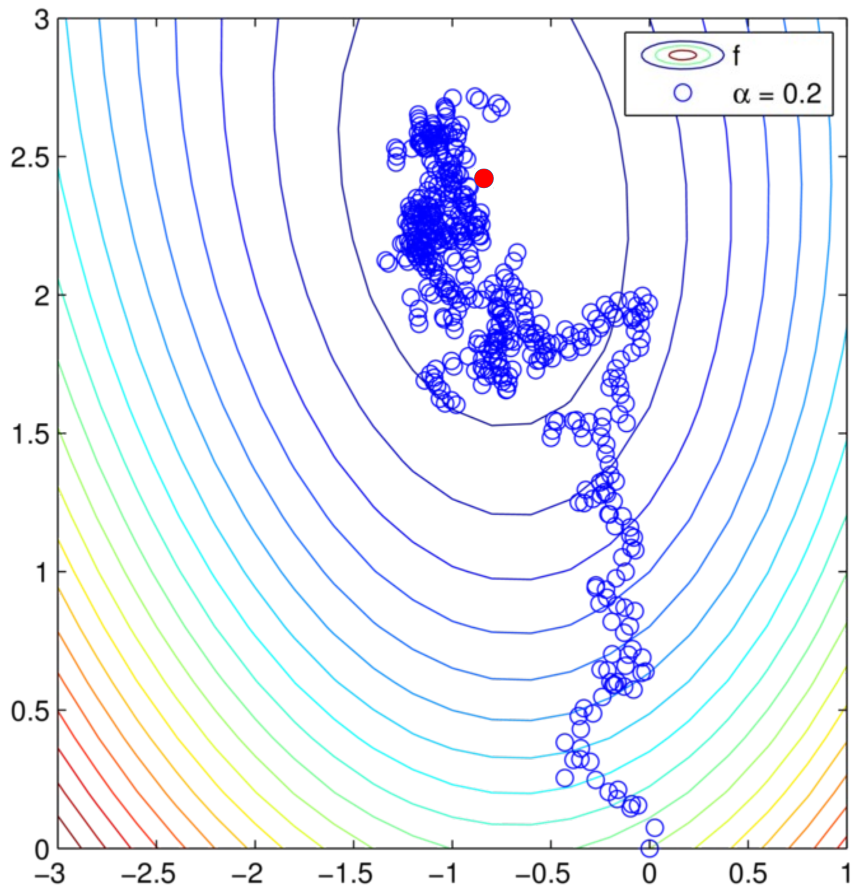
Stochastic Gradient Descent

α =0.1



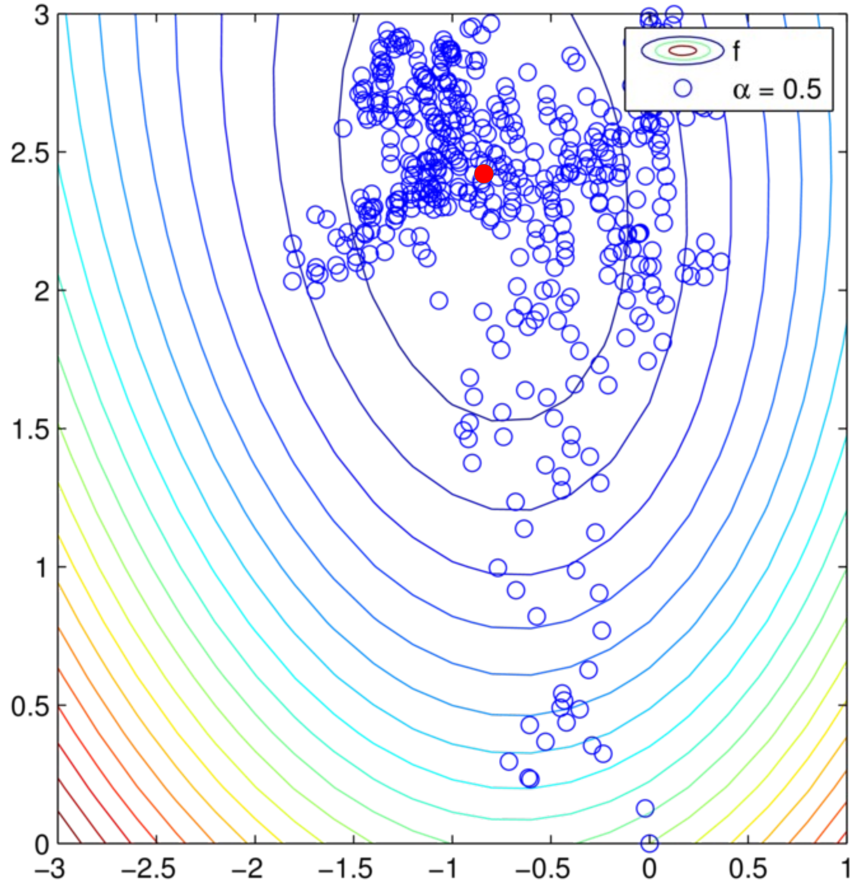
Stochastic Gradient Descent

α =0.2



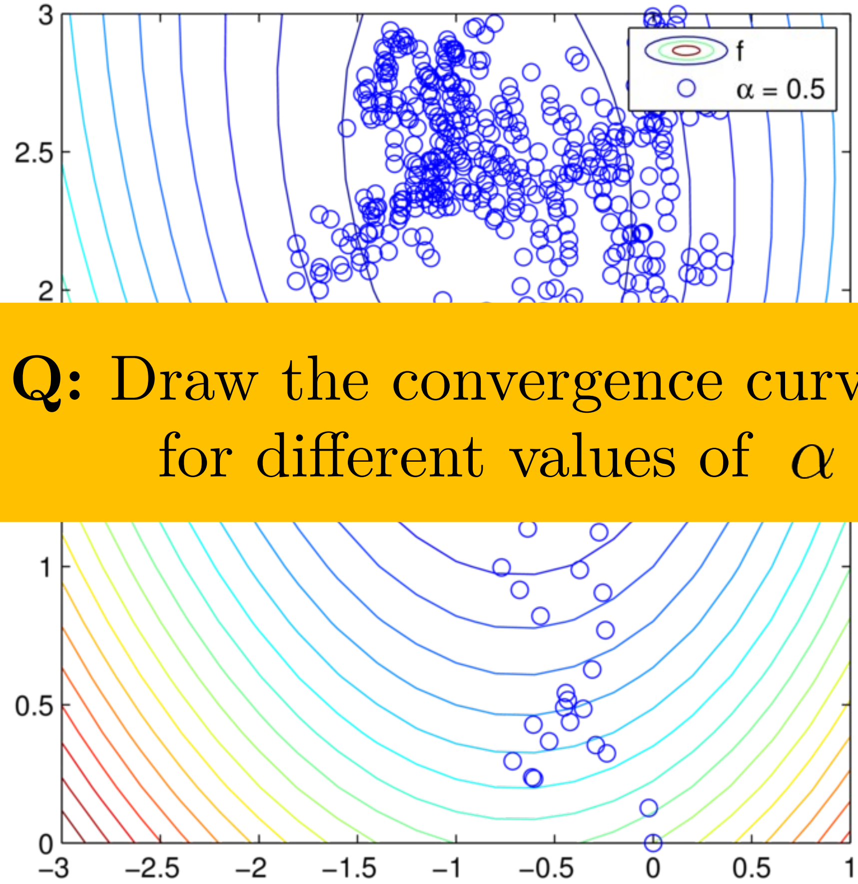
Stochastic Gradient Descent

α =0.5



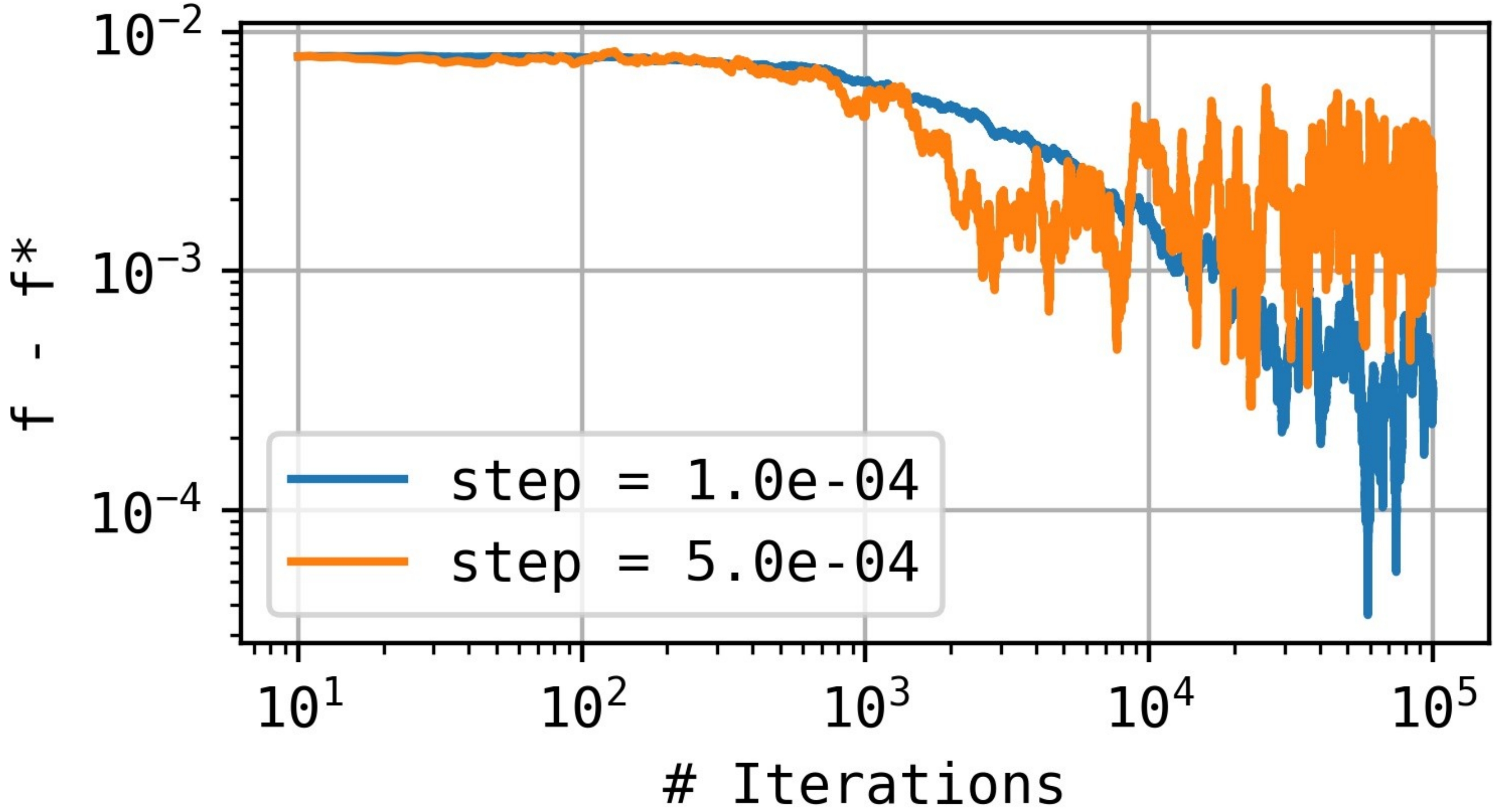
Stochastic Gradient Descent

α =0.5

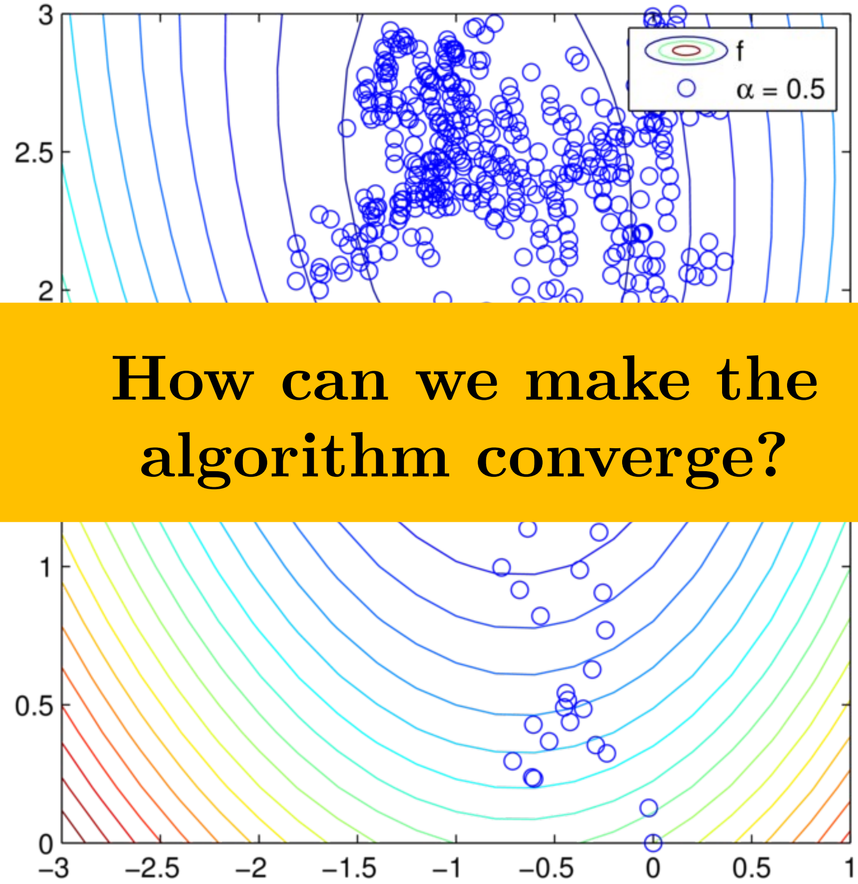


Q: Draw the convergence curves for different values of α

Stochastic Gradient Descent



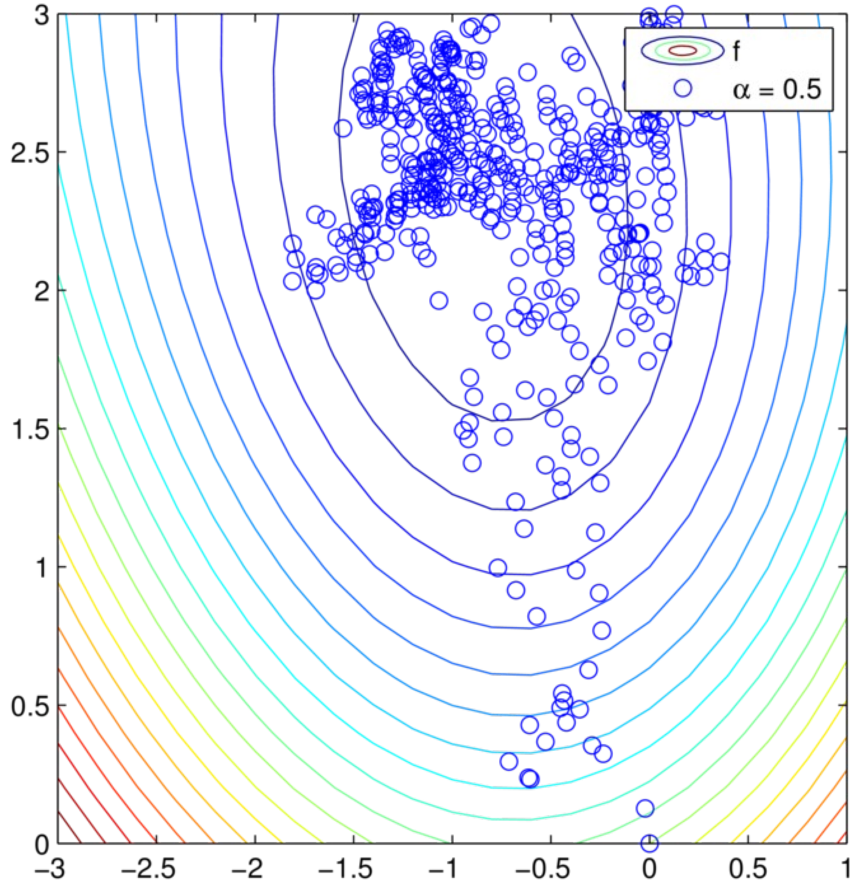
Stochastic Gradient Descent
α =0.5



How can we make the algorithm converge?

Stochastic Gradient Descent

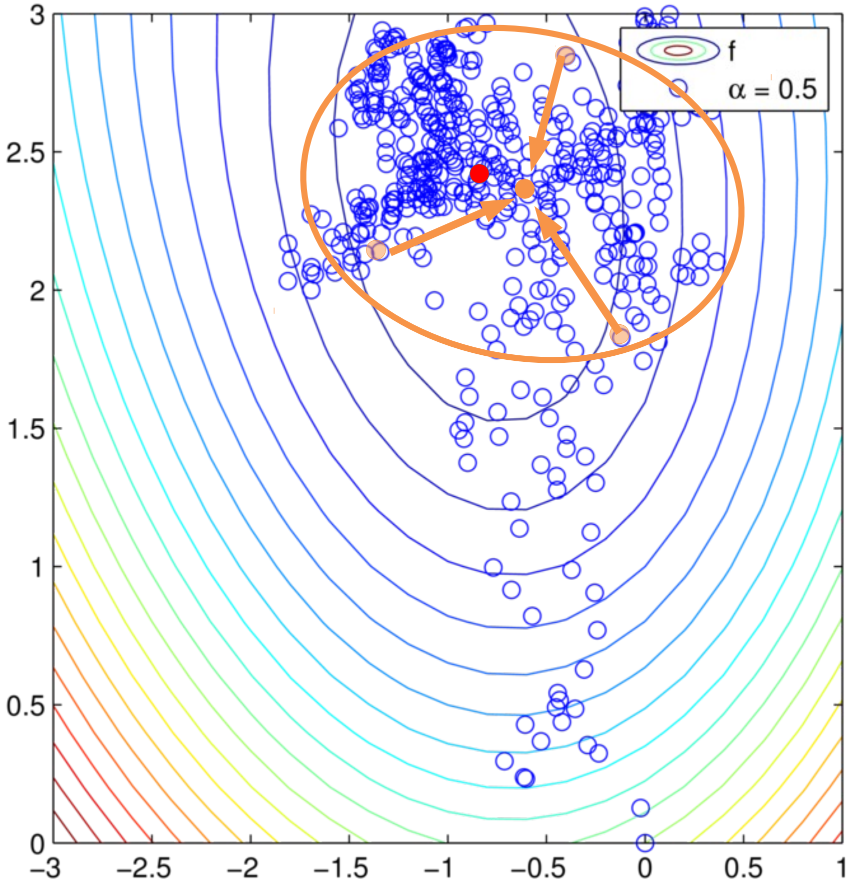
α =0.5



1) Start with big steps and end with smaller steps

Stochastic Gradient Descent

α =0.5



1) Start with big steps and end with smaller steps

2) Try averaging the points

SGD with decreasing stepsize

SGD with decreasing stepsize

Set $w^0 = 0$

Choose $\alpha_t > 0$,

for $t = 0, 1, 2, \ldots, T-1$

  sample $j \in \{1, \ldots, n\}$

  $w^{t+1} = w^t - \alpha_t \nabla f_j(w^t)$

Output w^T

Shrinking Stepsize

SGD with decreasing stepsize

SGD with decreasing stepsize

Set $w^0 = 0$

Choose $\alpha_t > 0$,

for $t = 0, 1, 2, \ldots, T-1$

  sample $j \in \{1, \ldots, n\}$

  $w^{t+1} = w^t - \alpha_t \nabla f_j(w^t)$

Output w^T

Shrinking Stepsize

What should the step-size be?

Step sizes should be small enough

Theorem If f is $\mu-$strongly convex and $\mathbb{E}[\|\nabla f_i(w)\|^2] \leq B^2$

If $0 < \alpha \leq \frac{1}{\mu}$ then the iterates of the SGD method satisfy

$$\mathbb{E}\left[||w^t - w^*||_2^2\right] \leq (1-\alpha\mu)^t ||w^0 - w^*||_2^2 + \frac{\alpha}{\mu} B^2$$

Step sizes should be large enough

Intuition: If step sizes are too small, the algorithm will stop moving before convergence.

Question:
Consider gradient descent on $w \to \frac{1}{2}||w||^2$ with step sizes α_t .

What is a condition for convergence to the correct limit?

Step sizes should be large enough

Intuition: If step sizes are too small, the algorithm will stop moving before convergence.

Question:
Consider gradient descent on $w \to \frac{1}{2}\|w\|^2$ with step sizes α_t .

What is a condition for convergence to the correct limit?

Answer: $w_{t+1} = w_t - \alpha_t w_t$, so $w_t = \prod_{i=1}^{t}(1-\alpha_i)w_0$

Condition : $\lim_{t\to+\infty} \prod_{i=1}^{t}(1-\alpha_i) = 0$, i.e. $\sum_{t=0}^{+\infty} \alpha_t = +\infty$

Decreasing step-sizes

Theorem If f is μ – strongly convex and $\mathbb{E}[\|\nabla f_i(w)\|^2] \leq B^2$

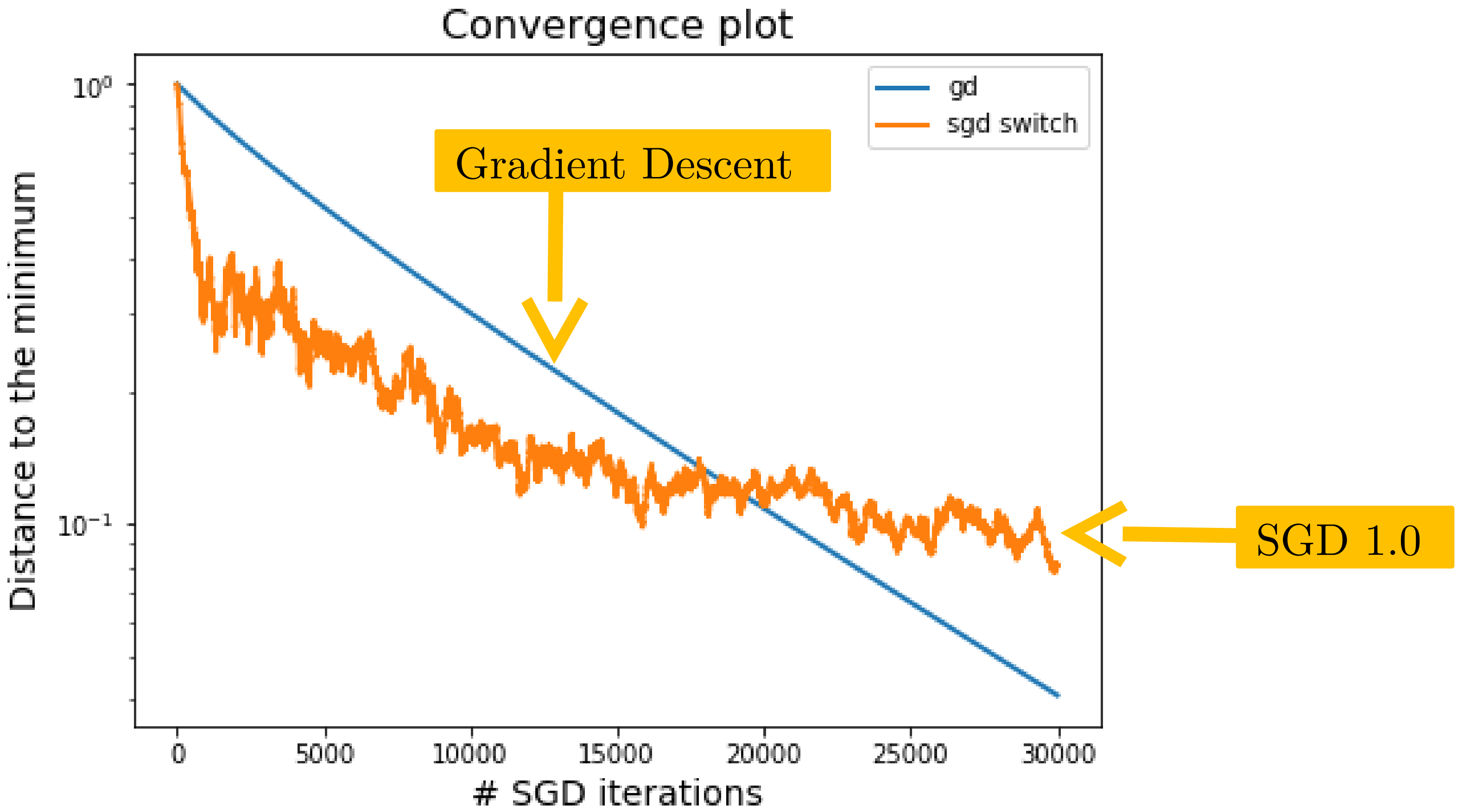
If α_t is such that $\sum_{t=0}^{+\infty} \alpha_t = +\infty, \; \sum_{t=0}^{+\infty} \alpha_t^2 = K \leq +\infty,$ then

$$\inf_{t \leq T} \mathbb{E}\left[\|w^t - w^*\|_2^2\right] \leq \left(\mu \sum_{t=0}^{T} \alpha_t\right)^{-1} \times (\|w^0 - w^*\|^2 + B^2 K)$$

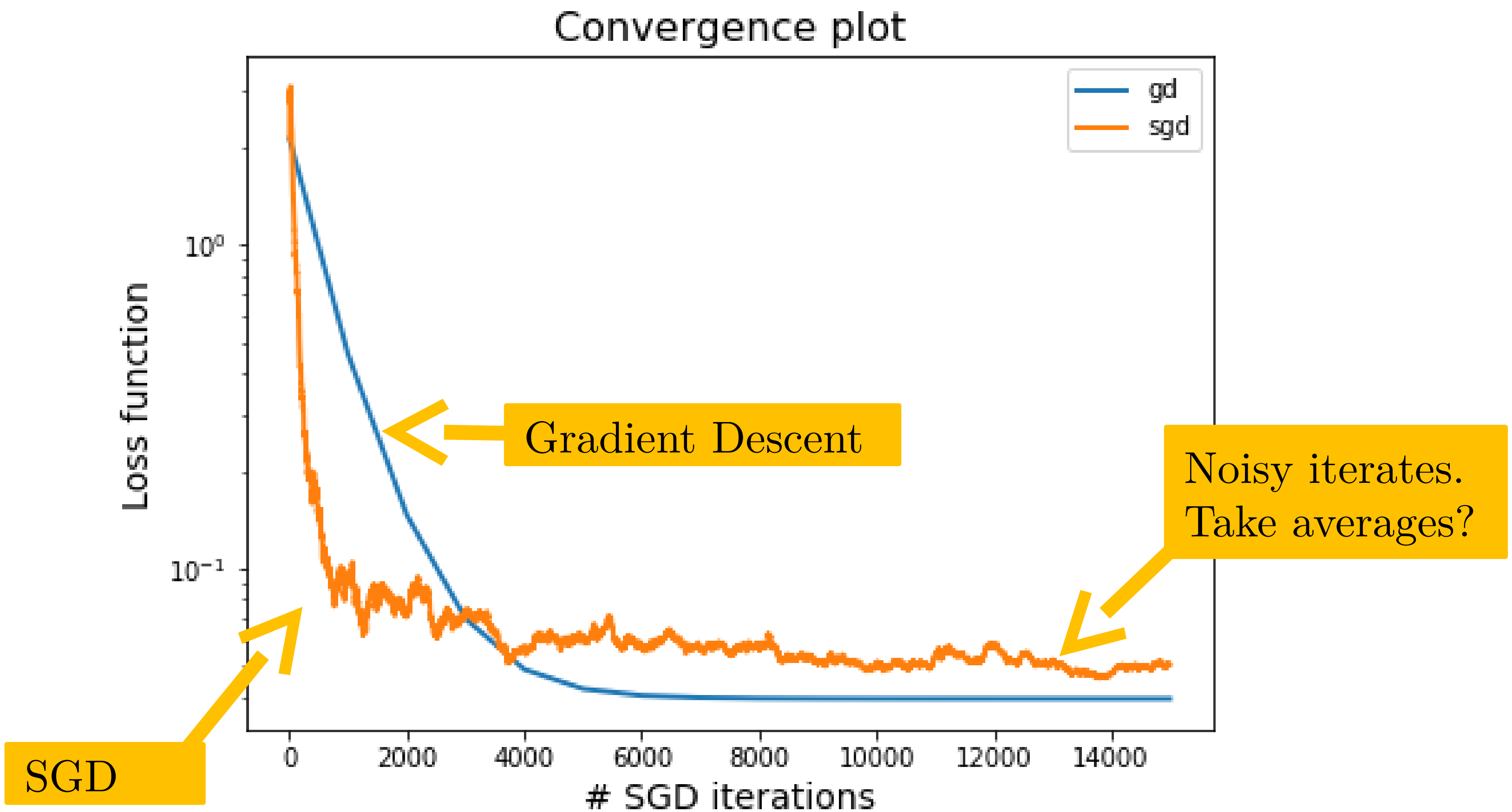
Question: Demonstrate the theorem.

If we take $\alpha_t = \dfrac{1}{(1+t)^{\beta}}$ what is the best value for β ?

SGD with shrinking stepsize



SGD with shrinking stepsize



SGD with (late start) averaging

SGDA 1.1

Set $w^0 = 0$

Choose $\alpha_t > 0,\ \alpha_t \rightarrow 0,\ \sum_{t=0}^{\infty} \alpha_t = \infty$

Choose averaging start $s_0 \in \mathbb{N}$

for $t = 0, 1, 2, \ldots, T-1$

  sample $j \in \{1, \ldots, n\}$

  $w^{t+1} = w^t - \alpha_t \nabla f_j(w^t)$

  if $t > s_0$

    $\overline{w} = \frac{1}{t - s_0} \sum_{i=s_0}^{t} w^t$

  else: $\overline{w} = w$

Output $\overline{w}$

B. T. Polyak and A. B. Juditsky, SIAM Journal on Control and Optimization (1992)
Acceleration of stochastic approximation by averaging

SGD with (late start) averaging

SGDA 1.1

Set $w^0 = 0$

Choose $\alpha_t > 0,\ \alpha_t \to 0,\ \sum_{t=0}^{\infty} \alpha_t = \infty$

Choose averaging start $s_0 \in \mathbb{N}$

for $t = 0, 1, 2, \ldots, T-1$

  sample $j \in \{1, \ldots, n\}$

  $w^{t+1} = w^t - \alpha_t \nabla f_j(w^t)$

  if $t > s_0$

    $\overline{w} = \frac{1}{t - s_0} \sum_{i=s_0}^{t} w^t$

  else: $\overline{w} = w$

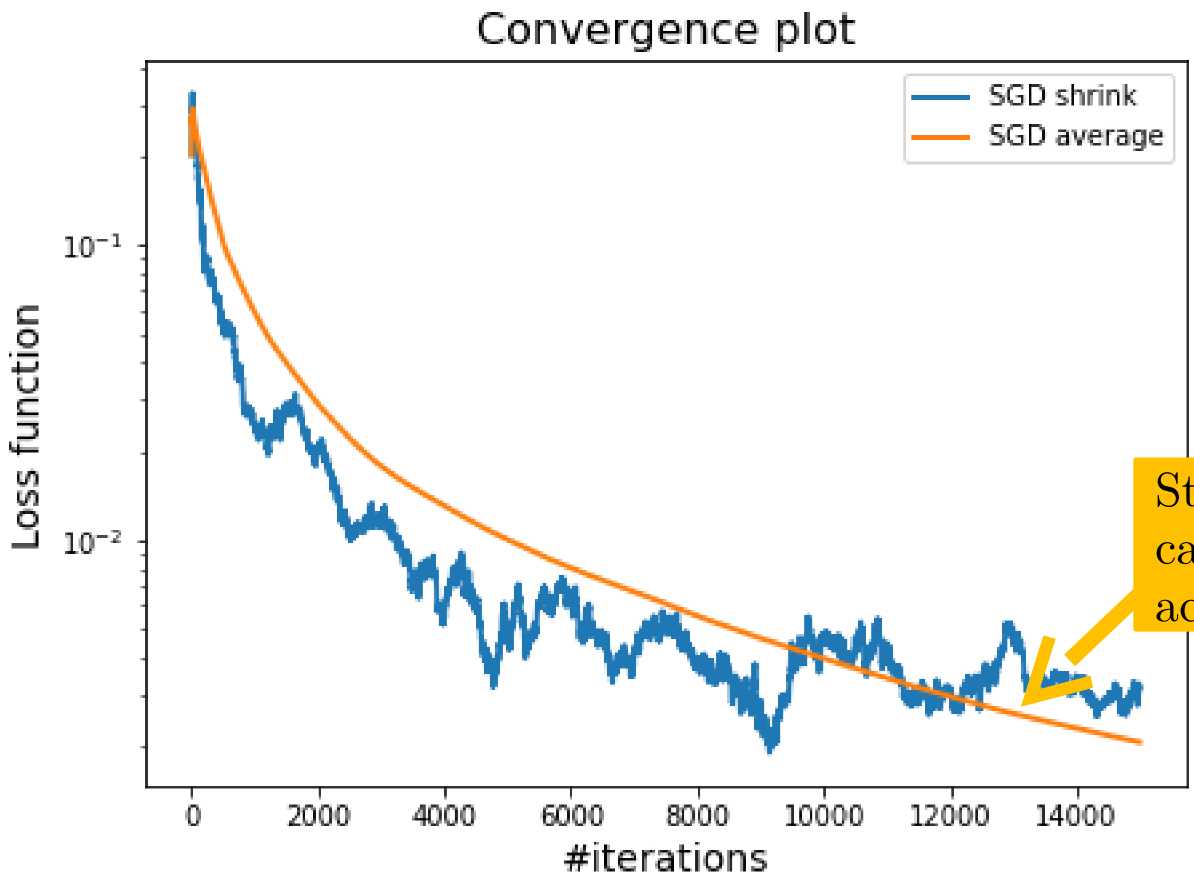
Output $\overline{w}$

This is not efficient. How to make this efficient?

B. T. Polyak and A. B. Juditsky, SIAM Journal on Control and Optimization (1992)
Acceleration of stochastic approximation by averaging

Stochastic Gradient Descent

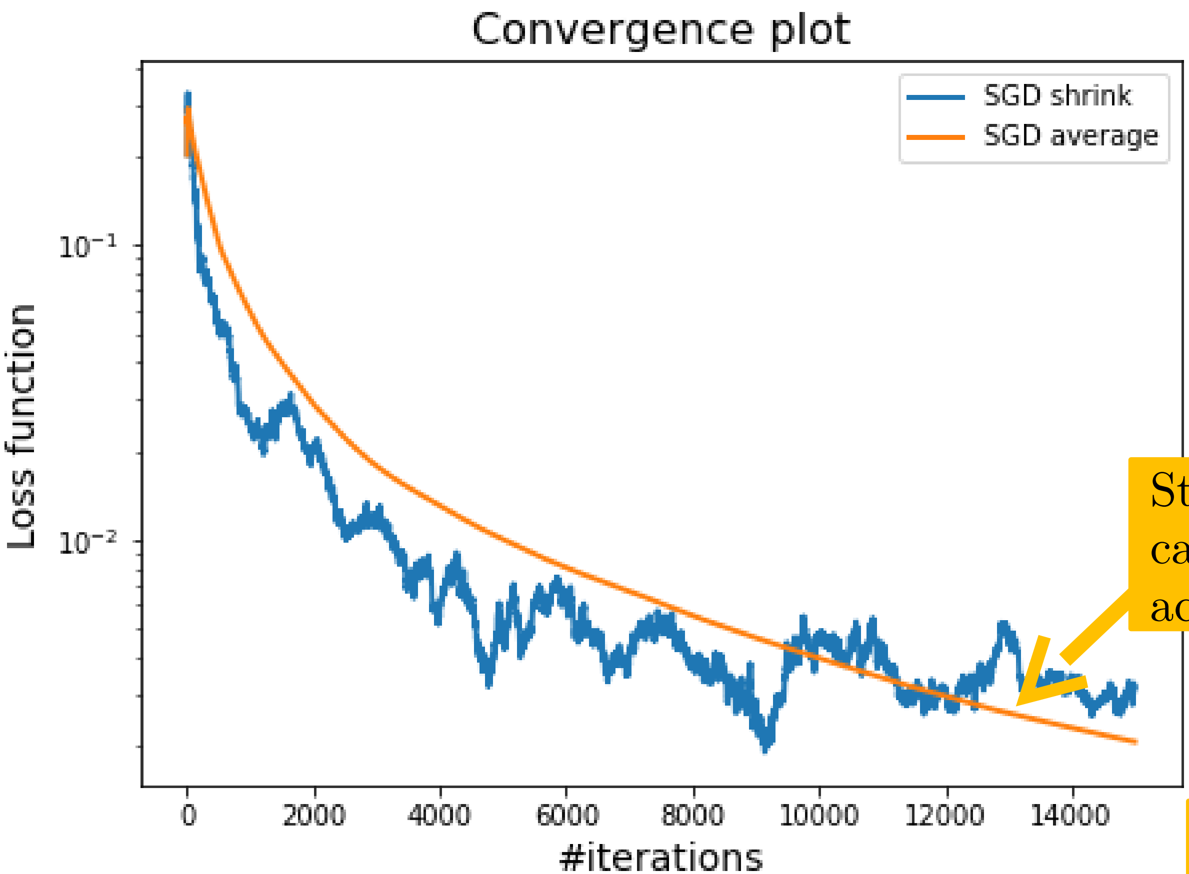
With and without averaging



Starts slow, but can reach higher accuracy

Stochastic Gradient Descent

With and without averaging

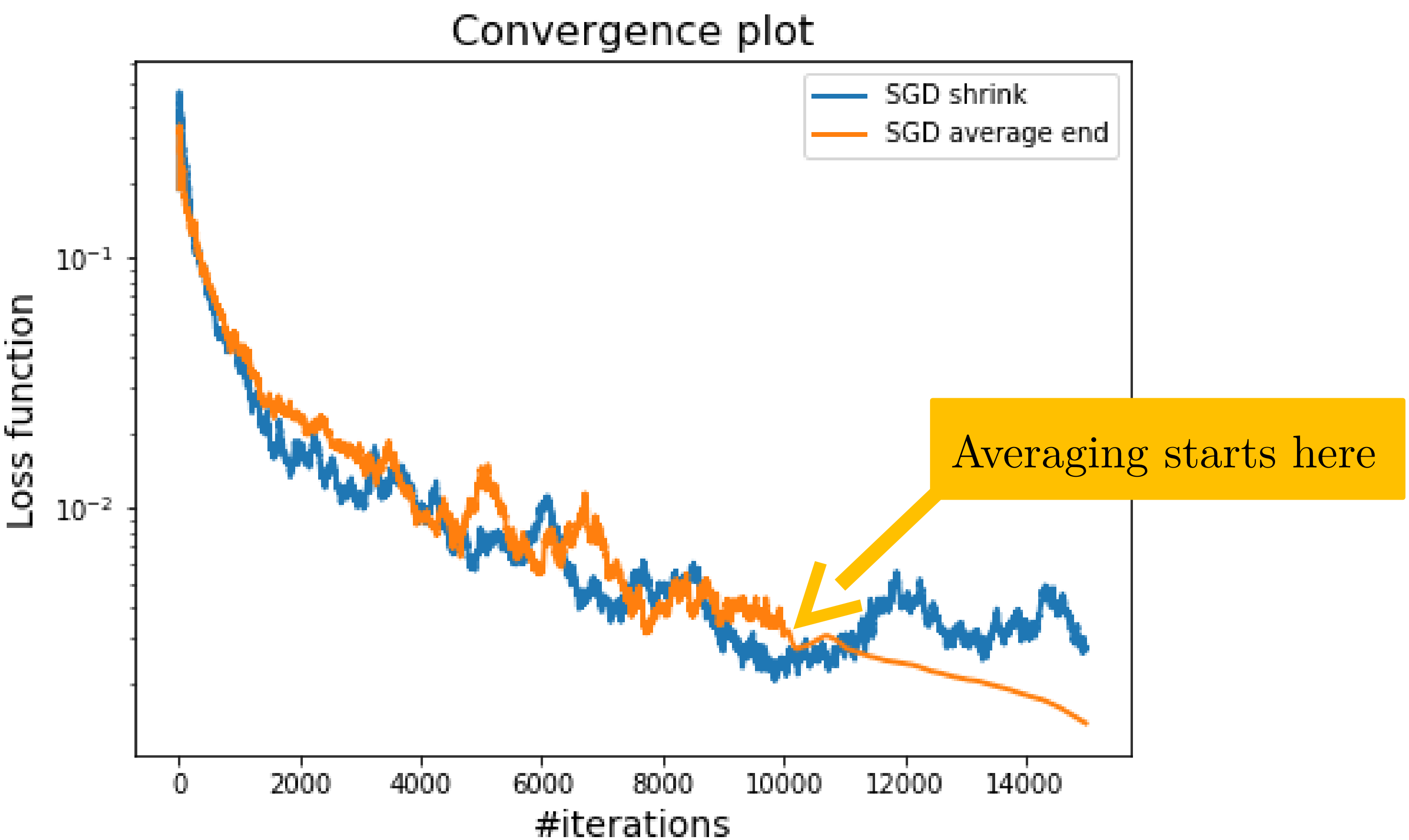


Starts slow, but can reach higher accuracy

Only use averaging towards the end?

Stochastic Gradient Descent

Averaging the last few iterates



Comparison GD and SGD for strongly convex

	SGD	**GD**
Iteration complexity	$O\left(\frac{1}{\epsilon}\right)$	$O\left(\log\left(\frac{1}{\epsilon}\right)\right)$

Comparison GD and SGD for strongly convex

	SGD	**GD**
Iteration complexity	$O\left(\frac{1}{\epsilon}\right)$	$O\left(\log\left(\frac{1}{\epsilon}\right)\right)$
Cost of an iteration	$O\left(1\right)$	$O\left(n\right)$

Comparison GD and SGD for strongly convex

	SGD	GD
Iteration complexity	$O\left(\frac{1}{\epsilon}\right)$	$O\left(\log\left(\frac{1}{\epsilon}\right)\right)$
Cost of an iteration	$O(1)$	$O(n)$
Total complexity*	$O\left(\frac{1}{\epsilon}\right)$	$O\left(n\log\left(\frac{1}{\epsilon}\right)\right)$

Comparison GD and SGD for strongly convex

	SGD	**GD**
Iteration complexity	$O\left(\frac{1}{\epsilon}\right)$	$O\left(\log\left(\frac{1}{\epsilon}\right)\right)$
Cost of an iteration	$O(1)$	$O(n)$
Total complexity*	$O\left(\frac{1}{\epsilon}\right)$	$O\left(n\log\left(\frac{1}{\epsilon}\right)\right)$

*Total complexity = (Iteration complexity) × (Cost of an iteration)

Comparison GD and SGD for strongly convex

	SGD	GD
Iteration complexity	$O\left(\frac{1}{\epsilon}\right)$	$O\left(\log\left(\frac{1}{\epsilon}\right)\right)$
Cost of an iteration	$O(1)$	$O(n)$
Total complexity*	$O\left(\frac{1}{\epsilon}\right)$	$O\left(n\log\left(\frac{1}{\epsilon}\right)\right)$

What happens if ϵ is small?

What happens if n is big?

*Total complexity = (Iteration complexity) $\times$ (Cost of an iteration)

Mini-batching

CPU / GPU parallelization : it is faster to compute ten gradients once than one gradient ten times

Mini-batching

CPU / GPU parallelization : it is faster to compute ten gradients once than one gradient ten times

SGD:

$$w_{t+1} = w_t - \alpha \nabla_i f(w_t)$$

Mini-batching

CPU / GPU parallelization : it is faster to compute ten gradients once than one gradient ten times

SGD:

$$w_{t+1} = w_t - \alpha \nabla_i f(w_t)$$

Can we be more efficient?

Mini-batching

CPU / GPU parallelization : it is faster to compute ten gradients once than one gradient ten times

SGD:

$$w_{t+1} = w_t - \alpha \nabla_i f(w_t)$$

Mini-batch SGD:

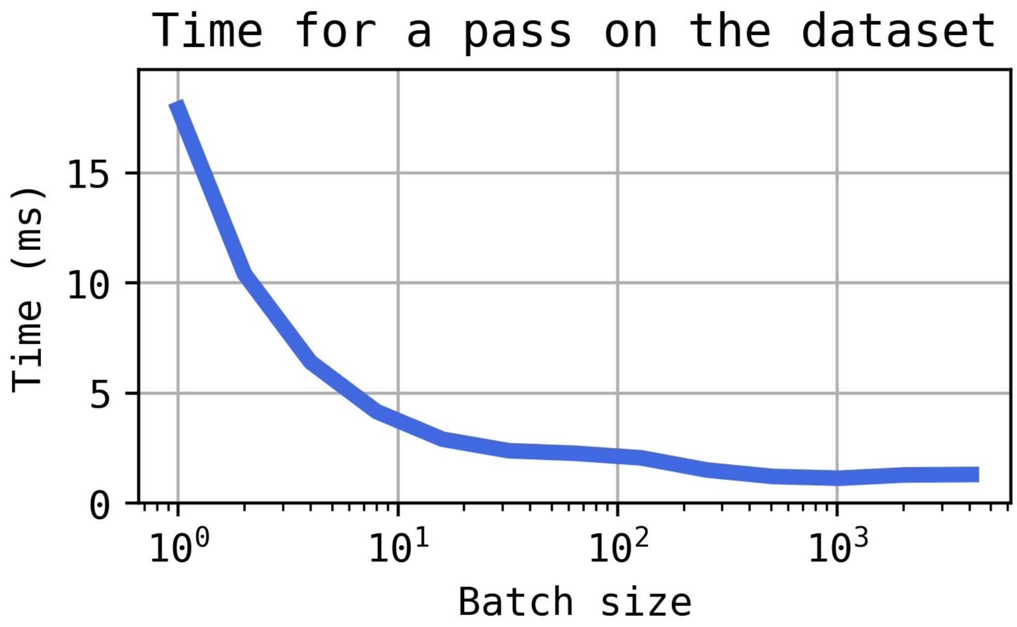
$$w_{t+1} = w_t - \alpha \frac{1}{b} \sum_{j=1}^{b} \nabla_{i_j} f(w_t)$$

Compute gradient over a mini batch

Mini-batching

Avantages:

- Uses parallelization



Mini-batching

Avantages:

- Reduces variance

Why Machine Learners Like SGD

Why Machine Learners like SGD

Though we solve:

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right) + \lambda R(w)$$

We want to solve:

The statistical learning problem:

Minimize the expected loss over an *unknown* expectation

$$\min_{w \in \mathbf{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell\left(h_w(x), y\right)\right]$$

SGD can solve the statistical learning problem!

Why Machine Learners like SGD

The statistical learning problem:

Minimize the expected loss over an *unknown* expectation

$$\min_{w \in \mathbf{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell\left(h_w(x), y\right)\right]$$

SGD ∞.0 for learning

Set $w^0 = 0$, $\alpha > 0$

for $t = 0, 1, 2, \ldots, T-1$

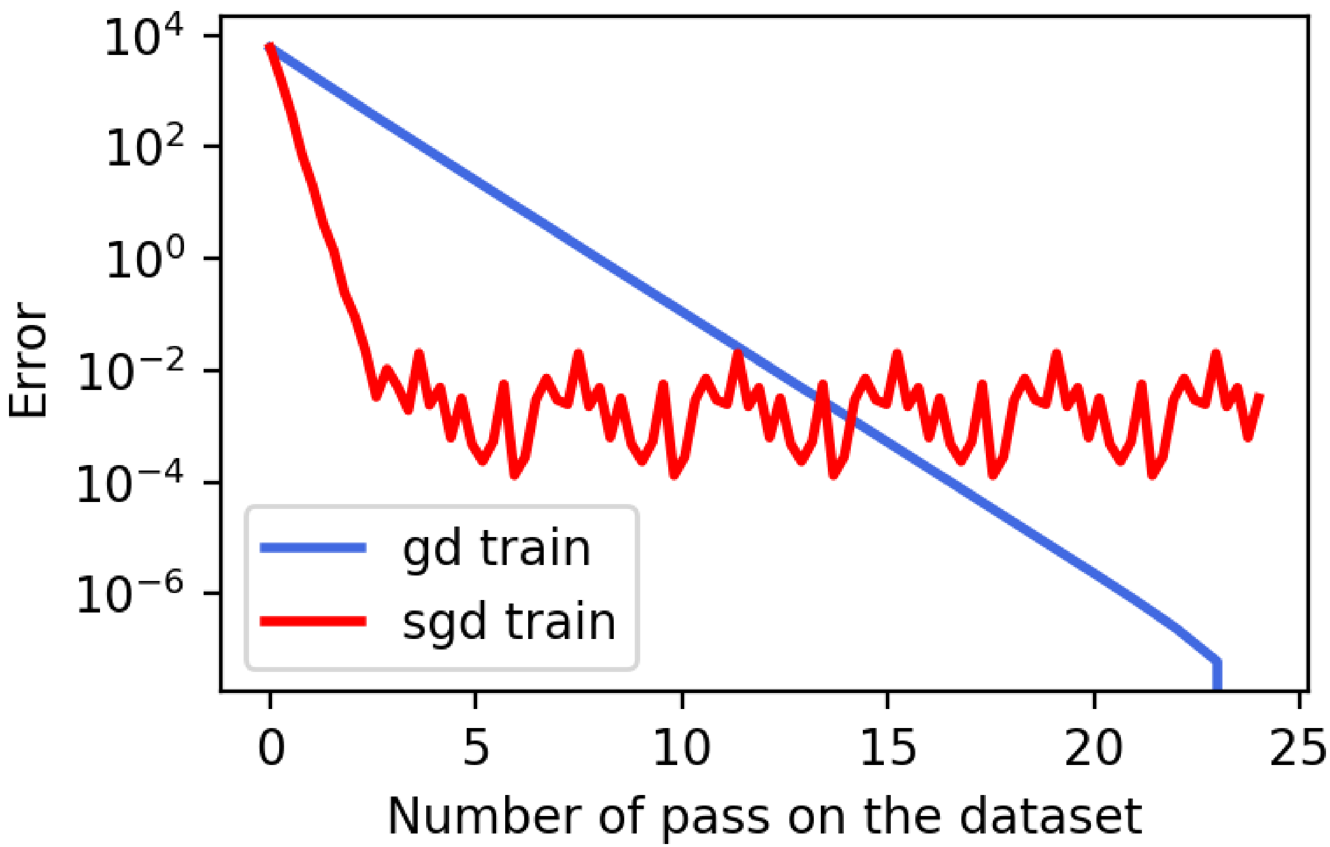
sample $(x, y) \sim \mathcal{D}$

calculate $v_t = \nabla_x \ell(h_{w^t}(x), y)$

$w^{t+1} = w^t - \alpha v_t$

Output $\overline{w}^T = \frac{1}{T}\sum_{t=1}^{T} w^t$

Train error


Error
10^4
10^2
10^0
10^-2
10^-4
10^-6
gd train
sgd train
0
5
10
15
20
25
Number of pass on the dataset

Train error and test error