Stochastic Gradient Descent

Qualitative Convergence Behavior

Pontus Giselsson

Outline

- **Stochastic gradient descent**
- Convergence and distance to solution
- Convergence and solution norms
- Overparameterized vs underparameterized setting
- Escaping not individually flat minima
- SGD step-sizes
- SGD convergence

Notation

- Optimization (decision) variable notation:
  - Optimization literature: x, y, z
  - Statistics literature: β
  - Machine learning literature: θ, w, b
- Data and labels in statistics and machine learning are x, y
- Training problems in supervised learning

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{N} L(m(x_i; \theta), y_i)$$

  optimizes over decision variable θ for fixed data $\{(x_i, y_i)\}_{i=1}^N$
- Optimization problem in standard optimization notation

$$\underset{x}{\text{minimize}}\, f(x)$$

  optimizes over decision variable x
- Will use optimization notation when algorithms not applied in ML

Gradient method

- Gradient method is applied problems of the form

$$\underset{x}{\text{minimize}}\, f(x)$$

  where f is differentiable and gradient method is

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

  where $\gamma_k > 0$ is a step-size
- f not differentiable in DL with ReLU but still say gradient method
- For large problems, gradient can be expensive to compute
  $\Rightarrow$ replace by unbiased stochastic approximation of gradient

Unbiased stochastic gradient approximation

- Stochastic gradient *estimator*:
  - notation: $\widehat{\nabla} f(x)$
  - outputs random vector in $\mathbb{R}^n$ for each $x \in \mathbb{R}^n$
- Stochastic gradient *realization*:
  - notation: $\widetilde{\nabla} f(x) : \mathbb{R}^n \to \mathbb{R}^n$
  - outputs, $\forall x \in \mathbb{R}^n$, vector in $\mathbb{R}^n$ drawn from distribution of $\widehat{\nabla} f(x)$
- An unbiased stochastic gradient estimator $\widehat{\nabla} f$ satisfies $\forall x \in \mathbb{R}^n$:

$$\mathbb{E}\widehat{\nabla} f(x) = \nabla f(x)$$

- If x is random vector in $\mathbb{R}^n$, unbiased estimator satisfies

$$\mathbb{E}[\widehat{\nabla} f(x)|x] = \nabla f(x)$$

(both are random vectors in $\mathbb{R}^n$)

Stochastic gradient descent (SGD)

- The following iteration generates $(x_k)_{k\in\mathbb{N}}$ of *random* variables:

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

  since $\widehat{\nabla} f$ outputs random vectors in $\mathbb{R}^n$
- Stochastic gradient descent finds a *realization* of this sequence:

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

  where $(x_k)_{k\in\mathbb{N}}$ here is a realization with values in $\mathbb{R}^n$
- Sloppy in notation for when x_k is *random variable* vs *realization*
- Can be efficient if evaluating $\widetilde{\nabla} f$ much cheaper than ∇f

Stochastic gradients – Finite sum problems

- Consider *finite sum problems* of the form

$$\underset{x}{\text{minimize}} \underbrace{\frac{1}{N}\left(\sum_{i=1}^{N} f_i(x)\right)}_{f(x)}$$

  where $\frac{1}{N}$ is for convenience and gives average loss
- Training problems of this form, where sum over training data
- Stochastic gradient: select f_i at random and take gradient step

Single function stochastic gradient

- Let I be a $\{1, \ldots, N\}$-valued random variable
- Let, as before, $\widehat{\nabla} f$ denote the stochastic gradient estimator
- Realization: let i be drawn from probability distribution of I

$$\widetilde{\nabla} f(x) = \nabla f_i(x)$$

where we will use uniform probability distribution

$$p_i = p(I = i) = \tfrac{1}{N}$$

- Stochastic gradient is unbiased:

$$\mathbb{E}[\widehat{\nabla} f(x)] = \sum_{i=1}^{N} p_i \nabla f_i(x) = \tfrac{1}{N} \sum_{i=1}^{N} \nabla f_i(x) = \nabla f(x)$$

Mini-batch stochastic gradient

- Let $\mathcal{B}$ be set of K-sample mini-batches to choose from:
  - Example: 2-sample mini-batches and $N = 4$:

$$\mathcal{B} = \{\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{3,4\}\}$$

  - Number of mini batches $\binom{N}{K}$, each item in $\binom{N-1}{K-1}$ batches
- Let $\mathbb{B}$ be $\mathcal{B}$-valued random variable
- Let, as before, $\widehat{\nabla} f$ denote stochastic gradient estimator
- Realization: let B be drawn from probability distribution of $\mathbb{B}$

$$\widetilde{\nabla} f(x) = \tfrac{1}{K} \sum_{i \in B} \nabla f_i(x)$$

  where we will use uniform probability distribution

$$p_B = p(\mathbb{B} = B) = \tfrac{1}{\binom{N}{K}}$$

- Stochastic gradient is unbiased:

$$\mathbb{E}\widehat{\nabla} f(x) = \tfrac{1}{\binom{N}{K}} \sum_{B \in \mathcal{B}} \tfrac{1}{K} \sum_{i \in B} \nabla f_i(x) = \tfrac{\binom{N-1}{K-1}}{\binom{N}{K} K} \sum_{i=1}^{N} \nabla f_i(x) = \tfrac{1}{N} \sum_{i=1}^{N} \nabla f_i(x) = \nabla f(x)$$

Stochastic gradient descent for finite sum problems

- The algorithm, choose $x_0 \in \mathbb{R}^n$ and iterate:
    1. Sample a mini-batch $B_k \in \mathcal{B}$ of K indices uniformly
    2. Update

$$x_{k+1} = x_k - \tfrac{\gamma_k}{K} \sum_{j \in B_k} \nabla f_j(x_k)$$

- Can have $\mathcal{B} = \{\{1\}, \ldots, \{N\}\}$ and sample only one function
- Gives realization of underlying stochastic process

Outline

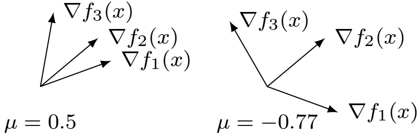
- Stochastic gradient descent
- **Convergence and distance to solution**
- Convergence and solution norms
- Overparameterized vs underparameterized setting
- Escaping not individually flat minima
- SGD step-sizes
- SGD convergence

Qualitative convergence behavior

- Consider single-function batch setting
- Assume that the individual gradients satisfy

$$(\nabla f_i(x))^T(\nabla f_j(x)) \geq \mu$$

for all i, j and for some $\mu \in \mathbb{R}$ (i.e., can be positive or negative)



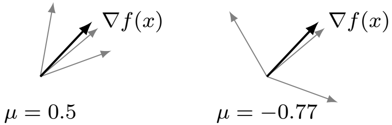
Will larger or smaller μ likely give better SGD convergence? Why?

Qualitative convergence behavior

- Consider single-function batch setting
- Assume that the individual gradients satisfy

$$(\nabla f_i(x))^T (\nabla f_j(x)) \geq \mu$$

for all i, j and for some $\mu \in \mathbb{R}$ (i.e., can be positive or negative)



Will larger or smaller μ likely give better SGD convergence? Why?

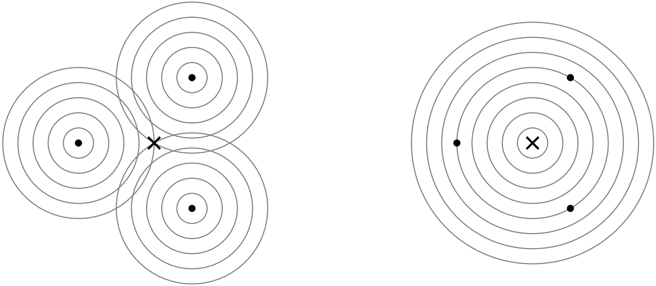
- Larger μ gives more similar to full gradient and faster convergence

Minibatch setting

- Larger minibatch gives larger μ and faster convergence
- Comes at the cost of higher per iteration count
- Limiting minibatch case is the gradient method
- Tradeoff in how large minibatches to use to optimize convergence
- Other reasons exist that favor small batches (later)

SGD – Example

- Let $c_1 + c_2 + c_3 = 0$
- Solve $\text{minimize}_x(\frac{1}{2}(\|x - c_1\|_2^2 + \|x - c_2\|_2^2 + \|x - c_3\|_2^2)) = \frac{3}{2}\|x\|_2^2 + c$
- How will trajectory look for SGD with $\gamma_k = 1/3$?

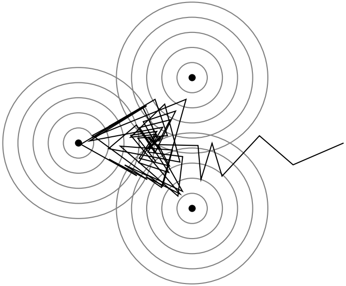


Levelsets of summands

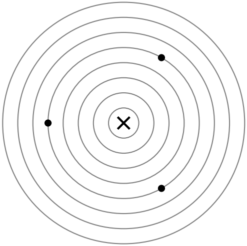
Levelset of sum

SGD – Example

- Let $c_1 + c_2 + c_3 = 0$
- Solve $\text{minimize}_x(\frac{1}{2}(\|x - c_1\|_2^2 + \|x - c_2\|_2^2 + \|x - c_3\|_2^2)) = \frac{3}{2}\|x\|_2^2 + c$
- How will trajectory look for SGD with $\gamma_k = 1/3$?



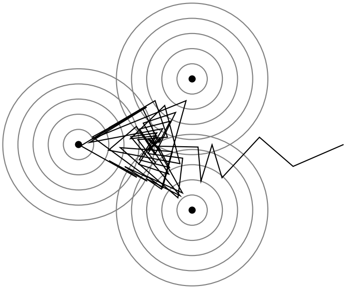
Levelsets of summands



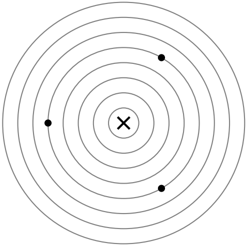
Levelset of sum

SGD – Example

- Let $c_1 + c_2 + c_3 = 0$
- Solve $\text{minimize}_x(\frac{1}{2}(\|x - c_1\|_2^2 + \|x - c_2\|_2^2 + \|x - c_3\|_2^2)) = \frac{3}{2}\|x\|_2^2 + c$
- How will trajectory look for SGD with $\gamma_k = 1/3$?



Levelsets of summands

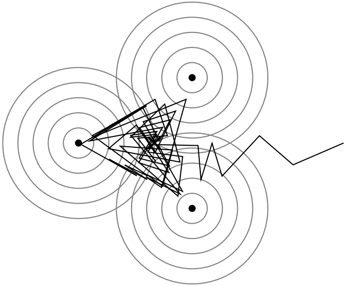


Levelset of sum

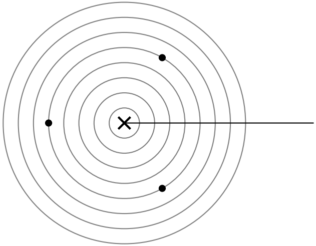
- Fast convergence outside "triangle" where gradients similar, slow inside
- Constant step SGD converges to noise ball

SGD – Example

- Let $c_1 + c_2 + c_3 = 0$
- Solve $\text{minimize}_x(\frac{1}{2}(\|x - c_1\|_2^2 + \|x - c_2\|_2^2 + \|x - c_3\|_2^2)) = \frac{3}{2}\|x\|_2^2 + c$
- How will trajectory look for SGD with $\gamma_k = 1/3$?



Levelsets of summands

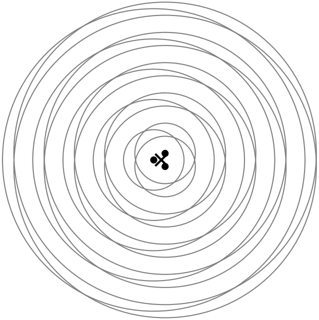


Levelset of sum

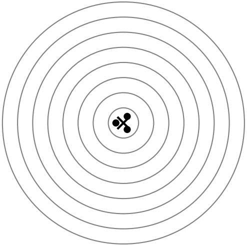
- Constant step GD converges (in this case straight to) solution (right)
- Difference is noise in stochastic gradient that can be measured by μ

SGD – Example zoomed out

- Same example but zoomed out
- Solve $\text{minimize}_x(\frac{1}{2}(\|x - c_1\|_2^2 + \|x - c_2\|_2^2 + \|x - c_3\|_2^2)) = \frac{3}{2}\|x\|_2^2 + c$
- How will trajectory look with $\gamma_k = 1/3$ from more global view?



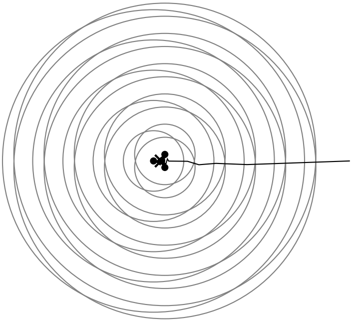
Levelsets of summands



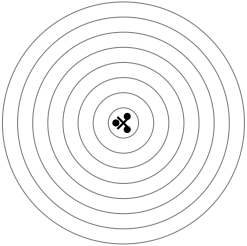
Levelset of sum

SGD – Example zoomed out

- Same example but zoomed out
- Solve $\text{minimize}_x(\frac{1}{2}(\|x-c_1\|_2^2 + \|x-c_2\|_2^2 + \|x-c_3\|_2^2)) = \frac{3}{2}\|x\|_2^2 + c$
- How will trajectory look with $\gamma_k = 1/3$ from more global view?



Levelsets of summands



Levelset of sum

- Far form solution ∇f_i more similar to ∇f, larger $\mu \Rightarrow$ faster convergence

Qualitative convergence behavior

- Often fast convergence far from solution, slow close to solution
- Fixed-step size converges to noise ball in general
- Need diminishing step-size to converge to solution in general

Drawback of diminishing step-size

- Diminishing step-size typically gives slow convergence
- Often better convergence with constant step (if it works)
- Is there a setting in which constant step-size works?

Outline

- Stochastic gradient descent
- Convergence and distance to solution
- **Convergence and solution norms**
- Overparameterized vs underparameterized setting
- Escaping not individually flat minima
- SGD step-sizes
- SGD convergence

Fixed step-size SGD does not converge to solution

- We can at most hope for finding point $\bar{x}$ such that

$$\nabla f(\bar{x}) = 0$$

- Let $x_k = \bar{x}$, and assume $\nabla f_i(x_k) \neq 0$, then

$$x_{k+1} = x_k - \gamma_k \nabla f_i(x_k) \neq x_k$$

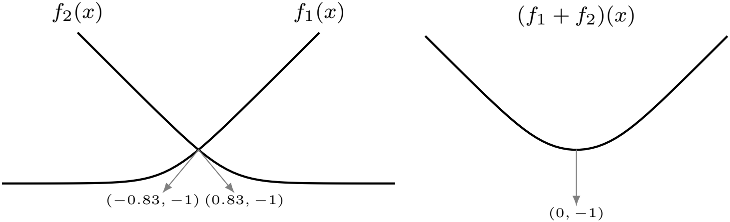
  i.e., moves away from solution $\bar{x}$
- Only hope with fixed step-size if all $\nabla f_i(\bar{x}) = 0$, since for $x_k = \bar{x}$

$$x_{k+1} = x_k - \gamma_k \nabla f_i(x_k) = x_k$$

  independent on γ_k and algorithm stays at solution
- How does norm of individual gradients affect local convergence?

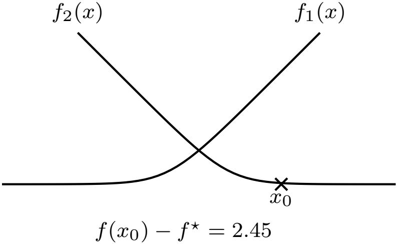
Example – Large gradients at solution

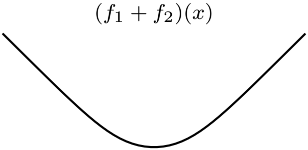
- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:



Example – Large gradients at solution

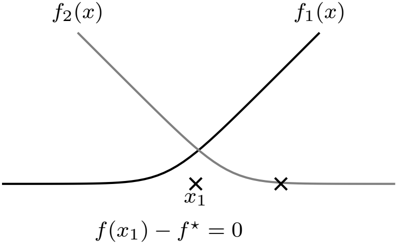
- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:

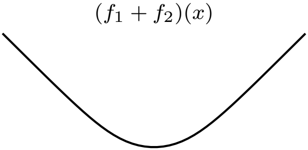




Example – Large gradients at solution

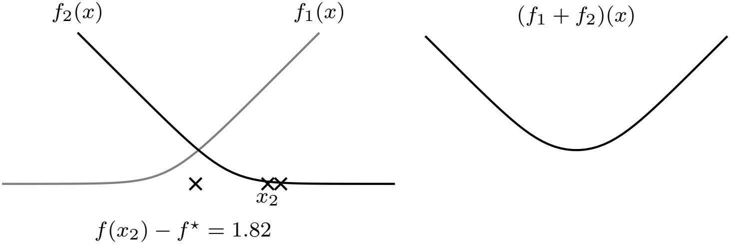
- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:





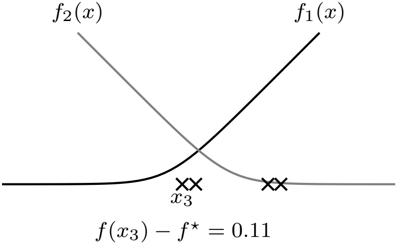
Example – Large gradients at solution

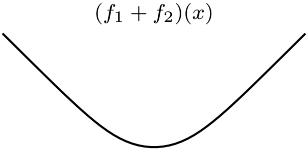
- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:



Example – Large gradients at solution

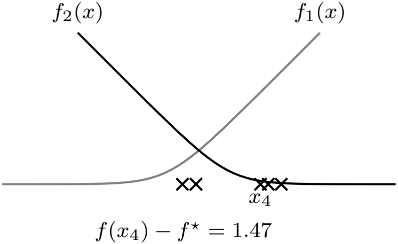
- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:

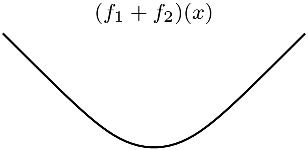




Example – Large gradients at solution

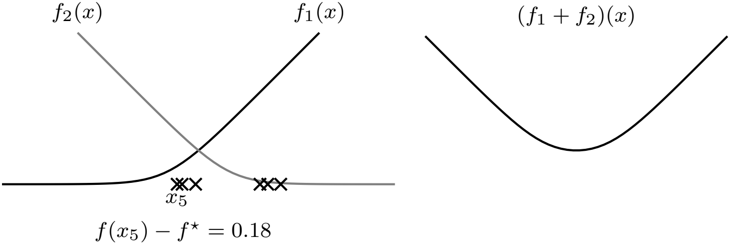
- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:





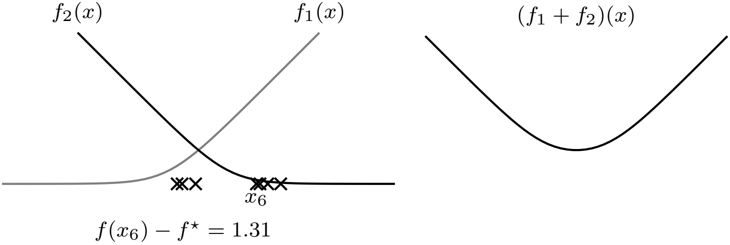
Example – Large gradients at solution

- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:



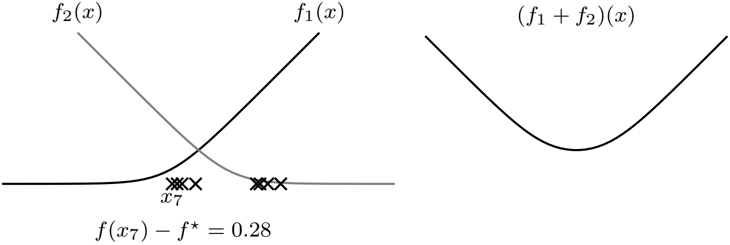
Example – Large gradients at solution

- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:



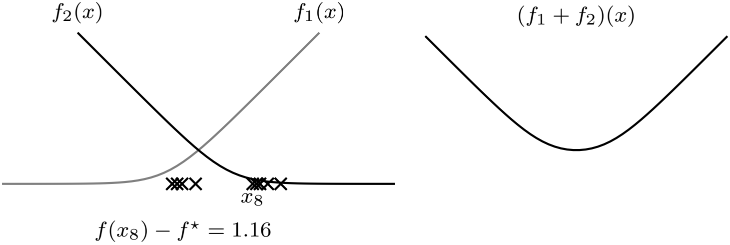
Example – Large gradients at solution

- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:



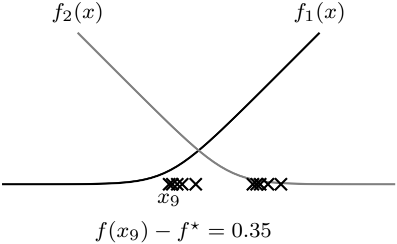
Example – Large gradients at solution

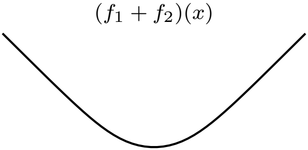
- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:



Example – Large gradients at solution

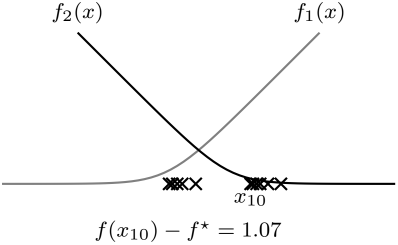
- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:

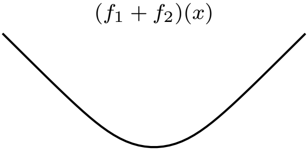




Example – Large gradients at solution

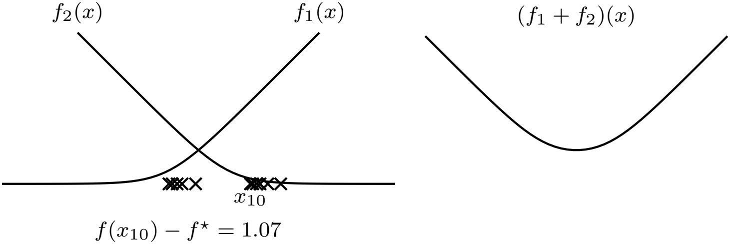
- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:





Example – Large gradients at solution

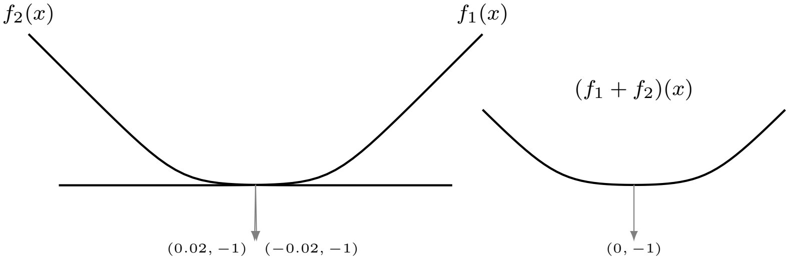
- Individal gradients at solution 0: $\nabla f_1(0) = 0.83$, $\nabla f_2(0) = -0.83$
- SGD with $\gamma = 0.07$ and cyclic update order:



- Will not converge to solution with constant step-size

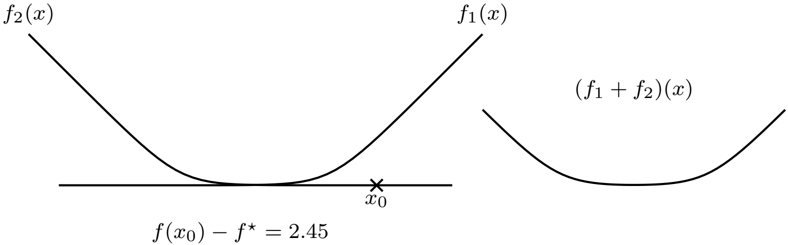
Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



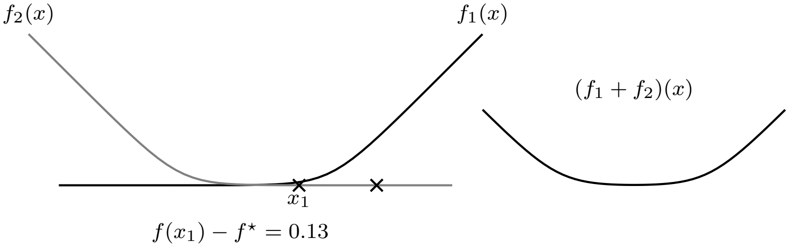
Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



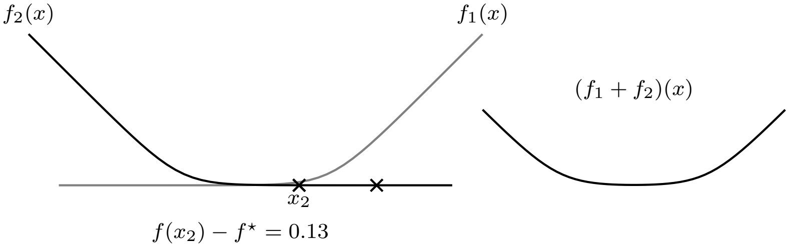
Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



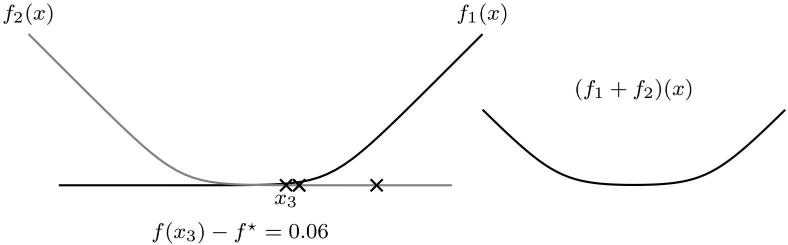
Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



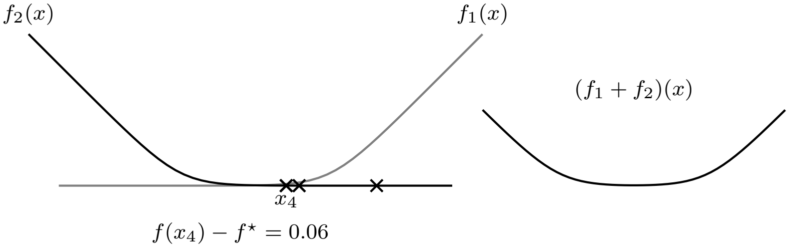
Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



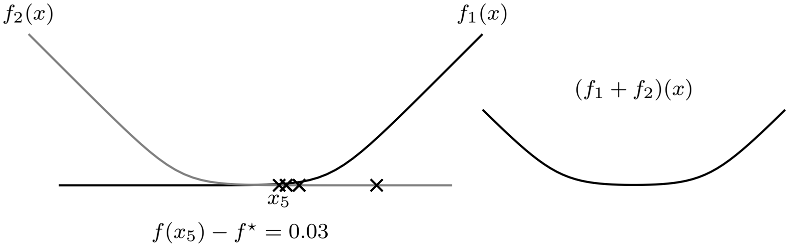
Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



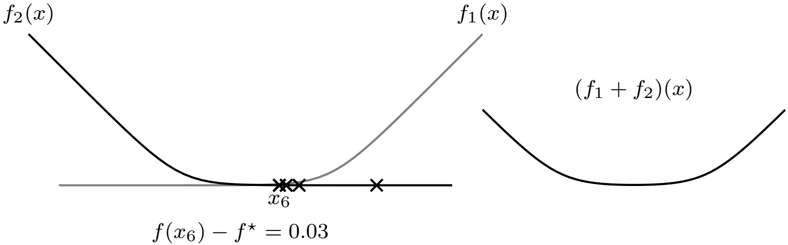
Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



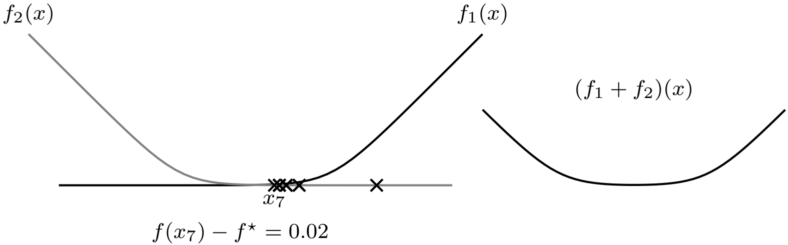
Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



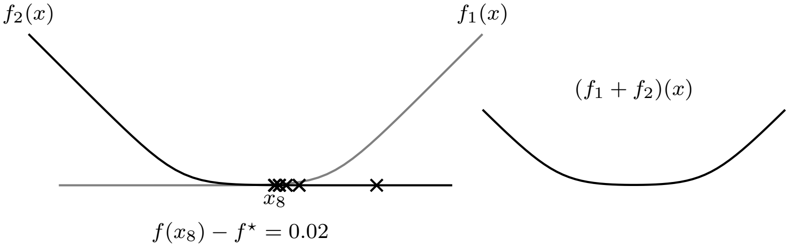
Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



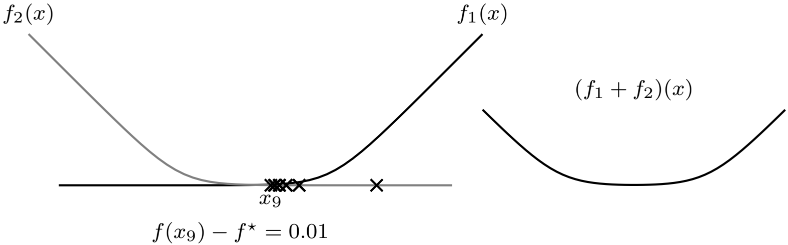
Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



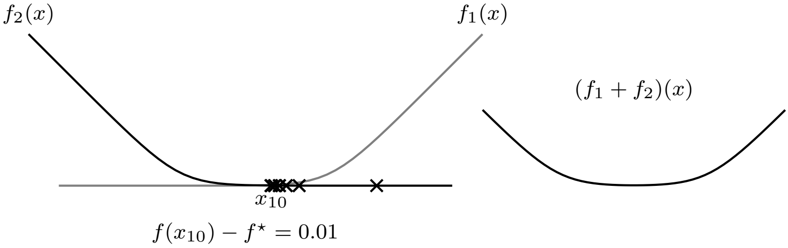
Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



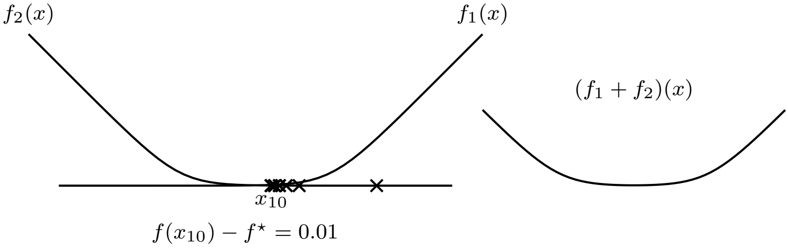
Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



Example – Small gradients at solution

- Shift f_1 and f_2 "outwards" to get new problem
- Individal gradients at solution 0: $\nabla f_1(0) = 0.02$, $\nabla f_2(0) = -0.02$
- SGD with $\gamma = 0.07$ and cyclic update order:



- Much faster to reach small loss

Convergence and individual gradient norm

Local convergence of stochastic gradient descent is:

- slow if individual functions do not agree on minima
  - individual norms "large" at and around minima
- faster if individual functions do agree on minima
  - individual norms "small" at and around minima

Outline

- Stochastic gradient descent
- Convergence and distance to solution
- Convergence and solution norms
- **Overparameterized vs underparameterized setting**
- Escaping not individually flat minima
- SGD step-sizes
- SGD convergence

Over- vs under-parameterized models

- Model overparameterized if:
  - in regression, zero loss is possible
  - in classification, correct classification with margin possible
    - logistic loss gives close to 0 loss
    - hinge loss gives 0 loss
- Model underparameterized if the above does not hold

Overparameterization – LS example

- Data $A \in \mathbb{R}^{N \times n}$, $b \in \mathbb{R}^N$, and $x \in \mathbb{R}^n$
- Consider least squares problem

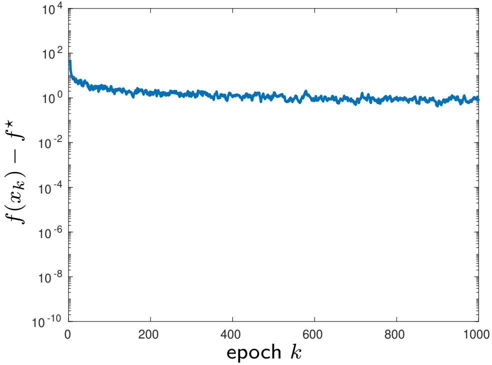
$$\underset{x}{\text{minimize}} \underbrace{\tfrac{1}{2}\|Ax - b\|_2^2}_{f(x)} = \sum_{i=1}^{N} \underbrace{\tfrac{1}{2}(a_i x - b_i)^2}_{f_i(x)}$$

where $a_i \in \mathbb{R}^{1 \times n}$ are rows in A and problem is

- overparameterized if $n > N$ (infinitely many 0-loss solutions)
- underparameterized if $n \leq N$ (unique solution if A full rank)

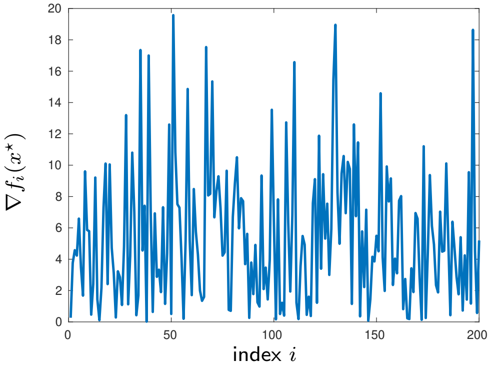
Convergence – LS example

- Random problem data: $A \in \mathbb{R}^{200\times 100}$, $b \in \mathbb{R}^{200}$ from Gaussian
- Underparameterized setting and unique solution
- Local convergence of SGD quite slow:



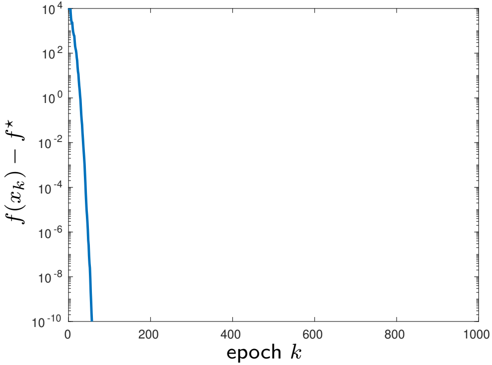
Convergence – LS example

- Random problem data: $A \in \mathbb{R}^{200\times 100}$, $b \in \mathbb{R}^{200}$ from Gaussian
- Underparameterized setting and unique solution
- Norms of $\nabla f_i(x^\star) = \frac{1}{2}(a_i x^\star - b_i)$ quite large:



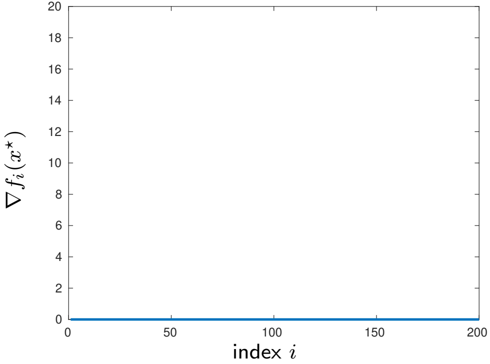
Convergence – LS example

- Random problem data: $A \in \mathbb{R}^{200\times 1000}$, $b \in \mathbb{R}^{200}$ from Gaussian
- Overparameterized, many 0-loss solutions, larger problem
- Convergence of SGD much faster:



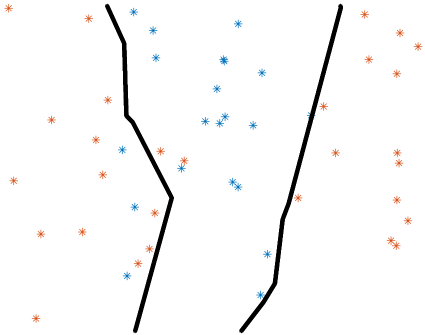
Convergence – LS example

- Random problem data: $A \in \mathbb{R}^{200\times 1000}$, $b \in \mathbb{R}^{200}$ from Gaussian
- Overparameterized, many 0-loss solutions, larger problem
- Individual norms $\nabla f_i(x^\star) = \frac{1}{2}(a_i x^\star - b_i) = 0$:



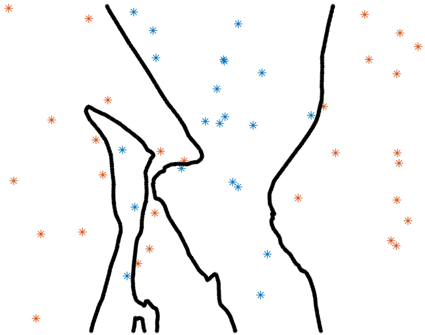
Convergence – DL example

- Classification problem: logistic loss
- Network: Residual, ReLU, 3x5,2,1 widths (5 layers)
- Underparameterized:



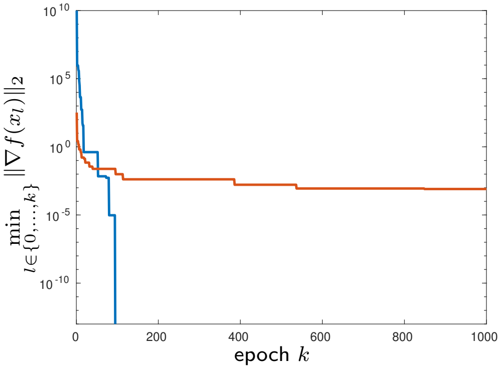
Convergence – DL example

- Classification problem: logistic loss
- Network: Residual, ReLU, 15x25,2,1 widths (17 layers)
- Overparameterized:



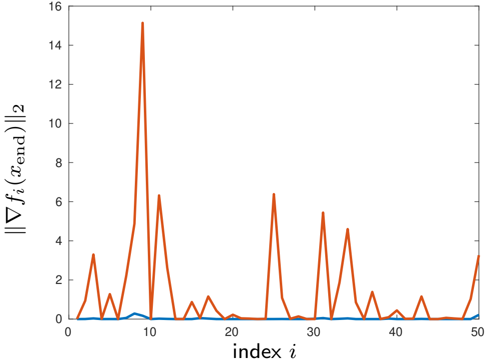
Convergence – DL example

- Classification problem: logistic loss
- Network: Residual, ReLU, 3x5,2,1 vs 15x25,2,1
- Convergence of "best gradient" (final loss: 0.17 vs 0.00018):



Convergence – DL example

- Classification problem: logistic loss
- Network: Residual, ReLU, 3x5,2,1 vs 15x25,2,1
- Final norm of individual gradients (final loss: 0.17 vs 0.00018):



Overparameterized networks and convergence

- Overparameterized models seems to give faster SGD convergence
- Reason: individual gradients agree better!

Outline

- Stochastic gradient descent
- Convergence and distance to solution
- Convergence and solution norms
- Overparameterized vs underparameterized setting
- **Escaping not individually flat minima**
- SGD step-sizes
- SGD convergence

Step-length

- The step-length in constant step SGD is given by

$$\|x_{k+1} - x_k\|_2 = \gamma \|\nabla f_i(x_k)\|_2$$

  i.e., proportional to individual gradient norm
- The step-length in constant step GD is given by

$$\|x_{k+1} - x_k\|_2 = \gamma \|\nabla f(x_k)\|_2$$

  i.e., proportional to full (average) gradient norm

Flatness of minima

- Is SGD or GD more likely to escape the sharp minima?

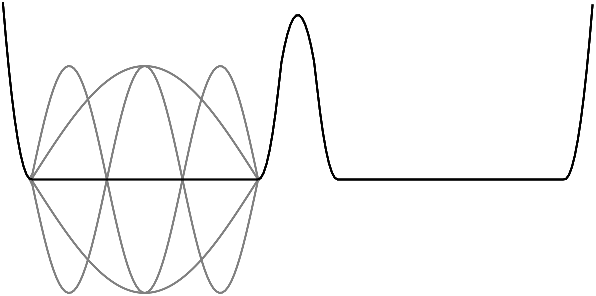
Flatness of minima

- Is SGD or GD more likely to escape the sharp minima?

- Impossible to say only from average training loss

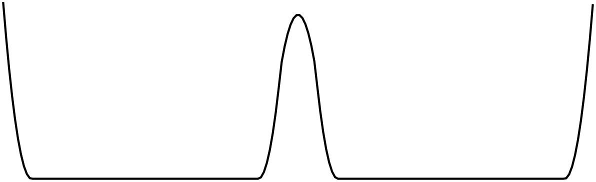
Example

- Flat (local) minima can be different
- Is SGD or GD more likely to escape right/left minima?



Example

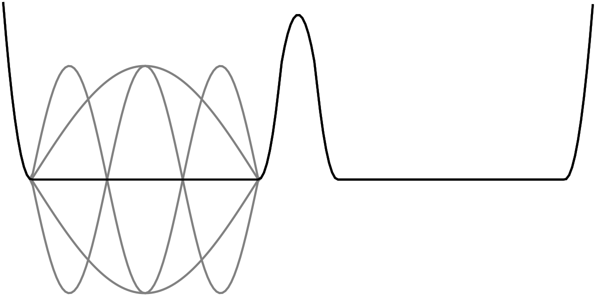
- Flat (local) minima can be different
- Is SGD or GD more likely to escape right/left minima?



- GD will stay in both minima ($\nabla f(x_k) = 0 \Rightarrow x_{k+1} = x_k$)

Example

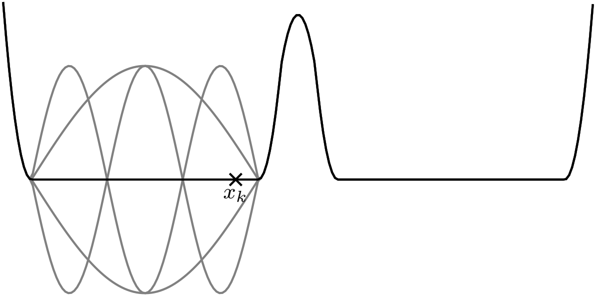
- Flat (local) minima can be different
- Is SGD or GD more likely to escape right/left minima?



- GD will stay in both minima ($\nabla f(x_k) = 0 \Rightarrow x_{k+1} = x_k$)
- SGD will stay in right minima ($\nabla f_i(x_k) = 0 \Rightarrow x_{k+1} = x_k$)
- SGD may escape left minima ($\|\nabla f_i(x_k)\|_2 \neq 0 \Rightarrow x_{k+1} \neq x_k$)

Example

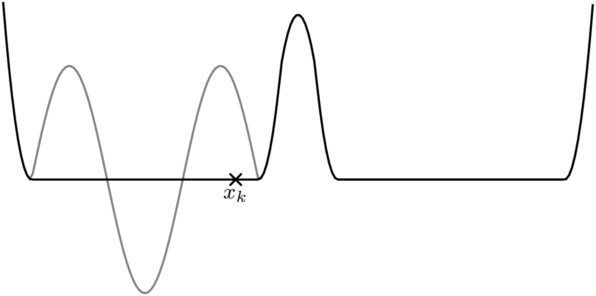
- Flat (local) minima can be different
- Is SGD or GD more likely to escape right/left minima?



- GD will stay in both minima ($\nabla f(x_k) = 0 \Rightarrow x_{k+1} = x_k$)
- SGD will stay in right minima ($\nabla f_i(x_k) = 0 \Rightarrow x_{k+1} = x_k$)
- SGD may escape left minima ($\|\nabla f_i(x_k)\|_2 \neq 0 \Rightarrow x_{k+1} \neq x_k$)
- $x_k = 0.8$ and $\gamma = 0.5$

Example

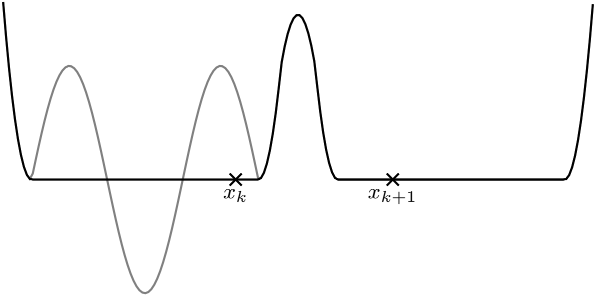
- Flat (local) minima can be different
- Is SGD or GD more likely to escape right/left minima?



- GD will stay in both minima ($\nabla f(x_k) = 0 \Rightarrow x_{k+1} = x_k$)
- SGD will stay in right minima ($\nabla f_i(x_k) = 0 \Rightarrow x_{k+1} = x_k$)
- SGD may escape left minima ($\|\nabla f_i(x_k)\|_2 \neq 0 \Rightarrow x_{k+1} \neq x_k$)
- $x_k = 0.8$ and $\gamma = 0.5$, $i = 4$ and $\nabla f_i(x_k) = -2.77$

Example

- Flat (local) minima can be different
- Is SGD or GD more likely to escape right/left minima?



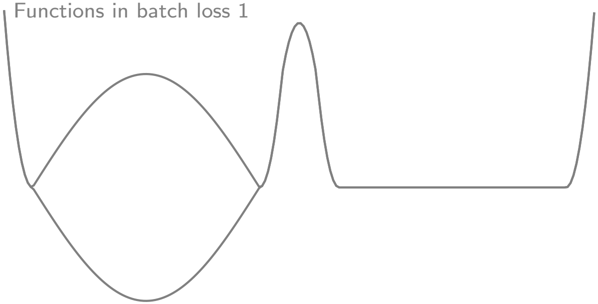
- GD will stay in both minima ($\nabla f(x_k) = 0 \Rightarrow x_{k+1} = x_k$)
- SGD will stay in right minima ($\nabla f_i(x_k) = 0 \Rightarrow x_{k+1} = x_k$)
- SGD may escape left minima ($\|\nabla f_i(x_k)\|_2 \neq 0 \Rightarrow x_{k+1} \neq x_k$)
- $x_k = 0.8$ and $\gamma = 0.5$, $i = 4$ and $\nabla f_i(x_k) = -2.77$, $x_{k+1} = 2.18$

Mini-batch vs single-batch

- Is escape property effected by mini-batch size?
- How large mini-batch size is best for escaping?

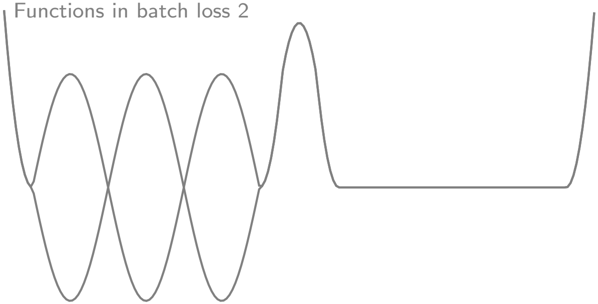
Mini-batch setting

- Use mini-batches of size 2:



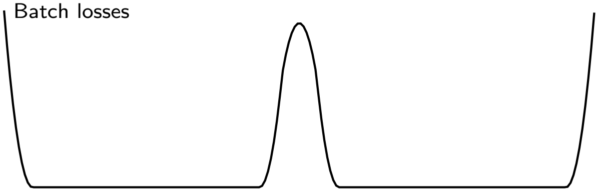
Mini-batch setting

- Use mini-batches of size 2:



Mini-batch setting

- Use mini-batches of size 2:



- Larger mini-batch ⇒ smaller gradients ⇒ worse at escaping
- Single-batch better at escaping

Connection to generalization

- Argued that individually flat minima generalize better, i.e.,

  all $\|\nabla f_i(x)\|_2$ small in region around minima

- SGD more likely to escape if individual gradients not small
- Smaller batch size increases chances of escaping "bad" minima

Have also argued for:

- Good convergence properties towards individually flat minima

In summary:

- Single-batch SGD well suited for overparameterized training

Outline

- Stochastic gradient descent
- Convergence and distance to solution
- Convergence and solution norms
- Overparameterized vs underparameterized setting
- Escaping not individually flat minima
- **SGD step-sizes**
- SGD convergence

Step-sizes

- Diminising step-sizes are needed for convergence in general
- Common static step-size rules
  - redude step-size every K epochs:

$$\gamma_k = \frac{\gamma_0}{1 + \lceil k/K \rceil} \qquad \gamma_k = \frac{\gamma_0}{1 + \sqrt{\lceil k/K \rceil}}$$

    where $\lceil k/K \rceil$ increases by 1 every K epochs
  - Convergence analysis under smoothness or convexity requires

$$\sum_{k=0}^{\infty} \gamma_k = \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \gamma_k^2 < \infty$$

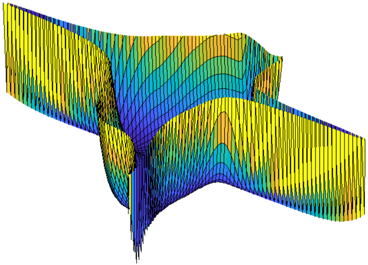
    which is satisfied by first but not second above
  - Refined analysis gives requirements

$$\sum_{k=0}^{\infty} \gamma_k = \infty \quad \text{and} \quad \frac{\sum_{k=0}^{\infty} \gamma_k}{\sum_{k=0}^{\infty} \gamma_k^2} = \infty$$

    which is satisfied by all the above

Large gradients

- Fixed step-size rules does not take gradient size into account
- Gradients can be very large:



- Step-size rule

$$\gamma_k = \frac{\gamma_0}{\alpha\|\widetilde{\nabla} f(x_k)\|_2 + 1}$$

with $\gamma_0, \alpha > 0$ gives
  - small steps if $\|\widetilde{\nabla} f(x_k)\|_2$ large
  - approximately γ_0 steps if $\|\widetilde{\nabla} f(x_k)\|_2$ small

Combined step-size rule

- Combination the two previous rules

$$\gamma_k = \frac{\gamma_0}{(1+\psi(\lceil k/K \rceil))(\alpha\|\widetilde{\nabla} f(x_k)\|_2 + 1)}$$

  where, e.g., $\psi(x) = \frac{1}{x}$ or $\psi(x) = \frac{1}{\sqrt{x}}$ (as before)
- Properties
  - $\|\widetilde{\nabla} f(x_k)\|_2$ large: small step-sizes
  - $\|\widetilde{\nabla} f(x_k)\|_2$ small: diminshing step-sizes according to $\frac{\gamma_0}{1+\psi(\lceil k/K \rceil)}$

Step-size rules and convergence

- Classification, Residual layers, ReLU, 15x25,2,1 widths (17 layers)
- Step-size parameters: $\psi(x) = 0.5\sqrt{x}$, $K = 50$, $\alpha = \gamma_0 = 0.1$
- Iteration data:

# epoch	step-size	batch norm	full norm
0	$4.8 \cdot 10^{-8}$	$2.1 \cdot 10^{7}$	$6.8 \cdot 10^{5}$
10	$1.4 \cdot 10^{-5}$	$7.2 \cdot 10^{4}$	$1.4 \cdot 10^{4}$
50	0.097	0.31	1.4
100	0.016	0.28	3.2
200	0.012	$6.8 \cdot 10^{-5}$	0.72
300	0.01	0.33	11.8
500	0.008	0	0.529
700	0.007	$1.2 \cdot 10^{-6}$	0.0008
1000	0.006	$3.1 \cdot 10^{-6}$	0.0003

- Large initial gradients dampened
- Diminishing step-size gives local convergence

Step-size rules and convergence

- Classification, Residual layers, ReLU, 15x25,2,1 widths (17 layers)
- Step-size parameters: $\psi(x) = 0.5\sqrt{x}$, $K = 50$, $\alpha = 0$, $\gamma_0 = 0.1$
- Iteration data:

# epoch	step-size	batch norm	full norm
1	0.1	$1.2 \cdot 10^6$	$6.8 \cdot 10^5$
2	-	NaN	NaN
50	-	NaN	NaN
100	-	NaN	NaN
200	-	NaN	NaN
300	-	NaN	NaN
500	-	NaN	NaN
700	-	NaN	NaN
1000	-	NaN	NaN

- No adaptation to large gradients – Gradient explodes
- Diminishing step-size does of course not help

Step-size rules and convergence

- Classification, Residual layers, ReLU, 15x25,2,1 widths (17 layers)
- Step-size parameters: $\psi \equiv 0$, $\alpha = \gamma_0 = 0.1$
- Iteration data:

# epoch	step-size	batch norm	full norm
0	$1.4 \cdot 10^{-7}$	$7.0 \cdot 10^{6}$	$4.7 \cdot 10^{5}$
10	0.004	257	39.4
50	0.10	$6.2 \cdot 10^{-10}$	4.1
100	0.087	1.5	1.3
200	0.089	1.2	0.26
300	0.1	$2.0 \cdot 10^{-12}$	1.3
500	0.1	$5.1 \cdot 10^{-12}$	0.198
700	0.1	$2.4 \cdot 10^{-13}$	0.16
1000	0.087	1.5	0.013

- Large initial gradients dampened
- Larger final full norm than first choice since not diminishing γ_k

Outline

- Stochastic gradient descent
- Convergence and distance to solution
- Convergence and solution norms
- Overparameterized vs underparameterized setting
- Escaping not individually flat minima
- SGD step-sizes
- **SGD convergence**

Convergence analysis

- Need some inequality that function satisfies to analyze SGD
- Convexity inequality not applicable in deep learning
- Smoothness inequality not applicable in deep learning in general
  - ReLU networks are not differentiable and therefore not smooth
  - Tanh networks with smooth loss are cont. diff. $\Rightarrow$ locally smooth
- We have seen that training problem is piece-wise polynomial if
  - L2 loss and piece-wise linear activation functions
  - hinge loss and piece-wise linear activation functions

  but does not provide an inequality for proving convergence

Error bound

- In absence of convexity, an *error bound* is useful in analysis:

$$\delta(f(x) - f(x^\star)) \le \|\nabla f(x)\|_2^2$$

  that holds locally around solution $x^\star$ with $\delta > 0$
- Gradient in error bound can be replaced by
  - sub-gradient for convex nondifferentiable f
  - limiting sub-gradient for nonconvex nondifferentiable f

Kurdyka-Lojasiewicz

- Error bound is instance of the Kurdyka-Lojasiewicz (KL) property
- KL property has exponent $\alpha \in [0, 1)$, $\alpha = \frac{1}{2}$ gives error bound
- Examples of KL functions:
  - Continuous (on closed domain) semialgebraic functions are KL:

    $$\mathrm{graph} f = \cup_{i=1}^{r} \left(\cap_{j=1}^{q} \{x : h_{ij}(x) = 0\} \cap_{l=1}^{p} \{x : g_{il}(x) < 0\} \right)$$

    graph is union of intersection, where h_{ij} and g_{il} polynomials
  - Continuous piece-wise polynomials (some DL training problems)
  - Strongly convex functions
- Often difficult to decide KL-exponent
- Result: descent methods on KL functions converge
  - sublinearly if $\alpha \in (\frac{1}{2}, 1)$
  - linearly if $\alpha \in (0, \frac{1}{2}]$ (the error bound regime)

Strongly convex functions satisfy error bound

- $s + \sigma x \in \partial f(x)$ with $s \in \partial g(x)$ for convex $g = f - \frac{\sigma}{2}\|\cdot\|_2^2$
- Therefore

$$\begin{aligned}
\|s + \sigma x\|_2^2 &= \|s\|_2^2 + 2\sigma s^T x + \sigma^2 \|x\|_2^2 \\
&\geq \|s\|_2^2 + 2\sigma s^T x^\star + 2\sigma(g(x) - g(x^\star)) + \sigma^2\|x\|_2^2 \\
&= \|s\|_2^2 + 2\sigma s^T x^\star + \sigma\|x^\star\|_2^2 + 2\sigma(f(x) - f(x^\star)) \\
&= \|s + \sigma x^\star\|_2^2 + 2\sigma(f(x) - f(x^\star)) \\
&\geq 2\sigma(f(x) - f(x^\star))
\end{aligned}$$

where we used

- subgradient definition $g(x^\star) \geq g(x) + s^T(x^\star - x)$ in first inequality
- nonnegativity of norms in the second inequality

Implications of error bound

- Restating error bound for differentiable case

$$\delta(f(x) - f(x^\star)) \leq \|\nabla f(x)\|_2^2$$

- Assume it holds for all x in some ball X around solution $x^\star$
- What can you say about local minima and saddle-points in X?

Implications of error bound

- Restating error bound for differentiable case

$$\delta(f(x) - f(x^\star)) \leq \|\nabla f(x)\|_2^2$$

- Assume it holds for all x in some ball X around solution $x^\star$
- What can you say about local minima and saddle-points in X?
- There are none! Proof by contradiction:
  - Assume local minima or saddle-point $\bar{x}$
  - Then $\nabla f(\bar{x}) = 0 \Rightarrow f(\bar{x}) = f(x^\star)$ and $\bar{x}$ is global minima

Convergence analysis – Smoothness and error bound

- Convergence analysis of gradient method
- β-smoothness and error bound assumptions ($f^\star = f(x^\star)$):

$$
\begin{aligned}
f(x_{k+1}) - f^\star &\leq f(x_k) - f^\star + \nabla f(x_k)^T(x_{k+1} - x_k) + \tfrac{\beta}{2}\|x_k - x_{k+1}\|_2^2 \\
&= f(x_k) - f^\star - \gamma_k\|\nabla f(x_k)\|_2^2 + \tfrac{\beta\gamma_k^2}{2}\|\nabla f(x_k)\|_2^2 \\
&= f(x_k) - f^\star - \gamma_k(1 - \tfrac{\beta\gamma_k}{2})\|\nabla f(x_k)\|_2^2 \\
&\leq (1 - \gamma_k\delta(1 - \tfrac{\beta\gamma_k}{2}))(f(x_k) - f^\star)
\end{aligned}
$$

  where

  - β-smoothness of f is used in first inequality
  - gradient update $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$ in first equality
  - error bound is used in the final inequality

- Linear convergence in function values if $\gamma_k \in [\epsilon, \frac{2}{\beta} - \epsilon]$, $\epsilon > 0$

Semi-smoothness

- Typical DL training problems are not smooth
  - E.g.: overparameterized ReLU networks with smooth loss
- But semi-smooth[1] in neighborhood around random initialization[2]:

$$f(x) \leq f(y) + \nabla f(y)^T(x-y) + c\|x-y\|_2\sqrt{f(y)} + \tfrac{\beta}{2}\|x-y\|_2^2$$

  for some constants c and β
  - Holds locally for large enough c, β if cont. piece-wise polynomial
  - Constants and neighborhood quantified in [1][2]
- $c = 0$ gives smoothness
- c small gives close to smoothness but allows nondifferentiable

---

1 Semismoothness definition not a standard semismoothness definition
2 [1] A Convergence Theory for Deep Learning via Over-Parameterization. Z. Allen-Zhu et al.

Convergence – Error bound and semi-smoothness

- Convergence analysis of gradient descent method
- Assumptions: (c,β)-semi-smooth, δ-error bound, $f^\star = 0$ (w.l.o.g.)
- Parameters $c \leq \frac{\sqrt{\delta}\gamma\beta}{2}$ and $\gamma \in (0, \frac{1}{\beta})$:

$$
\begin{aligned}
& f(x_{k+1}) \\
& \leq f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) + c\|x_{k+1} - x_k\|\sqrt{f(x_k)} + \tfrac{\beta}{2}\|x_{k+1} - x_k\|_2^2 \\
& = f(x_k) - \gamma\|\nabla f(x_k)\|_2^2 + c\gamma\|\nabla f(x_k)\|\sqrt{f(x_k)} + \tfrac{\beta\gamma^2}{2}\|\nabla f(x_k)\|_2^2 \\
& \leq f(x_k) - \gamma\|\nabla f(x_k)\|_2^2 + \tfrac{c\gamma}{\sqrt{\delta}}\|\nabla f(x_k)\|^2 + \tfrac{\beta\gamma^2}{2}\|\nabla f(x_k)\|_2^2 \\
& \leq f(x_k) - \gamma\|\nabla f(x_k)\|_2^2 + \beta\gamma^2\|\nabla f(x_k)\|^2 \\
& \leq f(x_k) - \gamma(1 - \beta\gamma)\|\nabla f(x_k)\|_2^2 \\
& \leq (1 - c\gamma(1 - \beta\gamma))f(x_k)
\end{aligned}
$$

  which shows linear convergence to 0 loss
- Need the nonsmooth part of upper bound c to be small enough
- Can analyze SGD in similar manner

Convergence in deep learning

- Setting: ReLU network, fully connected, smooth loss
- c is small enough when model overparameterized enough [1][1]
- Linear convergence (with high prob.) for random initialization [1]
- In practice:
  - β will be big – relies on small enough ($\leq \frac{1}{\beta}$) constant step-size
  - need to find "correct" step-size by diminishing rule
  - need to control steps to not depart from linear convergence region
  - hopefully achieved by previous step-size rule

[1] [1] A Convergence Theory for Deep Learning via Over-Parameterization. Z. Allen-Zhu et al.