



MSC. DATA SCIENCE & ARTIFICIAL INTELLIGENCE

ADVANCED DEEP LEARNING

Dr Alessandro BETTI

---

## Assignment 1

---

*Author:* Joris LIMONIER

joris.limonier@gmail.com

*Due:* October 5, 2022

# Contents

<b>1</b>	<b>Exercise 1</b>	<b>1</b>
<b>2</b>	<b>Exercise 2</b>	<b>1</b>
2.1	Question 1 . . . . .	1
2.2	Question 2 . . . . .	1
2.3	Question 3 . . . . .	1
2.4	Question 4 . . . . .	1
2.5	Question 5 . . . . .	1
2.6	Question 6 . . . . .	1
2.7	Question 7 . . . . .	1

## 1 Exercise 1

## 2 Exercise 2

### 2.1 Question 1

I was not familiar with the PyTorch library, so I had to perform some research in order to know how its methods/classes work. I consulted several sources, including the following, which were useful in order to answer this question:

1. <https://discuss.pytorch.org/t/how-sgd-works-in-pytorch/8060/2>
2. <https://discuss.pytorch.org/t/performing-mini-batch-gradient-descent-or-stochastic-gradient-descent-on-a-mini-batch/21235>
3. <https://pytorch.org/docs/stable/generated/torch.optim.SGD.html>

According to those sources, PyTorch's SGD actually computes a full-batch (vanilla) Gradient Descent, based on the data that is passed to it. It is my understanding that in order to perform actual mini-batch (*i.e.* where  $1 < \text{batch size} < \text{number of observations}$ ), one simply needs to give subsets of the data a each iteration.

In our case, we use the full dataset in `outputs = net(inputs)`, which is why we perform full-batch GD, although we call the `optim.SGD` class.

### 2.2 Question 2

### 2.3 Question 3

### 2.4 Question 4

### 2.5 Question 5

### 2.6 Question 6

### 2.7 Question 7