Exercises on Optimization for Machine Learning

M2 IASD 2025-2026

Version 4.2 - November 5, 2025*

Dauphine | PSL
UNIVERSITÉ PARIS

1. Basics of optimization

Exercise 1.1: Support vector machines

Given a dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$, finding a binary linear classifier for the data using support vector machines amounts to considering the problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}} \frac{1}{n}\sum_{i=1}^n \max\left\{1 - y_i \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}, 0\right\} + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2, \tag{1}$$

where $\lambda \geq 0$.

Show that the objective of (1) is a convex function for any $\lambda \geq 0$, and that it is λ-strongly convex when $\lambda > 0$.

Exercise 1.2: Least-squares problems

Let $\boldsymbol{x} \in \mathbb{R}^d$ such that $\|\boldsymbol{x}\|_2 \neq 0$ and $\boldsymbol{y} \in \mathbb{R}^d$. The goal of the exercise is to compute a linear model that best fits $\boldsymbol{x}$ to $\boldsymbol{y}$, and to show the interest of using overparameterization in that setting.

a) Consider first the one-dimensional problem

$$\underset{\boldsymbol{w}\in\mathbb{R}}{\text{minimize}} \frac{1}{2}\|w\boldsymbol{x} - \boldsymbol{y}\|_2^2. \tag{2}$$

Is problem (2) convex? Is its optimal value equal to 0?

b) Consider now the matrix problem

$$\underset{\boldsymbol{W}\in\mathbb{R}^{d\times d}}{\text{minimize}} \frac{1}{2}\|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{y}\|_2^2. \tag{3}$$

Reformulate the problem so as to show that the objective function is convex in the sense defined in class, i.e. for vector-valued functions.

*Thanks to the students who spotted typos in earlier versions!

c) Show that

$$\underset{\boldsymbol{W}\in\mathbb{R}^{d\times d}}{\text{minimize}}\ \frac{1}{2}\|\boldsymbol{W}\boldsymbol{x}-\boldsymbol{y}\|_2^2 = 0$$

and that the problem does not have a unique global minimum.

Exercise 1.3: Strong convexity

Let $f:\mathbb{R}^d\to\mathbb{R}$ be $\mathcal{C}^1$ and μ-strongly convex, and denote by $\boldsymbol{w}^*$ the minimum of f.

a) For any $\boldsymbol{w}\in\mathbb{R}^d$, justify that the function

$$\varphi_{\boldsymbol{w}}:\boldsymbol{z}\ \longmapsto\ f(\boldsymbol{w})+\nabla f(\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{z}-\boldsymbol{w})+\frac{\mu}{2}\|\boldsymbol{z}-\boldsymbol{w}\|^2$$

is $\mathcal{C}^1$ and strongly convex.

b) Using $\nabla\varphi_{\boldsymbol{w}}(\boldsymbol{z})=\nabla f(\boldsymbol{w})+\mu(\boldsymbol{z}-\boldsymbol{w})$ for any $\boldsymbol{z}\in\mathbb{R}^d$, compute $\min_{\boldsymbol{z}}\varphi_{\boldsymbol{w}}(\boldsymbol{z})$ and $\operatorname{argmin}_{\boldsymbol{z}}\varphi_{\boldsymbol{w}}(\boldsymbol{z})$.

c) Conclude from the previous questions that

$$\|\nabla f(\boldsymbol{w})\|_2^2\quad\geq\quad 2\mu\,\left(f(\boldsymbol{w})-f(\boldsymbol{w}^*)\right).$$

Exercise 1.4: Co-coercivity

Let $f:\mathbb{R}^d\to\mathbb{R}$ be $\mathcal{C}_L^{1,1}$ and convex. Suppose that $\boldsymbol{w}^*\in\operatorname{argmin}_{\boldsymbol{w}} f(\boldsymbol{w})$ and let $f^*=f(\boldsymbol{w}^*)$.

a) Let $\boldsymbol{w}\in\mathbb{R}^d$. Show that $f(\boldsymbol{w})-f(\boldsymbol{w}^*)\geq\frac{1}{2L}\|\nabla f(\boldsymbol{w})\|_2^2$.

b) Let $(\boldsymbol{w},\boldsymbol{v})\in(\mathbb{R}^d)^2$. Show that

$$\left(\nabla f(\boldsymbol{v})-\nabla f(\boldsymbol{w})\right)^{\mathrm{T}}(\boldsymbol{v}-\boldsymbol{w})\geq\frac{1}{L}\|\nabla f(\boldsymbol{v})-\nabla f(\boldsymbol{w})\|_2^2.$$

Consider $\boldsymbol{z}\mapsto f(\boldsymbol{z})-\nabla f(\boldsymbol{v})^{\mathrm{T}}\boldsymbol{z}$ and $\boldsymbol{z}\mapsto f(\boldsymbol{z})-\nabla f(\boldsymbol{w})^{\mathrm{T}}\boldsymbol{z}$.

2. Derivatives and differentiation

Exercise 2.1: A simple function (Adapted from 2019-2020 exam)

Consider the one-dimensional function $f:\mathbb{R}\to\mathbb{R}$ defined by

$$f(w)=\sqrt{\sin(w)}+\sin(w). \tag{4}$$

a) Write a directed acylic graph representing the computation of $f(w)$.

b) Write down the instructions for computing $f'(w)$ in forward mode. Use it to evaluate $f'(\frac{\pi}{2})$.

c) Write down the instructions for computing $f'(w)$ in backward mode. Use it to evaluate $f'(\frac{\pi}{2})$.

Exercise 2.2: Sigmoid-type functions (Adapted from 2020-2021 exam)

Given $\boldsymbol{x} \in \mathbb{R}^d$ and $y \in [0,1]$, we consider the two functions $f, g : \mathbb{R}^d \to \mathbb{R}$ defined by

$$f(\boldsymbol{w}) = y \ln\left(\sigma(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w})\right) \quad \text{and} \quad g(\boldsymbol{w}) = (1-y) \ln\left(1-\sigma(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w})\right), \tag{5}$$

where $\sigma(t) = \frac{1}{1+\mathrm{e}^{-t}}$. Recall that $\sigma'(t) = \sigma(t)\,(1-\sigma(t))$.

a) Write a computational graph representing the calculation of $f = f(\boldsymbol{w})$ by considering dot product, $\sigma(\cdot)$, logarithm and product as elementary operations.

b) Write down the chain rule for computing $\frac{\partial f}{\partial \boldsymbol{w}}$ in forward mode.

c) Write down the chain rule for computing $\frac{\partial f}{\partial \boldsymbol{w}}$ in backward mode.

d) Write a computational graph representing the calculation of $h = f(\boldsymbol{w}) + g(\boldsymbol{w})$ based on the graph from question a).

e) Write down the chain rule for computing $\frac{\partial h}{\partial \boldsymbol{w}}$ in backward mode.

Exercise 2.3: Two-layer neural networks

Consider the function $f : \mathbb{R}^{h\times d_x} \times \mathbb{R}^{d_y \times h} \times \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \to \mathbb{R}$ defined by

$$f(\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\left\|\boldsymbol{W}_2 \boldsymbol{\sigma}(\boldsymbol{W}_1 \boldsymbol{x}) - \boldsymbol{y}\right\|_2^2, \tag{6}$$

where $\boldsymbol{\sigma}(\boldsymbol{v}) = [\tanh(v_i)]_{i=1,\dots,h}$. The derivative of $\tanh$ is given by $\frac{\partial \tanh}{\partial t}(t) = 1 + \tanh(t)^2$.

a) Give a formula for the gradient $\nabla_{\boldsymbol{x}} f = \frac{\partial f}{\partial \boldsymbol{x}}^{\mathrm{T}}$.

b) Give a formula for the derivative $\frac{\partial f}{\partial \boldsymbol{W}_1}$. What are this object's dimensions?

3. Gradient descent

Exercise 3.1: A strongly convex problem

Consider the minimization problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{w}) := \frac{1}{2}\|\boldsymbol{w}\|^2. \tag{7}$$

For any $\boldsymbol{w} \in \mathbb{R}^d$, we have $\nabla f(\boldsymbol{w}) = \boldsymbol{w}$ and $\nabla^2 f(\boldsymbol{w}) = \boldsymbol{I}_d$.

a) Justify that $f \in \mathcal{C}_1^{1,1}(\mathbb{R}^d)$ and that f is 1-strongly convex.

b) Adapt the general convergence rates of gradient descent and accelerated gradient on strongly convex problems to this particular case. What do the results suggest about the problem?

c) Write down the first iteration of gradient descent and accelerated gradient for the problem at hand using a stepsize of 1: is the result in agreement with the convergence rates?

Exercise 3.2: Convergence rates

Let $f : \mathbb{R}^d \to \mathbb{R}$. Suppose that $f \in \mathcal{C}_L^{1,1}(\mathbb{R}^d)$ and that f is μ-strongly convex. Suppose that we apply gradient descent with a constant stepsize $\alpha_k = \frac{1}{\mu+L}$ to this problem.

a) Using the descent property stated in class, justify that one has

$$f(\boldsymbol{w}_{k+1}) \leq f(\boldsymbol{w}_k) - \frac{2\mu + L}{2(\mu + L)^2} \|\nabla f(\boldsymbol{w}_k)\|_2^2 \tag{8}$$

holds at every iteration of gradient descent.

b) Using the previous question, show that gradient descent achieves a linear convergence rate with this stepsize. Are additional assumptions on μ and/or L needed?

c) Gradient descent with stepsize $\frac{1}{L}$ achieves a linear rate of convergence in $\left(1 - \frac{\mu}{L}\right)^K$ on this problem. Compare this rate with that obtained in question b).

d) Accelerated gradient (with the appropriate parameterization for strongly convex functions) achieves a linear rate of convergence in $\left(1 - \sqrt{\frac{\mu}{L}}\right)^K$ on this problem. Compare this rate with that obtained in question b).

Exercise 3.3: Saddle points

Consider the two-dimensional function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$f(\boldsymbol{w}) := w_1^2 \left(w_1^2 - 2\right) + w_2^4.$$

a) For every $\boldsymbol{w} \in \mathbb{R}^2$, we have $\nabla f(\boldsymbol{w}) = \begin{bmatrix} 4w_1^3 - 4w_1 \\ 4w_2^3 \end{bmatrix}$. Find all first-order stationary points of f.

b) The formula for the Hessian matrix gives

$$\forall \boldsymbol{w} \in \mathbb{R}^d, \nabla^2 f(\boldsymbol{w}) = \begin{bmatrix} 12w_1^2 - 4 & 0 \\ 0 & 12w_2^2 \end{bmatrix}.$$

Evaluate the Hessian at the points found in the previous question. What can you conclude about the nature of those points?

c) Without using derivatives, show that the critical points are either global minima or strict saddle points.

Exercise 3.4: Gradient descent with line search

Consider the minimization of a nonconvex, $\mathcal{C}^1$ function $f : \mathbb{R}^d \to \mathbb{R}$, assuming that f is bounded below by $f_{\text{low}} \in \mathbb{R}$. Suppose that we use gradient descent with a stepsize chosen through backtracking line search. More precisely, given $\theta \in (0, 1)$ and $c \in (0, 1/2)$, we select the stepsize at iteration k as the largest value in $\{\theta^j\}_{j \in \mathbb{N}}$ such that

$$f(\boldsymbol{w}_k - \theta^j \nabla f(\boldsymbol{w}_k)) < f(\boldsymbol{w}_k) - c\theta^j \|\nabla f(\boldsymbol{w}_k)\|_2^2. \tag{9}$$

a) Suppose that $\|\nabla f(\boldsymbol{w}_k)\|_2 > 0$ at iteration k. Show then that the line search terminates within a fixed number of iterations (independent of $\boldsymbol{w}_k$) and give a bound on the value of the stepsize α_k.

b) Show that the convergence rate of gradient descent in this setting is $\mathcal{O}(\frac{1}{\sqrt{K}})$, as in the fixed-stepsize case seen in class. Give the precise values for the constants.

c) What is the maximum number of backtracking steps at each iteration? How does this number illustrate the additional cost of line-search techniques?

Exercise 3.5: Hölder continuity (Adapted from 2021-2022 exam)

In this exercise, we consider a general, unconstrained optimization problem of the form

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{w}), \tag{10}$$

where $f \in \mathcal{C}^1$. Rather than assuming that the gradient is Lipschitz continuous (as in the lectures), we will assume a more general property called *Hölder continuity*. More precisely, for any $\nu \in (0, 1]$ and $L > 0$, we write $f \in \mathcal{C}_L^{1,\nu}$ if $f \in \mathcal{C}^1$ and

$$\forall(\boldsymbol{v}, \boldsymbol{w}) \in (\mathbb{R}^d)^2, \qquad \|\nabla f(\boldsymbol{v}) - \nabla f(\boldsymbol{w})\| \leq L\|\boldsymbol{v} - \boldsymbol{w}\|^\nu. \tag{11}$$

We say that ∇f is (L, ν)-Hölder continuous.

For any $f \in \mathcal{C}_L^{1,\nu}$, we can show that

$$\forall(\boldsymbol{v}, \boldsymbol{w}) \in (\mathbb{R}^d)^2, \qquad f(\boldsymbol{v}) \leq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^T(\boldsymbol{v} - \boldsymbol{w}) + \frac{L}{1+\nu}\|\boldsymbol{v} - \boldsymbol{w}\|_2^2. \tag{12}$$

Moreover, since $f \in \mathcal{C}^1$, we can apply gradient descent to problem (10): the kth iteration of this method is

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k). \tag{13}$$

Part 1 We first study the case $\nu = 1$ (i.e. $f \in \mathcal{C}_L^{1,1}$), corresponding to the gradient being L-Lipschitz continuous.

a) Recalling that $f \in \mathcal{C}_L^{1,1}$ for this question, justify the choice $\alpha_k = \frac{1}{L}$ for every k.

b) Give the convergence rate result for gradient descent on problem (10), assuming that the function f is nonconvex. What quantity does this rate apply to?

c) Give the convergence rate result for gradient descent on problem (10), assuming that the function f is convex. What quantity does this rate apply to?

d) Assuming that f is convex, name a method with a better convergence rate than gradient descent, and give the associated rate.

Part 2 We now consider the more general setting $\nu \in (0, 1]$ (i.e. $f \in \mathcal{C}_L^{1,\nu}$) and f nonconvex.

a) Based on (12) and your answer to Part 1, justify the choice

$$\alpha_k = \left[\frac{\|\nabla f(w_k)\|^{1-\nu}}{L}\right]^{\frac{1}{\nu}} \tag{14}$$

for the stepsize used in the iteration (13).

b) Using the sequence $\{\alpha_k\}_k$ from the previous question, it is possible to show a convergence rate in $\mathcal{O}\left(\frac{1}{K^{\frac{\nu}{1+\nu}}}\right)$ for gradient descent applied to $f \in \mathcal{C}_L^{1,\nu}$ nonconvex and bounded below. Compare this bound with the results seen in class for the case $\nu = 1$.

c) In addition to $f \in \mathcal{C}_L^{1,\nu}$, suppose that f is twice continuously differentiable.

   i) Without any convexity assumption on f, are we guaranteed to converge towards a local minimum?

   ii) What guarantee seen in class can we provide about the limit points of gradient descent?

4. Nonsmooth and regularized optimization

Exercise 4.1: Robust linear regression (Adapted from 2024-2025 exam)

We consider a linear regression problem of the form

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}} \;\; f(\boldsymbol{w}) := \frac{1}{n}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_1 = \frac{1}{n}\sum_{i=1}^{n} |\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i|, \tag{15}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1 \;\cdots\; \boldsymbol{x}_n]^{\mathrm{T}} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{y} \in \mathbb{R}^n$ form the problem data.

a) Justify that the objective function of (15) is convex.

b) Let $\boldsymbol{w}^*$ and $\bar{\boldsymbol{w}}$ be two solutions of (15). Justify that the vector $\frac{\boldsymbol{w}^*+\bar{\boldsymbol{w}}}{2}$ is also a solution of the problem.

c) Due to the use of the ℓ_1 norm, the function f is not necessarily differentiable.

   i) Give a condition on the data $(\boldsymbol{X}, \boldsymbol{y})$ under which f is actually continuously differentiable.

   ii) Assuming that the condition from the previous question does not hold, give a point in $\mathbb{R}^d$ at which the function is *not* differentiable.

d) Given any real $t \in \mathbb{R}$, we let $\operatorname{sgn}(t) = 1$ if $t > 0$, $\operatorname{sgn}(t) = 0$ if $t = 0$ and $\operatorname{sgn}(t) = -1$ if $t < 0$.

   i) For any $\boldsymbol{w} \in \mathbb{R}^d$ and $i \in \{1, \ldots, n\}$, show that the vector $\operatorname{sgn}(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i)\boldsymbol{x}_i$ is a subgradient for the function $\boldsymbol{w} \mapsto \left|\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i\right|$ at $\boldsymbol{w}$.

   ii) Consequently, justify that the vector

$$\boldsymbol{g}(\boldsymbol{w}) = \frac{1}{n}\boldsymbol{X}^{\mathrm{T}}\operatorname{sgn}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}), \quad \text{where} \quad \operatorname{sgn}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) = \begin{bmatrix} \operatorname{sgn}(\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{w} - y_1) \\ \vdots \\ \operatorname{sgn}(\boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{w} - y_n) \end{bmatrix} \tag{16}$$

   is a subgradient for f at $\boldsymbol{w}$.

iii) Under the assumptions of question c)i), let $\boldsymbol{w}_0$ be the point considered in that question. Give another example of a subgradient for f at $\boldsymbol{w}_0$ other than $\boldsymbol{g}(\boldsymbol{w}_0)$, where $\boldsymbol{g}(\boldsymbol{w}_0)$ is defined by (16).

e) We now consider solving (15) using a subgradient method.

i) Write down the iteration of a subgradient method using a constant stepsize and the vector $\boldsymbol{g}(\boldsymbol{x})$ defined by (16).

ii) Do you expect the function values corresponding to the method's iterates to be monotonically decreasing? Justify your answer.

iii) What kind of guarantees seen in class can we provide about that method for a sufficiently small stepsize?

Exercise 4.2: Proximal operator (Adapted from 2022-2023 exam)

In this exercise, we revisit the proximal operator and the proximal gradient method on a specific problem. Given a vector $\boldsymbol{w} \in \mathbb{R}^d$, we consider

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}} \ \|\boldsymbol{w}\|_1 + \frac{1}{2\alpha}\|\boldsymbol{w}-\boldsymbol{z}\|_2^2, \tag{17}$$

where $\|\boldsymbol{w}\|_1 = \sum_{i=1}^d |[\boldsymbol{w}]_i|$, $\|\boldsymbol{w}\|_2^2 = \sum_{i=1}^d [\boldsymbol{w}]_i^2$ and $\alpha > 0$.

a) Explain why the objective function of (17) cannot be optimized by gradient-type techniques.

b) Justify that problem (17) and

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}} \ \alpha\|\boldsymbol{w}\|_1 + \frac{1}{2}\|\boldsymbol{w}-\boldsymbol{z}\|_2^2 \tag{18}$$

have the same solution set (argmin). Since both functions are strongly convex, what can be said about this solution set?

c) Write down an optimality condition for problem (17).

d) In this question, we view problem (17) as computing a proximal operator.

i) Using the definition of the proximal operator, write down the solution of problem (17) as the value of a proximal operator of a certain function.

ii) By repeatedly solving instances of problem (17) using the last solution found as $\boldsymbol{z}$, what algorithm do we obtain?

e) In this question, we change perspective and view problem (18) in a composite form, where $\frac{1}{2}\|\boldsymbol{w}-\boldsymbol{z}\|_2^2$ is a data-fitting term and $\alpha\|\boldsymbol{w}\|_1$ is a regularization term.

i) What is the purpose of such a regularization term? Why is it computationally worth using?

ii) Write down an iteration of proximal gradient applied to this problem with $\boldsymbol{w}_k = \boldsymbol{z}$ and stepsize α. What do you observe then? Is it to be expected?

Exercise 4.3: Regularization and sparsity (Adapted from 2024-2025 exam)

In this exercise, we consider an optimization problem with elastic net regularization of the form

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{w}) + \lambda\left(\|\boldsymbol{w}\|_1 + \frac{\mu}{2}\|\boldsymbol{w}\|_2^2\right), \tag{19}$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm, $\lambda > 0$ and $\mu > 0$.

a) Recall the purpose of the regularization terms $\|\cdot\|_1$ and $\|\cdot\|_2^2$ in optimization.

b) In this section, we consider the function $\Omega : \boldsymbol{w} \mapsto \lambda\|\boldsymbol{w}\|_1 + \frac{\lambda\mu}{2}\|\boldsymbol{w}\|_2^2$.

   i) Recall the definition of the proximal operator associated with Ω, denoted by $\mathrm{prox}_\Omega(\cdot)$. Justify that this particular "prox" can be considered as a function from $\mathbb{R}^d$ to $\mathbb{R}^d$.

   ii) Using properties of the proximal operator, one obtains

$$\forall \boldsymbol{w} \in \mathbb{R}^d, \qquad \mathrm{prox}_\Omega(\boldsymbol{w}) = \frac{1}{\lambda\mu + 1}\mathrm{prox}_{\lambda\|\cdot\|_1}(\boldsymbol{w}). \tag{20}$$

   where $\mathrm{prox}_{\lambda\|\cdot\|_1}(\cdot)$ corresponds to the proximal operator for (a multiple of) the ℓ_1 norm. Justify then that $\mathrm{prox}_\Omega(\boldsymbol{w})$ can be computed in closed form.

c) Suppose for this question that f is a continuously differentiable function.

   i) Write down an iteration of the proximal gradient algorithm applied to problem (17) using the prox_Ω operator.

   ii) Using the answer from the previous question, justify that the use of elastic net regularization is likely to produce sparse iterates that decay in norm for large values of the regularization parameter λ.

5. Stochastic gradient methods

Exercise 5.1: Recap on stochastic gradient

We consider a finite-sum optimization problem:

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{w}), \tag{21}$$

where each function f_i depends on a single item of a dataset consisting in n elements. We suppose that every function f_i is continuously differentiable, and we define a family of algorithms methods based on a starting point $\boldsymbol{w}_0 \in \mathbb{R}^d$ as well as the recursion

$$\forall k \geq 0, \quad \boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha \boldsymbol{g}_k, \tag{22}$$

where $\alpha > 0$ is a given stepsize and $\boldsymbol{g}_k$ is an estimate of the gradient $\nabla f(\boldsymbol{w}_k)$.

a) How should $\boldsymbol{g}_k$ be chosen in order for the recursion (22) to correspond to an instance of:

i) gradient descent?

ii) stochastic gradient?

b) Recall the definition of an epoch: what is the equivalent of this unit in terms of:

i) iterations of gradient descent?

ii) iterations of stochastic gradient?

c) We now focus on using of batch variants of stochastic gradient. Given a batch size $n_b \in \{1, \dots, n\}$, we draw a batch index set $\mathcal{S}_k \subseteq \{1, \dots, n\}$ based on the following distribution :

$$\forall \mathcal{S} \subseteq \{1, \dots, n\}, \quad \mathbb{P}(\mathcal{S}_k = \mathcal{S}) := \begin{cases} \frac{1}{\binom{n}{n_b}} = \frac{n_b!(n-n_b)!}{n!} & \text{if } |\mathcal{S}| = n_b \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

We then set $\boldsymbol{g}_k = \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} \nabla f_i(\boldsymbol{w}_k)$ in the recursion (22).

i) Show that $\mathbb{E}_{\mathcal{S}_k}[g_k] = \nabla f(\boldsymbol{w}_k)$.

ii) Suppose that the function f is strongly convex; in that case, and under the appropriate assumptions on the problem, one can show that

$$\lim_{k \to \infty} \mathbb{E}[f(\boldsymbol{w}_k) - f^*] \in \left[0, c\alpha \frac{M^2}{n_b}\right], \tag{24}$$

where $c > 0$ is a problem-dependent constant and $M^2 > 0$ is a bound on $\|\nabla f_i(\boldsymbol{w})\|$ for any $i \in \{1, \dots, n\}$ and $\boldsymbol{w} \in \mathbb{R}^d$. What property of (batch) stochastic gradient methods does this result illustrate?

d) Describe two modifications of the algorithm (among those covered in the lectures) that can lead to a stronger guarantee than (24).

e) *Practical situation :* Suppose that we want to compare stochastic gradient for several values of the batch size. While running on a given problem, we observe that $n_b = 1$ gives better convergence than $n_b = n$, while increasing the batch size from $n_b = 1$ to $n_b = n/10$ consistently improves the results, in that the method converges faster and to a smaller value of f. We then observe that the convergence slows down as we increase the batch size from $n/10$ to n. How can you explain these observations?

Exercise 5.2 : Importance sampling

We consider the finite-sum problem

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimize}} \, f(\boldsymbol{w}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{w}), \tag{25}$$

where for every $i = 1, \dots, n$, the function f_i is $\mathcal{C}^1$ and its gradient is L_i-Lipschitz continuous. We also assume that the function f is μ-strongly convex. In what follows, let $c_i = \frac{nL_i}{\sum_{j=1}^{n} L_j}$.

We consider a variant of stochastic gradient where we suppose that the random indices i_k are drawn according to an *importance sampling distribution* defined by

$$\forall i \in \{1, \dots, n\}, \qquad \mathbb{P}(i_k = i) \; = \; \frac{c_i}{\sum_{j=1}^n c_j}. \tag{26}$$

In addition, we suppose that the update is replaced by

$$\boldsymbol{w}_{k+1} \leftarrow \boldsymbol{w}_k - \frac{\alpha_k}{c_{i_k}} \nabla f_{i_k}(\boldsymbol{w}_k).$$

a) What can be the interest of such a strategy?

b) Show that $\mathbb{E}_{i_k}\left[\frac{1}{c_{i_k}} \nabla f_{i_k}(\boldsymbol{w}_k)\right] = \nabla f(\boldsymbol{w}_k)$.

c) It can be shown that ∇f is L-Lipschitz continuous with $L = \frac{1}{n}\sum_{i=1}^n L_i$. Suppose that we select a constant stepsize $\alpha_k = \frac{1}{L} \forall k$. Given a sampled index i_k, we wish to compare the classical stochastic gradient iteration to iteration (26).

   i) Show that $\frac{\alpha_k}{c_{i_k}} = \frac{1}{L_{i_k}}$.

   ii) When can we get $\frac{\alpha_k}{c_{i_k}} \geq \alpha_k$? What does this imply on the iteration (26) ?

d) Suppose that we know a Lipschitz constant L on ∇f with $L \geq \max_{1 \leq i \leq n} L_i$: with constant stepsize $\alpha \leq \frac{1}{L}$, we know that convergence guarantees can be obtained. How can the knowledge of the $\{L_i\}$s allow for a larger stepsize?

Exercise 5.3: Randomized Kaczmarz (Adapted from 2024-2025 exam)

Consider the linear regression problem

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimize}} \; f(\boldsymbol{w}) = \frac{1}{2n}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 = \frac{1}{2n}\sum_{i=1}^n (\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i)^2, \tag{27}$$

where $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are unit vectors in $\mathbb{R}^d$ ($\|\boldsymbol{x}_i\|_2 = 1$ for every $i \in \{1, \dots, n\}$) and $y_i \in \mathbb{R}$.
We suppose that there exists a vector $\boldsymbol{w}^* \in \mathbb{R}^d$ to problem (27) such that $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}^* = y_i$.

a) Justify that $\boldsymbol{w}^*$ is both a minimum of f and of every function f_i with $f_i(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i)^2$. What is the optimal value of problem (27)?

b) We first consider consider a general batch stochastic gradient scheme applied to problem (27) of the form

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \boldsymbol{g}_k, \tag{28}$$

where $\boldsymbol{g}_k$ is a stochastic gradient estimate computed from a batch of indices in $\{1, \dots, n\}$.

   i) Recall the formula for $\boldsymbol{g}_k$ introduced in class. How should $\boldsymbol{g}_k$ be chosen for iteration (28) to correspond to vanilla stochastic gradient? To gradient descent?

   ii) Suppose that our main computational cost is accessing the vectors $\{\boldsymbol{x}_i\}$. With that metric, what is the cost of the iteration (28)?

iii) How does that cost compare to that of a gradient descent iteration?

c) Suppose that the $\boldsymbol{g}_k$s are computed using a batch of n_b indices at every iteration. Under appropriate assumptions on the stochastic indices and the stepsize, one can obtain an expected convergence rate for iteration (28) of the form

$$\forall K \geq 1, \quad \mathbb{E}[f(\boldsymbol{w}_K)] - \min_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w}) \leq c_1 t^K + \frac{c_2}{n_b}, \tag{29}$$

where c_1 and c_2 are positive constants, and $t \in (0,1)$ that we assume are independent of n_b.

i) Given this result, what behavior are we likely to observe for an average run of the method?

ii) For a fixed budget of accesses to vectors $\{a_i\}$, what rate does (29) imply for batch stochastic gradient? Is that rate better than that of gradient descent?

d) In this question, we consider the *randomized Kaczmarz* method applied to problem (27). The algorithm is specific to linear regression problems, and is defined through the following iteration:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - (\boldsymbol{x}_{i_k}^{\mathrm{T}}\boldsymbol{w}_k - y_{i_k})\boldsymbol{x}_{i_k}, \tag{30}$$

where i_k is an index drawn at random in $\{1,\ldots,n\}$.

i) What is the connection between this method and that described by iteration (27)?

ii) Suppose that the indices $\{i_k\}$ are independent. Show that

$$\mathbb{E}_{i_k}[\boldsymbol{w}_{k+1} - \boldsymbol{w}^*] = \mathbb{E}_{i_k}[\boldsymbol{I}_d - \boldsymbol{x}_{i_k}\boldsymbol{x}_{i_k}^{\mathrm{T}}](\boldsymbol{w}_k - \boldsymbol{w}^*),$$

where $\boldsymbol{I}_d$ is the identity matrix in $\mathbb{R}^{d\times d}$.

iii) Under the assumptions of the previous question, suppose further that the data vectors $\{a_i\}$ satisfy $\mathbb{E}_{i_k}[\boldsymbol{I}_d - \boldsymbol{x}_{i_k}\boldsymbol{x}_{i_k}^{\mathrm{T}}] = \frac{1}{2}\boldsymbol{I}_d$. Show then that $\boldsymbol{w}_k - \boldsymbol{w}^*$ converges in expectation towards 0.

iv) The result from the previous question is (much) stronger than the convergence rate (29). How can you explain this discrepancy?

e) For this last question, we suppose that the vectors $\boldsymbol{x}_i$ are nonzero but not necessarily of norm 1. In that case, the randomized Kaczmarz method chooses an index i_k so that

$$\forall i \in \{1,\ldots,n\}, \qquad \mathbb{P}(i_k = i) = \frac{\|\boldsymbol{x}_i\|_2^2}{\sum_{\ell=1}^n \|\boldsymbol{x}_\ell\|_2^2}, \tag{31}$$

and the iteration (30) is modified as follows

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \frac{\boldsymbol{x}_{i_k}^T\boldsymbol{w}_k - y_{i_k}}{\|\boldsymbol{x}_{i_k}\|_2^2}\boldsymbol{x}_{i_k}.$$

i) Under this distribution (31), show that

$$\mathbb{E}_{i_k}\left[\frac{\boldsymbol{x}_{i_k}^{\mathrm{T}}\boldsymbol{w}_k - y_{i_k}}{\|\boldsymbol{x}_{i_k}\|_2^2}\boldsymbol{x}_{i_k}\right] = \gamma\nabla f(\boldsymbol{w}_k) = \frac{\gamma}{n}\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}_k - \boldsymbol{y})$$

for some positive constant γ that does not depend on the index k (give the value of that constant).

ii) Going back to the general method (28), it may be that stochastic gradient steps vary significantly in magnitude across all coordinates. What advanced variant of stochastic gradient would you recommend to tackle this issue? Justify your answer.

Solutions

Solutions for Part 1(Basics of optimization)

Solutions for Exercise 1.1: Support vector machines

We use basic axioms to show that the function with $\lambda = 0$ is convex.

For any $i = 1, \dots, n$, the functions $\boldsymbol{w} \mapsto 0$ and $\boldsymbol{w} \mapsto 1 - y_i \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w}$ are linear in $\boldsymbol{w}$, hence they are convex. The maximum of two convex functions is convex, thus $\boldsymbol{w} \mapsto \max\{1 - y_i \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w}, 0\}$ is a convex function. The sum of convex functions is convex, and multiplying by $\frac{1}{n} > 0$ preserves convexity. Thus, the objective function with $\lambda = 0$ is convex.

We finally use the fact that $\boldsymbol{w} \mapsto \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2$ is λ-strongly convex to conclude that the objective is λ-strongly convex.

Solutions for Exercise 1.2(Least-squares problems)

a) The objective function of problem (2) is a composition of the linear function $w \mapsto w\boldsymbol{x} - \boldsymbol{y}$ with the convex function $\frac{1}{2}\|\cdot\|_2^2$, hence it is a convex function. For its optimal value to be 0, there must exist a real value w^* such that

$$\frac{1}{2}\|w^*\boldsymbol{x} - \boldsymbol{y}\|_2^2 \;\Leftrightarrow\; w^*\boldsymbol{x} = \boldsymbol{y},$$

which implies that either $\boldsymbol{y} = \boldsymbol{0}$ or $\boldsymbol{x}$ and $\boldsymbol{y}$ are colinear. Therefore, the optimal value is not necessarily 0.

b) By writing $\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{w}_d^{\mathrm{T}} \end{bmatrix}$, we have

$$\frac{1}{2}\|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{y}\|_2^2 = \frac{1}{2}\sum_{i=1}^{d}(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x} - y_i).$$

By rewriting the variables as $\boldsymbol{v} = \begin{bmatrix} \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_d \end{bmatrix} \in \mathbb{R}^{d^2}$, it follows that

$$\frac{1}{2}\|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{y}\|_2^2 = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{v} - \boldsymbol{y}\|_2^2,$$

where

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}^{\mathrm{T}} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{x}^{\mathrm{T}} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \cdots & \boldsymbol{0} & \boldsymbol{x}^{\mathrm{T}} \end{bmatrix}$$

By the same argument as in question a), it follows that the objective function is convex.

c) Consider the matrix

$$\boldsymbol{W}^* = \frac{\boldsymbol{y}\boldsymbol{x}^{\mathrm{T}}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}.$$

We have

$$\boldsymbol{W}^*\boldsymbol{x} = \frac{1}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}\boldsymbol{y}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}\boldsymbol{y} = \boldsymbol{y}.$$

It follows that

$$\min_{\boldsymbol{W}\in\mathbb{R}^{d\times d}} \frac{1}{2}\|\boldsymbol{W}\boldsymbol{x} - \boldsymbol{y}\|_2^2 = 0.$$

To show that the problem does not have a unique solution, it suffices to note that for any $\boldsymbol{u} \in \mathbb{R}^d$ such that $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x} = 0$ (e.g. $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}\boldsymbol{e}_1 - \boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}\boldsymbol{x}$), we have

$$(\boldsymbol{W}^* + \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}})\boldsymbol{x} = \boldsymbol{W}^*\boldsymbol{x} = \boldsymbol{y},$$

hence $\boldsymbol{W}^* + \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}$ is also a global minimum, showing that they are infinitely many global minima.

Solutions for Exercise 1.3(Strong convexity)

a) The function $\boldsymbol{z} \mapsto f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{w})$ is linear, hence convex. The function $\boldsymbol{z} \mapsto \frac{\mu}{2}\|\boldsymbol{z} - \boldsymbol{w}\|_2^2$ is μ-strongly convex, hence $\varphi_{\boldsymbol{w}}$ is μ-strongly convex. The function is $\mathcal{C}^1$ because it is quadratic in $\boldsymbol{z}$.

b) Since the function is strongly convex, the minimum is the only vector $\boldsymbol{z}^*$ such that

$$\nabla\varphi_{\boldsymbol{w}}(\boldsymbol{z}^*) = \boldsymbol{0} \;\Leftrightarrow\; \boldsymbol{z}^* = \boldsymbol{w} - \frac{1}{\mu}\nabla f(\boldsymbol{w}).$$

It follows that

$$\min_{\boldsymbol{z}} \varphi_{\boldsymbol{w}}(\boldsymbol{z}) = \varphi_{\boldsymbol{w}}(\boldsymbol{z}^*) = f(\boldsymbol{w}) - \frac{1}{2\mu}\|\nabla f(\boldsymbol{w})\|^2.$$

c) By strong convexity of f, we have

$$f(\boldsymbol{z}) \geq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{w}) + \frac{\mu}{2}\|\boldsymbol{z} - \boldsymbol{w}\|_2^2 = \varphi_{\boldsymbol{w}}(\boldsymbol{z}). \tag{32}$$

By minimizing both sides of the inequality (32) with respect to $\boldsymbol{z}$, we get

$$f^* = \min_{\boldsymbol{z}} f(\boldsymbol{z}) \geq \varphi_{\boldsymbol{w}}(\boldsymbol{z}^*) = f(\boldsymbol{w}) - \frac{1}{2\mu}\|\nabla f(\boldsymbol{w})\|^2.$$

Re-arranging the terms gives the desired result.

Solutions for Exercise 1.4(Co-coercivity)

a) By recalling the descent lemma, we have for every $\boldsymbol{w} \in \mathbb{R}^d$ that

$$f(\boldsymbol{w} - \frac{1}{L}\nabla f(\boldsymbol{w})) \leq f(\boldsymbol{w}) - \frac{1}{2L}\|\nabla f(\boldsymbol{w})\|_2^2.$$

Rearranging the terms give

$$\frac{1}{2L}\|\nabla f(\boldsymbol{w})\|_2^2 \leq f(\boldsymbol{w}) - f(\boldsymbol{w} - \frac{1}{L}\nabla f(\boldsymbol{w})) \leq f(\boldsymbol{w}) - f(\boldsymbol{w}^*),$$

proving the desired result.

b) Consider the function $\phi_{\boldsymbol{v}} : \boldsymbol{z} \mapsto f(\boldsymbol{z}) - \nabla f(\boldsymbol{v})^{\mathrm{T}}\boldsymbol{z}$. This function is convex, as a sum of the convex function f and a linear function. Moreover, $\phi_{\boldsymbol{v}} \in \mathcal{C}_L^{1,1}$. It follows that any minimum $\boldsymbol{z}_{\boldsymbol{v}}$ of $\phi_{\boldsymbol{v}}$ satisfies

$$\nabla\phi_{\boldsymbol{v}}(\boldsymbol{z}_{\boldsymbol{v}}) = \nabla f(\boldsymbol{z}_{\boldsymbol{v}}) - \nabla f(\boldsymbol{v}) = \mathbf{0}.$$

Since this equality is satisfied for $\boldsymbol{z}_{\boldsymbol{v}} = \boldsymbol{v}$, we conclude that $\boldsymbol{v} \in \operatorname{argmin}_{\boldsymbol{z}} \phi_{\boldsymbol{v}}(\boldsymbol{z})$ with

$$\min_{\boldsymbol{z}} \phi_{\boldsymbol{v}}(\boldsymbol{z}) = \phi_{\boldsymbol{v}}(\boldsymbol{v}) = f(\boldsymbol{v}) - \nabla f(\boldsymbol{v})^{\mathrm{T}}\boldsymbol{v}.$$

Similarly, we have $\boldsymbol{w} \in \operatorname{argmin}_{\boldsymbol{z}} \phi_{\boldsymbol{w}}(\boldsymbol{z})$.

We now apply question a) to the convex functions $\phi_{\boldsymbol{v}}$ and $\phi_{\boldsymbol{w}}$ using $\boldsymbol{v}$ and $\boldsymbol{w}$ in different roles. We thus obtain

$$\begin{aligned} \phi_{\boldsymbol{v}}(\boldsymbol{w}) - \phi_{\boldsymbol{v}}(\boldsymbol{v}) &\geq \frac{1}{2L}\|\nabla\phi_{\boldsymbol{v}}(\boldsymbol{w})\|_2^2 \\ \iff f(\boldsymbol{w}) - \nabla f(\boldsymbol{v})^{\mathrm{T}}\boldsymbol{w} - f(\boldsymbol{v}) + \nabla f(\boldsymbol{v})^{\mathrm{T}}\boldsymbol{v} &\geq \frac{1}{2L}\|\nabla f(\boldsymbol{v}) - \nabla f(\boldsymbol{w})\|_2^2. \end{aligned} \tag{33}$$

Similarly, we have

$$f(\boldsymbol{v}) - \nabla f(\boldsymbol{w})^{\mathrm{T}}\boldsymbol{v} - f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}}\boldsymbol{w} \geq \frac{1}{2L}\|\nabla f(\boldsymbol{v}) - \nabla f(\boldsymbol{w})\|_2^2. \tag{34}$$

Summing (33) and (34) gives the desired inequality.

Solutions for Part 2(Derivatives and differentiation)

Solutions for Exercise 2.1(A simple function)

a) We first note that there are several ways to decompose the calculation of $f(w)$. We choose here the following:

$$\begin{aligned} v_1 &= w \\ v_2 &= \sin(v_1) \\ v_3 &= \sqrt{v_2} \\ v_4 &= v_2 + v_3 \\ f &= v_4. \end{aligned}$$

Note: The vertices v_1 and v_4 can be directly replaced with w and f in some variants of AD, so these can be omitted.

b) In forward mode, we evaluate the derivative starting from the vertices corresponding to the

smallest indices.

$$\begin{aligned}
\frac{\partial v_1}{\partial w} &= 1 \\
\frac{\partial v_2}{\partial w} &= \frac{\partial v_2}{\partial v_1}\frac{\partial v_1}{\partial w} = \cos(v_1) \times 1 = \cos(v_1) \\
\frac{\partial v_3}{\partial w} &= \frac{\partial v_3}{\partial v_2}\frac{\partial v_2}{\partial w} = \frac{1}{2\sqrt{v_2}}\cos(v_1) \\
\frac{\partial v_4}{\partial w} &= \frac{\partial v_4}{\partial v_2}\frac{\partial v_2}{\partial w} + \frac{\partial v_4}{\partial v_3}\frac{\partial v_3}{\partial w} = 1 \times \cos(v_1) + \frac{1}{2\sqrt{v_2}}\cos(v_1) \\
\frac{\partial f}{\partial w} &= \frac{\partial f}{\partial v_4}\frac{\partial v_4}{\partial w} = \left(1 + \frac{1}{2\sqrt{v_2}}\right)\cos(v_1).
\end{aligned}$$

By setting $w = \frac{\pi}{2}$, a pass for evaluating $f(w)$ gives $v_1 = \frac{\pi}{2}$, $v_2 = v_3 = 1$, $v_4 = f = 2$, and thus

$$f'(w) = \frac{\partial f}{\partial w} = \left(1 + \frac{1}{2}\right)\cos\left(\tfrac{\pi}{2}\right) = 0.$$

c) In backward mode, we start by the vertices of the graph corresponding to the largest indices. This gives:

$$\begin{aligned}
\frac{\partial f}{\partial v_4} &= 1 \\
\frac{\partial f}{\partial v_3} &= \frac{\partial f}{\partial v_4}\frac{\partial v_4}{\partial v_3} = 1 \times 1 \\
\frac{\partial f}{\partial v_2} &= \frac{\partial f}{\partial v_4}\frac{\partial v_4}{\partial v_2} + \frac{\partial f}{\partial v_3}\frac{\partial v_3}{\partial v_2} = 1 \times 1 + 1 \times \frac{1}{2\sqrt{v_2}} \\
\frac{\partial f}{\partial v_1} &= \frac{\partial f}{\partial v_2}\frac{\partial v_2}{\partial v_1} = \left(1 + \frac{1}{2\sqrt{v_2}}\right)\cos(v_1) \\
\frac{\partial f}{\partial w} &= \frac{\partial f}{\partial v_1}\frac{\partial v_1}{\partial w} = \left(1 + \frac{1}{2\sqrt{v_2}}\right)\cos(v_1) \times 1.
\end{aligned}$$

We recover the fomula from question b), and thus the same value for $f'(\frac{\pi}{2})$.

Solutions for Exercise 2.2(Sigmoid-type functions)

a) Unlike the previous exercise, we will not consider adding an initial vertex to the graph. We thus decompose the calculation as follows:

$$\begin{aligned}
v_1 &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{w} \\
v_2 &= \sigma(v_1) \\
v_3 &= \ln(v_2) \\
v_4 &= y\, v_3 \\
f &= v_4.
\end{aligned}$$

b) The input $\boldsymbol{w}$ is a vector, and thus the derivative $\frac{\partial f}{\partial \boldsymbol{w}}$ is a row vector in $\mathbb{R}^{1\times d}$ (its transpose vector being $\nabla f(\boldsymbol{w})$). Any derivative $\frac{\partial v_i}{\partial \boldsymbol{w}}$ has the same dimension. The chain rule in forward mode gives

$$\begin{aligned}
\frac{\partial v_1}{\partial \boldsymbol{w}} &= \boldsymbol{x}^{\mathrm{T}} \in \mathbb{R}^{1\times d} \\
\frac{\partial v_2}{\partial \boldsymbol{w}} &= \frac{\partial v_2}{\partial v_1}\frac{\partial v_1}{\partial \boldsymbol{w}} = \sigma(v_1)(1-\sigma(v_1))\,\boldsymbol{x}^{\mathrm{T}} \\
\frac{\partial v_3}{\partial \boldsymbol{w}} &= \frac{\partial v_3}{\partial v_2}\frac{\partial v_2}{\partial \boldsymbol{w}} = \frac{1}{v_2}\sigma(v_1)(1-\sigma(v_1))\,\boldsymbol{x}^{\mathrm{T}} \\
\frac{\partial v_4}{\partial \boldsymbol{w}} &= \frac{\partial v_4}{\partial v_3}\frac{\partial v_3}{\partial \boldsymbol{w}} = y\frac{1}{v_2}\sigma(v_1)(1-\sigma(v_1))\,\boldsymbol{x}^{\mathrm{T}} \\
\frac{\partial f}{\partial \boldsymbol{w}} &= \frac{\partial f}{\partial v_4}\frac{\partial v_4}{\partial \boldsymbol{w}} = y\frac{1}{v_2}\sigma(v_1)(1-\sigma(v_1))\,\boldsymbol{x}^{\mathrm{T}}.
\end{aligned}$$

c) The backward-mode differentiation gives

$$\begin{aligned}
\frac{\partial f}{\partial v_4} &= 1 \\
\frac{\partial f}{\partial v_3} &= \frac{\partial f}{\partial v_4}\frac{\partial v_4}{\partial v_3} = 1\times y \\
\frac{\partial f}{\partial v_2} &= \frac{\partial f}{\partial v_3}\frac{\partial v_3}{\partial v_2} = y\times\frac{1}{v_2} \\
\frac{\partial f}{\partial v_1} &= \frac{\partial f}{\partial v_2}\frac{\partial v_2}{\partial v_1} = \frac{y}{v_2}\sigma(v_1)(1-\sigma(v_1)) \\
\frac{\partial f}{\partial \boldsymbol{w}} &= \frac{\partial f}{\partial v_1}\frac{\partial v_1}{\partial \boldsymbol{w}} = \frac{y}{v_2}\sigma(v_1)(1-\sigma(v_1))\,\boldsymbol{x}^{\mathrm{T}}.
\end{aligned}$$

d) We add the following vertices to those from question a)

$$\begin{aligned}
v_5 &= 1-v_2 \\
v_6 &= \ln(v_5) \\
v_7 &= (1-y)v_6 \\
h &= v_4+v_7.
\end{aligned}$$

NB: One could also consider adding a vertex g and define h as $f+g$.

e) Building on the derivative formulas from question c), we compute

$$\begin{aligned}
\frac{\partial h}{\partial v_7} &= 1 \\
\frac{\partial h}{\partial v_6} &= \frac{\partial h}{\partial v_7}\frac{\partial v_7}{\partial v_6} = 1 \times (1-y) \\
\frac{\partial h}{\partial v_5} &= \frac{\partial h}{\partial v_6}\frac{\partial v_6}{\partial v_5} = (1-y)\frac{1}{v_5} \\
\frac{\partial h}{\partial v_4} &= 1 \\
\frac{\partial h}{\partial v_3} &= \frac{\partial h}{\partial v_4}\frac{\partial v_4}{\partial v_3} = 1 \times y \\
\frac{\partial h}{\partial v_2} &= \frac{\partial h}{\partial v_5}\frac{\partial v_5}{\partial v_2} + \frac{\partial h}{\partial v_3}\frac{\partial v_3}{\partial v_2} = (1-y)\frac{1}{v_5} \times (-1) + y \times \frac{1}{v_2} \\
\frac{\partial h}{\partial v_1} &= \frac{\partial h}{\partial v_2}\frac{\partial v_2}{\partial v_1} = \left(\frac{y-1}{v_5} + \frac{y}{v_2}\right)\sigma(v_1)(1-\sigma(v_1)) \\
\frac{\partial h}{\partial \boldsymbol{w}} &= \frac{\partial h}{\partial v_1}\frac{\partial v_1}{\partial \boldsymbol{w}} = \left(\frac{y-1}{v_5} + \frac{y}{v_2}\right)\sigma(v_1)(1-\sigma(v_1))\boldsymbol{x}^{\mathrm{T}} \in \mathbb{R}^{1\times d}.
\end{aligned}$$

Solutions for Exercise 2.3(Two-layer neural networks)

The exercise does not specify a precise way of computing the derivatives, which is intentional. We present here a solution involving backward-mode differentiation.

a) We decompose the computation of f as follows:

$$\begin{aligned}
\boldsymbol{v}_1 &= \boldsymbol{W}_1\boldsymbol{x} \in \mathbb{R}^h \\
\boldsymbol{v}_2 &= \boldsymbol{\sigma}(\boldsymbol{v}_1) \in \mathbb{R}^h \\
\boldsymbol{v}_3 &= \boldsymbol{W}_2\boldsymbol{v}_2 \in \mathbb{R}^{d_y} \\
\boldsymbol{v}_4 &= \boldsymbol{v}_3 - \boldsymbol{y} \in \mathbb{R}^{d_y} \\
f &= \frac{1}{2}\|\boldsymbol{v}_4\|_2^2 \in \mathbb{R}.
\end{aligned}$$

By backward-mode differentiation, it follows that

$$\begin{aligned}
\frac{\partial f}{\partial \boldsymbol{v}_4} &= \boldsymbol{v}_4^{\mathrm{T}} \in \mathbb{R}^{1\times d_y} \\
\frac{\partial f}{\partial \boldsymbol{v}_3} &= \frac{\partial f}{\partial \boldsymbol{v}_4}\underbrace{\frac{\partial \boldsymbol{v}_4}{\partial \boldsymbol{v}_3}}_{\in\mathbb{R}^{d_y\times d_y}} = \boldsymbol{v}_4^{\mathrm{T}}\boldsymbol{I}_{d_y} \in \mathbb{R}^{1\times d_y} \\
\frac{\partial f}{\partial \boldsymbol{v}_2} &= \frac{\partial f}{\partial \boldsymbol{v}_3}\underbrace{\frac{\partial \boldsymbol{v}_3}{\partial \boldsymbol{v}_2}}_{\in\mathbb{R}^{d_y\times h}} = \boldsymbol{v}_4^{\mathrm{T}}\boldsymbol{W}_2 \in \mathbb{R}^{1\times h} \\
\frac{\partial f}{\partial \boldsymbol{v}_1} &= \frac{\partial f}{\partial \boldsymbol{v}_2}\underbrace{\frac{\partial \boldsymbol{v}_2}{\partial \boldsymbol{v}_1}}_{\in\mathbb{R}^{h\times h}} = \boldsymbol{v}_4^{\mathrm{T}}\boldsymbol{W}_2\boldsymbol{\Sigma} \in \mathbb{R}^{1\times h},
\end{aligned}$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{h \times h}$ is a diagonal matrix with diagonal entries equal to $1 + \tanh([\boldsymbol{v}_1]_i)^2$. The final backward step gives

$$\frac{\partial f}{\partial \boldsymbol{x}} = \frac{\partial f}{\partial \boldsymbol{v}_1} \underbrace{\frac{\partial \boldsymbol{v}_1}{\partial \boldsymbol{x}}}_{\in \mathbb{R}^{h \times d_x}} = \boldsymbol{v}_4^{\mathrm{T}} \boldsymbol{W}_2 \boldsymbol{\Sigma} \boldsymbol{W}_1 \in \mathbb{R}^{1 \times d_x}.$$

And thus

$$\nabla_{\boldsymbol{x}} f = \frac{\partial f}{\partial \boldsymbol{x}}^{\mathrm{T}} = \boldsymbol{W}_1 \boldsymbol{\Sigma} \boldsymbol{W}_2 \boldsymbol{v}_4 = \boldsymbol{W}_1 \mathrm{diag}\left(1 + \tanh([\boldsymbol{W}_1 \boldsymbol{x}]_i)^2\right) \boldsymbol{W}_2 \left(\boldsymbol{W}_2 \boldsymbol{\sigma}(\boldsymbol{W}_1 \boldsymbol{x}) - \boldsymbol{y}\right).$$

b) To compute the derivative with respect to $\boldsymbol{W}_1$, we simply consider a different final backward step in the reasoning of question a):

$$\underbrace{\frac{\partial f}{\partial \boldsymbol{W}_1}}_{\in \mathbb{R}^{1 \times (h \times d_x)}} = \frac{\partial f}{\partial \boldsymbol{v}_1} \underbrace{\frac{\partial \boldsymbol{v}_1}{\partial \boldsymbol{W}_1}}_{\in \mathbb{R}^{h \times (h \times d_x)}}.$$

Note: Computing the last derivative requires to use tensor structures. However, by considering only the product of $\frac{\partial \boldsymbol{v}_1}{\partial \boldsymbol{W}_1}$ with $\frac{\partial f}{\partial \boldsymbol{W}_1}$, one can show that this product reduces to

$$\frac{\partial f}{\partial \boldsymbol{W}_1} = \boldsymbol{\Sigma} \boldsymbol{W}_2^{\mathrm{T}} \boldsymbol{v}_4 \boldsymbol{x}^{\mathrm{T}} \in \mathbb{R}^{h \times d_x} \equiv \mathbb{R}^{1 \times (h \times d_x)}.$$

which can be written without tensor notations.

Solutions for Part 3(Gradient descent)

Solutions for Exercise 3.1: A strongly convex problem

i) The function f is quadratic in each variable, therefore it is (in particular) $\mathcal{C}^1$ and $\mathcal{C}^2$. Moreover, for any $(\boldsymbol{v}, \boldsymbol{w}) \in (\mathbb{R}^d)^2$, we have

$$\|\nabla f(\boldsymbol{v}) - \nabla f(\boldsymbol{w})\| = \|\boldsymbol{v} - \boldsymbol{w}\| \leq \|\boldsymbol{v} - \boldsymbol{w}\|,$$

therefore the gradient is 1-Lipschitz continuous. Finally, we have $\nabla^2 f(\boldsymbol{w}) = \boldsymbol{I}_d \geq 1 \times \boldsymbol{I}_d$, implying that f is 1-strongly convex.

ii) For a $\mathcal{C}_L^{1,1}$, μ-strongly convex function, the convergence rate of gradient descent is $\mathcal{O}((1 - \frac{\mu}{L})^K)$, in that for any $K \geq 1$,

$$f(\boldsymbol{w}_K) - \min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) \leq \left(1 - \frac{\mu}{L}\right)^K \left(f(\boldsymbol{w}_0) - \min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})\right).$$

For accelerated gradient, the rate is $\mathcal{O}((1 - \sqrt{\frac{\mu}{L}})^K)$. On our particular problem, we have $\mu = L = 1$: this leads to $f(\boldsymbol{w}_K) - \min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) = 0$ for $K \geq 1$, suggesting that gradient descent would converge in 1 iteration. The same observation can be made for accelerated gradient.

iii) We begin by pointing out that the first iterations of gradient descent and accelerated gradient are identical, and consist in taking a gradient step. With a stepsize of $\frac{1}{L}$, the first iteration is

$$\boldsymbol{w}_1 = \boldsymbol{w}_0 - \frac{1}{L}\nabla f(\boldsymbol{w}_0).$$

In our case, we have $L = 1$ and $\nabla f(\boldsymbol{w}_0) = \boldsymbol{w}0$, hence

$$\boldsymbol{w}_1 = \boldsymbol{w}_0 - \boldsymbol{w}_0 = \mathbf{0}_{\mathbb{R}^d}.$$

As a result, regardless of the choice $\boldsymbol{w}_0$, both gradient descent and accelerated gradient reach the minimum of the problem in one iteration : this confirms the result of the previous question.

Solutions for Exercise 3.2: Convergence rates

a) Using the Taylor expansion of f, we have

$$\begin{aligned}
f\left(\boldsymbol{w}_k - \frac{1}{\mu+L}\nabla f(\boldsymbol{w}_k)\right) &\leq f(\boldsymbol{w}_k) - \frac{1}{\mu+L}\nabla f(\boldsymbol{w}_k)^{\mathrm{T}}\nabla f(\boldsymbol{w}_k) + \frac{L}{2(\mu+L)^2}\|\nabla f(\boldsymbol{w}_k)\|^2 \\
f\left(\boldsymbol{w}_k - \frac{1}{\mu+L}\nabla f(\boldsymbol{w}_k)\right) &\leq f(\boldsymbol{w}_k) + \left(\frac{L}{2(\mu+L)^2} - \frac{1}{\mu+L}\right)\|\nabla f(\boldsymbol{w}_k)\|^2 \\
f\left(\boldsymbol{w}_k - \frac{1}{\mu+L}\nabla f(\boldsymbol{w}_k)\right) &\leq f(\boldsymbol{w}_k) - \frac{2\mu+L}{2(\mu+L)^2}\|\nabla f(\boldsymbol{w}_k)\|^2. \qquad (35)
\end{aligned}$$

As $-\frac{2\mu+L}{2(\mu+L)^2} < 0$, the above inequality guarantees a decrease in the function value as long as $\|\nabla f(\boldsymbol{w}_k)\| > 0$.

b) For any $k \geq 1$, using the result of Exercise 1.3 gives $\|\nabla f(\boldsymbol{w}_{k-1})\| \geq 2\mu(f(\boldsymbol{w}_{k-1}) - f^*)$ with $f^* = \min_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w})$. Injecting this inequality into that of question a) (applied at iteration $k-1$) gives

$$\begin{aligned}
f(\boldsymbol{w}_k) &\leq f(\boldsymbol{w}_{k-1}) - \frac{2\mu+L}{2(\mu+L)^2}\|\nabla f(\boldsymbol{w}_{k-1})\|^2 \\
f(\boldsymbol{w}_k) &\leq f(\boldsymbol{w}_{k-1}) - \frac{\mu(2\mu+L)}{(\mu+L)^2}\left(f(\boldsymbol{w}_{k-1}) - f^*\right) \\
f(\boldsymbol{w}_k) - f^* &\leq \left(1 - \frac{\mu(2\mu+L)}{(\mu+L)^2}\right)\left(f(\boldsymbol{w}_{k-1}) - f^*\right).
\end{aligned}$$

A recursive argument then yields

$$f(\boldsymbol{w}_k) - f^* \leq \left(1 - \frac{\mu(2\mu+L)}{(\mu+L)^2}\right)^k (f(\boldsymbol{w}_0) - f^*),$$

which also holds for $k = 0$. As a result, we have established a convergence rate in $\mathcal{O}\left(\left(1 - \frac{\mu(2\mu+L)}{(\mu+L)^2}\right)^k\right)$ for this variant of gradient descent. Note that $\frac{\mu(2\mu+L)}{(\mu+L)^2} < 1$ since

$$(\mu+L)^2 - \mu(2\mu+L) = L^2 - \mu^2 + L\mu > 0,$$

thus no additional assumption on L and μ is required.

c) The rate seen in class (for a constant stepsize $\frac{1}{L}$) is $\mathcal{O}\left(\left(1-\frac{\mu}{L}\right)^k\right)$. Comparing the two ratios, we find that

$$\frac{\mu(2\mu+L)}{(\mu+L)^2} < \frac{\mu}{L},$$

hence the rate is slower for the approach proposed in this exercise, regardless of the value of μ and L. *Note : This can be explained by the fact that the stepsize $\frac{1}{\mu+L}$ is smaller than $\frac{1}{L}$, and can lead to a smaller decrease in the objective.*

d) The rate established in the previous questions is provably worse than that of gradient descent with stepsize $\frac{1}{L}$: since accelerated gradient achieves the optimal rate in $\mathcal{O}\left(\left(1-\sqrt{\frac{\mu}{L}}\right)^k\right)$, provably better than that of gradient descent, the proposed method cannot be better than accelerated gradient in terms of convergence rates.

Solutions for Exercise 3.3: Saddle points

a) The set of first-order stationary points correspond to the points with zero gradient. Solving $\nabla f(\boldsymbol{w}) = \mathbf{0}$ gives to

$$\begin{cases} 4w_1^3 - 4w_1 &= 0 \\ 4w_2^3 &= 0. \end{cases}$$

The second equation immediately gives $w_2 = 0$. The first equation has three solutions, namely $w_1 \in \{0, -1, 1\}$. As a result, the set of first-order critical points is

$$\left\{\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}-1\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}\right\}.$$

b) If the Hessian matrix of a first-order critical point is positive definite (resp. negative definite), we know that the point is a local minimum (resp. a local maximum). If it is positive semi-definite, the point is also a second-order critical point, but it may be a local minimum, a saddle point or a local maximum (the latter case being possible only when the Hessian matrix is the zero vector).

For the three critical points studied here, we find that

$$\begin{aligned} \nabla^2 f\left(\begin{bmatrix}0\\0\end{bmatrix}\right) &= \begin{bmatrix}-4 & 0\\ 0 & 0\end{bmatrix} \\ \nabla^2 f\left(\begin{bmatrix}-1\\0\end{bmatrix}\right) &= \begin{bmatrix}8 & 0\\ 0 & 0\end{bmatrix} \succeq \mathbf{0} \\ \nabla^2 f\left(\begin{bmatrix}1\\0\end{bmatrix}\right) &= \begin{bmatrix}8 & 0\\ 0 & 0\end{bmatrix} \succeq \mathbf{0}. \end{aligned}$$

c) For any $\boldsymbol{v} \in \mathbb{R}^2$ such that $|v_1| \in (0,1)$ and $v_2 = 0$, we have :

$$f(\boldsymbol{v}) = v_1^2(v_1^2 - 1) + v_2^4 = v_1^2(v_1^2 - 1) < 0.$$

As a result, for any $\epsilon \in (-1, 1)$, the point $[\epsilon\ 0]^{\mathrm{T}}$ gives a better function value than $\mathbf{0}$. This implies that $\mathbf{0}$ cannot be a local minimum. A similar reasoning shows that $f([0\ v_2]^{\mathrm{T}}) > 0$ for any $v_2 > 0$, so $\mathbf{0}$ cannot be a local maximum either. We thus conclude that $\mathbf{0}$ is a saddle point of the function.

Solutions for Exercise 3.4

Solutions for Exercise 3.5: Gradient descent with line search

a) Suppose that $\|\nabla f(\boldsymbol{w}_k)\| > 0$. By performing backtracking line search, we seek the largest stepsize α_k such that

$$f(\boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)) < f(\boldsymbol{w}_k) \tag{36}$$

among the sequence $\{1, \frac{1}{2}, \frac{1}{4}, \dots\}$.

We know per the descent lemma that any choice $\alpha < \frac{2}{L}$ satisfies the condition (36). As a result, for any value $\alpha = 2^{-j}$ with $j \in \mathbb{N}$ that does not satisfy (36), we must have

$$2^{-j} \geq \frac{2}{L} \quad \Leftrightarrow \quad j \leq \log_2\left(\frac{L}{2}\right).$$

Since we are testing every value of the form 2^{-j}, there will exist a value for which the inequality will be violated, implying that (36) will eventually be satisfied. More precisely, after no more than

$$\bar{j} = \left\lceil \log_2\left(\frac{L}{2}\right) \right\rceil$$

iterations of backtracking, the stepsize $\alpha = 2^{-\bar{j}}$ will satisfy $\alpha < \frac{2}{L}$ with $\alpha > \frac{1}{L}$ (otherwise the line-search would have terminated at the previous step of backtracking).

Overall, we have shown that the line-search terminates in a finite number of backtracking iterations (at most $\left\lceil \log_2\left(\frac{L}{2}\right)\right\rceil$) and outputs a stepsize α_k with $\alpha_k \geq \frac{1}{L}$.

b) To adapt the proof in the case of a constant stepsize, it suffices to observe that our bound $\alpha_k \geq \frac{1}{L}$ from the previous question leads to

$$f(\boldsymbol{w}_{k+1}) \leq f(\boldsymbol{w}_k) + \left(-\alpha_k + \frac{L}{2}\alpha_k^2\right)\|\nabla f(\boldsymbol{w}_k)\|^2 \leq f(\boldsymbol{w}_k) - \frac{1}{2L}\|\nabla f(\boldsymbol{w}_k)\|^2,$$

since $\alpha \mapsto -\alpha + \frac{L}{2}\alpha^2$ is increasing on $\left[\frac{1}{L}, 1\right]$. The rest of the proof follows by applying the same reasoning. Namely, after $K \geq 1$ iterations, we have

$$\min_{0 \leq k \leq K-1} \|\nabla f(\boldsymbol{w}_k)\| \leq \frac{2L(f(\boldsymbol{w}_0) - f_{\text{low}})}{\sqrt{K}}.$$

c) A complexity result is obtained by defining $\epsilon > 0$ and bounding the number of iterations required to produce a point with gradient norm smaller than ϵ. Suppose that this has not happened by iteration $K \geq 1$: in that case, we have

$$\epsilon \leq \min_{0 \leq k \leq K-1} \|\nabla f(\boldsymbol{w}_k)\| \leq \frac{2L(f(\boldsymbol{w}_0) - f_{\text{low}})}{\sqrt{K}},$$

where the last inequality comes from the previous question. This inequality is only valid as long as

$$K \leq \frac{2L(f(\boldsymbol{w}_0) - f_{\text{low}})}{\epsilon^2}.$$

Therefore, the number of iterations needed to produce a point with gradient norm smaller than ϵ is at most [1]

$$\left\lceil \frac{2L(f(\boldsymbol{w}_0) - f_{\text{low}})}{\epsilon^2} \right\rceil.$$

We thus say that the complexity of gradient descent on nonconvex problems is $\mathcal{O}(\epsilon^{-2})$.

d) In the worst-case, every iteration requires $\left\lceil \log_2 \left(\frac{L}{2}\right) \right\rceil$ backtracking iterations/function evaluations, while there can be at most

$$\left\lceil \frac{2L(f(\boldsymbol{w}_0) - f_{\text{low}})}{\epsilon^2} \right\rceil$$

iterations. Overall, the total number of function evaluations is bounded above by

$$\left\lceil \log_2 \left(\frac{L}{2}\right) \right\rceil \times \left\lceil \frac{2L(f(\boldsymbol{w}_0) - f_{\text{low}})}{\epsilon^2} \right\rceil.$$

The cost of the line-search approach in terms of number of function evaluations is thus increased compared to that of the constant stepsize technique. *Note : The line-search approach does not require knowledge of L.*

Solutions for Exercise 3.6(Hölder continuity)

a) By choosing $\alpha = \frac{1}{L}$, the gradient iteration (13) becomes

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \frac{1}{L}\nabla f(\boldsymbol{w}_k).$$

Using this relation in (12) with $\nu = 1$, we obtain:

$$\begin{aligned} f(\boldsymbol{w}_{k+1}) &\leq f(\boldsymbol{w}_k) + \nabla f(\boldsymbol{w}_k)^T(\boldsymbol{w}_{k+1} - \boldsymbol{w}_k) + \frac{L}{2}\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_k\|^2 \\ &= f(\boldsymbol{w}_k) - \frac{1}{L}\|\nabla f(\boldsymbol{w}_k)\|^2 + \frac{1}{2L}\|\nabla f(\boldsymbol{w}_k)\|^2 \\ &= f(\boldsymbol{w}_k) - \frac{1}{2L}\|\nabla f(\boldsymbol{w}_k)\|^2. \end{aligned}$$

The choice $\alpha_k = \frac{1}{L}$ for every k thus guarantees a decrease in the function value as long as $\|\nabla f(\boldsymbol{w}_k)\| \neq 0$. *Note: The value $\frac{1}{L}$ actually minimizes the right-hand side of the first inequality.*

b) The convergence rate of gradient descent for $\mathcal{C}^{1,1}$ functions is

i) $\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$ if f is nonconvex, in the sense that for any $K \geq 1$, we have

$$\min_{0 \leq k \leq K-1} \|\nabla f(\boldsymbol{w}_k)\| \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right).$$

ii) $\mathcal{O}(\frac{1}{K})$ if f is convex, in the sense that for any $K \geq 1$, we have

$$f(\boldsymbol{w}_k) - \min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) \leq \mathcal{O}\left(\frac{1}{K}\right).$$

[1] If we reach this number of iterations, then necessarily $\|\nabla f(\boldsymbol{w}_K)\| < \epsilon$.

c) Provided f is convex, the accelerated gradient method (also called Nesterov's method) possesses a better convergence rate than gradient descent on convex problems: this rate is of order $\mathcal{O}\left(\frac{1}{K^2}\right)$.

d) We apply (12) for the prescribed choice of α_k:

$$\begin{aligned} f(\boldsymbol{w}_{k+1}) &\leq f(\boldsymbol{w}_k) + \nabla f(\boldsymbol{w}_k)^T(\boldsymbol{w}_{k+1} - \boldsymbol{w}_k) + \frac{L}{1+\nu}\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_k\|^{1+\nu} \\ &= f(\boldsymbol{w}_k) - \left[\frac{\|\nabla f(\boldsymbol{w}_k)\|^{1-\nu}}{L}\right]^{\frac{1}{\nu}} \|\nabla f(\boldsymbol{w}_k)\|^2 + \frac{L}{1+\nu}\left[\frac{\|\nabla f(\boldsymbol{w}_k)\|^{1-\nu}}{L}\right]^{\frac{1+\nu}{\nu}} \|\nabla f(\boldsymbol{w}_k)\|^{1+\nu} \\ &= f(\boldsymbol{w}_k) - \frac{\|\nabla f(\boldsymbol{w}_k)\|^{\frac{1+\nu}{\nu}}}{L^{\frac{1}{\nu}}} + \frac{1}{1+\nu}\frac{\|\nabla f(\boldsymbol{w}_k)\|^{\frac{1+\nu}{\nu}}}{L^{\frac{1}{\nu}}} \\ &= f(\boldsymbol{w}_k) - \frac{\nu}{1+\nu}\frac{\|\nabla f(\boldsymbol{w}_k)\|^{\frac{1+\nu}{\nu}}}{L^{1/\nu}}. \end{aligned}$$

As a result, the chosen value for α_k yields a decrease in the function value as long as $\|\nabla f(\boldsymbol{w}_k)\| \neq 0$: this justifies to use such a value as a stepsize. *Note: As for the case $\nu = 1$, the chosen value actually minimizes the right-hand side of the first inequality.*

e) For $f \in \mathcal{C}_L^{1,1}$ nonconvex and bounded below, the convergence rate of gradient descent is $\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$. Replacing $\mathcal{C}_L^{1,1}$ by $\mathcal{C}_L^{1,\nu}$, we obtain that

$$\min_{0\leq k\leq K-1} \|\nabla f(\boldsymbol{w}_k)\|_2^{\frac{1+\nu}{\nu}} \leq \mathcal{O}\left(\frac{1}{K}\right),$$

thus $\min_{0\leq k\leq K-1}\|\nabla f(\boldsymbol{w}_k)\|_2$ decreases at a rate $\mathcal{O}\left(\frac{1}{K^{\frac{\nu}{1+\nu}}}\right)$. This rate is slower than the rate for $\mathcal{C}^{1,1}$ functions.

f) *This question assumes $f \in \mathcal{C}_L^{1,\nu}$ and $f \in \mathcal{C}^2$.*

i) The method will converge to a first-order stationary point, i.e. a point with zero gradient. Without any convexity assumptions on f, this point could be a local minimum, a saddle point, or a local maximum.

ii) Since the objective is twice continuously differentiable, we know that for almost any initial point $\boldsymbol{w}_0 \in \mathbb{R}^d$, gradient descent will reach a second-order stationary point $\boldsymbol{w}^* \in \mathbb{R}^d$ such that $\nabla f(\boldsymbol{w}^*) = 0$ and $\nabla^2 f(\boldsymbol{w}^*) \succeq \mathbf{0}$.

Solutions for Part 4(Nonsmooth and regularized optimization)

Solutions for Exercise 4.1

a) A norm is a convex function, hence $\|\cdot\|_1$ is convex. The composition of a convex function with a linear function is convex, hence $\|\boldsymbol{X}\cdot - \boldsymbol{y}\|_1$ is convex. Multiplying by the positive constant $\frac{1}{n}$ preserves convexity.

b) Let $f^* = f(\boldsymbol{w}^*) = f(\bar{\boldsymbol{w}})$. By convexity of f, we have

$$f\left(\frac{\boldsymbol{w}^* + \bar{\boldsymbol{w}}}{2}\right) \quad \leq \quad \frac{1}{2}f(\boldsymbol{w}^*) + \frac{1}{2}f(\bar{\boldsymbol{w}}) = f^*,$$

from which we conclude that $f\left(\frac{\boldsymbol{w}^*+\bar{\boldsymbol{w}}}{2}\right) = f^*$.

c) i) When $\boldsymbol{X}$ is the zero matrix in $\mathbb{R}^{n\times d}$, the function f is constant, hence continuously differentiable.

ii) Let $\boldsymbol{w}_0 = \boldsymbol{w}^*$, since $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}^* - y_i = 0$, the function $\boldsymbol{w} \mapsto |\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i|$ is not differentiable at $\boldsymbol{w}_0$.

d) i) By inspection, we note that for any $\boldsymbol{z} \in \mathbb{R}^d$,

$$|\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{z} - y_i| \geq |\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i| + \operatorname{sgn}(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i)\boldsymbol{x}_i^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{w}). \tag{37}$$

Indeed, if $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i = 0$, the inequality (37) clearly holds. If $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i > 0$,

$$|\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i| + \operatorname{sgn}(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i)\boldsymbol{x}_i^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{w}) = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i + \boldsymbol{x}_i^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{w}) = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{z} - y_i < |\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{z} - y_i|,$$

hence (37) holds. In the remaining case $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i < 0$, we have

$$|\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i| + \operatorname{sgn}(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i)\boldsymbol{x}_i^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{w}) = -\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{z} + y_i - \boldsymbol{x}_i^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{w}) = -\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{z} + y_i < |\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{z} - y_i|,$$

proving (37).

ii) It suffices to do a weighted combination of the subgradient inequality (37) and to notice that

$$\frac{1}{n}\boldsymbol{X}^{\mathrm{T}} \operatorname{sgn}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n} \operatorname{sgn}(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i)\boldsymbol{x}_i.$$

iii) Since $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_0 - y_i = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}^* - y_i = 0$, any subgradient $\boldsymbol{g} \in \mathbb{R}^d$ for $\boldsymbol{w} \mapsto |\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i|$ satisfies

$$|\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{z} - y_i| \geq |\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_0 - y_i| + \boldsymbol{g}^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{w}_0) = \boldsymbol{g}^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{w}^*) \quad \forall \boldsymbol{z} \in \mathbb{R}^d.$$

Using $\boldsymbol{g} = \boldsymbol{x}_i$ gives $\boldsymbol{x}_i^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{w}^*) = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{z} - \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}^* = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{z} - y_i$, so the inequality is satisfied. As a result, the vector $\frac{1}{n}\boldsymbol{X}^{\mathrm{T}}1_n$ is a subgradient for f at $\boldsymbol{w}_0$.

e) i) $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha \boldsymbol{g}(\boldsymbol{x}_k)$, where $\alpha > 0$.

ii) In general, a subgradient is not a descent direction, hence there is no guarantee that a subgradient method will decrease function values monotonically. On the other hand, the subgradient chosen here is a minimum-norm subgradient, and thus decrease will be guaranteed for a sufficiently small stepsize.

iii) For a sufficiently small stepsize, the function gap $f(\bar{\boldsymbol{w}}_K) - f^*$, where $\bar{\boldsymbol{w}}_K = \frac{1}{K}\sum_{k=0}^{K-1} \boldsymbol{w}_k$, decreases like $\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$.

Solutions for Exercise 4.2

a) The function $\boldsymbol{w} \mapsto \|\boldsymbol{w}\|_1$ is nonsmooth in the sense that its gradient is not defined at any point that has a zero coefficient. This property precludes from applying gradient-type techniques to optimize this function and, consequently, to the objective function of problem (17).

b) Multiplying the objective function by a positive constant does not modify the set of solutions of an optimization problem. Since $\alpha > 0$, this justifies that

$$\operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^d} \left\{ \alpha\|\boldsymbol{w}\|_1 + \frac{1}{2}\|\boldsymbol{w}-\boldsymbol{z}\|_2^2 \right\} = \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^d} \left\{ \|\boldsymbol{w}\|_1 + \frac{1}{2\alpha}\|\boldsymbol{w}-\boldsymbol{z}\|_2^2 \right\}.$$

Both functions are continuous and strongly convex, therefore this set contains a single element.

c) An optimality condition for problem (17) can be formulated using the subdifferential of the objective. More precisely, the unique solution of the problem, call it $\boldsymbol{w}^*$, is the unique vector such that

$$0_{\mathbb{R}^d} \in \partial\left(\|\cdot\|_1 + \frac{1}{2\alpha}\|\cdot - \boldsymbol{z}\|_2^2 \right)(\boldsymbol{w}^*).$$

Extra: Since only the ℓ_1 norm is not necessarily differentiable at $\boldsymbol{w}^*$ while $\nabla\left(\frac{1}{2\alpha}\|\cdot-\boldsymbol{z}\|_2^2\right)(\boldsymbol{w}^*) = \frac{1}{\alpha}(\boldsymbol{w}^*-\boldsymbol{z})$, this condition is equivalent to

$$\frac{1}{\alpha}(\boldsymbol{z}-\boldsymbol{w}^*) \in \partial\left(\|\cdot\|_1\right)(\boldsymbol{w}^*).$$

d) i) By definition, the proximal operator of a function h is

$$\operatorname{prox}_h(\boldsymbol{w}) = \operatorname*{argmin}_{\boldsymbol{u}\in\mathbb{R}^d} \left\{ h(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{w}\|_2^2 \right\}.$$

The objective of problem (18) has the structure appearing in the definition of the proximal operator. As a result, we can write the solution (which is unique by Question 2) as

$$\boldsymbol{w}^* = \operatorname{prox}_{\alpha\|\cdot\|_1}(\boldsymbol{w}).$$

ii) Repeating the iteration

$$\boldsymbol{w} \leftarrow \operatorname{prox}_{\alpha\|\cdot\|_1}(\boldsymbol{w})$$

corresponds to performing iterations of the proximal point algorithm.

e) In this question, we view problem (17) in a composite form, where $\frac{1}{2\alpha}\|\boldsymbol{w}-\boldsymbol{z}\|_2^2$ is a data-fitting term and $|\boldsymbol{w}\|_1$ is a regularization term.

i) The purpose of an ℓ_1 regularization term is to promote sparsity of the solution. Its computational interest is that it is a convex, continuous relaxation term (unlike the ℓ_0 norm which is both nonconvex and discontinuous) for which the proximal subproblems can be solved explicitly.

ii) The proximal gradient iteration for problem 17 can be written as

$$\boldsymbol{w}_{k+1} = \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^d} \left\{ \frac{1}{2}\|\boldsymbol{w}_k - \boldsymbol{z}\|_2^2 + \nabla\|\cdot\|_2^2(\boldsymbol{w}_k)^{\mathrm{T}}(\boldsymbol{w} - \boldsymbol{w}_k) + \frac{1}{2\alpha}\|\boldsymbol{w} - \boldsymbol{w}_k\|_2^2 + \|\boldsymbol{w}\|_1 \right\}.$$

Since $\boldsymbol{w}_k = \boldsymbol{z}$, we have

$$\nabla\|\cdot\|_2^2(\boldsymbol{w}_k) = \tfrac{1}{\alpha}(\boldsymbol{w}_k - \boldsymbol{z}) = 0_{\mathbb{R}^d},$$

and thus the first two terms vanish, resulting in the expression

$$\boldsymbol{w}_{k+1} = \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^d} \left\{ \frac{1}{2\alpha}\|\boldsymbol{w} - \boldsymbol{z}\|_2^2 + \|\boldsymbol{w}\|_1 \right\}.$$

This iteration of proximal gradient is therefore equivalent to solving the original problem. This could have been expected because the original problem is a quadratic problem with an ℓ_1 regularization term, and thus the proximal subproblems have the same form.

Solutions for Exercise 4.3

a) The ℓ_1 norm is used to enforce sparsity of the solution, while the ℓ_2 regularization reduces the sensitivity to small changes in the data defining f. *Other answers matching what was discussed in the course can also be accepted.*

b) i) For any $\boldsymbol{w} \in \mathbb{R}^d$, we recall that $\operatorname{prox}_{\Omega}(\boldsymbol{w}) = \operatorname{argmin}_{\boldsymbol{u}\in\mathbb{R}^d} \Omega(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{w}\|^2$. In general, this is a set-valued operator. However, the function Ω is convex, and thus the optimization problem defining the proximal operator is strongly convex. As a result, it possesses a unique solution, and one can define that solution as the output of the proximal operator, making it a function from $\mathbb{R}^d$ to $\mathbb{R}^d$.

ii) As seen in class, the proximal operator of (a multiple of) the ℓ_1 norm can be computed in close form through the soft-thresholding operator. It follows that the proximal operator for elastic net also satisfies this property.

c) i) An iteration of proximal gradient reads

$$\boldsymbol{w}_{k+1} = \operatorname{prox}_{\alpha_k\Omega}\left(\boldsymbol{w}_k - \alpha_k\nabla f(\boldsymbol{w}_k)\right),$$

where $\alpha_k > 0$.

ii) The formula from the previous question gives

$$\boldsymbol{w}_{k+1} = \frac{1}{\alpha_k\lambda\mu + 1}\operatorname{prox}_{\alpha_k\lambda\|\cdot\|_1}(\boldsymbol{w}_k - \alpha_k\nabla f(\boldsymbol{w}_k)).$$

Thus, for every coordinate $j \in \{1, \ldots, d\}$, it follows that

$$[\boldsymbol{w}_{k+1}]_j = \begin{cases} \frac{1}{\alpha_k\lambda\mu+1}\left([\boldsymbol{w}_k - \alpha_k\nabla f(\boldsymbol{w}_k)]_j - \alpha_k\lambda\right) & \text{if } [\boldsymbol{w}_k - \alpha_k\nabla f(\boldsymbol{w}_k)]_j > \alpha_k\lambda \\ \frac{1}{\alpha_k\lambda\mu+1}\left([\boldsymbol{w}_k - \alpha_k\nabla f(\boldsymbol{w}_k)]_j + \alpha_k\lambda\right) & \text{if } [\boldsymbol{w}_k - \alpha_k\nabla f(\boldsymbol{w}_k)]_j < -\alpha_k\lambda \\ 0 & \text{otherwise.} \end{cases}$$

One then sees that large values of λ will tend to zero more components of the gradient step, and therefore of the next proximal gradient iterate. In addition, the remaining coordinates are decreased by a factor $\frac{1}{\alpha_k\lambda\mu+1}$, suggesting (especially for a fixed stepsize) that all coordinates will decrease (smoothly), and thus the ℓ_2 norm will also decrease. *NB: The answer did not need to be as detailed as this, but a justification for both properties should be provided.*

Solutions for Part 5(Stochastic gradient)

Solutions of Exercise 5.1: Recap on stochastic gradient

a) In $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha \boldsymbol{g}_k$, the vector $\boldsymbol{g}_k$ should be set as

i) $\nabla f(\boldsymbol{w}_k)$ for the method to be an instance of gradient descent;

ii) $\nabla f_{i_k}(\boldsymbol{w}_k)$, with i_k drawn at random in $\{1, \dots, n\}$, to be an instance of stochastic gradient.

b) An epoch is a unit of cost equivalent to n accesses to data points. As a result, this unit corresponds to

i) 1 iteration of gradient descent, in which all n data points need to be accessed in order to compute the gradient;

ii) n iterations of stochastic gradient, since every iteration of that form only requires access to 1 data point. Equivalently, an iteration of stochastic gradient corresponds to $\frac{1}{n}$ epoch.

c) (Batch variant.)

i) By definition of the expected value, one has

$$\begin{aligned}
\mathbb{E}_{\mathcal{S}_k}[\boldsymbol{g}_k] &= \sum_{\mathcal{S} \subseteq \{1,\dots,n\}} \mathbb{P}(\mathcal{S}_k = \mathcal{S}) \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \nabla_i f(\boldsymbol{w}_k) \\
&= \sum_{\substack{\mathcal{S} \subseteq \{1,\dots,n\} \\ |\mathcal{S}| = n_b}} \frac{1}{\binom{n}{n_b}} \frac{1}{n_b} \sum_{i \in \mathcal{S}} \nabla_i f(\boldsymbol{w}_k) \\
&= \frac{1}{\binom{n}{n_b}} \frac{1}{n_b} \sum_{\substack{\mathcal{S} \subseteq \{1,\dots,n\} \\ |\mathcal{S}| = n_b}} \sum_{i \in \mathcal{S}} \nabla_i f(\boldsymbol{w}_k)
\end{aligned}$$

The random set $\mathcal{S}_k$ takes $\binom{n}{n_b}$ possible values. If we consider any index $i \in \{1, \dots, n\}$, this index appears in exactly $\binom{n-1}{n_b-1}$ index sets of cardinality n_b out of the possible $\binom{n}{n_b}$. Therefore,

$$\sum_{\substack{\mathcal{S} \subseteq \{1,\dots,n\} \\ |\mathcal{S}| = n_b}} \sum_{i \in \mathcal{S}} \nabla_i f(\boldsymbol{w}_k) = \binom{n-1}{n_b-1} \sum_{i=1}^{n} \nabla f_i(\boldsymbol{w}_k).$$

Using $\binom{n}{n_b} = \frac{n}{n_b}\binom{n-1}{n_b-1}$, we thus obtain

$$\begin{aligned}
\mathbb{E}_{\mathcal{S}_k}[\boldsymbol{g}_k] &= \frac{1}{\binom{n}{n_b}} \frac{1}{n_b} \binom{n-1}{n_b-1} \sum_{i=1}^{n} \nabla f_i(\boldsymbol{w}_k) \\
&= \frac{1}{\binom{n}{n_b}} \frac{1}{n_b} \frac{n_b}{n} \binom{n}{n_b} \sum_{i=1}^{n} \nabla f_i(\boldsymbol{w}_k) \\
&= \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\boldsymbol{w}_k) = \nabla f(\boldsymbol{w}_k),
\end{aligned}$$

which is the desired result.

ii) This result shows that stochastic gradient methods with a constant step size can only be guaranteed to converge to a neighborhood of the optimal value. It also shows that this neighborhood becomes tighter as n_b grows. *Note that the bound is overly pessimistic when $n_b = n$, since the method is equivalent to gradient descent in that case.*

d) Below are two possibilities for guaranteeing better convergence properties:

- Use a decreasing step size sequence instead of a constant one;
- Average out the iterate.

e) If stochastic gradient ($n_b = 1$) improves over gradient descent ($n_b = n$), this indicates there is enough correlation in the data to converge using random subsets of it at every iteration. Using more than one data point yet significantly less than n (mini-batching) can reduce the variance of the gradient estimates while remaining significantly cheaper than a full gradient estimation: this can explain why $n_b = n/10$ yields better performance than $n_b = 1$. When the batch size gets closer to n, its cost also gets closer to that of a full gradient iteration, and the method becomes at risk of suffering from redundancies in the data. This can explain why the behavior of the method worsens when $n_b > n/10$. *Note: This is an open question. Students should be able to a) provide intuition as to why stochastic gradient works better than gradient descent and b) distinguish the mini-batch regime (n_b relatively small) from the regime $n_b \approx n$, where the method tends to behave like gradient descent.*

Solutions of Exercise 5.2: Importance sampling

a) This sampling strategy will favor components that have larger Lipschitz constants and thus are more likely to vary between successive iterates.

b) Using the distribution of i_k, we have that

$$\begin{aligned}
\mathbb{E}_{i_k}\left[\frac{1}{c_{i_k}}\nabla f_{i_k}(\boldsymbol{w}_k)\right] &= \sum_{i=1}^{n} \mathbb{P}\left(i_k = i\right)\frac{1}{c_i}\nabla f_i(\boldsymbol{w}_k) \\
&= \sum_{i=1}^{n} \frac{c_i}{\sum_{j=1}^{n} c_j}\frac{1}{c_i}\nabla f_i(\boldsymbol{w}_k) \\
&= \sum_{i=1}^{n} \frac{1}{\sum_{j=1}^{n} c_j}\nabla f_i(\boldsymbol{w}_k) \\
&= \sum_{i=1}^{n} \frac{1}{n}\nabla f_i(\boldsymbol{w}_k) = \nabla f(\boldsymbol{w}_k),
\end{aligned}$$

where we used $\sum_{j=1}^{n} c_j = \sum_{j=1}^{n} \frac{nc_j}{\sum_{\ell=1}^{n} c_\ell} = n$.

c) (Comparison with classical SG).

i) Since $\alpha_k = \frac{1}{L}$, we have

$$\frac{\alpha_k}{c_{i_k}} = \frac{1}{L}\frac{\sum_{j=1}^{n} L_j}{nL_{i_k}} = \frac{n}{\sum_{j=1}^{n} L_j}\frac{\sum_{j=1}^{n} L_j}{nL_{i_k}} = \frac{1}{L_{i_k}}.$$

ii) If i_k is the random index drawn at iteration k, the kth iteration of stochastic gradient reads

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f_{i_k}(\boldsymbol{w}_k) = \boldsymbol{w}_k - \frac{1}{L}\nabla f_{i_k}(\boldsymbol{w}_k),$$

while the iteration (26) becomes

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \frac{\alpha_k}{c_{i_k}} \nabla f_{i_k}(\boldsymbol{w}_k) = \boldsymbol{w}_k - \frac{1}{L_i}\nabla f_{i_k}(\boldsymbol{w}_k).$$

As a result, the second iteration will take a smaller stepsize in the direction $-\nabla f_{i_k}(\boldsymbol{w}_k)$ if $L_i \geq \frac{1}{n}\sum_{j=1}^n L_j$, i.e. when the ith Lipschitz constant is larger than the average. This is precisely what importance sampling aims at achieving: the stepsize is adjusted according to the Lipschitz constant, in order to reduce the impact of the components with an excessively large Lipschitz constant.

Solutions of Exercise 5.3: Randomized Kaczmarz

a) All functions f_is as well as f are nonnegative functions, i.e. $f_i(\boldsymbol{w}) \geq 0$ and $f(\boldsymbol{w}) \geq 0$ for all $\boldsymbol{w} \in \mathbb{R}^d$. Since $f_1(\boldsymbol{w}^*) = \cdots = f_n(\boldsymbol{w}^*) = f(\boldsymbol{w}^*) = 0$, we conclude that $\boldsymbol{w}^*$ is the minimum of f and all f_is, with the optimal value being 0.

b) i) The formula is $\boldsymbol{g}_k = \frac{1}{|S_k|}\sum_{i\in S_k} \nabla f_i(\boldsymbol{w}_k)$, with S_k a subset of n_b indices drawn at random with or without replacement in $\{1,\ldots,n\}$. Using $|S_k| = 1$ corresponds to stochastic gradient, while drawing n indices without replacement corresponds to gradient descent.

ii) With the notations of the previous question, the iteration (28) costs $|S_k|$ accesses to $\boldsymbol{x}_i$s.

iii) Such an iteration is $\frac{n}{|S_k|}$ times cheaper than an iteration of gradient descent.

c) i) This result suggests that on average, the method will converge at a linear rate towards a neighborhood of the optimal value.

ii) Let B represent a given budget of accesses. Given that an iteration corresponds to n_b accesses to $\boldsymbol{x}_i$s, the budget is used in $\frac{B}{n_b}$ iterations, hence (29) yields a rate $\mathcal{O}(t^{B/n_b})$. For gradient descent, a similar argument shows that the rate must be in $\mathcal{O}(t^{B/n})$, which is worse.

d) i) The randomized Kaczmarz method corresponds to vanilla stochastic gradient (i.e. batch size 1) used with unit stepsize.

ii) Using that $\boldsymbol{x}_i^T \boldsymbol{w}^* = y_i$ for any $i \in \{1,\ldots,n\}$, we have

$$\begin{aligned}
\boldsymbol{w}_{k+1} - \boldsymbol{w}^* &= \boldsymbol{w}_k - \boldsymbol{w}^* - (\boldsymbol{x}_{i_k}^{\mathrm{T}}\boldsymbol{w}_k - y_{i_k})\boldsymbol{x}_{i_k} \\
&= \boldsymbol{w}_k - \boldsymbol{w}^* - (\boldsymbol{x}_{i_k}^{\mathrm{T}}\boldsymbol{w}_k - \boldsymbol{x}_{i_k}^{\mathrm{T}}\boldsymbol{w}^*)\boldsymbol{x}_{i_k} \\
&= \boldsymbol{w}_k - \boldsymbol{w}^* - \boldsymbol{x}_{i_k}\boldsymbol{x}_{i_k}^{\mathrm{T}}(\boldsymbol{w}_k - \boldsymbol{w}^*) \\
&= (\boldsymbol{I}_d - \boldsymbol{x}_{i_k}\boldsymbol{x}_{i_k}^{\mathrm{T}})(\boldsymbol{w}_k - \boldsymbol{w}^*).
\end{aligned}$$

Taking the expected values on both sides and using the independence of i_k with respect to $i_0,\ldots,i_{k-1}$ and thus x_k gives the desired result.

iii) Using the new assumption together with the result of question b) leads to

$$\mathbb{E}_{i_k}[\boldsymbol{w}_{k+1} - \boldsymbol{w}^*] = \frac{1}{2}(\boldsymbol{w}_k - \boldsymbol{w}^*).$$

Consider $K \geq 1$. Applying the argument recursively (thanks to independence of the indices, we get

$$\mathbb{E}_{i_0,\ldots,i_{K-1}}[\boldsymbol{w}_K - \boldsymbol{w}^*] = \frac{1}{2^K}(\boldsymbol{w}_0 - \boldsymbol{w}^*),$$

showing that $\boldsymbol{w}_k - \boldsymbol{w}^*$ converges in expected value towards the zero vector.

iv) This result explicitly uses the existence of $\boldsymbol{w}^*$ and the fact that it gives zero error on all f_i, i.e. that $f_i(\boldsymbol{w}^*) = 0$. It also makes use of the quadratic nature of the function. Overall, it is an analysis specialized to the particular setup, and it is thus poised to provide better guarantees.

e) i) We have

$$\begin{aligned}
\mathbb{E}_{i_k}\left[\frac{\boldsymbol{x}_{i_k}^{\mathrm{T}}\boldsymbol{w}_k - y_{i_k}}{\|\boldsymbol{x}_{i_k}\|^2}\boldsymbol{x}_{i_k}\right] &= \sum_{i=1}^{n} \mathbb{P}(i_k = i)\frac{\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_k - y_i}{\|\boldsymbol{x}_i\|^2}\boldsymbol{x}_i \\
&= \sum_{i=1}^{n} \frac{\|\boldsymbol{x}_i\|^2}{\sum_{\ell=1}^{n}\|\boldsymbol{x}_\ell\|^2}\frac{\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_k - y_i}{\|\boldsymbol{x}_i\|^2}\boldsymbol{x}_i \\
&= \frac{1}{\sum_{\ell=1}^{n}\|\boldsymbol{x}_\ell\|^2}\sum_{i=1}^{n}(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_k - y_i)\boldsymbol{x}_i \\
&= \frac{1}{\sum_{\ell=1}^{n}\|\boldsymbol{x}_\ell\|^2}\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}_k - \boldsymbol{y}) \\
&= \frac{n}{\sum_{\ell=1}^{n}\|\boldsymbol{x}_\ell\|^2}\frac{1}{n}\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}_k - \boldsymbol{y}).
\end{aligned}$$

The desired conclusion follows by setting $\gamma = \frac{n}{\sum_{\ell=1}^{n}\|\boldsymbol{x}_\ell\|^2}$.

ii) To cope with stochastic gradient coordinates that vary in magnitude, one can use algorithms like Adagrad, RMSProp or Adam. These methods all use a diagonal scaling for the stepsize based on past information about the value of those coordinates.